Whole-slide Image Classification in Digital Pathology using Deep Learning

Pham, Paul

Award date:
2021

Department of Mathematics and Computer Science
UMC Utrecht

Whole-slide Image Classification in Digital Pathology using Deep Learning

Master Thesis

Paul Pham

Supervisors:

Dr. Meng Fang, TU/e
Dr. Mitko Veta, TU/e
Msc. Nikolas Stathonikos, UMC Utrecht

Eindhoven, November 2021

Abstract

In this thesis we focus on Deep Learning models that classify whole-slide images without the need of region of interest annotations. With these models, we would like to enhance the workflow of a pathologist and essentially let the model be a second pair of eyes. We choose weakly-supervised models, since annotating whole-slide images is a tedious and time-consuming job due to the large resolution of these images (e.g. 100.000 $\times$ 100.000 pixels).

We evaluate the models on three different tasks, which are breast cancer grade classification in young women, lymph node metastases detection and melanoma pathway classification. Furthermore, we propose our own top-k average pooling multiple-instance learning method which outperforms various other weakly-supervised methods on the breast cancer grade classification task. We also explore the field of self-supervised learning for feature extraction and compare this with a feature extractor pre-trained on ImageNet.

From our results we can conclude that building models without region of interest annotations requires expressive feature extractors which can extract valuable information from tissue sections and use these features in a context-aware manner. Furthermore, we think that models should provide a level of interpretation for the pathologist, since it is important to know what the model finds relevant for the classification.

Preface

I would like to thank Meng Fang for his guidance during my thesis project. His expertise helped me become self-critical of the work in line of research for Artificial Intelligence. Furthermore, I would like to thank Nikolas Stathonikos and Mitko Veta for their guidance and support. Without them this project would not have existed. I would also like to thank Suzanne Wetstein for her help and input for the breast cancer grade classification task. Lastly, I would like to thank my family and friends for their moral support.

Contents

Abbreviations

Abbreviation	Explanation
CLAM	Clustering-constrained-attention multiple-instance learning
CNN	Convolutional Neural Network
MIL	Multiple-instance learning
MTL	Multi-task learning
NIC	Neural Image Compression
ROI	Region of Interest(s)
WSI	Whole-slide image(s)
YBCP	Young Breast Cancer Project

Table 1: Commonly used abbreviations

Chapter 1

Introduction

Pathology is the study of disease. When a patient is suspected to have a certain disease, the patient can get a biopsy to get a better understanding of the situation. The pathologists can examine the tissue of the patient and possibly diagnose the patient in collaboration with other medical experts.

A technological development within pathology is the use of digital slide scanners. Given a tissue sample on a glass slide, these scanners produce a digitized image called a whole-slide image (WSI) [1]. Examiners use WSI image viewers to inspect WSI, since these images frequently have a large resolution, e.g. 100.000×100.000 pixels; and can zoom in at different magnifications. Often they also include tools to make comments and annotations, which examiners can use to mark regions of interest (ROI) within a WSI.

Since WSI are digital they have several benefits over examining tissue slides using microscopes, namely they can be shared more easily, the quality stays constant and it allows for automated image analysis [2]. Examples of adapting artificial intelligence (AI) within automated image analysis are nuclei segmentation/classification [3] and slide-level classification [4].

1.1 Motivation

Classification tasks on WSI for digital pathology faces different kinds of problems then conventional image classification tasks, such as classification on the ImageNet dataset [5]. The main challenge of WSI classification is that the resolution of WSI are vastly greater than that of classical image datasets. Although many Deep Learning architectures vary greatly from each other, they often incorporate convolutional layers to extract features from feature maps [6]. Applying *state-of-the-art* architectures for image classification directly on a WSI is infeasable due to the computational load.

Another challenge that the size of WSI introduces is the lack of annotated data [7]. Since WSI are very large, it is infeasible to annotate ROI of large sets of WSI. Annotating WSI requires a vast amount of time/concentration and expert medical knowledge, which is not reasonable.

We want to give pathologists a second pair of eyes for classification tasks on WSI to enhance their workflow. Having an automated image analysis tool that can classify WSI and possibly indicate ROI would be a major benefit for pathologists.

1.2 Objective

Our thesis project focuses on WSI classification without using ROI annotations. We apply our experiments on breast cancer grade classification in young women (≤ 40 years of age), lymph node metastases detection and melanoma pathway classification.

1.3 Contributions

Our main contributions are:

- Our custom multiple-instance-learning (MIL) with top-k average pooling implementation, which outperforms the MIL with max-pooling and clustering-constrained-attention multiple-instance learning (CLAM) models on the breast cancer grade classification task.
- A comparison between different feature extractors and their impact on model performance. We mainly focus on two types of feature extractors, a ResNet-50 pre-trained on ImageNet and a self-supervised feature extractor based on SimCLR.
- We extend the options for data augmentation of Neural Image Compression by adding arbitrary angle rotation, horizontal/vertical flipping, Gaussian blurring, brightness change and color jitter on a tissue region scale.

1.4 Outline

The remainder of this work is presented as follows. Chapter 2 contains the problem statement. In Chapter 3, the literature analysis is given. In Chapter 4, the Materials and Methods are described. In Chapter 5, we evaluate the models on the different tasks. Finally, in Chapter 6, we give our conclusions, limitations and possible future work.

Chapter 2

Problem statement

In this section, we explain the underlying tasks of breast cancer grade classification in young women, lymph node metastases detection and melanoma pathway classification. Furthermore, we elaborate on the main research question of this project.

2.1 Classification tasks

2.1.1 Breast Cancer Grade Classification in Young Women

Breast cancer is the most prevalent cancer in women, with 2.261.419 new occurrences and 684.996 deaths counted in 2020 alone [8]. In case of young women ($\leq$ 40 years of age), breast cancer is often more aggressive and their prognosis is less favorable than older women with breast cancer [9]. However, there is a proportion of young women that fall into the low-risk category, where their prognosis is favorable after locoregional treatment without the need of adjuvant systemic therapy [10]. Applying adjuvant systemic therapy to low-risk patients is considered overtreatment and might cause age-related side effects [11]. Therefore, to reduce overtreatment in young women with breast cancer, it is important to distinguish low-risk patients from high-risk patients.

An initiative called PAtients with bReAst cancer DIaGnosed preMenopausally (PARADIGM) aims to reduce overtreatment of patients with breast cancer under the age of 40 [11]. PARADIGM created a dataset that contains 3525 patients that were diagnosed between 1989 and 2000, ages $\leq$ 40, and who did not receive adjuvant systemic or hormonal therapy. Henceforth, we call this dataset the YBCP dataset. Similarly to Wetstein et al. [12], we consider patients with main diagnosis of the case invasive ductal (no special type) carcinoma, with no other cancer types present. Furthermore, pathologists of the PARADIGM initiative assigned these patients a grade based on the Nottingham modification of the Bloom-Richardson system [13], which indicates the severity of the cancer. Since the grade of the cancer is linked to the risk factor, our goal is to classify WSI into low/intermediate grade and high grade.

2.1.2 Lymph Node Metastases Detection in Women with Breast Cancer

Detecting metastases in the lymph nodes of the breast is an important aspect in determining the breast cancer stage in patients. A well-known challenge within the field of automated image analysis for digital pathology is Cancer Metastases in Lymph Nodes Challenge 2016 (Camelyon16), which purpose is two-fold; (I) identification of individual metastases and (II) classification of metastases on slide-level. For this thesis project, we are only interested in task II. Our goal is to classify a WSI in either normal lymph node or lymph node containing metastases.

2.1.3 Melanoma Pathway Classification

Melanoma [14] is a deadly form of skin cancer, which is a malignancy of melanocytes [15]; skin cells which are responsible for the pigmentation of the skin and offer protection against ultraviolet radiation. According to [8], the number of new melanoma cases in 2020 alone are 324.635 and the number of new deaths are 57.043, which makes melanoma the deadliest form of skin cancer.

The World Health Organization (WHO) defined 9 pathways/subtypes of melanoma [16], each described by their epidemiology, morphology and genomic characteristics. The WHO further grouped the 9 pathways into two groups; Cumulative Solar Damage (CSD) for pathways I-III and non-CSD for pathways IV-IX. For the purpose of this project, we are primarily interested in cases which belong to pathways I or IV, since they are the most occurring diagnoses within patients.

Pathway I consists of superficial spreading melanoma (SSM), which are melanoma with low-CSD. Low-CSD melanoma are characterized by the relative low amount of solar elastosis [17], opposed to pathways II and III, which contain cases of high-CSD. SSM is a spreading lesion primarily characterized by its radial growth phase (RGP), and according to [18] the neoplastic cells in the epidermis make a pattern that resembles the Paget's disease of the breast [19]. A tissue lesion is said to be in RGP if it grows alongside the horizontal axis within the skin, opposed to a vertical growth phase (VGP), which grows alongside the vertical axis below and/or above the skin. On the other hand, Pathway IV consists of spitz melanoma, which is a non-CSD melanoma. According to [16], Spitz melanoma are characterized by lesions containing large spindle and/or epithelioid melanocytes.

Our project focuses on the classification of melanoma of pathway I, IV and Remaining (WSI that fall into the remaining pathways).

2.2 Formalization

A WSI x has a variable resolution and typically is around 100.000 by 100.000 pixels. An example of a WSI can be seen in Figure 2.1. To demonstrate the size of the WSI we extract 3 patches from this slide at the same location with three different magnification levels, see Figure 2.2. The patch of 5× magnification is marked green in Figure 2.1.

In this example, the slide was scanned using a Philips UFS scanner 1.6.1.3 RA, where the micrometer per pixel ratio is 0.25 at 40× magnification.

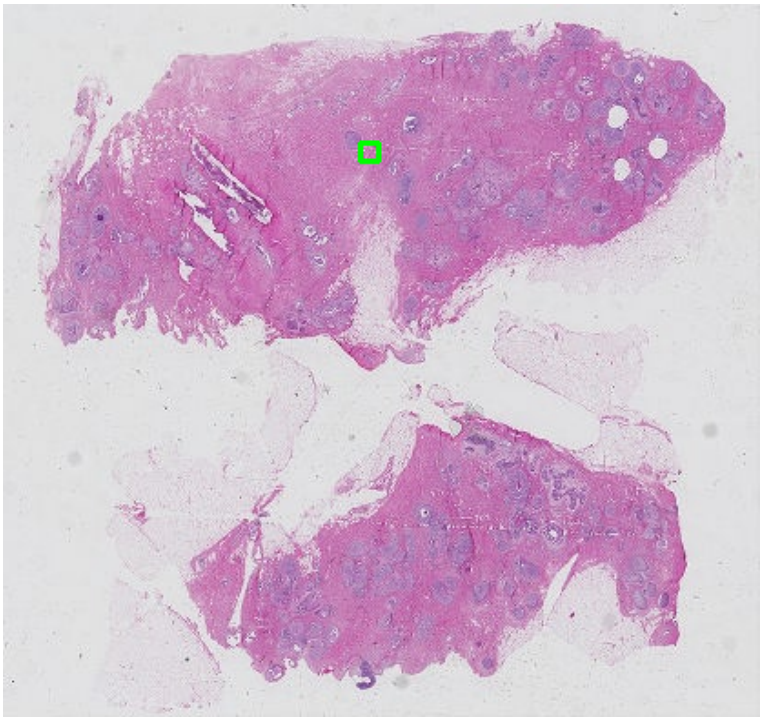
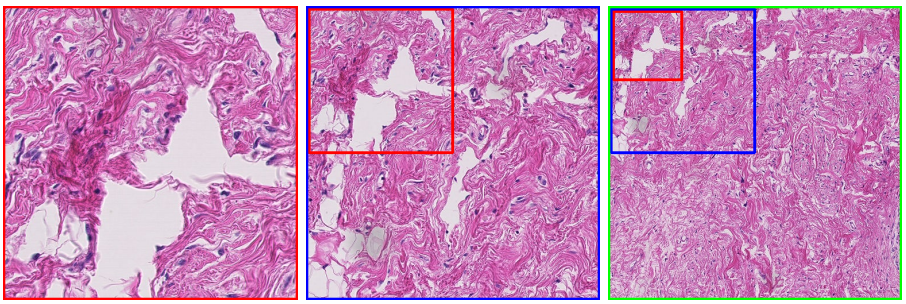


Figure 2.1: A WSI of size 100,352 by 106,496



(a) Patch at 20× magnification with 0.5 μm/pixel

(b) Patch at 10× magnification with 1 μm/pixel

(c) Patch at 5× magnification with 2 μm/pixel

Figure 2.2: Patches from $x = 50,000$ and $y = 20,000$ with the same image size, at different magnification levels. We can see how the patches at higher magnifications fit into the lower magnification patches

Given a WSI x and a model f, we make a prediction $f(x) = \hat{y}$, where $\hat{y}$ is an array of probabilities where each entry $\hat{y}_i$ denotes the probability of slide x being of class i. More formally, $\hat{y} = [\hat{y}_1, \ldots, \hat{y}_n]$, where $(\sum_{i=1}^{n} \hat{y}_i) = 1$ and $\forall \hat{y}_i \in \hat{y} : (0 \leq \hat{y}_i \leq 1)$, where n is the number of classes.

We evaluate the performance of the model by calculating three classification metrics, namely Area-Under-the-Curve (AUC), accuracy and F1-score. The AUC metric tells us how well the model performs under different thresholds, the accuracy tells us a percentage of how many samples were correctly classified out of the whole dataset and the F1-score is an aggregate metric, which takes into account the precision and recall of the model.

2.3 Research question

Since we have three different tasks and datasets, we face different kinds of challenges. However, there should be some common ground in the characteristics of the dataset and models used. Therefore, our main research question for this project is:

Which factors play an important role for classification on histopathology slides?

Chapter 3

Literature Analysis

We first elaborate on the field of Deep Learning and in particular Convolutional Neural Networks (CNNs). Thereafter, since we are interested in classifying WSI without using ROI annotations, we explore weakly-supervised learning, which is a sub-field of machine learning (ML) that deals with inexact, incomplete and inaccurate data. Next, we will discuss several frameworks that classify WSI, which are based on MIL and/or spatial context-aware approaches. Thereafter, we will discuss several works that quantify the performance of feature extractors. Finally, we will summarize our findings of the literature analysis.

3.1 Deep Learning

Deep Learning is a field in AI which enables models to automatically learn representations from data [20]. Prior to Deep Learning, many conventional machine learning methods required hand-crafted feature extractors build by experts with domain knowledge.

The rise of Deep Learning in recent years has been contributed to the availability of big datasets, computational power by graphical processing units (GPUs) and improving techniques/architectures by the research community [20]. It has seen a wide variety of applications, namely in visual, audio and text-related tasks.

In a supervised setting for a classification task, the goal is to learn a function f, such that given an input x, f predicts $\hat{y}$ that gives a probability score which ideally aligns with the true label y. Initially, the function f does not perform well as it has not been trained. In a neural network setting, f is the whole neural network, which contains the network parameters (weights and biases of the network). To achieve good predictions, these internal parameters need to be tuned during the training process of the network using stochastic gradient descent (SGD). With SGD, the network predicts the labels of the input, calculates the errors of the prediction using a loss function and adjusts the internal parameters of the network based on the accumulated gradients. The model stops training when the loss converges. Thereafter, the model can be used for inference. Note that during inference, the weights and biases will not be updated as the training procedure has finished.

CNNs have been shown to work well in the area of computer vision the past decade [20]. What characterizes CNNs is the series of convolutional layers, followed by a non-linearity and pooling layers. To predict the class of an image, the neural network flattens the feature maps into a vector, applies a linear fully connected layer followed by a softmax layer to get the predictions per class. An example of a CNN can be seen in Figure 3.1. Naturally, there are many variations to CNN architectures, e.g. ResNets [21] and VGG-networks [22], however a commonality between these networks is that they use convolutional and pooling layers.

A convolutional layer applies a series of convolutional operations on the input feature maps using a learnable filter to produce the next feature map. Within a convolutional layer, n filters are used to produce the next n feature maps. These learnable filter are often 3×3 or 5×5 in size, and therefore can learn local patterns within feature maps. A pooling layer reduces the dimensionality of a feature map by extracting the most significant features within a feature map. By applying a series of convolutional/pooling layers, the network can extract import features from the input image.

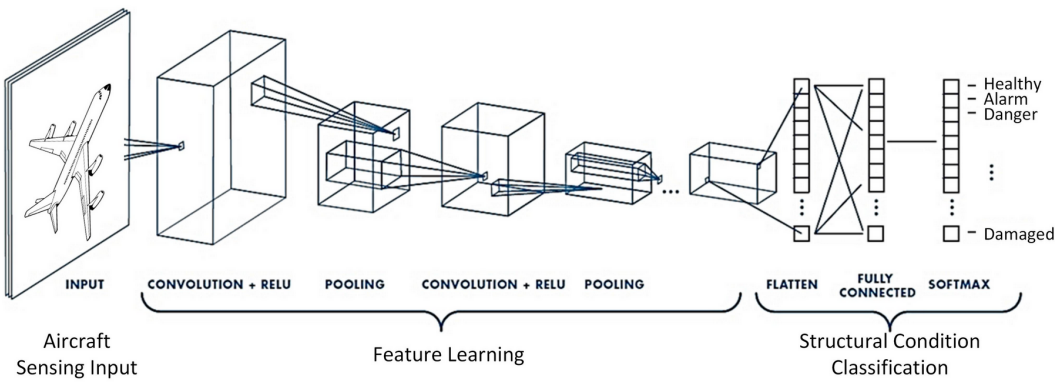


Figure 3.1: An example of a CNN which classifies the structural condition of an aircraft. Image is from [23]

3.2 Weakly-Supervised Learning

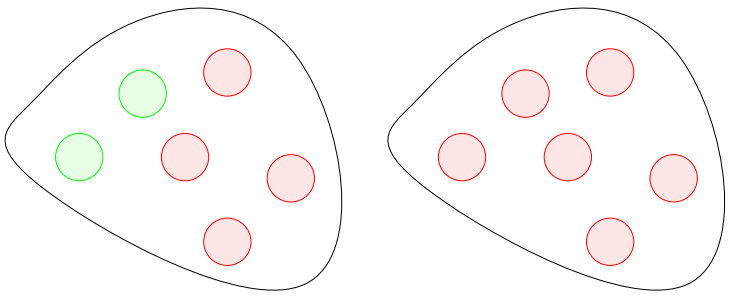
AI practitioners use weakly-supervised learning when labeling the dataset is tedious, time-consuming or infeasible. In [24], the author distinguished three types of weakly-supervised learning techniques, which includes inexact, incomplete and inaccurate supervision.

Inexact supervision deals with datasets where a set of instances — often referred to as bag — receives a single label. An example of inexact supervision is MIL [25]. MIL is a binary classification problem and is defined as follows: Given a bag X containing instances $x_1, ..., x_n$, then X is positive if at least one instance of X is positive, and X is negative if all instances of X are negative, see Figure 3.2. However, the labels of the dataset are not on instance-level, but on bag-level, which makes MIL an inexact weakly-supervised problem. MIL-classification problems are mainly solved in two ways; (I) aggregating the instance-level predictions (e.g. max-pooling) or (II) creating a bag-level feature representation.

In incomplete supervision only a subset of the dataset contains labels. Two main techniques fall under incomplete supervision, which are active learning and semi-supervised learning. Active learning requires a human-in-the-loop, where the machine learning algorithm can query the human for labels for unlabeled data [26]. Semi-supervised learning on the other hand requires no human intervention — the algorithm tries to learn from both the labeled and unlabeled data [27].

Inaccurate supervision deals with datasets that contain incorrect labels [24], an example of this is crowdsourcing [28]. In crowdsourcing, machine learning algorithms rely on workers who are outsourced to label the dataset. Naturally, not every worker can produce correct labels consistently due to difference in human judgement.

In our case, the most relevant form of weakly-supervised learning is inexact supervision. As already mentioned in the Introduction, the main challenge of WSI is the image resolution. For pathologists, it is infeasible to annotate detailed ROI in WSI by hand as it is very time-consuming. MIL is a viable option for WSI classification and has often been used with success [4; 29; 30]. In the task of WSI classification, the WSI can be seen as the bag, whereas patches sampled from the WSI can be seen as instances of that bag.



(a) A bag is positive if at least one instance of that bag is positive

(b) A bag is negative if all instances in that bag are negative

Figure 3.2: The multiple-instance learning principle in a binary classification task.

3.3 WSI classification methods

In this section we elaborate on WSI classification models which are MIL-based and spatial context-aware.

3.3.1 MIL-based approaches

3.3.1.1 Aggregating patch features by using recurrent neural networks

Campanella et al. uses the MIL-framework to classify WSI using only slide-level labels. Their model f first segments tissue from non-tissue area in a slide and then creates patches of size $P \times P$. Then for the training procedure, they do a full inference pass on all the patches of a slide and calculate their probabilities of being positive or negative (where positive indicates presence of the class). Thereafter, they update the model by using the top-1 ranked patch in the slide. They applied their model on a prostate cancer classification task containing roughly 12.000 slides and achieved an AUC of 0.977 with the model that uses VGG11-BN as backbone.

MIL-RNN [4] is an extension that builds on the MIL approach of Campanella et al. [31]. The framework has been tested on three different binary classification tasks: prostatic carcinoma, basal cell carcinoma and breast cancer metastasis in axillary lymph nodes. In total, the authors evaluated MIL-RNN on 44.732 WSI from 15.187 patients without data curation, thus the WSI contained common scanning artifacts due to irregularities. Examples of these artifacts include air bubbles, tissue folding and pen markings.

The MIL-RNN framework consists of two models, which are a tile-based classifier f and RNN aggregator g. The model f differs slightly from the model in their previous work. Instead of updating the model with the top-1 ranked patch, the model updates its weights with the top-K ranked patches.

After training the tile-based classifier f, the model trains the RNN-based aggregator g. Similarly to f, the model segments the WSI and extracts patches from it. Thereafter, it does a forward pass of those patches using f to retrieve the patch predictions. Next, the model uses the second-to-last layer of f to extract embeddings from the S most probable patches in the slide. Finally, aggregator g does a forward pass using these embeddings to retrieve the slide-level prediction.

The authors experimented on the prostate cancer dataset to check the influence of the dataset size on model performance. They set aside 2.000 validation WSI and trained their model using different training set sizes: 100, 200, 500, 1.000, 2.000, 4.000, 6.000, 8.000 WSI, where each training set was a superset of the previous datasets. They observed that an increase in training set size also improves model performance and concluded that at least 10.000 slides are needed to achieve satisfactory results. However, access to that many slides may not be feasible in classification tasks of rare diseases.

3.3.1.2 Multi-task learning with top-k MIL pooling

Wetstein et al. [12] uses only the tile-based classifier f of the MIL-RNN approach [4]. They extended their approach by adding a multi-task learning (MTL) approach [32]. In MTL we try to learn auxiliary tasks alongside the main task, which gives extra supervisory signal to the model. Wetstein et al. uses a hard-parameter sharing approach, where the base of the model is shared among the different tasks, while the different classification heads remain separate, see Figure 3.3. They applied the model to breast cancer grade classification in young patients, which is the task described in Section 2.1.1. Their single-head classification model achieved an accuracy of 0.77 ± 0.05, whereas their multi-task model, which uses the component grades (nuclear, tubular, and mitosis scores) as additional tasks, achieved an accuracy of 0.80 ± 0.05. This study showed that the supervisory signal of additional task could improve the overall performance of the model.

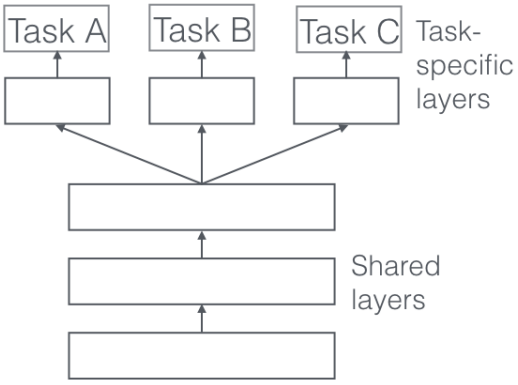


Figure 3.3: Hard parameter sharing between tasks. From [33]

3.3.1.3 Attention-based MIL models

Ilse et al. introduced Attention-based Deep MIL [29]. The attention-mechanism allows Attention-based Deep MIL to find key instances that it attends to for the final slide-level prediction. Their framework consists of three components, which include a dimensionality reduction of patches to low-level embeddings, a permutation-invariant attention-based pooling mechanism and a bag-level classifier.

Given a bag of embeddings $H = \{h_1, ..., h_k\}$, the pooling operation $\boldsymbol{z}$ is defined as follows:

$$\boldsymbol{z} = \sum_{k=1}^{K} a_k h_k \tag{3.1}$$

where a_k is defined as:

$$a_k = \frac{\exp\{\boldsymbol{w}^T \tanh(\boldsymbol{V} h_k^T)\}}{\sum_{j=1}^{K} \exp\{\boldsymbol{w}^T \tanh(\boldsymbol{V} h_j^T)\}} \tag{3.2}$$

and $\boldsymbol{w}$ and $\boldsymbol{V}$ are trainable parameters.

However, the authors noted that the pooling mechanism was suboptimal for $tanh(x)$ when $x \in [-1, 1]$ due to the linearity that it introduces. Therefore, the authors added the gating mechanism [34] to the pooling operation, which would retain the model's ability to learn complex relations. The gated attention mechanism changes a_k to the following:

$$a_k = \frac{\exp\{\boldsymbol{w}^T \tanh(\boldsymbol{V} h_k^T) \odot \text{sigmoid}(\boldsymbol{U} h_k^T)\}}{\sum_{j=1}^{K} \exp\{\boldsymbol{w}^T \tanh(\boldsymbol{V} h_j^T) \odot \text{sigmoid}(\boldsymbol{U} h_j^T)\}} \tag{3.3}$$

where $\boldsymbol{U}$ is a set of trainable parameters and $\odot$ is an element-wise multiplication.

What makes Attention-based Deep MIL interpretable is that the attention scores in the pooling operation $\boldsymbol{z}$ can be used to visualize patches that are important for the bag-level prediction. This is especially important in a clinical setting, since the model can justify its decisions to a certain extend.

CLAM [30] is an extension of Attention-based Deep MIL, which is suitable for multi-class classification problems. Furthermore, it introduces instance-level clustering over high- and low diagnostic-valued patches to refine the feature space.

The authors evaluated CLAM on three classification tasks, which are renal-cell carcinoma (RCC) subtyping, non-small-cell-lung-cancer (NSCLC) subtyping and breast cancer lymph node

metastases detection. The authors highlight in the paper that data efficiency is an important aspect with classification tasks of rare diseases, as there are limited WSI available. They concluded that different tasks require varying number of slides to achieve an AUC > 0.9.

Possible improvements for both Attention-based Deep MIL and CLAM is that they use fixed feature extractors to compute embeddings of patches. These feature extractors are often pre-trained ResNets and are not adapted to the dataset. A possible improvement for the feature extractor is to make it domain specific.

The authors of CLAM proposed a different method called Tumor Origin Assessment via Deep Learning (TOAD), which uses a similar attention mechanism for WSI prediction as CLAM. However, the use case of TOAD is different, as it is a multi-task classification network which can predict whether a WSI contains primary or metastatic tumor and the site of origin of the tumor e.g. breast, lung and skin. In addition to the WSI as input, the authors also provide the network the sex of the patient as additional input, which has been shown to be beneficial for the slide-level prediction according to their ablation studies.

TOAD can be applied in cases of cancer of unknown primary (CUP) origin, which are cases where the primary site of origin of the tumor is undetermined. This will help pathologist narrow down possible sites of origins, and thus help in the process of finding a diagnosis.

Another extension of Attention-based Deep MIL is Self-supervised pre-training and heterogeneity-aware deep Multiple Instance LEarning (DeepSMILE) [35]. The main limitations that DeepSMILE addresses in Attention-based Deep MIL are: (1) the feature extractor trained on ImageNet is sub-optimal to extract features from medical images and (2) the attention-based pooling is unable to capture global patterns in WSI.

To address the first limitation, DeepSMILE uses SimCLR [36], which is a self-supervised feature extractor, to extract features from patches. In their paper, they evaluated SimCLR on 12 different classification tasks and it achieved results on par with supervised baselines. Furthermore, DeepSMILE addresses the second limitation by calculating the variability between features across tiles, which allows the model to capture global features within a WSI.

The authors evaluated DeepSMILE on Homologous recombination deficiency (HRD) and microsatellite instability (MSI) prediction and achieved for both tasks state-of-the-art performance with an AUC of 0.8379 and 0.9032 respectively. Furthermore, their model with SimCLR as feature extractor performed better than their model with an pre-trained ImageNet feature extractor.

3.3.1.4 Large margin principle MIL

The authors of [37] take a more traditional machine learning approach to solve the binary WSI classification problem. They adapt the large-margin principle in the MIL framework, where the large margin classifier tries to maximize the margin between two different classes, while minimizing the false positives. They use two loss functions to optimize their model. In case of a negative case, they incur loss when at least one instance of the ROI within a slide is positive. In case of a positive slide, they incur a loss when all ROI within a slide is negative. Both loss functions align with the MIL framework. The ROI are extracted by first converting them into CIELAB color space, and consequently applying a clustering algorithm to segment cytological components from the ROI. Thereafter, Haralick texture descriptors are used to extract a feature vector of length 9 from the ROI. A limitation of this work is that the feature extractor is untrainable unlike feature extractors such as neural networks.

3.3.2 Spatial context-aware approaches

Most MIL models for WSI classification are patch-based and do not consider a larger spatial context. In this section we elaborate on several methods that do take into account a larger spatial context and thus can extract more global features from WSI.

3.3.2.1 Compression-based models

Neural Image Compression (NIC) is a framework that compresses a WSI x into x' and consequently classify x' [38]. The authors applied NIC on a synthetic dataset, the CAMELYON16 dataset [39], TUPAC16 dataset [40] and a rectal carcinoma dataset [41].

Given x with dimensions $W \times H \times 3$, NIC compresses x into x' with dimensions $\frac{W}{S} \times \frac{H}{S} \times C$, where W is the width, H is the height, S is the stride and C is the compression length. NIC creates a grid of x by dividing it into patches of size $P \times P \times 3$ with a stride of S. For each patch in the grid, it compresses the patch into an embedding/feature vector of length C using an encoder. In the paper, they explore unsupervised learning methods to extract embeddings from the patches — these methods include Variational Auto-encoders (VAEs) [42], Bidirectional Generative Adversarial Networks (BiGANs) [43] and contrastive learning [44]. See Figure 3.4 for an overview of compressing a WSI.

After training the encoder in the compression step, NIC trains a convolutional neural network (CNN) to classify x'. By using a CNN on x', the authors argue that NIC is able to capture not only local features of patches, but also the spatial relations among patches.

The authors noticed several drawbacks of NIC. Firstly, NIC has troubles classifying WSI with small tumor lesions, which might be caused by the expressiveness of the encoders. A better encoder may lead to better results in slides with low witness rates. Secondly, the CNN of NIC can overfit easily due to the lack of data augmentation on the WSI (only 90-degrees rotations and mirroring on the WSI) and small dataset size. Lastly, NIC is computationally expensive which makes optimizing hyperparameters unfeasible.

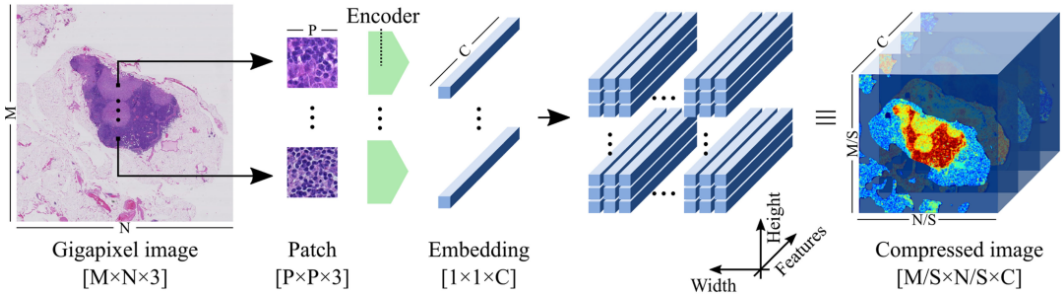


Figure 3.4: NIC compresses each patch in the WSI by using the trained encoder. Image is from [38]

An extension of NIC is presented in [45], where the authors use a MTL approach for encoding patches. In their method, they have a network consisting of an encoder and four separate classification heads. The goal of the network is to classify patches of four separate tasks, using the same encoder, which would make the encoder highly transferable among different tasks. After training the encoder network, they followed the same procedure as NIC for the classifier. The authors argue that training an encoder in a multi-task supervised manner produces more discriminative embeddings of the patches and therefore enhances the classification accuracy, compared to an unsupervised feature extraction. For the TUPAC16 challenge [40], they achieved state-of-the-art performance with an AUC of 0.632 on the test set. However supervised encoding for NIC may be infeasible when labeled datasets of patches are not available. This is usually the case as these datasets are time-consuming and tedious to construct.

Shaban et al. [46] propose a technique which is similar to NIC. In their work, they also make

a compression of a region, however this region is considerable smaller than of NIC 1.792×1.792 versus 50.000×50.000. Furthermore, Shaban et al. employ an attention mechanism which can localize areas that of high importance within a compressed tissue region. The slide-level prediction is based on majority vote, whereas NIC averages the predictions of each compressed tissue region. Both approaches may suffer from slides which contain a small tissue region.

3.3.2.2 Multi-scale approach

Hashimoto et al. [47] propose a framework called Multi-scale Domain-adversarial Multiple-instance CNN, which they applied to malignant lymphoma sub-type classification. Their contributions are two-fold; (I) extracting local and global features from multiple magnifications from a WSI and (II) robustness against differences in staining conditions among hospitals/specimens by using domain adversarial normalization.

Their approach consists of training n feature extractors for n magnification-levels. The feature extractors in their framework are also referred to as DA-MIL. For each feature extractor, they incur two losses; a classification loss and a loss for the domain predictor. The classification is calculated using the attention mechanism of Ilse et al. [29], in which they calculate an attention score per patch and then compute an attention-weighted bag-level feature vector. Furthermore, the domain predictor gives a prediction on the possible domains, where in their case each patient is seen as a single domain.

After training the n DA-MIL models, they train a multi-scale model, which they call MS-DA-MIL. This model incorporates the n DA-MIL models trained previously. This training process for MS-DA-MIL is similar to training DA-MIL, however they only compute the classification task and disregard the domain adversarial training. For the classification task, they extract features from multiple magnifications and apply the attention mechanism to compute the attention-weighted bag-level feature vector, which is consequently used for classification.

In their work, they applied MS-DA-MIL with $10\times$ and $20\times$ magnification, however it would be interesting how the model would perform by using more than 2 magnification levels, e.g. $5\times$ or $40\times$. Furthermore, they did not include an ablation study on the effect of the domain adversarial normalization.

Wetteland et al. [48] takes a different approach to incorporating multiple magnifications into their network. Their model has a branch for each magnification level input and concatenate the features from all the branches before the classification layer. They experimented with combining 2 and 3 different magnification levels for their models and achieved better performance than using a single magnification level.

However, the models in this work are only patch-based classifiers and have not been applied in a WSI setting.

3.3.2.3 Context by large area

Bejnordi et al. [49] propose a framework called Context-aware stacked CNN (CAS-CNN) for the classification of breast WSI into normal, ductal carcinoma in situ and invasive ductal carcinoma. The model incorporates two CNNs. The first CNN extracts features from patches of size 224×224 with a wide-ResNet-architecture, which they call WRN-4-2. After training this network, the filters are used to train the CAS-CNN model. The authors use a larger patch size for this network to capture more global/contextual features. They experimented with patch sizes of (512×512, 786×786 and 1024×1024) for the CAS-CNN.

The authors found that increasing input size for the CAS-CNN model improves the model performance, which comes at the cost of computational efficiency. It would be interesting to see how this method compares to a model that uses the same patch regions as CAS-CNN, but with lower magnification levels. By using a lower magnification level, we reduce the computational load of the model, while maintaining high contextual information. However, we might lose some level of detail since we use a lower magnification level, which does not pose a problem for CAS-CNN, since it uses the same magnification level as WRN-4-2.

3.3.2.4 Clustering approach

Gueréndel et al. [50] explore a novel approach in creating context-aware WSI classification models. Their main contribution is a slide-level representation of a slide based on k-means clustering of feature vectors of instances.

Firstly, they extract tiles from a WSI by using a segmentation network and then pass these tiles through two tile-validation networks. These networks validate whether the tile is in-focus and whether they have pen-markings. These tile-validation networks can be seen as a pre-processing step, which filters out tiles containing scanning artifacts. Secondly, they extract features from these tiles using a ResNet-18 which was trained for tissue classification. Thirdly, they apply a k-means clustering algorithm to cluster patch features. Thereafter, they construct a slide-level representation by checking the proportion of patches within a cluster. Finally, this slide-level representation is used for training a WSI classifier based on XGBoost [51].

3.4 Feature Extraction

We discussed several ways of feature extraction within WSI classification models in Section 3.3 and came across strategies which use supervised learning, weakly-supervised learning and unsupervised learning, see Figure 3.5. However, these papers often do not compare the performance of feature extractors or quantify their expressiveness. In this section, we address these two problems with feature extractors using MTL and self-supervised learning.

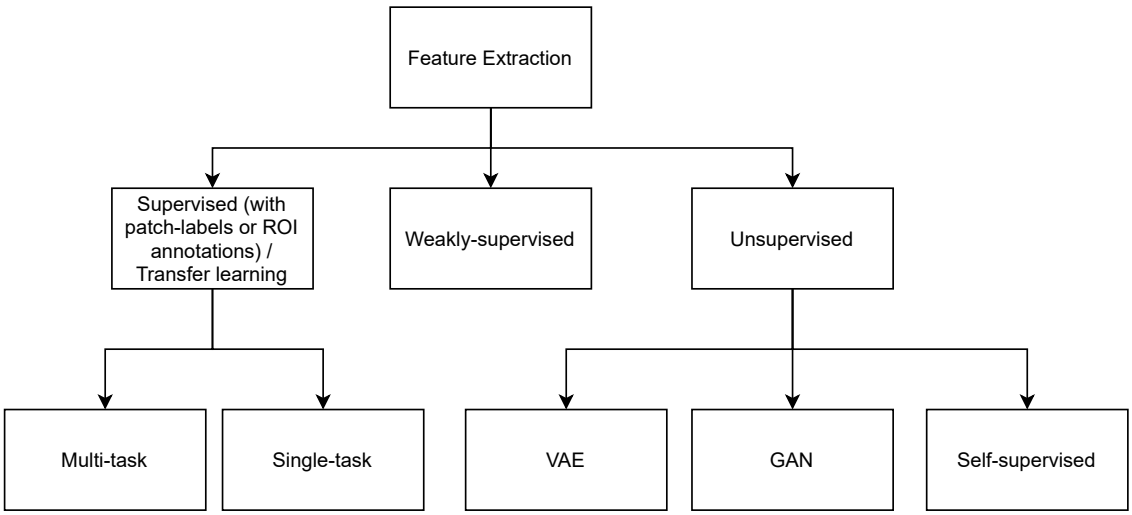


Figure 3.5: Various feature extraction strategies

Performance by accuracy metrics
Mormont et al. [52] did a study on the performance of different deep transfer learning strategies for pathology images. They concluded that fine-tuning networks outperforms pre-trained networks feature extractors.

Mormont et al. [53] extended their previous work in feature extraction by exploring the field of pre-training models using MTL. They constructed a large dataset from many digital pathology datasets to create 22 classification tasks with roughly $900k$ images.

They adopt a leave-one-task-out (LOTO) protocol to evaluate the transferability of their MTL model. Given task $t \in \mathcal{T}$, they leave t out of the MTL model and train the MTL model on $\mathcal{T} \setminus t$. The MTL model contains base layers and classification heads for the $\mathcal{T} \setminus t$ tasks. They adopt the LOTO strategy to check how the MTL model performs for task t, and thus are not necessarily interested in the performance of the MTL model on the tasks of $\mathcal{T} \setminus t$.

After training the MTL model, there are two different strategies they explored for using the MTL model; (I) train a linear support vector machine classifier on the features extracted from MTL model for task t or (II) remove the classification heads of the MTL model and attach a fully connected and softmax layer to further fine-tune the network for task t. In (I), they only use the MTL network as feature extraction, whereas in (II) they further fine-tune the network.

Their results show that using a MTL model as feature extractor achieves comparable results to a pre-trained model on ImageNet and in some instances improve upon that. Furthermore, their fine-tuned networks improve upon the MTL models which only act as feature extractors. It would be interesting to see how the performance of these models would compare on features extracted from different layers of the network.

Ciga et al. [54] compares feature extractors of three different types; self-supervised learning using SimCLR, pre-trained models on ImageNet and randomly initialized models. In their experiments, all three model types had a similar backbone based on ResNet. They applied their feature extractors on classification, segmentation and regression tasks and concluded that self-supervised

pre-training outperforms pre-trained models on ImageNet and randomly initialized models. Furthermore, they found that using a wide variety of datasets from different organs improves the quality of feature extraction in the self-supervised feature extractor. The increase in performance by using data from multiple organ sites aligns with the findings of Mormont et al. [53] and Tellez et al. [45].

Performance by similarity

Gildenblat et al. [55] explore the field of self-supervised learning for feature extraction and quantifies the expressiveness of their feature extractor. Their model is build on the Siamese network framework which learns a similarity function based on pairs of patches. The pairs of patches are either similar or non-similar, where similarity is based on the spatial distance between patches. The authors assume spacial continuity in patches, thus patches that are close to each other are similar and patches that are distant to each other non-similar. They defined a metric called global Average Descriptor Distance Ratio (ADDR) as: '*the ratio of the average descriptor distance of non-similar pairs and the average descriptor distance*', where the descriptor is an embedding of a patch.

They compared their Siamese network with a ResNet-50 pre-trained on ImageNet and a Non-Parametric Instance Discrimination method and found that their Siamese network achieved best performance based on the ADDR metric.

However, the authors noted that the spatial continuity assumption in WSI is not completely sound, as their approach could lead to false-positive pairs. For instance, in case we select a patch in the border of a tumor cluster, then a similar patch is within the proximity of that tumor patch, which could be either a tumor patch or a non-tumor patch. We could mitigate this problem by enforcing additional constraints on the pair construction of the dataset. For instance, we could use the same strategy for measuring performance (distance between descriptors) as a prerequisite for pairs, for which we would then use a ResNet-50 pre-trained on ImageNet. This way, we take into account both the spatial distance, as well as the similarity between patches. Note that this is limited by the expressiveness of the pre-trained model.

Another area that could be further looked into is the encoding size. In this particular study, they used an encoder length of 128 for the descriptor/patch, however a larger encoding size could lead to better performance. Note that this does come at the cost of computation.

In their work, they have performed their analysis on the Camelyon16 challenge using its annotations. It would be interesting to see whether such self-supervised method could also be used in a weakly-supervised manner and how it would compare with using these annotations.

3.5 Summary

In this literature analysis, we first elaborated on the field of Deep Learning and weakly-supervised learning. We argue that a weakly-supervised approach to solving classification tasks on WSI is necessary in case we do not have ROI annotations. This is usually the case as annotating ROI for pathologists is a time-consuming and tedious job due to the large resolution size of WSI. We mainly focused on techniques which use neural networks, as their automatic representation learning approach has been shown to be effective in the field of computer vision.

Thereafter, we explored weakly-supervised WSI classification models which we grouped into MIL-based methods and spatial context-aware methods.

For the MIL-based category, we explored a traditional MIL approach by Campanella et al. [31] which uses max-pooling as an aggregation method for pooling the patch based predictions. They extended their work by using a model on top of their patch-based classifier, which could aggregate patch features by using an RNN. A downside of these models is that they need thousands of slides to achieve good performance.

Thereafter, we explored MIL models which were based on attention. These models could give patches an attention scores which tells the relative importance of the patch for the slide-level diagnosis. Lu et al. [30] in particular found that their CLAM method was more data-efficient than MIL with max-pooling and thus needed less slides to train their model.

Within these models, various different feature extractors were used, e.g. a ResNet-50 pre-trained on ImageNet and a self-supervised SimCLR model in DeepSMILE [35]. They found that in DeepSMILE that their model performed better with the self-supervised feature extractor than an ImageNet feature extractor.

For the spatial context-aware category, we explored methods that could be aware of a larger spatial context within a slide. The first context-aware methods we discussed were compression-based models. These models consist of a compression step and a classification step. Since WSI are large, MIL-models often create patches out of WSI to reduce the computational load. However, when we apply a permutation-invariant aggregation technique in MIL, we lose spatial context. Compression-based models try to extract features from WSI both on local (patch-level) as well as global (grid of patches) features. The compression-based models would compress a large region in a slide, where this region is a grid of patches. Thereafter, the compressed tissue region in a slide is passed through a classification model, which then gives the prediction of that tissue region in a slide. To predict a WSI, they classify all the tissue regions within a slide and aggregate the results (e.g. average-pooling or majority vote). However, the disadvantage of these models is that small ROI might go unnoticed due to expressiveness of the feature extractor or due to the sub-optimal aggregation strategy.

Thereafter, we have seen models which tries to achieve context by extracting features from multiple scales/magnifications. They found that the multi-scale approaches often would perform better than single-scale approaches.

Furthermore, we have seen an approach which tries to achieve context by taking in consideration a large area. Typically in WSI models, the input patches range from 100×100 pixels to 256×256 pixels, however in this case, the authors experimented with patch sizes of 512×512, 786×786 and 1024×1024. They found that models which use a larger patch size achieve better performance than models which use a smaller patch size, however this does come at the cost of computational load. Lastly, we have seen a novel approach which is based on k-means clustering of patch features to create a bag-level representation.

From these MIL-based and context-aware models, different feature extraction strategies were used. A common feature extractor that we have seen is a pre-trained model on ImageNet, but we have also seen feature extractors based on supervised learning, weakly-supervised learning, unsupervised learning and self-supervised learning. We saw two main ways to quantify the performance of these feature extractors, which are based on performance metrics (such as accuracy or F1-score) and feature similarity based on spatial proximity.

Chapter 4

Materials and Methods

This chapter elaborates on the datasets and methods used for this project.

4.1 Materials

4.1.1 Young Breast Cancer Patients Dataset

The dataset for training and validation of the model consists of 706 slides and there is an independent test set of 686 slides, see Table 4.1. We use the same data split as in the work by Wetstein et al. [12] and also combine the low/intermediate class with each other. Furthermore, we only consider cases of invasive ductal carcinoma.

Dataset	**Low/Intermediate**	**High**
Train/validation	357	349
Test	327	359
Total	684	708

Table 4.1: Number of slides for the train/validation set and test set for the YBCP dataset

Pathologists used the Nottingham modification of the Bloom-Richardson system [56] to grade the slides. The factors that determine the breast cancer grade is the level of nuclear atypia, the mitotic count and tubule formation. An overall grade is assigned to the slide based on the summation of the sores of these three factors, which is either Grade 1, 2 and 3. In our dataset, Grade 1, 2 and 3 are labeled *low*, *intermediate* and *high*-grade respectively.

4.1.2 Camelyon16 Dataset

The Camelyon16 dataset is collected from the Radboud University Medical Center (RUMC) and the UMCU, where the train/validation set contains 270 images and the independent test set contains 129 images, see Table 4.2. For the train/validation set, we have 160 slides with no metastases and 110 slides with metastases, and for the test set we have 80 slides with no metastases and 49 with metastases.

The cases containing metastatic lymph nodes are further divided into macro- and micrometastases, where macrometastases has a tumor cell cluster $\geq$ 2mm and micrometastases has a tumor cell cluster between 0.2mm and 2mm. For the Camelyon16 challenge, the cases containing macro- or micrometastases are combined into a single group, thus making the problem a binary classification problem with the classes *Normal* and *Metastases*.

Dataset	Hospital	No metastases	Macro metastases	Micro metastases
Train/Validation	RUMC	100	35	35
	UMCU	60	26	14
Test	RUMC	50	14	15
	UMCU	30	8	12
Total		240	83	76

Table 4.2: The number of slides collected per institution and their labels.

4.1.3 Melanoma Dataset

The melanoma dataset contains 505 WSI sampled from 225 patients, see Table 4.3 for the data distribution. These slides were scanned during the period of September 2018 till February 2021 at the University Medical Center Utrecht (UMCU) in the Netherlands.

	Pathway I	Pathway IV	Remaining
Count	245	215	43

Table 4.3: Number of slides per pathway and their diagnosis for the melanoma dataset

4.2 Methods

In this section we explain the different variations of CLAM and elaborate on our approach of NIC. We chose CLAM because it is highly interpretable due to the attention mechanism and we chose NIC since this approach could in theory capture local features from patches and combine these features on a more global level, which other methods often lack.

4.2.1 CLAM and MIL

The code for CLAM comes from a public GitHub repository and is available at `https://github.com/mahmoodlab/CLAM`. The CLAM workflow consists of WSI tissue segmentation, WSI patching, patch feature extraction and model training.

4.2.1.1 Tissue segmentation and patching

For each WSI in the dataset, CLAM segments tissue from non-tissue area. CLAM computes the segmentation mask on a downscaled version of the WSI, since this is more efficient in terms of memory. Firstly, CLAM converts the WSI from RGB to HSV, then blurs the image and creates a binary mask by thresholding the saturation channel. The binary mask serves as the segmentation mask. After the segmentation step, CLAM can extract tissue patches from the WSI. The segmentation and patching step can be seen in Figure 4.1.

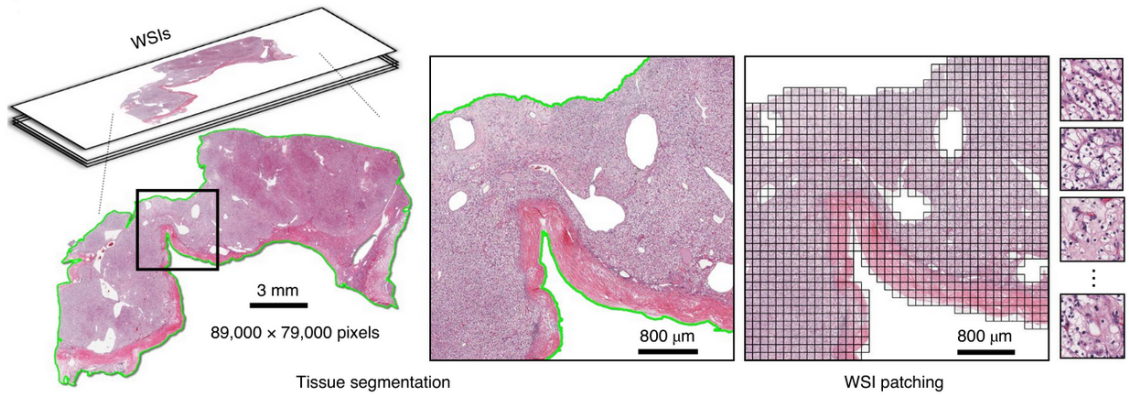


Figure 4.1: CLAM segments tissue from non-tissue area and extracts patches from the WSI. From [30].

4.2.1.2 Feature extraction

After localizing the patches for every slide, CLAM extracts features from these patches by feeding patches of size $3\times P\times P$ through a ResNet-50 [21] pre-trained on ImageNet [5]. The dimensionality reduction serves two purposes: (I) extracting important features from patches and (II) reduce computational load.

As an alternative feature extractor, we use SimCLR. SimCLR is a self-supervised method which aims to learn representations of the dataset using contrastive learning [36]. We choose SimCLR since it has been shown that this self-supervised method performed better than a pre-trained network in various works [35; 54]. Furthermore, this method is one of the *state-of-the-art* methods in self-supervised learning [36].

The SimCLR model has two different types of input; pairs of patches that are from the same location, or pairs of patches that are different from different locations. In the first scenario, SimCLR applies two random data augmentations on a patch x, which creates two different images $\hat{x}_i$ and $\hat{x}_j$. In the second scenario, two different patches are augmented. The data augmentations

include cropping, resizing, color distortion (drop and jitter), rotation (90°, 180° and 270°), cutouts, Gaussian blurring and sobel filter.

Therafter, SimCLR extracts features from both augmented images with a function $f(\cdot)$, which is a RESNET [21]. Next, the model inputs these representation into a projection head $g(\cdot)$. The outputs of the projection head are used to calculate the loss. The procedure of this process can be seen in Figure 4.2.

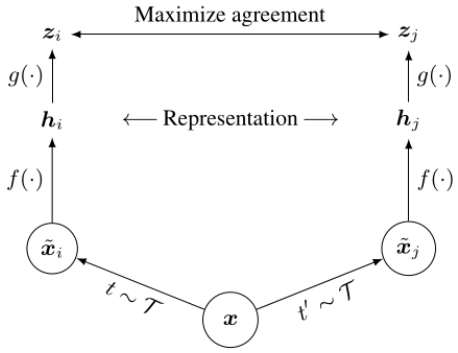


Figure 4.2: SimCLR augments a patch x, extracts a representation from it and at the end passes it through a projection head to calculate the loss. From [36]

The loss function that SimCLR uses is called NT-Xent, which tries to maximize agreement between pairs of patches that are from the same location, with respect to pairs of patches from different locations. The loss of a positive pair is $l(i, j)$ is defined as:

$$l_{i,j} = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(z_i, z_k)/\tau)} \tag{4.1}$$

For every epoch, SimCLR samples a mini-batch of size N and augments every patch, which results into $2N$ augmented patches. Given the augmented set, we take the cross product of G and calculate a similarity measure between every pair.

We calculate the total loss of a single mini batch by averaging the loss of all positive pairs, see Equation 4.2.

$$\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} [l(2k-1, 2k) + l(2k, 2k-1)] \tag{4.2}$$

After training the SimCLR model, we remove the projection head g and keep the feature extractor f as patch encoder.

4.2.1.3 Model architectures

There are two main architectures within CLAM, which are CLAM single-branch (SB) and CLAM multi-branch (MB). Aside from the CLAM models, the authors also provide two MIL approaches, which are binary-MIL and n-arity MIL. Both MIL-versions come with max-pooling and our own average-pooling implementation.

The architecture of the CLAM SB model can be seen in Figure 4.3. The input of the model is a bag of feature vectors, in the diagram this bag is called x, the number of patches in the bag is denoted as b and 1024 is the length of each feature vector. x passes through a fully connected linear layer with a *relu*-activation function with 512 neurons and dropout of 25%, which produces h. Thereafter, the model feeds h into the gated attention layer, which is marked in blue. h passes through separate layers in the gated attention layer, which produce a and b. These layers only differ in activation function, where the layer that produces a uses a *tanh*-activation function, whereas the layer that produces b uses a *sigmoid* activation function. Next, a pairwise multiplication of a and b produces c, which is then fed to a fully connected linear layer with a *softmax*-activation to produce A, which can be seen as the attention score per feature vector in x. An entry a_k in A is formalized in Equation 3.3.

After the gated attention layer, we apply a matrix multiplication of A' and h to produce M, where M can be seen as the vector containing the features that are important according to the attention scores of the patches. The model feeds M into a fully connected layer with N_CLASSES as number of neurons to produce the raw logits of the model. We obtain the class probabilities by applying a softmax layer on the logits.

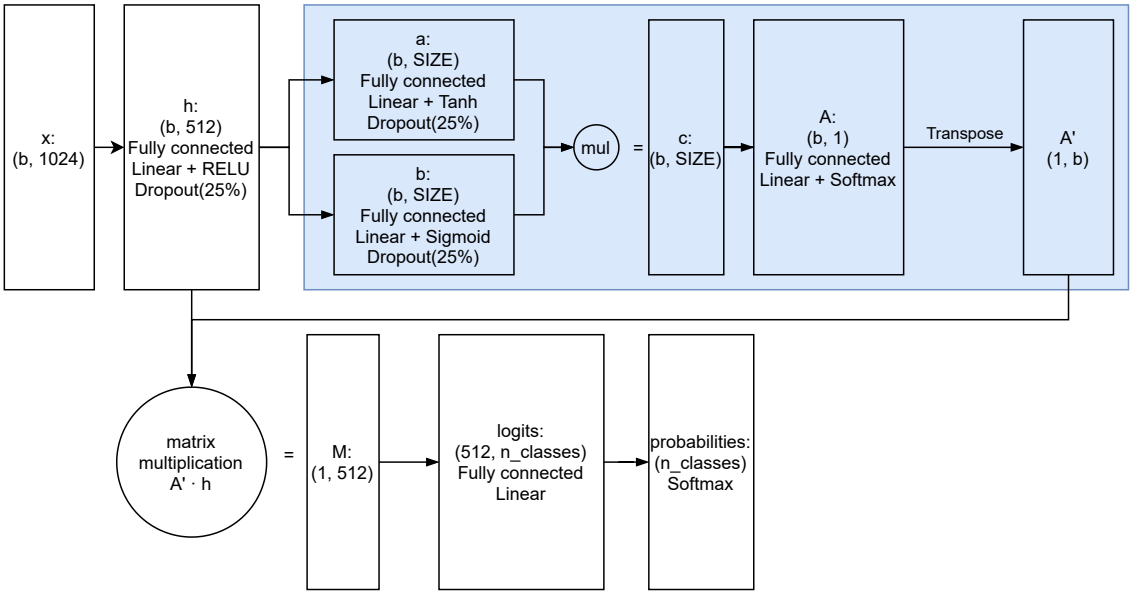


Figure 4.3: CLAM SB architecture

The CLAM MB architecture can be seen in Figure 4.4. A difference between the CLAM MB model and the CLAM SB model is that the CLAM MB produces attention scores for each class, whereas CLAM SB produces only a single attention vector. The logits are also calculated differently. Given M of size (N_CLASSES, 512), we feed each row its own fully connected layer to produce a single entry in the logits vector. Similarly to the CLAM SB model, we obtain the class probabilities by applying a softmax layer on the logits.

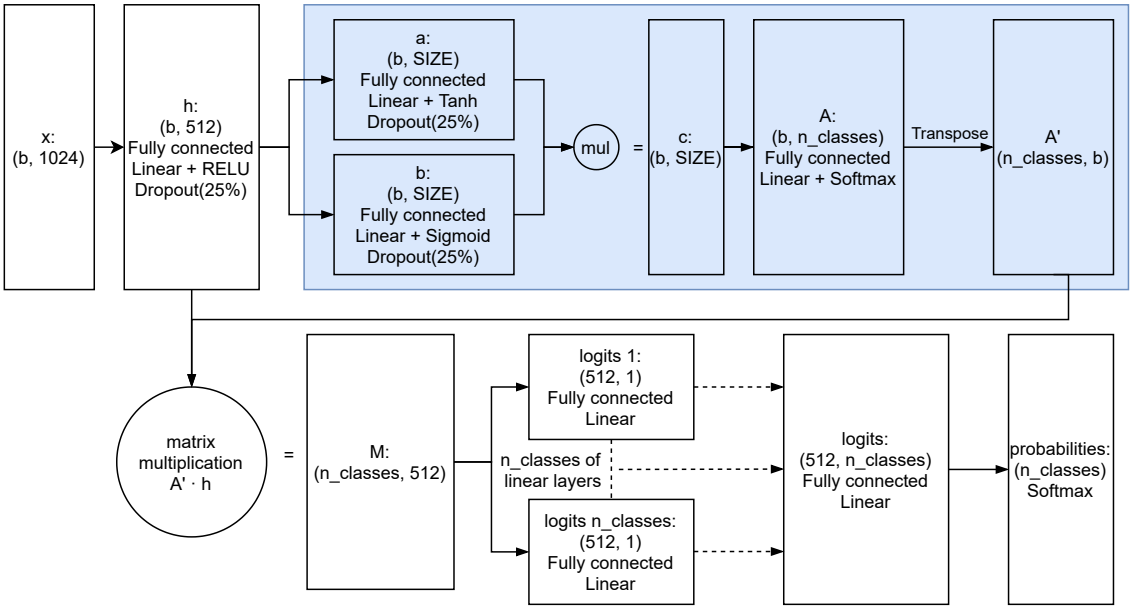


Figure 4.4: CLAM MB architecture

The third and fourth architecture are the binary MIL-architecture and the n-arity MIL-architecture, which can be seen in Figures 4.5 and 4.6. Both models produce a matrix h, which represents the features for each instance of the WSI. The raw logits of the models are calculated differently, the binary MIL-model creates logits by passing forward h to a linear layer with 2 neurons, whereas the n-arity MIL-model has n linear layers of 1 neuron to produce the logits matrix. Thereafter, both models use a softmax layer to calculate the probability scores for each instance in the bag. Finally, to obtain the class prediction, we apply max-pooling over the predictions of the whole set of instances within that bag. The main difference between the CLAM and MIL architecture is that CLAM creates a slide-level embedding M to make the slide prediction, whereas MIL derives its prediction from individual patch-level predictions (with either max-pooling or average pooling in our case).

We extend the MIL-method by allowing an average pooling operation over the top-k most probable patches. In the case of max-pooling, the model is more susceptible for false positive patches, whereas in average-pooling we try to reduce noise of the supervisory signal by taking into account the top-k patches. We incorporate domain knowledge into the pooling operation, since we know that a slide that contains tumor has more than 1 patch containing tumor, hence max-pooling would not be sufficient. By taking into account a larger context (top-k patches), we make a more representative WSI prediction. Naturally, we try to seek the optimal k, such that the noise in the supervisory signal is reduced, as well as keeping this supervisory signal strong.

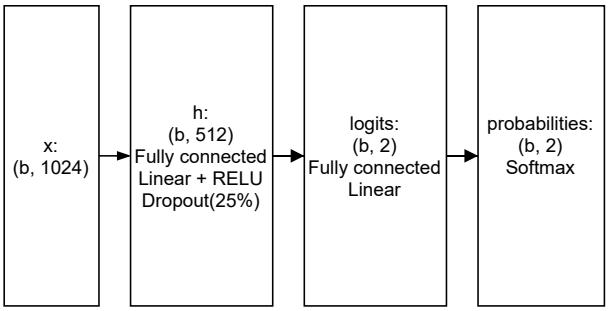


Figure 4.5: Binary MIL architecture

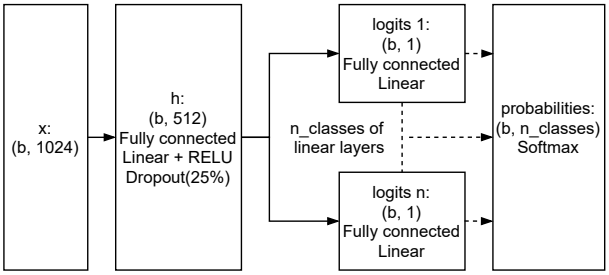


Figure 4.6: n-arity MIL architecture

4.2.1.4 Instance-level clustering

CLAM provides an instance-level clustering feature, which constraints the feature space by clustering lowly attended patches against highly attended patches. By constraining the feature space, the model can better differentiate between important and less important patches, and hence the performance of the model increases.

4.2.1.5 Visualization of the attention scores

The multi- and single-attention branch models are capable of generating attention heatmaps of a WSI. These attention heatmaps visualize the attention score per patch of a WSI, where a high attention score indicates great importance of the patch for the slide-level prediction, and a low attention score indicates that the patch was not of importance for the prediction. To make the attention heatmap fine-grained, CLAM chooses a high overlap percentage between patches and averages the attention scores in regions.

4.2.2 NIC

As already discussed in Section 3.3.2, NIC is a framework which compresses a WSI x into x' and consequently classifies x'. However, there are some slight nuances to our implementation, as x is variable in resolution, we classify fixed size compressed tissue regions of a slide, and in the end aggregate the results by averaging the predictions. In our case, a tissue region can be considered a grid of patches of size $M \times N \times P \times P \times 3$, where M is the number of rows, N is the number of columns and P is the patch size. Furthermore, a compressed tissue region has dimensions $M \times N \times C$, where C is the encoder length.

Firstly, NIC trains a patch encoder, which is extracts important features from a tissue patch. Secondly, for each WSI, NIC proposes tissue regions, augments these regions and compresses them. Lastly, NIC uses the compressed tissue regions to train the WSI classifier.

4.2.2.1 Patch encoder

The goal of the encoder is to reduce the dimensionality of a patch and extract features from it. Given a patch of size $3 \times P \times P$, the encoder compresses the patch to an embedding vector of size C, where P is the patch width/height and C is the encoder length.

We implement an adaptation of the contrastive model presented in the NIC paper. The dataset for the contrastive model consists of three kinds of pairs of patches; pairs from the exact same location, pairs from different locations that are close to each other and pairs from different locations that are located far away from each other. In the second and third case, the method calculates the distance between patches using the euclidean distance between the coordinates of the centers of those patches.

The goal of the contrastive model is to distinguish the three different classes. This dataset construction deviates from the one described in the NIC paper, where originally they only have two kinds of pairs; pairs from the same location and pairs from different locations. The intuition behind our dataset construction is that the model should extract more discriminative features from the patches, since the task has become harder. We argue that patches that are close to each other look similar, while patches that are far away from each other do not look similar. This principle is based on spatial contintuity, similarly to [55].

Figure 4.7 describes a single forward pass of the model. After training the contrastive model, NIC uses the feature extractor $f(\cdot)$ as an encoder and throws away the head $g(\cdot)$.

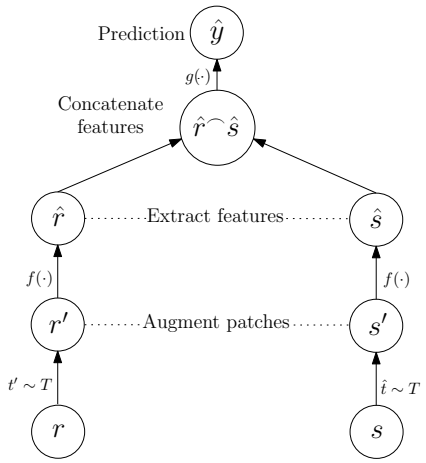


Figure 4.7: The method applies data augmentation sampled from T on patches r and s. From those augmented patches, the model extract features using $f(\cdot)$ and concatenates them. Finally, the model does a forward pass of the concatenated features using $g(\cdot)$, which gives a prediction $\hat{y}$.

4.2.2.2 Tissue region localization, augmentation and compression

For each WSI, we apply a region extraction algorithm. Given a WSI, we first segment tissue from non-tissue area using Otsu's method [57]. To reduce the computational load of the segmentation step, we segment the WSI on a downscaled version. Thereafter, we pick parameters W and H, which indicate the width and the height of the tissue region box. Since we know the tissue borders of the WSI, we can apply a sliding window approach to extract tissue regions of the slide. This sliding window approach places a bounding box on the tissue area and slides it horizontally and vertically, thereby extracting all tissue regions of a tissue area. See Figure 4.8 for an example of the tissue region proposals on a WSI.

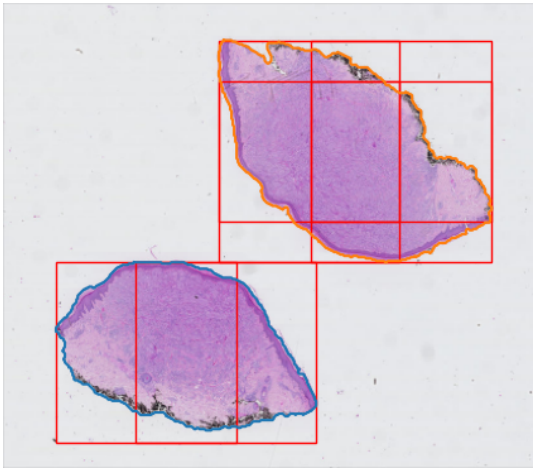


Figure 4.8: Tissue region proposals of a WSI. Note that blue and orange borders are drawn around the tissue area, which indicate the output of the segmentation. The red borders indicate the fixed size tissue region boxes. As can be seen in this example, overlap between tissue regions may occur

After localizing the tissue regions, we augment and compress the tissue regions. We extend the work of NIC [38] with their data augmentation approach, since their data augmentation is limited to 90-degree rotations and mirroring of tissue regions. Our data augmentation on the other hand includes arbitrary angle rotation, horizontal/vertical flipping, Gaussian blurring, brightness change and color jitter. For each patch in the grid/tissue region, we apply the random augmentations and compress the patch using the encoder as mentioned in 4.2.2.1. See Appendix 6.1 for the pseudocode for WSI compression.

4.2.2.3 Offline versus online data augmentation/compression of tissue regions

Data augmentation/compression on tissue regions can be applied either offline or online. In an offline setting, NIC compresses the tissue regions n times before training. The reason why offline data augmentation/compression is a viable option is due to the large image resolution of WSI. Augmenting and compressing the tissue regions for every epoch is not feasible due to computational load and training time, especially in high magnifications such as $40\times$.

However, data augmentation/compression of tissue regions can also be applied online, during every epoch. A prerequisite of applying online data augmentation/compression is that the dimensionality of the data (number of patches in a slide) is small. This is the case when we apply NIC on a smaller magnification level such as $10\times$. We would then need far less operations to compress a single tissue region.

4.2.2.4 NIC classifier

Given a single WSI, the NIC classifier takes as input m randomly sampled compressed tissue regions and for each tissue region predicts the class. The final slide-level prediction is the average

of all the tissue region prediction probabilities.

The NIC classifier consists of 5 blocks of strided convolutional layers of 128 filters with a kernel size of 3×3, batch normalization layers, leaky relu activation layers, dropout layers with a probability of 20% dropout. These blocks are followed by a dense layer of 128 neurons with batch normalization and leaky relu, and in the end a classification head. This architecture is based on the architecture presented in the Supplementary work of NIC [38].

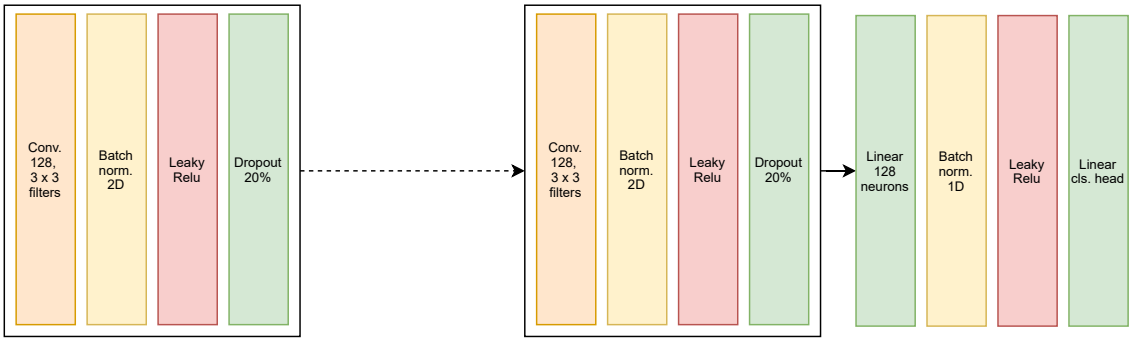


Figure 4.9: NIC classifier architecture

Chapter 5

Evaluation

In this chapter we elaborate on the results of the experiments on breast cancer grade classification, lymph node metastases detection and melanoma pathway classification. We evaluated the performance of the models using the AUC metric, accuracy and F1-Score, which are common metrics for classification tasks.

We ran experiments on the High-Performance-Cluster (HPC) of the UMCU, which has dedicated GPUs available that were randomly assigned to the jobs that were submitted. These GPUs include Nvidia RTX2080TIs, Tesla V100s and RTX6000s.

5.1 Grade Classification for Young Breast Cancer Patients

For the YBCP dataset, we trained our model on 5 different train/validation folds and evaluate the model on an independent test set. The ratio of slides for the training and validation sets were 80% and 20% respectively.

Our experiments included models using a magnification of 10× and 20× with the default feature extractor[1] and models using a magnification of 10× with 3 custom SimCLR feature extractors. As mentioned in [36], SimCLR benefits from a large model, a large batch size and many epochs. The SimCLR models that we evaluated are the SimCLR models based on ResNet-18 with a batch size of 128 running for 100 epochs, ResNet-18 with a batch size of 300 running for 100 epochs and a ResNet-50 with a batch size of 128 running for 100 epochs. Due to time constraints, we did not train feature extractors with more training epochs, and for the SimCLR MIL Top-k average pooling models, we only considered the model with $k = 50$. We chose $k = 50$, since for this k, the performance was best for the models which use the default feature extractor at 10× magnification. We used 15% of the training set for the SimCLR models and made sure that within each training fold, the 15% for the SimCLR models is contained within the training fold. We do not consider this strategy as data leakage, as the SimCLR model uses a subset of the training fold of the classification models. Hence, there is no data leakage from the SimCLR model to the validation and test set of the classification model. We applied this strategy, since we wanted to compare different classification models under the same training fold. Another strategy that could have been chosen was that 15% of the training set was set aside only for the SimCLR models to use, however in this strategy the classification models would train under different folds, which is not ideal.

For the experiments done on 20× magnification, we tried various MIL top-k with average pooling models with larger numbers of k compared to the experiments done at 10× magnification, since a patch at 20× magnification fits 4 times in a patch of 10× magnification of the same image resolution. Aside from the AUC, accuracy and F1-score, we also report the accuracy of each specific grade class (low, intermediate and high) separately. We report these accuracies to gain a better understanding of what our models find easy/difficult to classify.

[1] The default feature extractor is a ResNet-50 pre-trained on ImageNet

5.1.1 Results

Model	AUC	Accuracy	F1-Score	Accuracy (Low)	Accuracy (Intermediate)	Accuracy (High)
CLAM SB - Small	0.842 ± 0.01	0.762 ± 0.01	0.778 ± 0.01	0.861 ± 0.04	0.668 ± 0.04	0.799 ± 0.03
CLAM SB - Big	0.834 ± 0.01	0.749 ± 0.01	0.762 ± 0.02	0.865 ± 0.03	0.674 ± 0.06	0.770 ± 0.05
CLAM MB - Small	0.834 ± 0.01	0.756 ± 0.01	0.763 ± 0.00	0.876 ± 0.02	0.650 ± 0.02	0.797 ± 0.01
CLAM MB - Big	0.840 ± 0.01	0.763 ± 0.00	0.783 ± 0.01	0.870 ± 0.03	0.640 ± 0.03	**0.817 ± 0.02**
MIL Max-pooling	0.806 ± 0.04	0.731 ± 0.03	0.774 ± 0.02	0.827 ± 0.04	0.642 ± 0.08	0.767 ± 0.03
MIL Top-50 avg-pooling	**0.858 ± 0.00**	**0.785 ± 0.01**	0.796 ± 0.02	0.872 ± 0.01	**0.721 ± 0.04**	0.805 ± 0.04
MIL Top-75 avg-pooling	0.857 ± 0.00	0.784 ± 0.01	**0.797 ± 0.01**	**0.876 ± 0.03**	0.706 ± 0.04	0.812 ± 0.03
MIL Top-100 avg-pooling	0.857 ± 0.00	0.779 ± 0.01	0.793 ± 0.01	0.874 ± 0.03	0.701 ± 0.03	0.808 ± 0.03

Table 5.1: 10× Magnification models using a ResNet-50 pre-trained on ImageNet

Model	AUC	Accuracy	F1-Score	Accuracy (Low)	Accuracy (Intermediate)	Accuracy (High)
CLAM SB - Small	0.848 ± 0.01	0.775 ± 0.01	0.793 ± 0.02	0.879 ± 0.02	0.659 ± 0.05	0.826 ± 0.04
CLAM SB - Big	0.849 ± 0.00	0.768 ± 0.00	0.783 ± 0.01	0.885 ± 0.02	0.673 ± 0.04	0.802 ± 0.04
CLAM MB - Small	0.844 ± 0.01	0.778 ± 0.01	0.796 ± 0.01	0.874 ± 0.03	0.667 ± 0.07	0.827 ± 0.04
CLAM MB - Big	0.850 ± 0.01	0.790 ± 0.01	0.807 ± 0.01	0.892 ± 0.02	0.679 ± 0.03	0.838 ± 0.02
MIL Max-pooling	0.731 ± 0.06	0.671 ± 0.06	0.687 ± 0.07	0.789 ± 0.07	0.578 ± 0.10	0.703 ± 0.11
MIL Top-150 avg-pooling	0.859 ± 0.01	0.799 ± 0.01	0.814 ± 0.01	0.906 ± 0.02	0.696 ± 0.01	**0.842 ± 0.03**
MIL Top-200 avg-pooling	0.860 ± 0.01	0.801 ± 0.01	0.815 ± 0.01	0.912 ± 0.00	0.696 ± 0.02	**0.842 ± 0.02**
MIL Top-250 avg-pooling	0.864 ± 0.01	**0.803 ± 0.01**	0.816 ± 0.01	0.903 ± 0.02	0.711 ± 0.02	0.838 ± 0.02
MIL Top-300 avg-pooling	0.865 ± 0.00	0.797 ± 0.01	0.807 ± 0.02	0.915 ± 0.01	0.731 ± 0.01	0.812 ± 0.03
MIL Top-350 avg-pooling	0.866 ± 0.00	0.799 ± 0.01	0.809 ± 0.01	**0.917 ± 0.01**	**0.734 ± 0.01**	0.813 ± 0.02
MIL Top-400 avg-pooling	**0.869 ± 0.01**	0.802 ± 0.01	**0.817 ± 0.01**	0.901 ± 0.02	0.704 ± 0.04	**0.842 ± 0.03**

Table 5.2: 20× Magnification models using a ResNet-50 pre-trained on ImageNet

Model	AUC	Accuracy	F1-Score	Accuracy (Low)	Accuracy (Intermediate)	Accuracy (High)
CLAM SB - Small	0.712 ± 0.01	**0.673 ± 0.01**	0.685 ± 0.01	**0.825 ± 0.02**	**0.604 ± 0.03**	0.681 ± 0.03
CLAM SB - Big	0.712 ± 0.01	0.665 ± 0.00	0.684 ± 0.02	0.784 ± 0.03	0.576 ± 0.06	0.695 ± 0.05
CLAM MB - Small	0.714 ± 0.01	0.663 ± 0.01	0.682 ± 0.00	0.782 ± 0.06	0.574 ± 0.07	0.692 ± 0.05
CLAM MB - Big	0.723 ± 0.00	0.667 ± 0.00	0.692 ± 0.01	0.820 ± 0.05	0.592 ± 0.06	0.697 ± 0.05
MIL Max-pooling	0.674 ± 0.01	0.631 ± 0.01	0.657 ± 0.04	0.654 ± 0.10	0.537 ± 0.12	0.689 ± 0.11
MIL Top-50 avg-pooling	**0.731 ± 0.01**	**0.673 ± 0.01**	**0.701 ± 0.02**	0.769 ± 0.03	0.543 ± 0.04	**0.735 ± 0.05**

Table 5.3: 10× Magnification models using a SimCLR feature extractor with a ResNet-18 backbone ran trained with a batch-size of 128 for 100 epochs

Model	AUC	Accuracy	F1-Score	Accuracy (Low)	Accuracy (Intermediate)	Accuracy (High)
CLAM SB - Small	0.704 ± 0.01	0.654 ± 0.01	0.676 ± 0.02	0.755 ± 0.05	0.559 ± 0.05	0.691 ± 0.05
CLAM SB - Big	0.705 ± 0.01	0.651 ± 0.01	0.677 ± 0.01	0.744 ± 0.03	0.543 ± 0.03	0.700 ± 0.03
CLAM MB - Small	0.705 ± 0.01	0.647 ± 0.01	0.669 ± 0.02	0.751 ± 0.06	0.552 ± 0.08	0.684 ± 0.06
CLAM MB - Big	0.706 ± 0.00	0.645 ± 0.00	0.666 ± 0.02	0.751 ± 0.07	0.555 ± 0.09	0.679 ± 0.06
MIL Max-pooling	0.676 ± 0.01	0.609 ± 0.03	0.633 ± 0.05	0.600 ± 0.26	0.524 ± 0.25	0.668 ± 0.18
MIL Top-50 avg-pooling	**0.735 ± 0.01**	**0.683 ± 0.01**	**0.700 ± 0.02**	**0.827 ± 0.02**	**0.591 ± 0.04**	**0.709 ± 0.04**

Table 5.4: 10× Magnification models using a SimCLR feature extractor with a ResNet-18 backbone ran trained with a batch-size of 300 for 100 epochs .

Model	AUC	Accuracy	F1-Score	Accuracy (Low)	Accuracy (Intermediate)	Accuracy (High)
CLAM SB - Small	0.703 ± 0.01	0.656 ± 0.01	0.675 ± 0.02	0.744 ± 0.07	0.578 ± 0.05	0.685 ± 0.04
CLAM SB - Big	0.703 ± 0.01	0.654 ± 0.01	0.675 ± 0.02	0.737 ± 0.09	0.567 ± 0.07	0.691 ± 0.06
CLAM MB - Small	0.704 ± 0.02	0.650 ± 0.01	0.688 ± 0.01	0.665 ± 0.06	0.513 ± 0.05	0.738 ± 0.03
CLAM MB - Big	0.720 ± 0.01	0.657 ± 0.02	**0.695 ± 0.01**	0.681 ± 0.10	0.514 ± 0.07	**0.747 ± 0.04**
MIL Max-pooling	0.652 ± 0.01	0.595 ± 0.04	0.622 ± 0.08	0.616 ± 0.25	0.466 ± 0.21	0.676 ± 0.20
MIL Top-50 avg-pooling	**0.742 ± 0.01**	**0.670 ± 0.01**	0.687 ± 0.03	**0.780 ± 0.08**	**0.588 ± 0.07**	0.697 ± 0.07

Table 5.5: 10× Magnification models using a SimCLR feature extractor with a ResNet-50 backbone ran trained with a batch-size of 128 for 100 epochs

Results that are marked bold are the highest performing with respect to their experiment and results that are colored red are the highest performing over all the experiments.

We noticed that the models which use a ResNet-50 pre-trained on ImageNet as the feature extractor outperformed the models which use a variant of SimCLR as the feature extractor. On both 10× as well as 20× magnification, our MIL top-k average-pooling method beat the CLAM and MIL max-pooling methods in terms of AUC, accuracy and F1-Score. Furthermore, we see an increase in these classification metrics if we go from 10× magnification to 20× magnification, mainly in the *low* and *high*-grade accuracy. Furthermore, we outperform the model of Wetstein et al. [12], where they achieved for the exact same task an accuracy of 0.77 ± 0.05 on the test set.

We noticed that there was not much difference in model performance in the CLAM models which used the SimCLR variants as feature extractor, see Tables 5.3, 5.4 and 5.5. In [36], the authors state that contrastive learning benefits from larger models and a larger batch size. However, in our case we did not experience these claims, which may be caused by that our increase in model- and batch-size were insignificant. In [36], the authors experiment with batch sizes ranging from 256 all the way to 8192 and with models which have up till roughly 400 million network parameters, which is computationally infeasible in our case. Another aspect that could have impacted the performance of the SimCLR feature extractors is the imbalance in healthy and tumor patches in the dataset.

In general, the difference in model performance of 'small' models and 'big' models was unnoticeable, which implies that the 'small' model-variants have enough model-capacity to learn. Therefore, we favor the small model over the big model versions, since a smaller model requires less training/inference time and is less likely to overfit. We also do not see any noticeable difference between the SB and MB variants of CLAM, even though the MB models should be more expressive, since they assign attention scores to a patch per class, whereas the SB version only assigns a single attention score.

A pattern that we see in the results of the models is that the max-pooling MIL-models performed the worst. The reason why this might be the case is that tubule formation can not be captured in a single patch, which is an important factor for determining the breast cancer grade. The CLAM and MIL with average pooling models did not suffer from this problem, as their predictions depended on a number of patches, thus taking into account a larger context of the WSI. We have not found a clear explanation why MIL with average pooling performed better than the CLAM models, as in theory the CLAM models would be more flexible than the average pooling MIL method. The CLAM models assigns attention scores to every patch, which indicate the patch importance, whereas average pooling uses a fixed k for the k most probable patches.

Lastly, we noticed that the most misclassified cases belong to the *intermediate*-grade within our *low/intermediate* class along all the different models. The reason for this is that the *intermediate*-grade is morphologically not as well defined as *low* or *high*-grade, since *intermediate*-grade is in between those grades. Due to the low inter-class variability of the *intermediate*-grade between the *low* and *high*-grades, it is difficult to train the model and hence the dataset construction is a limiting factor in model performance.

5.1.2 Visualization of attention scores

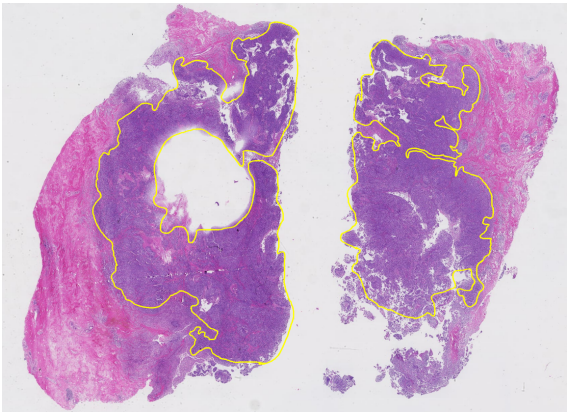
We use the CLAM model that has the highest accuracy on 10× magnification, due to computational efficiency, see Table 5.1. See Section 4.2.1.5 for an explanation of the attention scores. Furthermore, we compare the heatmaps with the annotations of the pathologists to check for correspondence. The examples given in this section are from slides from the test set.

Normal
Tumor

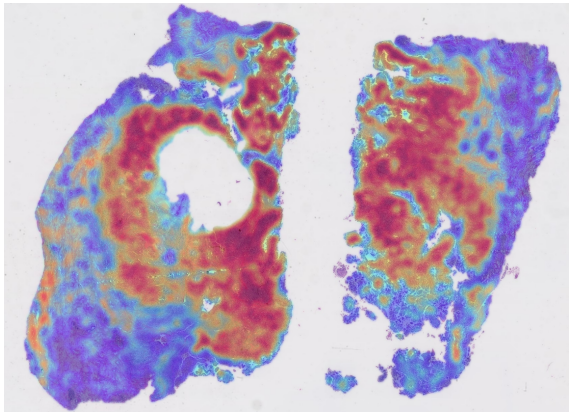
Table 5.6: Annotation color legend

5.1.2.1 High correspondence cases

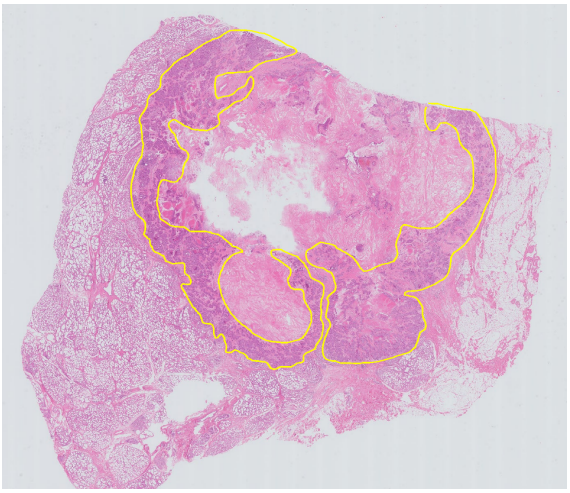
Figure 5.1 shows 2 cases of *high*-grade tumor, in which there is a high correspondence of ROI between the pathologist and the model. This indicates that the model has an understanding of what *high-grade* tumor tissue looks like.



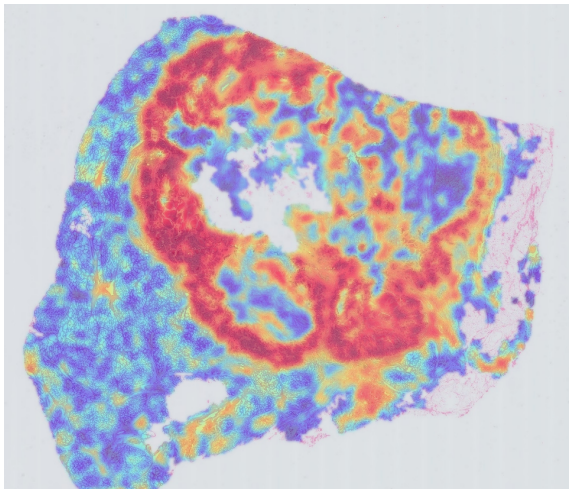
(a) Annotated WSI



(b) Attention heatmap of Figure 5.1a.



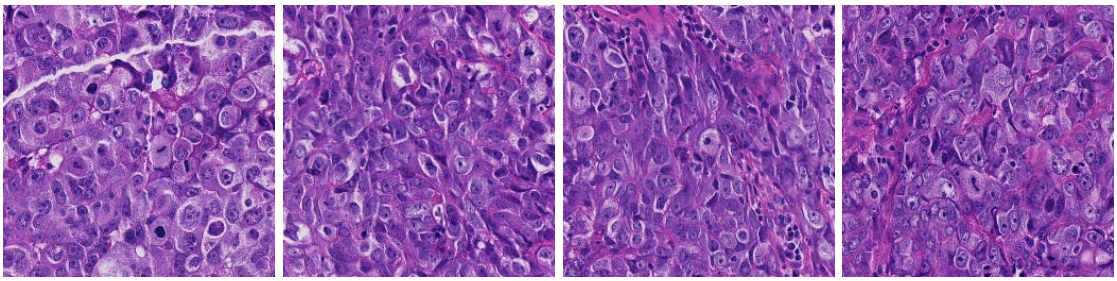
(c) Annotated WSI



(d) Attention heatmap of Figure 5.1c.

Figure 5.1: Two *high-grade* cases which contain high attention scores in the annotated area

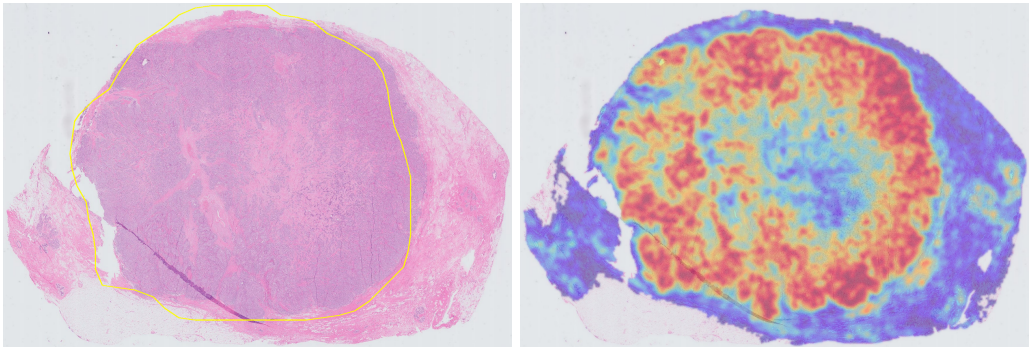
To examine the malignant areas of the slide in Figure 5.1a we extracted the top-4 patches with the highest attention score of the WSI prediction, see Figure 5.2. In these patches, we see low-levels of tubule formation, prominent nucleoli and a high mitotic count, which are all factors of *high-grade* tumor.



(a) Patch 1 (b) Patch 2 (c) Patch 3 (d) Patch 4

Figure 5.2: The top-4 patches with highest attention scores.

Figures 5.3a and 5.3b show a *low/intermediate grade* case in which the model captured the healthy tissue in the center of the tissue well. Annotating a WSI is a very time-consuming task for pathologists, therefore they often make a rough annotation of the ROI.

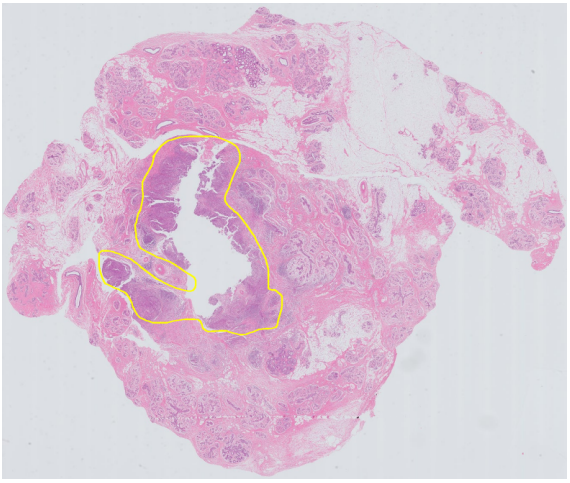


(a) Annotated WSI (b) Attention heatmap of Figure 5.3a

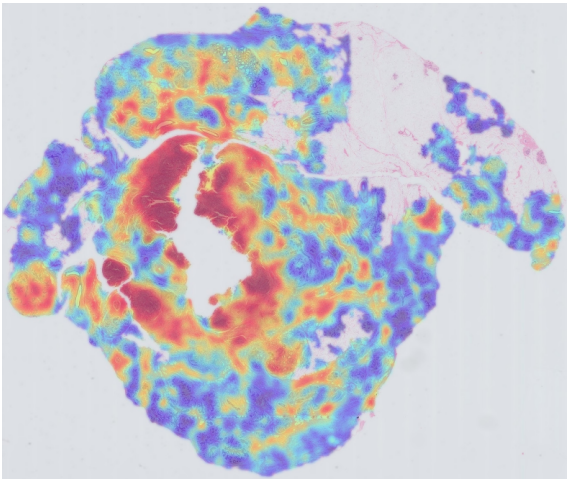
Figure 5.3: An example case where the attention heatmap detects healthy tissue within the annotated area of the pathologist

5.1.2.2 High correspondence with moderate attention outside of the ROI

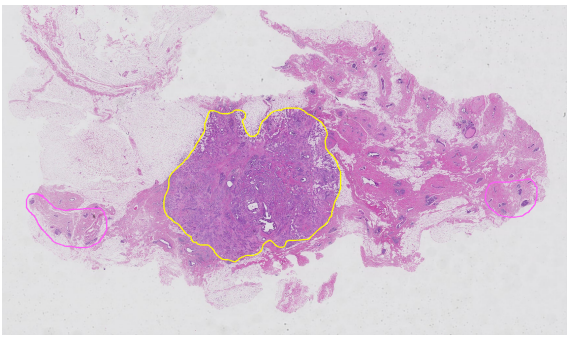
Figures 5.4a and 5.4c show a *high-grade* and a *low/intermediate grade* WSI respectively. In both WSI, we can see that the model captures the tumor in the annotated area, however there are areas outside of the annotation which contain healthy tissue that seems to be of moderate attention as well. This implies that the model could assign moderate attention scores to areas which are not relevant for the diagnosis. The heatmaps of these specific instances are still usable by a pathologist, since there is a clear distinction between high attention and moderate attention.



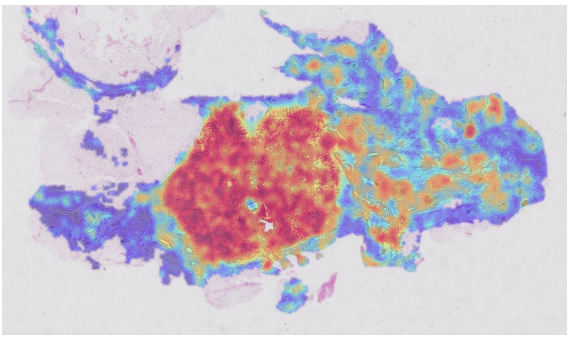
(a) Annotated WSI with *high-grade* tumor



(b) Attention heatmap of Figure 5.4a.



(c) Annotated WSI with *low/intermediate grade* tumor

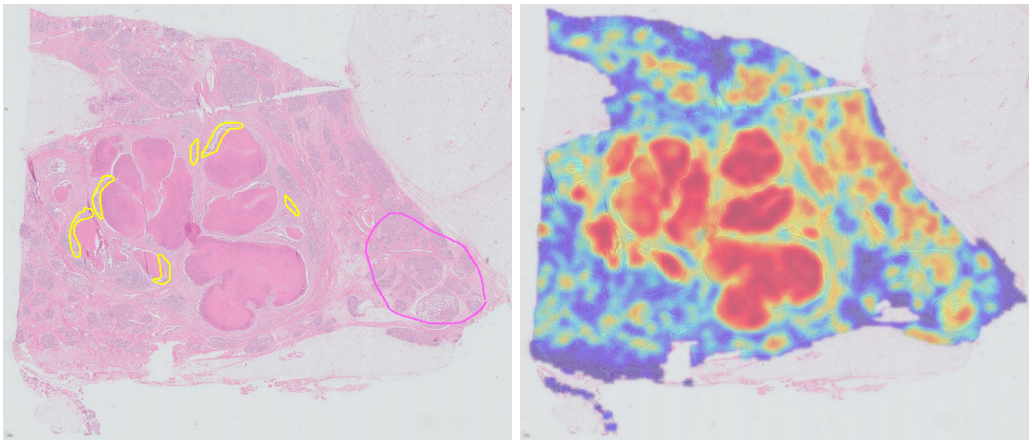


(d) Attention heatmap of Figure 5.4c.

Figure 5.4: An example case which contains high attention scores in the annotated area, but also outside of that area

5.1.2.3 Invalid cases

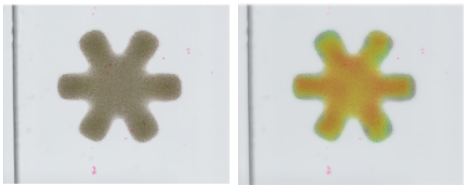
Figures 5.5a and 5.5b show a *high-grade* WSI where the model places high attention in areas that are not of importance, and places low attention in the annotated areas. This means that the model interpreted the slide completely different from the pathologist. Even though this discrepancy exists, the model and pathologist both gave the slide a diagnosis of *high-grade*.



(a) Annotated WSI (b) Attention heatmap of Figure 5.5a

Figure 5.5: An example case where the attention heatmap is completely different from the annotations of the pathologist

Figures 5.5a and 5.5b show a scanning artifact and its heatmap. During the process of obtaining a WSI, we can encounter scanning artifacts such as bubbles, cuts and other imperfections. In this specific case, the model gave high attention on the artifact, which is undesirable.



(a) Artifact (b) Attention heatmap

Figure 5.6: A scanning artifact that contains moderately high attention

5.1.2.4 Conclusions of the attention score visualization

We have shown scenarios in which the attention heatmaps corresponded well with the annotations that the pathologist made. However, there were cases for which the model predicted moderate attention in areas that were not of importance. Furthermore, in some instances the attention scores and annotations of the pathologist do not correspond at all, such as in scanning artifacts. The faulty attention heatmaps are likely a consequence of the sub-optimal accuracy of the model, which was $76.3\% \pm 0.0$.

We can conclude that it is important to know how the model interprets the data. Even though it might be the case that the model made a correct WSI prediction, the attention map of the WSI can still be invalid as has been shown in Figure 5.5. Understanding of the interpretations of the model leads to more confidence in what the model can and cannot do, and therefore is an important factor in employing these models in practice.

5.2 Detecting Lymph Node Metastases in Breast Cancer Patients

The main reason why we use CAMELYON16 is to validate our custom average pooling MIL-method, since this method was the best performing in the YBCP task, see Table 5.2. For the experiments, we used a pre-trained RESNET-50 on ImageNet as feature extractor, since the models that used this feature extractor in the breast cancer grade classification task achieved the best performance.

We performed three types of experiments; (I) 10× magnification, (II) 20× magnification and (III) 20× magnification with Otsu's thresholding [57]. Experiments I and II both use the default segmentation algorithm as described in Section 4.2.1.1, while experiment III uses Otsu's thresholding as tissue segmentation method. We performed experiment III since we noticed that the attention heatmaps of the CLAM models on experiment II would often have high attention in fat tissue, which was not ignored properly by the models. The masks produced by using Otsu's thresholding would often not include fat tissue, thus mitigating the problem. Moreover, we wanted to see the effects of using a higher magnification level, thus we performed experiments on 20× magnification.

The results of experiment I, II and III can be seen in Tables 5.7, 5.8 and 5.9 respectively. Aside from the AUC, accuracy, and F1-score, we also include the accuracy percentages of normal lymph nodes, macrometastases and micrometastases.

5.2.1 Results

Model	AUC	Accuracy	F1-Score	Accuracy (Normal)	Accuracy (Macrometastases)	Accuracy (Micrometastases)
CLAM SB - Small	0.747 ± 0.02	0.757 ± 0.03	0.613 ± 0.02	0.910 ± 0.05	**0.964 ± 0.02**	0.133 ± 0.04
CLAM SB - Big	**0.751 ± 0.01**	0.743 ± 0.02	0.595 ± 0.04	0.890 ± 0.05	0.936 ± 0.04	0.148 ± 0.10
CLAM MB - Small	0.750 ± 0.02	0.726 ± 0.03	0.578 ± 0.02	0.868 ± 0.07	0.936 ± 0.04	0.133 ± 0.06
CLAM MB - Big	0.743 ± 0.02	0.741 ± 0.02	0.581 ± 0.04	0.905 ± 0.03	0.918 ± 0.03	0.111 ± 0.05
MIL Max-pooling	0.713 ± 0.07	0.747 ± 0.09	0.622 ± 0.07	0.882 ± 0.14	0.873 ± 0.19	0.244 ± 0.14
MIL Top-2 avg-pooling	0.673 ± 0.08	0.684 ± 0.10	0.579 ± 0.08	0.765 ± 0.18	0.836 ± 0.20	**0.319 ± 0.17**
MIL Top-5 avg-pooling	0.725 ± 0.01	**0.767 ± 0.04**	**0.629 ± 0.03**	**0.922 ± 0.07**	0.955 ± 0.00	0.156 ± 0.05
MIL Top-10 avg-pooling	0.715 ±0.02	0.760 ±0.02	0.611 ±0.03	0.920 ± 0.03	0.955 ± 0.00	0.126 ± 0.07
MIL Top-50 avg-pooling	0.665 ±0.05	0.705 ±0.08	0.540 ±0.04	0.865 ± 0.15	0.818 ± 0.09	0.141 ± 0.15

Table 5.7: Results on CAMELYON16 test set using 10× magnification

Model	AUC	Accuracy	F1-Score	Accuracy (Normal)	Accuracy (Macrometastases)	Accuracy (Micrometastases)
CLAM SB - Small	0.814 ± 0.04	0.797 ± 0.02	0.691 ± 0.04	0.918 ± 0.04	0.945 ± 0.02	**0.319 ± 0.12**
CLAM SB - Big	**0.839 ± 0.02**	**0.803 ± 0.03**	**0.693 ± 0.04**	**0.932 ± 0.03**	**0.955 ± 0.00**	0.296 ± 0.12
CLAM MB - Small	0.805 ± 0.09	0.794 ± 0.03	0.677 ± 0.06	0.925 ± 0.02	**0.955 ± 0.00**	0.274 ± 0.16
CLAM MB - Big	0.817 ± 0.05	0.797 ± 0.02	0.690 ± 0.05	0.915 ± 0.03	**0.955 ± 0.00**	**0.319 ± 0.15**
MIL Max-pooling	0.678 ± 0.12	0.690 ± 0.11	0.478 ± 0.26	0.845 ± 0.17	0.609 ± 0.36	0.296 ± 0.16
MIL Top-2 avg-pooling	0.661 ± 0.07	0.665 ± 0.07	0.411 ± 0.21	0.847 ± 0.15	0.455 ± 0.32	0.296 ± 0.21
MIL Top-5 avg-pooling	0.639 ± 0.09	0.639 ± 0.08	0.397 ± 0.18	0.817 ± 0.13	0.391 ± 0.29	0.311 ± 0.19
MIL Top-10 avg-pooling	0.678 ±0.12	0.690 ±0.11	0.478 ±0.26	0.845 ± 0.17	0.609 ± 0.36	0.296 ± 0.16
MIL Top-50 avg-pooling	0.641 ± 0.10	0.668 ± 0.13	0.513 ± 0.13	0.812 ± 0.17	0.727 ± 0.28	0.193 ± 0.12
MIL Top-100 avg-pooling	0.641 ±0.09	0.673 ±0.11	0.515 ±0.11	0.818 ± 0.14	0.727 ± 0.28	0.200 ± 0.14
MIL Top-200 avg-pooling	0.727 ±0.01	0.744 ±0.02	0.605 ±0.03	0.883 ± 0.05	0.882 ± 0.04	0.222 ± 0.08

Table 5.8: Results on CAMELYON16 test set using 20× magnification

Model	AUC	Accuracy	F1-Score	Accuracy (Normal)	Accuracy (macrometastases)	Accuracy (Micrometastases)
CLAM SB - Small	0.825 ± 0.03	0.840 ± 0.03	0.759 ± 0.03	0.950 ± 0.03	0.936 ± 0.02	0.437 ± 0.01
CLAM SB - Big	0.823 ± 0.02	0.828 ± 0.02	0.752 ± 0.03	0.918 ± 0.05	0.945 ± 0.02	**0.467 ± 0.05**
CLAM MB - Small	**0.837 ± 0.01**	**0.842 ± 0.01**	**0.765 ± 0.03**	0.945 ± 0.04	**0.955 ± 0.03**	0.444 ± 0.02
CLAM MB - Big	0.829 ± 0.02	0.837 ± 0.02	0.753 ± 0.02	0.950 ± 0.03	0.927 ± 0.02	0.430 ± 0.06
MIL Max-pooling	0.639 ± 0.14	0.682 ± 0.14	0.466 ± 0.28	0.845 ± 0.18	0.573 ± 0.36	0.289 ± 0.15
MIL Top-2 avg-pooling	0.599 ± 0.11	0.633 ± 0.11	0.361 ± 0.23	0.823 ± 0.20	0.427 ± 0.34	0.237 ± 0.15
MIL Top-5 avg-pooling	0.700 ± 0.09	0.726 ± 0.10	0.408 ± 0.34	**0.960 ± 0.05**	0.555 ± 0.45	0.170 ± 0.16
MIL Top-10 avg-pooling	0.742 ± 0.08	0.747 ± 0.07	0.527 ± 0.26	0.932 ± 0.07	0.764 ± 0.38	0.185 ± 0.12
MIL Top-50 avg-pooling	0.711 ± 0.01	0.766 ± 0.01	0.599 ± 0.02	0.953 ± 0.01	**0.955 ± 0.00**	0.059 ± 0.04
MIL Top-100 avg-pooling	0.698 ± 0.01	0.758 ± 0.01	0.587 ± 0.02	0.945 ± 0.01	**0.955 ± 0.00**	0.044 ± 0.03
MIL Top-200 avg-pooling	0.683 ± 0.01	0.746 ± 0.01	0.566 ± 0.02	0.935 ± 0.01	0.909 ± 0.03	0.052 ± 0.03

Table 5.9: Results on CAMELYON16 test set using 20× magnification with Otsu's thresholding as segmentation algorithm

Results that are marked bold are the highest performing with respect to their experiment and results that are colored red are the highest performing over all the experiments.

Models from experiment III had an overall higher total accuracy, F1-score, normal lymph node accuracy and micro-metastases accuracy. With regards to detecting macrometastases, the CLAM SB small model from experiment I has the best accuracy, followed by a shared second place for experiments II and III. Finally, the AUC of the CLAM-SB Big model from experiment II had the best AUC, followed by experiment III and I.

We suspect that the experiments on 20× magnification perform better than the experiments on 10× magnification due to the extra level of detail that the 20× magnification images provide, which is supported by the micrometastases accuracy of the CLAM models in experiment III. Furthermore, we also suspect that experiment III performs better than experiment II, since the input data is less noisy due to Otsu's thresholding. Otsu's thresholding technique disregards fat tissue, which is irrelevant for detecting metastases in lymph nodes. By not including the fat tissue

in the slides, we feed the model less noisy data and therefore experience an increase in model performance. Furthermore, we suspect that the small tumor regions of micrometastases in a slide makes this case especially hard to classify.

CLAM models perform better on this dataset, since the models can assign attention scores to patches, thus making it more flexible than top-k avg pooling models. Therefore, CLAM models can deal with both macrometastases as well as micrometastases.

A trend we see in top-k avg pooling-models is that with increasing k, the accuracy of macrometastases increases, whereas the accuracy of micrometastases decreases. The reason why this occurs is that macrometastases are tumor cell clusters $\geq$ 2mm, whereas micrometastases are tumor cell clusters between 0.2 and 2mm. Thus, detecting macrometastases requires a larger k, while detecting micrometastases can be detected with smaller k. This makes top-k avg pooling models sub-optimal, since finding an optimal k is contradictory due to the nature of the different sizes of metastases.

From the experiments, we can also conclude that MIL with top-k avg pooling outperforms MIL with max-pooling, in the case of detecting normal lymph nodes and lymph nodes containing macrometastases. We suspect this behaviour occurs due to the larger area that the top-k avg pooling MIL models takes into account. However, we see that this behaviour does not count for the accuracy of micrometastases, since micrometastases can likely be captured within a single patch of a slide.

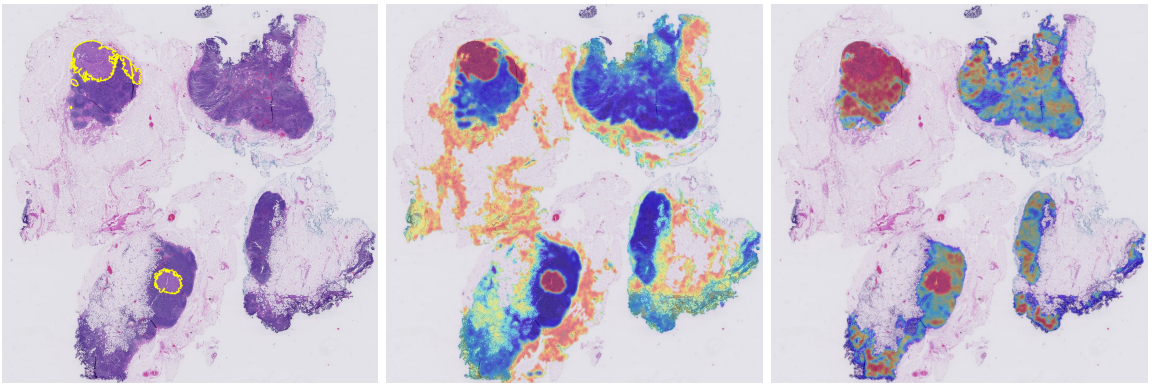
We found that the authors of CLAM had an AUC of 0.938 [30] using 40$\times$ magnification, Campanella et al. achieved an AUC of 0.899 at 20$\times$ magnification [4] and that Tellez et al. achieved an AUC of 0.704 at 20$\times$ magnification [38] with NIC. These models were all trained in a weakly-supervised manner, thus without ROI annotations. Our best performing MIL Top-k avg-pooling model in terms of AUC was for $k = 10$ on experiment III, which had an AUC of 0.742 $\pm$ 0.08, which is slightly higher than NIC. Tellez et al. also stated that NIC had difficulties with detecting micrometastases. We suspect that due to averaging the predictions from k-tissue regions to get the slide-level prediction, the supervisory signal from the micrometastases gets lost.

5.2.2 Visualization of attention scores

We visualize the attention heatmaps of two models; the CLAM SB - Big model at 20× magnification and the CLAM MB - Small model at 20× magnification using Otsu's thresholding, let us refer to these models as model $\mathcal{A}$ and model $\mathcal{B}$ respectively. Both models have the highest AUC and accuracy for their respecting experiment. Furthermore, we would like to see the difference in attention heatmaps between a model that uses the default segmentation method versus a model that uses Otsu's thresholding. We compare the results of the models with the annotations of the pathologists, which are annotated in yellow.

In Figure 5.7, we can see the annotated WSI and the attention heatmaps of models $\mathcal{A}$ and $\mathcal{B}$. This specific case contains macrometastases. We can see that both models have high correspondence with the annotation of the pathologist, however both models suffer from different problems. Model $\mathcal{A}$ has moderate attention in fat tissue, which is unimportant for the classification task. On the other hand, model $\mathcal{B}$ has high attention surrounding the ROI annotation, which is undesirable.

We can say that the attention heatmap of model $\mathcal{A}$ is better than the one of model $\mathcal{B}$ in Figure 5.7 due to two reasons; (I) attention in fat tissue can be ignored in the heatmap of model $\mathcal{A}$, and (II) model $\mathcal{B}$ has high attention in tissue area outside of the annotation.



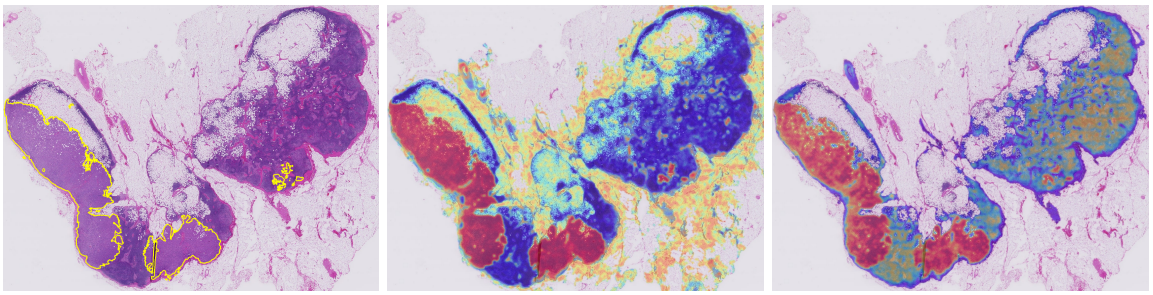
(a) Annotated WSI with macrometastases

(b) Attention heatmap of Figure 5.7a using model $\mathcal{A}$

(c) Attention heatmap of Figure 5.7a using model $\mathcal{B}$

Figure 5.7: A WSI with macrometastases in the lymph nodes

Figure 5.8 shows a WSI containing macrometastases. Both models have a high overlap between the annotation of the pathologist and the attention scores.



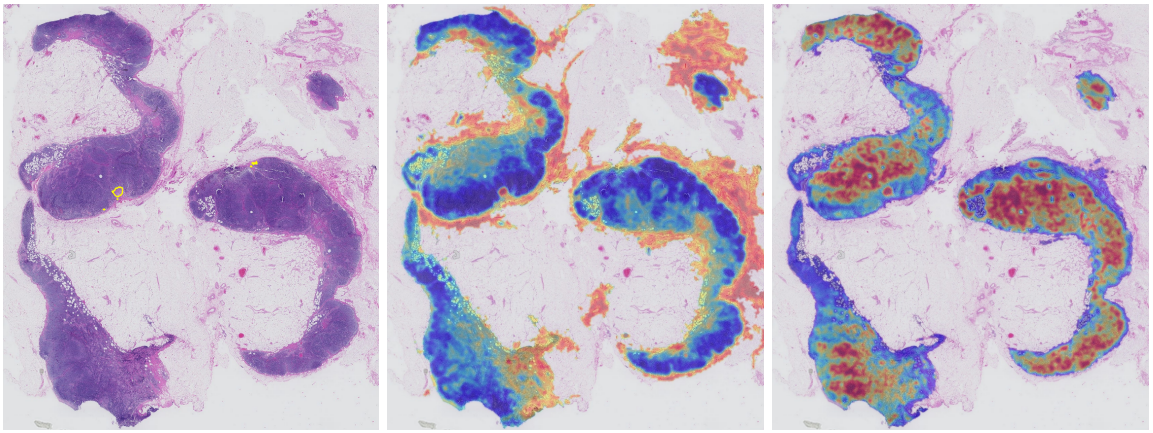
(a) Annotated WSI with macrometastases

(b) Attention heatmap of Figure 5.8a using model $\mathcal{A}$

(c) Attention heatmap of Figure 5.8a using model $\mathcal{B}$

Figure 5.8: A WSI with macrometastases in the lymph nodes

Figure 5.9 shows a case containing micrometastases. Both models captured the annotation on the left side, however both cases missed the small right annotation. In this WSI, both models suffer from the same problem as in Figure 5.7, where in case of model $\mathcal{B}$ the effect is more exaggerated. As expected due to the model performance, the areas containing micrometastases are difficult to capture.



(a) Annotated WSI with micrometastases

(b) Attention heatmap of Figure 5.9a using model $\mathcal{A}$

(c) Attention heatmap of Figure 5.9a using model $\mathcal{B}$

Figure 5.9: A WSI with micrometastases in the lymph nodes

We can conclude that the attention heatmaps of both models can capture macrometastases well. However, both models suffer from their own problems. Model $\mathcal{A}$ often assigns high attention to fat tissue, which is irrelevant for the diagnosis and model $\mathcal{B}$ often gives high attention to tissue area outside of the ROI. These problems arise due to the sub-optimal performance of the network, especially in the case of detecting micrometastases.

5.3 Melanoma pathway prediction

For this task, we apply 5-fold cross-validation with 70-15-15% for training, validation and testing respectively. We evaluate the CLAM SB, MB and MIL models on this task, see Table 5.10. Note that the AUC and F1-score are calculated with a one-versus-rest approach and averaging the results.

Model	AUC	Accuracy	F1-Score
CLAM SB - Small	0.601 ± 0.08	0.568 ± 0.08	0.427 ± 0.15
CLAM SB - Big	**0.641 ± 0.04**	0.536 ± 0.04	0.374 ± 0.11
CLAM MB - Small	0.589 ± 0.11	0.599 ± 0.08	**0.478 ± 0.07**
CLAM MB - Big	0.570 ± 0.11	**0.605 ± 0.11**	0.464 ± 0.13
MIL Max-pooling	0.581 ± 0.10	0.594 ± 0.10	0.416 ± 0.04
MIL Top-50 avg-pooling	0.622 ± 0.11	0.525 ± 0.11	0.378 ± 0.11

Table 5.10: 40× Magnification models using a ResNet-50 feature extractor pre-trained on ImageNet

We noticed that the dataset for melanoma pathway prediction is a very difficult dataset to work with due to the several reasons. Pathways I and IV both contain subtypes of melanoma, which increases the intra-class variability. This makes the prediction of pathways I and IV a difficult task since these classes are not well-defined. Moreover, the Remaining class is the remainder of the dataset that does not fall into pathways I or IV, and inherently since this class contains a wide variety of melanoma subtypes, the same problem of high intra-class variability exists. A more concrete indication that the dataset is poor is that the model performance is highly-dependant on the split that was used, as the standard deviations for the classification metrics are very high.

We also tried to apply our NIC model on this task, but results were also not favorable and terminated this approach. We suspect that the main issues of applying the NIC model are:

- Patch encoder does not extract important features from the tissue region. From our self-supervised feature extractor experiments on the breast cancer grade classification task, we noticed that these feature extractors perform worse than a pre-trained ResNet-50 on ImageNet. However, further work must be done to compare different feature extractors.
- Lack of data augmented tissue regions due to offline data augmentation, see Section 4.2.2.3 for an explanation of offline versus online data augmentation.
- We average the predictions of each tissue region within a slide, however this operation assumes that every tissue region is of equal importance, which is not the case.

Chapter 6

Conclusions

Automated WSI classification can be used by pathologists as a second pair of eyes and thus help pathologist in their workflow. In our work we investigated WSI classification models on three different tasks; breast cancer grade classification in young women, lymph node metastases detection and melanoma pathway classification. We only consider weakly-supervised learning methods, since annotating WSI is a time-consuming and tedious job for pathologists, therefore it is often infeasible to obtain datasets which contain ROI annotations.

For breast cancer grade classification on the YBCP dataset, we can conclude that our custom implemented MIL Top-250 average pooling model on 20× magnification has the best performance in terms of accuracy. It has an accuracy of 0.803 ± 0.01, which outperforms the CLAM and MIL with max-pooling models of [30], as well as the model accuracy of Wetstein et al. [12].

For this task, we also experimented with self-supervised learning for feature extraction using several SimCLR models [36], however the performance did not exceed that of a ResNet-50 pre-trained on ImageNet. We suspect that it might be due to the limited dataset that is used for the SimCLR models, as well as the limitation in compute power. SimCLR benefits from a large batch size, large model capacity and many training epochs, which was infeasible in our case.

For lymph node metastases detection on the Camelyon16 dataset, the CLAM MB - Small model at 20× magnification using Otsu's thresholding as tissue segmentation method has the best accuracy of 0.842 ± 0.01. We think CLAM outperforms MIL with Top-k average-pooling due to the flexible nature of assigning attention scores to patches, whereas MIL Top-k average-pooling has a constant k. This constant k negatively impacts the performance, since we want to classify macrometastases (tumor cell cluster $\geq$ 2mm), as well as micrometastases (tumor cell cluster between 0.2mm and 2mm). In case of macrometastases we require a large k, whereas in case of micrometastases we need a small k, therefore both cases contradict each other, which limits model performance. Furthermore, we noticed that the models at 20× magnification perform better in terms of micrometastases accuracy than the models at 10× magnification, which is likely due to the extra level of detail that the 20× magnification patches provide.

Our main research question that we defined in Section 2.3 was:

Which factors play an important role for classification on histopathology slides?

From our experiments we can conclude that feature extraction plays an important role for model performance. Most WSI classification methods work in a patch-based manner, in which the model extracts features from patches and consequently uses these features in various ways, see Section 3.3. We have experimented with two kinds of feature extractors, and saw a significant difference in model performance due to their expressiveness.

Aside from feature extraction of patches, it is important how these features are used. Various models have different takes on this manner. We think that context-awareness is an important

factor in model performance. For instance, MIL with max-pooling only uses a single patch for their WSI classification, and thus does not take into account a larger context. CLAM and MIL with top-k average pooling on the other hand do take into account a larger collection of patches and thus could make a more representative WSI prediction, thus can be considered more context-aware approaches.

From our results we can conclude that ambiguously defined classes are a limiting factor in model performance. We experienced this behavior in the *intermediate*-grade within the *low/intermediate* class for the breast cancer grade classification task and all the classes of the melanoma pathway classification task. This ambiguity led the melanoma pathway classification task to be infeasible.

We can also conclude that visual inspection of the model interpretations is of great importance to understand what the model finds relevant for the slide-level prediction. We have shown cases in which there is high correspondence between the heatmaps of the model and the annotations of the pathologist, however there could be cases in which there is no correspondence even though the prediction of the model may be correct. Therefore, it is important to evaluate the model's interpretation before using it in practice, to make sure that the ROI of the model aligns with the corresponding class.

6.1 Future work

Robustness of Models

A limiting factor of CLAM is that there is no data augmentation of patches. Naturally, the reason for this is due to the huge computational load of feature extraction of patches. We require roughly 14 hours to extract features from patches over our entire YBCP dataset at 10× magnification. If we would train our CLAM model with data augmentation, then we would need to augment the patches and extract features from these patches during every epoch, which is unfeasible. Future work may include on how to apply data augmentation in an efficient manner.

Furthermore, on the topic of robustness it would be interesting to see how models would perform on data from different institutions. There is often variability in the staining/scanning procedure, which could lead to a drop in performance. Data augmentation and in particular stain normalization should make these models more robust against these types of variabilities [45].

Neural Image Compression

Unfortunately, due to the infeasibility of the melanoma pathway classification task, we also had to terminate the development of NIC, as NIC was developed initially for this specific task and there was no time to fix its issues and apply NIC on a different dataset. An area that has been unexplored in NIC is the use of a pre-trained network on ImageNet as feature extractor. We have experienced that this feature extractor performed the best in case of breast cancer grade classification for CLAM and MIL-models, and therefore it might be an interesting feature extractor for NIC.

When developing NIC we noticed that we could only use offline data augmentation, since we used a 40× magnification level for the melanoma pathway prediction task. However, we think that online data augmentation may be feasible when using a smaller magnification level, since we need less tissue compressions.

For NIC, we average the predictions of the tissue regions within a slide to compute our slide-level prediction. However, this approach assumes that the importance of each tissue region is equal, which is not realistic since particular regions in a slide are more important than others (malignant tumor tissue versus healthy tissue). Therefore, future work may include on how to give importance to tissue regions. An example on how this can be achieved is by using the attention mechanism of CLAM.

Pre-training Feature Extractors from Multiple Organ Sites

The work from [45; 54; 53] experimented with pre-training feature extractors using data from multiple organ sites and experienced an increase in performance of their models, opposed to using only the data from the main task. Future models should consider a similar strategy, as it has

been shown that domain-specific feature extractors could perform better than non-domain specific feature extractors (e.g. a model pre-trained on ImageNet).

Automated Machine Learning
Another area that could be investigated is the use of intelligent hyperparameter tuning and Neural Architecture Search (NAS). WSI classification models take a long time to train, which implies that exhaustive tuning of the architecture/hyperparameters is often infeasible. A method to solve these problems could be in the form of Automated Machine Learning (AutoML) [58], which is a field that focuses on building Deep Learning models without human assistance. By leveraging on AutoML, we could optimize WSI models in a more efficient and guided manner.

Bibliography

[1] N. Farahani, A. V. Parwani, and L. Pantanowitz, "Whole slide imaging in pathology: advantages, limitations, and emerging perspectives," *Pathology and Laboratory Medicine International*, vol. 7, pp. 23–33, 2015. 1

[2] S. Al-Janabi, A. Huisman, and P. J. Van Diest, "Digital pathology: current status and future perspectives," *Histopathology*, vol. 61, no. 1, pp. 1–9, 2012. 1

[3] S. Graham, Q. D. Vu, S. E. A. Raza, A. Azam, Y. W. Tsang, J. T. Kwak, and N. Rajpoot, "Hover-net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images," *Medical Image Analysis*, vol. 58, p. 101563, 2019. 1

[4] G. Campanella, M. G. Hanna, L. Geneslaw, A. Miraflor, V. W. K. Silva, K. J. Busam, E. Brogi, V. E. Reuter, D. S. Klimstra, and T. J. Fuchs, "Clinical-grade computational pathology using weakly supervised deep learning on whole slide images," *Nature medicine*, vol. 25, no. 8, pp. 1301–1309, 2019. 1, 9, 10, 38

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009. 1, 21

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012. 1

[7] H. R. Tizhoosh and L. Pantanowitz, "Artificial intelligence and digital pathology: challenges and opportunities," *Journal of pathology informatics*, vol. 9, 2018. 1

[8] H. Sung, J. Ferlay, R. L. Siegel, M. Laversanne, I. Soerjomataram, A. Jemal, and F. Bray, "Global cancer statistics 2020: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries," *CA: a cancer journal for clinicians*, vol. 71, no. 3, pp. 209–249, 2021. 3, 4

[9] C. K. Anders, R. Johnson, J. Litton, M. Phillips, and A. Bleyer, "Breast cancer before age 40 years," in *Seminars in oncology*, vol. 36, pp. 237–249, Elsevier, 2009. 3

[10] J. A. van der Hage, J. S. D. Mieog, C. J. van de Velde, H. Putter, H. Bartelink, and M. J. van de Vijver, "Impact of established prognostic factors and molecular subtype in very young breast cancer patients: pooled analysis of four eortc randomized controlled trials," *Breast Cancer Research*, vol. 13, no. 3, pp. 1–11, 2011. 3

[11] G. M. Dackus, N. D. Ter Hoeve, M. Opdam, W. Vreuls, Z. Varga, E. Koop, S. M. Willems, C. H. Van Deurzen, E. J. Groen, A. Cordoba, *et al.*, "Long-term prognosis of young breast cancer patients (40 years) who did not receive adjuvant systemic treatment: protocol for the paradigm initiative cohort study," *BMJ open*, vol. 7, no. 11, p. e017842, 2017. 3

[12] S. Wetstein, "Deep learning-based breast cancer grading and survival analysis on whole-slide histopathology images." unpublished, . 3, 10, 19, 31, 42

[13] C. W. Elston and I. O. Ellis, "Pathological prognostic factors in breast cancer. i. the value of histological grade in breast cancer: experience from a large study with long-term follow-up," *Histopathology*, vol. 19, no. 5, pp. 403–410, 1991. 3

[14] D. Schadendorf, D. E. Fisher, C. Garbe, J. E. Gershenwald, J.-J. Grob, A. Halpern, M. Herlyn, M. A. Marchetti, G. McArthur, A. Ribas, *et al.*, "Melanoma," *Nature reviews Disease primers*, vol. 1, no. 1, pp. 1–20, 2015. 4

[15] J. Y. Lin and D. E. Fisher, "Melanocyte biology and skin pigmentation," *Nature*, vol. 445, no. 7130, pp. 843–850, 2007. 4

[16] D. E. Elder, B. C. Bastian, I. A. Cree, D. Massi, and R. A. Scolyer, "The 2018 world health organization classification of cutaneous, mucosal, and uveal melanoma: detailed analysis of 9 distinct subtypes defined by their evolutionary pathway," *Archives of pathology & laboratory medicine*, vol. 144, no. 4, pp. 500–522, 2020. 4

[17] J. K. Heng, D. C. W. Aw, and K. B. Tan, "Solar elastosis in its papular form: uncommon, mistakable," *Case reports in dermatology*, vol. 6, no. 1, pp. 124–128, 2014. 4

[18] W. H. Clark, L. From, E. A. Bernardino, and M. C. Mihm, "The histogenesis and biologic behavior of primary human malignant melanomas of the skin," *Cancer research*, vol. 29, no. 3, pp. 705–727, 1969. 4

[19] R. Ashikari, K. Park, A. G. Huvos, and J. A. Urban, "Paget's disease of the breast," *Cancer*, vol. 26, no. 3, pp. 680–685, 1970. 4

[20] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015. 8

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. 8, 21, 22

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014. 8

[23] I. Tabian, H. Fu, and Z. Sharif Khodaei, "A convolutional neural network for impact detection and characterization of complex composite structures," *Sensors*, vol. 19, no. 22, p. 4933, 2019. 8

[24] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *National science review*, vol. 5, no. 1, pp. 44–53, 2018. 9

[25] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez, "Solving the multiple instance problem with axis-parallel rectangles," *Artificial intelligence*, vol. 89, no. 1-2, pp. 31–71, 1997. 9

[26] B. Settles, "Active learning literature survey," 2009. 9

[27] X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning," *Synthesis lectures on artificial intelligence and machine learning*, vol. 3, no. 1, pp. 1–130, 2009. 9

[28] D. C. Brabham, "Crowdsourcing as a model for problem solving: An introduction and cases," *Convergence*, vol. 14, no. 1, pp. 75–90, 2008. 9

[29] M. Ilse, J. Tomczak, and M. Welling, "Attention-based deep multiple instance learning," in *International conference on machine learning*, pp. 2127–2136, PMLR, 2018. 9, 11, 14

[30] M. Y. Lu, D. F. Williamson, T. Y. Chen, R. J. Chen, M. Barbieri, and F. Mahmood, "Data-efficient and weakly supervised computational pathology on whole-slide images," *Nature Biomedical Engineering*, pp. 1–16, 2021. 9, 11, 18, 21, 38, 42

[31] G. Campanella, V. W. K. Silva, and T. J. Fuchs, "Terabyte-scale deep multiple instance learning for classification and localization in pathology," *arXiv preprint arXiv:1805.06983*, 2018. 10, 18

[32] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017. 10

[33] "Multi-task learning hard-parameter sharing." `https://ruder.io/multi-task/index.html#hardparametersharing`. 11

[34] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *International conference on machine learning*, pp. 933–941, PMLR, 2017. 11

[35] Y. Schirris, E. Gavves, I. Nederlof, H. M. Horlings, and J. Teuwen, "Deepsmile: Self-supervised heterogeneity-aware multiple instance learning for dna damage response defect classification directly from h&e whole-slide images," *arXiv preprint arXiv:2107.09405*, 2021. 12, 18, 21

[36] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*, pp. 1597–1607, PMLR, 2020. 12, 21, 22, 29, 31, 42

[37] M. M. Dundar, S. Badve, V. C. Raykar, R. K. Jain, O. Sertel, and M. N. Gurcan, "A multiple instance learning approach toward optimal classification of pathology slides," in *2010 20th International Conference on Pattern Recognition*, pp. 2732–2735, IEEE, 2010. 12

[38] D. Tellez, G. Litjens, J. van der Laak, and F. Ciompi, "Neural image compression for gigapixel histopathology image analysis," *IEEE transactions on pattern analysis and machine intelligence*, 2019. 13, 27, 28, 38

[39] B. E. Bejnordi, M. Veta, P. J. Van Diest, B. Van Ginneken, N. Karssemeijer, G. Litjens, J. A. Van Der Laak, M. Hermsen, Q. F. Manson, M. Balkenhol, *et al.*, "Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer," *Jama*, vol. 318, no. 22, pp. 2199–2210, 2017. 13

[40] M. Veta, Y. J. Heng, N. Stathonikos, B. E. Bejnordi, F. Beca, T. Wollmann, K. Rohr, M. A. Shah, D. Wang, M. Rousson, *et al.*, "Predicting breast tumor proliferation from whole-slide images: the tupac16 challenge," *Medical image analysis*, vol. 54, pp. 111–121, 2019. 13

[41] F. Ciompi, O. Geessink, B. E. Bejnordi, G. S. De Souza, A. Baidoshvili, G. Litjens, B. Van Ginneken, I. Nagtegaal, and J. Van Der Laak, "The importance of stain normalization in colorectal tissue classification with convolutional networks," in *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, pp. 160–163, IEEE, 2017. 13

[42] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2014. 13

[43] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," 2017. 13

[44] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018. 13

[45] D. Tellez, D. Höppener, C. Verhoef, D. Grünhagen, P. Nierop, M. Drozdzal, J. Laak, and F. Ciompi, "Extending unsupervised neural image compression with supervised multitask learning," in *Medical Imaging with Deep Learning*, pp. 770–783, PMLR, 2020. 13, 17, 43

[46] M. Shaban, R. Awan, M. M. Fraz, A. Azam, Y.-W. Tsang, D. Snead, and N. M. Rajpoot, "Context-aware convolutional neural network for grading of colorectal cancer histology images," *IEEE transactions on medical imaging*, vol. 39, no. 7, pp. 2395–2405, 2020. 13

[47] N. Hashimoto, D. Fukushima, R. Koga, Y. Takagi, K. Ko, K. Kohno, M. Nakaguro, S. Nakamura, H. Hontani, and I. Takeuchi, "Multi-scale domain-adversarial multiple-instance cnn for cancer subtype classification with unannotated histopathological images," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3852–3861, 2020. 14

[48] R. Wetteland, K. Engan, T. Eftestøl, V. Kvikstad, and E. A. Janssen, "Multiscale deep neural networks for multiclass tissue classification of histological whole-slide images," *arXiv preprint arXiv:1909.01178*, 2019. 14

[49] B. E. Bejnordi, G. Zuidhof, M. Balkenhol, M. Hermsen, P. Bult, B. van Ginneken, N. Karssemeijer, G. Litjens, and J. van der Laak, "Context-aware stacked convolutional neural networks for classification of breast carcinomas in whole-slide histopathology images," *Journal of Medical Imaging*, vol. 4, no. 4, p. 044504, 2017. 14

[50] C. Gueréndel, P. Arnold, and B. Torben-Nielsen, "Creating small but meaningful representations of digital pathology images," in *MICCAI Workshop on Computational Pathology*, pp. 206–215, PMLR, 2021. 15

[51] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016. 15

[52] R. Mormont, P. Geurts, and R. Marée, "Comparison of deep transfer learning strategies for digital pathology," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 2262–2271, 2018. 16

[53] R. Mormont, P. Geurts, and R. Marée, "Multi-task pre-training of deep neural networks for digital pathology," *IEEE journal of biomedical and health informatics*, vol. 25, no. 2, pp. 412–421, 2020. 16, 17, 43

[54] O. Ciga, T. Xu, and A. L. Martel, "Self supervised contrastive learning for digital histopathology," *Machine Learning with Applications*, p. 100198, 2021. 16, 21, 43

[55] J. Gildenblat and E. Klaiman, "Self-supervised similarity learning for digital pathology," *arXiv preprint arXiv:1905.08139*, 2019. 17, 26

[56] J. S. Meyer, C. Alvarez, C. Milikowski, N. Olson, I. Russo, J. Russo, A. Glass, B. A. Zehnbauer, K. Lister, and R. Parwaresch, "Breast carcinoma malignancy grading by bloom–richardson system vs proliferation index: reproducibility of grade and advantages of proliferation index," *Modern pathology*, vol. 18, no. 8, pp. 1067–1078, 2005. 19

[57] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979. 27, 36

[58] X. He, K. Zhao, and X. Chu, "Automl: A survey of the state-of-the-art," *Knowledge-Based Systems*, vol. 212, p. 106622, 2021. 44

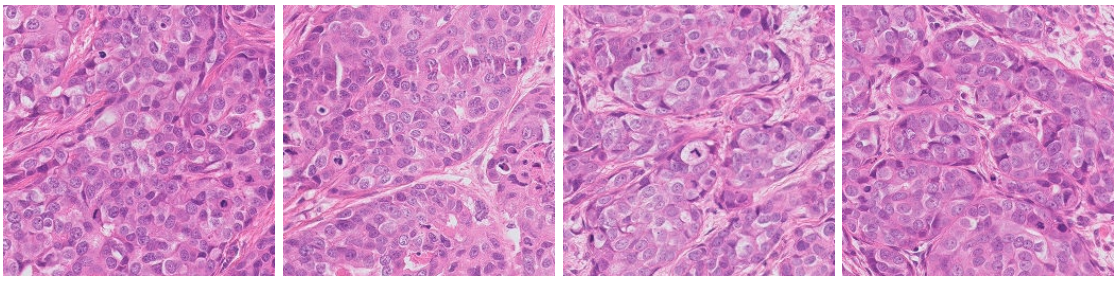
Appendix A: WSI Compression pseudo algorithm

Algorithm 1 Neural Image Compression of a single slide

```
1: procedure Compress(WSI, encoder)
2:     compressedRegions = list()
3:     mask = TissueSegmentation(WSI)
4:     regions = LocalizeRegions(mask)
5:     randomAugmentations = InitializeRandomAugmentations()
6:     for region ∈ regions do
7:         for y_i ∈ region.height/patch.height do
8:             offsetY = region.offsetY + patch.height * y_i
9:             for x_i ∈ region.width/patch.width do
10:                offsetX = region.offsetX + patch.width * x_i
11:                x = offsetX + patch.width/2
12:                y = offsetY + patch.height/2
13:                sourcePatchCenter = localizeSourcePatch(x, y, region.center, randomAugmentations.angle)
14:                patch = extractPatchFromWSI(WSI, sourcePatchCenter)
15:                augmentedPatch = patchAugmentations(patch, randomAugmentations)
16:                compressedPatch = encoder(augmentedPatch)
17:                compressedRegions[region][x_i][y_i] = compressedPatch
18:            end for
19:        end for
20:        compressedRegions[region] = wsiAugmentations(compressedRegions[region], randomAugmentations)
21:    end for
22:    return compressedRegions
23: end procedure
```

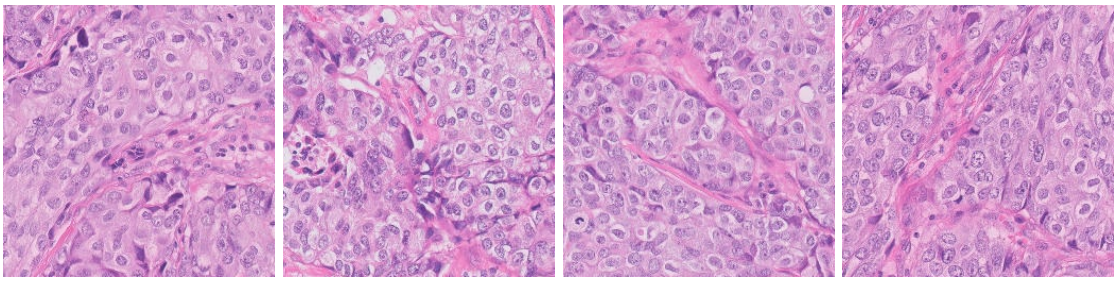
Appendix B: Top-k patches

Figures 6.1, 6.2 and 6.3 show the top-4 patches of slides of various breast cancer grades.



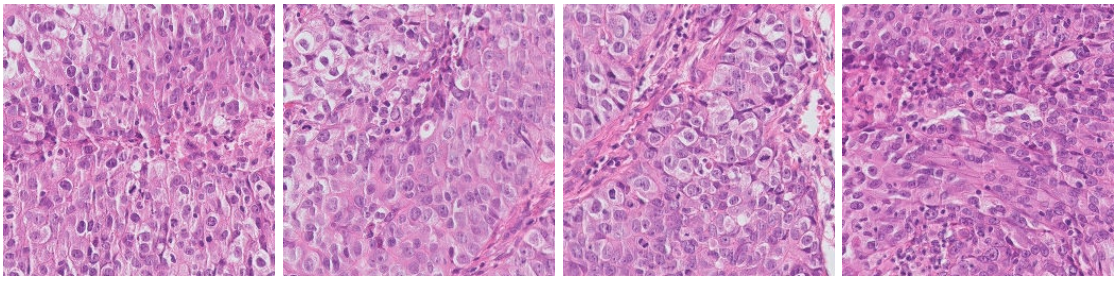
(a) Patch 1 (b) Patch 2 (c) Patch 3 (d) Patch 4

Figure 6.1: The top-4 patches of Figure 5.1d, which is a *high*-grade tumor



(a) Patch 1 (b) Patch 2 (c) Patch 3 (d) Patch 4

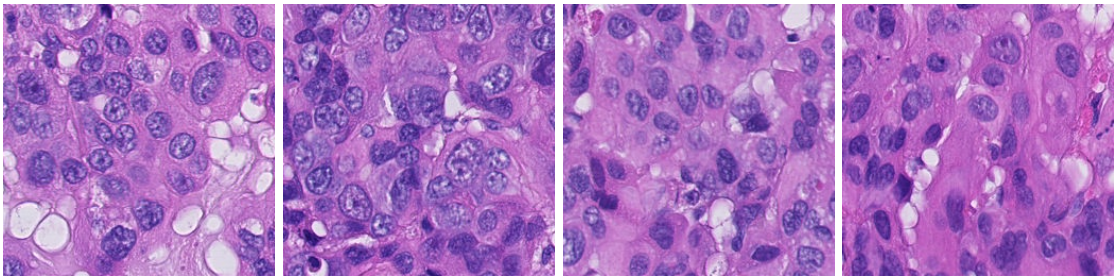
Figure 6.2: The top-4 patches of Figure 5.3b, which is a *low/intermediate*-grade tumor



(a) Patch 1 (b) Patch 2 (c) Patch 3 (d) Patch 4

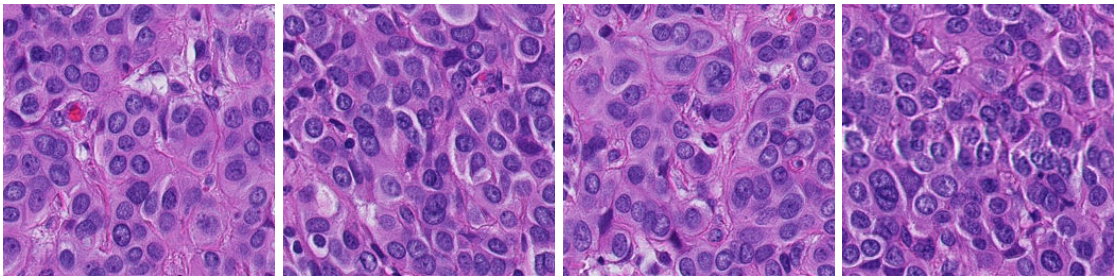
Figure 6.3: The top-4 patches of Figure 5.4b, which is a *high*-grade tumor

Figures 6.4 and 6.5 show the top-4 patches of slides which contain macrometastases in lymph nodes.



(a) Patch 1 (b) Patch 2 (c) Patch 3 (d) Patch 4

Figure 6.4: The top-4 patches with highest attention scores of Figure 5.7c



(a) Patch 1 (b) Patch 2 (c) Patch 3 (d) Patch 4

Figure 6.5: The top-4 patches with highest attention scores of Figure 5.9c