Multilevel feature fusion in digital pathology

Master of Science Thesis
University of Turku
Department of Computing, Faculty of Technology
Artificial Intelligence Retraining
2022
Joni Juvonen

Supervisors:
Assoc. Prof. Tapio Pahikkala
Ph.D. Antti Karlsson

UNIVERSITY OF TURKU
Department of Computing, Faculty of Technology

Joni Juvonen: Multilevel feature fusion in digital pathology

Master of Science Thesis, 51 p., 2 app. p.
Artificial Intelligence Retraining
April 2022

The breast cancer stage and prognosis are mainly diagnosed from surgically removed sentinel lymph nodes which are dissected, stained and scanned for the presence of tumor cells. The extent of tumor cells spreading to lymph nodes affects the treatment plan and prognosis of the patient. Currently, the best practice for scanning stained glass slides of lymph node tissue sections is a microscopic examination by a trained pathologist. The time-consuming examination is prone to subjective errors as the scanner produced images are typically gigapixel size and tumor areas are relatively small. An isolated tumor cell (ITC) especially, is hard to spot, and requires a trained eye.

The recent progress with convolutional neural networks (CNN) in the image processing area has also proven effective in detecting metastases from tissue sample images. Their performance has been on par with a group of expert pathologists. CNN's could be used routinely to highlight possible tumor areas for pathologists or even independently assess the level of metastatic regions in samples to ease human experts' workload.

What makes the tumor detection from tissue images challenging is the high resolution of images the commercial scanners export. Processing a whole gigapixel image requires a lot of memory so the image is typically split into smaller windows that are fed through CNN separately. The weakness of such procedure is that the field-of-view per sample window is narrow and the information about the surrounding context is lost.

This thesis examines the benefits of feeding image batch samples of different zoom levels, cropped from the same location, to a CNN tumor classifier.

Keywords: CNN, classification, tumor, cancer

Contents

List of Figures

List of Tables

Abbreviations

AI artificial intelligence. 4, 5

AJCC The American Joint Committee on Cancer. 1

AUC area under the ROC curve. 22, 29, 30, 34, 36, 40, 41, 43, 44, 46, 48, 49

CNN convolutional neural network. 2, 4, 5, 6, 7, 9, 14, 17, 22

CWZ Canisius-Wilhelmina Hospital. 16, 17

DSC Dice similarity coefficient. 22, 30, 44, 48, 49

H&E hematoxylin & eosin. 1, 16

HSV hue, saturation and value color space. 17

IoU intersection over union. 22, 30, 44, 48, 49

ITC isolated tumor cell. 2

LPON Laboratory of Pathology East-Netherlands. 16, 17

RGB red, green and blue image channels. 18, 22

ROC curve receiver operating characteristic curve. 21

RST Rijnstate Hospital. 16, 17

RUMC Radboud University Medical Center. 16, 17

TNM tumor-node-metastasis breast cancer staging. 1, 7

UMCU University Medical Center Utrecht. 16, 17

WHO World Health Organization. 1

WSI whole slide image. iv, 4, 11, 13, 14, 17, 21, 22, 30, 48, 51

1 Introduction

1.1 Background

Breast cancer is the leading cancer in women worldwide according to World Health Organization (WHO), and it annually causes more than 500,000 deaths [1]. Early detection is critical as the five-year survival rate is very low if cancer has spread outside the breast to other parts of the body (27% according to American Cancer Society) [2]. The stage of cancer affects the patient's prognosis, and treatment plan and one of the regional spreading indicators is whether cancer has spread to lymph nodes.

The nodal status and tumor size are the primary observations used to for patient prognosis. Larger tumor sizes decrease the patient survival rate, and the lymph node status gives indication about the tumors ability to spread [3]. The American Joint Committee on Cancer (AJCC) has established a breast cancer staging system widely known as TNM staging where T stands for the primary tumor, N for regional lymph nodes, and M for distant metastasis. Each of the three categories has several sub-categories, and defining the patient's cancer stage will determine prognosis and aid in choosing the right treatment plan. [3]–[5]

Getting the lymph node status involves a sentinel lymph node biopsy where the nodes that are first affected by regional tumor spread are removed and inspected [6]. The removed lymph nodes are sliced and stained with hematoxylin & eosin (H&E),

or if specific antigens (proteins) need to be identified, binding immunohistochemical stains. The slides are imaged with a microscope and visually inspected by an expert pathologist who looks for the presence, quantity, and type of metastases for cancer staging [4]. Metastasis, smaller than 2.0 mm, are regarded as micrometastases, and single metastatic cells smaller than 0.2 mm as isolated tumor cells ITC [5]. The latter ones especially can be hard to detect due to their small size, but they can be an important indicator for selecting effective treatment for early stage breast cancer [4].

1.2 Digital pathology

Digital whole slide scanning is the process of capturing a digital image from a microscope sample. Due to the high magnification and large filming area, the scanners may capture the sample in smaller tiles and digitally stitch them back to a large image. Some devices use line-scanning sensors that don't require tile splitting, but instead, film the whole area line by line from one side to another. When a glass slide sample is scanned, the device creates a digital image where the size of the scan file depends on scanned area, magnification level, color depth, and compression format. A slide scanned at 40X magnification and 24-bit color depth, for instance, may produce 48 MB of data for every square millimeter of the scanned area. [7] Disk space is quickly used in such high-resolution imaging and to reduce storage space requirements, scanners may use lossy JPEG compression. Some scanner models include pre-scan automatic tissue area detection that first scans the slide in lowest magnification to detect empty regions in glass slide. This allows to scan only tissue parts of the slide, and reduce filming time and final file size. However, automatic tissue detection adds a potential scanning failure case where part of a faint tissue gets left out if the contrast between the glass slide, and tissue is below a set threshold. Figure 1.1 demonstrates such case. [8]

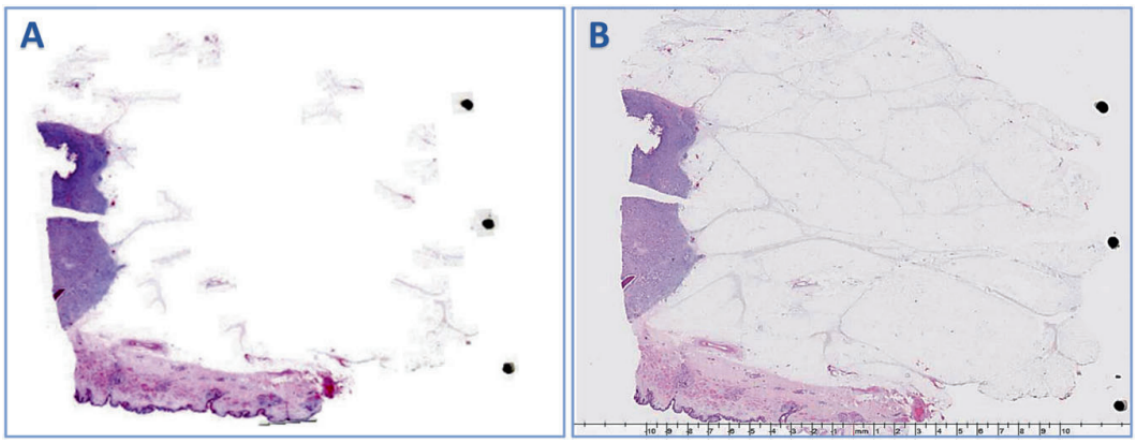


Figure 1.1: Example of the same tissue scanned at two different settings. (A) has a high threshold for automatic tissue detection and parts of the fatty tissue are left out. (B) has a lower threshold and the algorithm has not detected any non-tissue regions. [8]

Besides failure in automatic tissue area recognition, other potential scanning issues are out-of-focus tissue and compression artifacts, failed stitching operation, poor exposure and white balance adjustments, or other scanner settings-related causes. Some errors may propagate from slide preparation, such as uneven tissue staining, cuts and broken tissue, and bubbles in the slide. [7], [8] These error scenarios and even the differences between practices of different laboratories are affecting resulting scan quality and causing variation between illumination, sharpness, and color representation. Attempts have been made to calibrate scanning procedures to produce similar color representations, but since there are multiple sources of variation in several stages of scanning, it is difficult to achieve a consistent scan look within all samples taken in single laboratory, let alone across laboratories. The whole slide imaging protocol of an International Color Consortium in Figure 1.2 is a standard which aims to unify color representation.

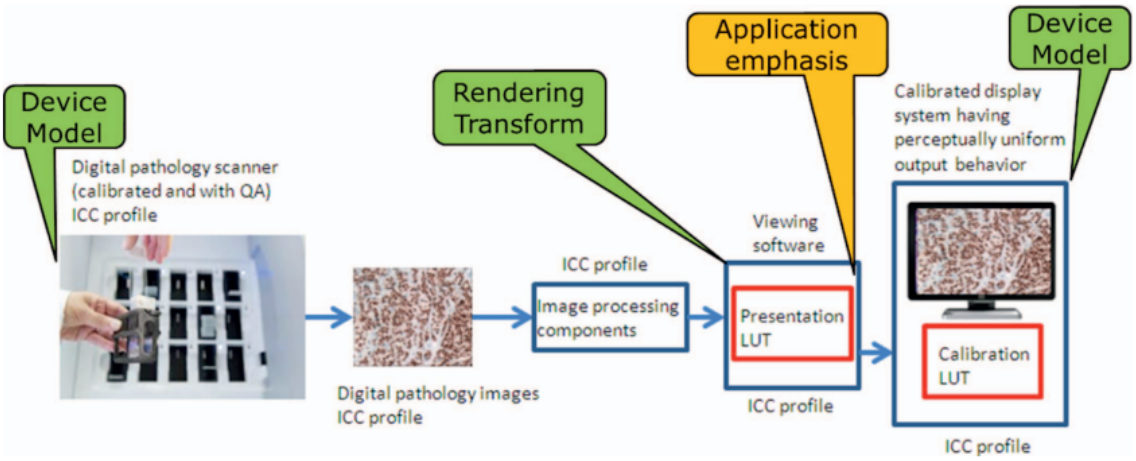


Figure 1.2: The color standardized whole slide imaging pipeline of the International Color Consortium. The scanner and image viewer software as well as the monitor are color calibrated to ensure similar color representation for the viewer. [7]

1.3 AI in digital pathology

1.3.1 Milestones

Advances in digital slide scanning microscopy and hard drive storage cost reductions have entirely digitized the slide analysis workflow [9]. Among benefits like remote analysis and more accessible archived samples, some health care facilities have made a combined effort to publish large whole slide image (WSI) collections such as the CAMELYON16 and CAMELYON17 datasets [10], [11]. These publicly accessible and large annotated datasets, and the advances in deep learning during the last decade, especially in the use of convolutional neural networks CNN in image processing, have advanced the use of artificial intelligence (AI) in digital pathology.

Figure 1.3 highlights some of the notable milestones that have affected modern AI practices in digital pathology. Starting from the AI branch of computer science being created by McCarthy [12], and the first convolutional neural network by Yann LeCun in 1988 [13] to Generative Adversarial Networks invented by Ian Goodfellow in 2014. These have been a few of the major milestones that have played part

to CNN architectures dominating many of the vision-related tasks. Their success comes from convolutional layers, which are translationally invariant. Thus, they are efficient with images where objects may be present in different picture regions. The same neurons that learn to recognize certain features will work in all image parts without needing to learn the same features several times for each location [13]. Each convolutional layer learns to detect features from the output feature maps of a previous layer, and the deeper the stack goes, the more complex the learned features will get. For instance, in dog and cat classification, the first convolutional layers may learn to detect simple edges and contours. The later ones could specialize in higher-level features such as ears or snout.

However, recent research and success of vision transformer architecture could very well lead to transformers replacing CNN architecture in some of the tasks in the coming years [14]. Especially in tasks where it's beneficial to perceive longer spatial distances than CNN receptive fields cover. Convolutional layers are relatively local, meaning a single point in the layer's output is affected only by the immediate neighborhood of the corresponding point in the input. When stacking these operations in sequence, the area in the image which can affect the response in certain one point in the last convolutional output is called its receptive field, and this may only cover part of the input image. Vision transformers don't have these spatial limitations and can simultaneously see all input regions.

The invention of whole slide scanner, photoacoustic microscopy [15] and microscopy with ultraviolet surface excitation [16] have enabled more scalable digital imaging for medical diagnostics. Modern AI technologies along with large digital record datasets have made assisted pathology software tools possible. The Philips IntelliSite Pathology Solution, which received De Novo pathway clearance and Food and Drug Administration approval in 2017 is one example of such AI assisted pathology software [17]. Apart from the major technology companies, there are also sev-

eral emerging startups that are using deep learning technologies in digital pathology such as Paige.AI [18], DeepLens [19], Proscia [20], PathAI [21], Inspirata [22] and DeePathology [23].

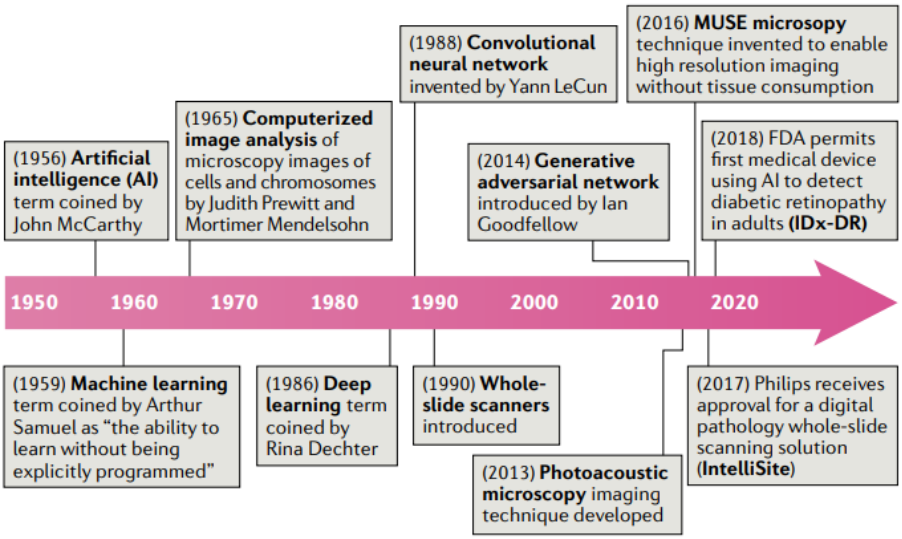


Figure 1.3: Milestones that have led to the use of AI in modern digital pathology. [24], [25], [26], [13], [15], [16], [17]

1.3.2 Modern methods

The performance of CNN models in image-related tasks such as the famous Imagenet classification benchmark challenge [27] have been improving in the last decade, and they have become the dominant algorithm in image classification, segmentation and object detection. The advances have come partially from the deeper and more efficient model architectures, but also from the improved practices in training procedures. These include adaptive optimizers such as Adam [28] and learning rate schedules. Namely cosine annealing, cyclical or one-cycle [29], [30]. Also, many minor model architecture changes have improved the accuracy, such as changing the stride of the first convolutional layers or including channel- and spatial-wise attention into CNN building blocks. [31], [32].

As for digital pathology, accessible large datasets like CAMELYON16 and 17 have possibly accelerated the use and adoption of CNN models in the field. Now, virtually anyone could develop computer vision algorithms to identify cancer in tissue samples, as previously this would have required access to patient biopsy records through a medical research institute or hospital and an expert pathologist to label samples. AI models have already been beating pathologist-level performance and proven to be useful in classifying and segmenting tumor regions, as well as assisting in grading biopsy samples [33], [34]. Lee & Paeng showed that a CNN-based AI algorithm could be used in predicting the TNM stage of a patient from lymph node biopsy. Such prediction was not achieved by end-to-end learning, meaning predicting the stage directly from images. Rather, they used statistical and shape features calculated from CNN segmentation maps of tumor regions and predicted the stage using a second-level random forest classifier [34]. It may be helpful for complex tasks like TNM staging, to split the problem in multiple parts to achieve good results and transparency in decision logic.

1.3.3 Generalization and robustness

Even though recent advances in AI have shown that computer algorithms can miss fewer tumor regions compared to human pathologists, leaving the diagnosis or grading decision solely to algorithms could be dangerous [35], [33]. These models have merely learned to transform the training data into a feature space that separates target label classes, but there are several pitfalls into this. The training data may not be representative enough of the population, the input data might not have the necessary information to make predictions of the target, or the training targets could be erroneously labeled. The latter is often present to some degree in medical image data, since labels may be ambiguous and pathologists don't necessarily agree on each case [35] [33]. To address this, majority voting of three or more pathologists

may be used to get better annotations.

One of the reasons why it is difficult to have generalizing models in digital pathology is the variation in the sample staining and scanning settings. Biopsies scanned in one laboratory may look quite different in color hue and contrast compared to samples from another laboratory. Image post color normalization using different computational methods have been proposed for bringing all color representations on the same scale before the analysis [36], [37], [38]. Even generative AI models have been proposed for the normalization [39]. Many of the conventional H&E normalization methods are computationally trying to estimate the fraction of hematoxylin and eosing color components. This is done for each sample or area of the sample and scaling fractions against a reference region. [37], [36].

A second approach for dealing with color variation is color augmentation. Instead of bringing the color component fractions of hematoxylin and eosin to static values, color augmentation varies the hue, saturation, and value of the color, and forces the AI model not to trust colors blindly. Color augmentation methods exist that target the H&E staining, and in a similar way that normalization tries to scale color components to fixed fractions, augmentation randomizes these within given limits [38].

AI models are often regarded as black boxes when interpreting the model's reasoning. In more conventional modeling where features are engineered based on domain knowledge, [13] models find patterns and features from the images by getting feedback from data labels. The objective of the training algorithm is to minimize the outcome of a loss function by adjusting weights and biases of the model layers by backward propagating the errors of predictions. It is difficult to regulate what features the model can learn, so the model may learn to use bias in training data as a shortcut. An example of such a case is when a pathologist marks abnormal tissue regions with a sharpie to glass slides, and these markings show in training

images. Then, instead of learning the features of abnormal tissue, the model may discover that a sharpie marking is a good feature for identifying abnormal regions. The outcome may be a model that works perfectly when glass slides have markings around abnormal tissue parts but fails to make correct predictions if they are missing.

Several tools and techniques have been developed to interpret and visualize the reasoning of CNN models. These include occlusion-based methods where part of images is hidden to see how the prediction changes or class activation mapping (CAM) methods such as Grad-CAM. [40] In Grad-CAM, class-specific positive activations are followed back to convolutional layers that hold spatial structure to highlight the regions of image that contributed to a class label outcome [41]. Figure 1.4 shows different visualization methods such as fine-grained and class-invariant Guided Backpropagation [42], class-specific Grad-CAM, their combination, and an occlusion based visualization method. These visualization techniques help debug the reasoning of a CNN classifier models and are useful in discovering data biases. One way of making sure that the model is learning right features is to check that the target class regions in images are showing actions. Figure 1.5 shows an example of data bias that is visible in activated regions. The gender-biased model mainly bases predictions in the face region that provides gender cues. [43]

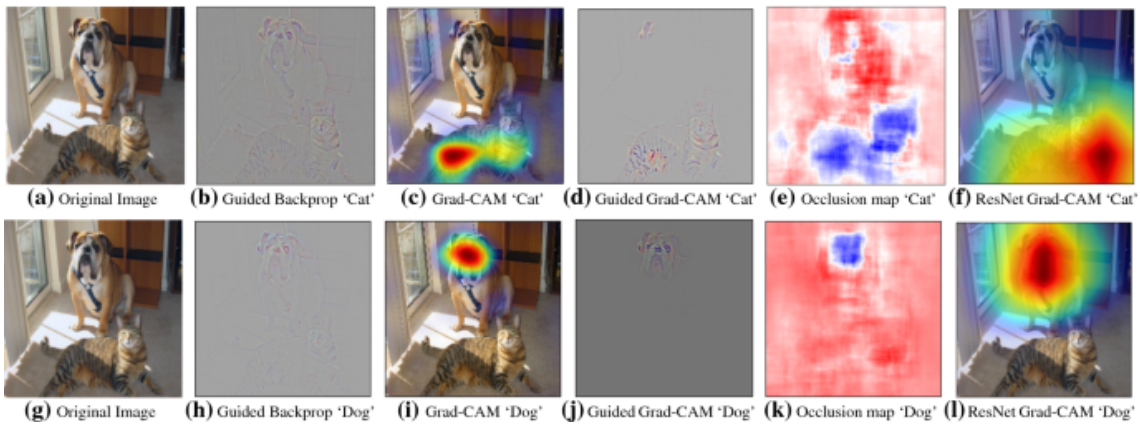


Figure 1.4: Top and bottom rows show the cat-specific and dog-specific activation visualizations methods. The first column shows (a and g) original images, second column (d and h) high-resolution Guided Backpropagation, third column (c and i) Grad-CAM, fourth column (d and j) Guided Grad-CAM, fifth column (e and k) occlusion maps and final column (f and l) ResNet model's Grad-CAM activations overlaid on top of the original images. [41]

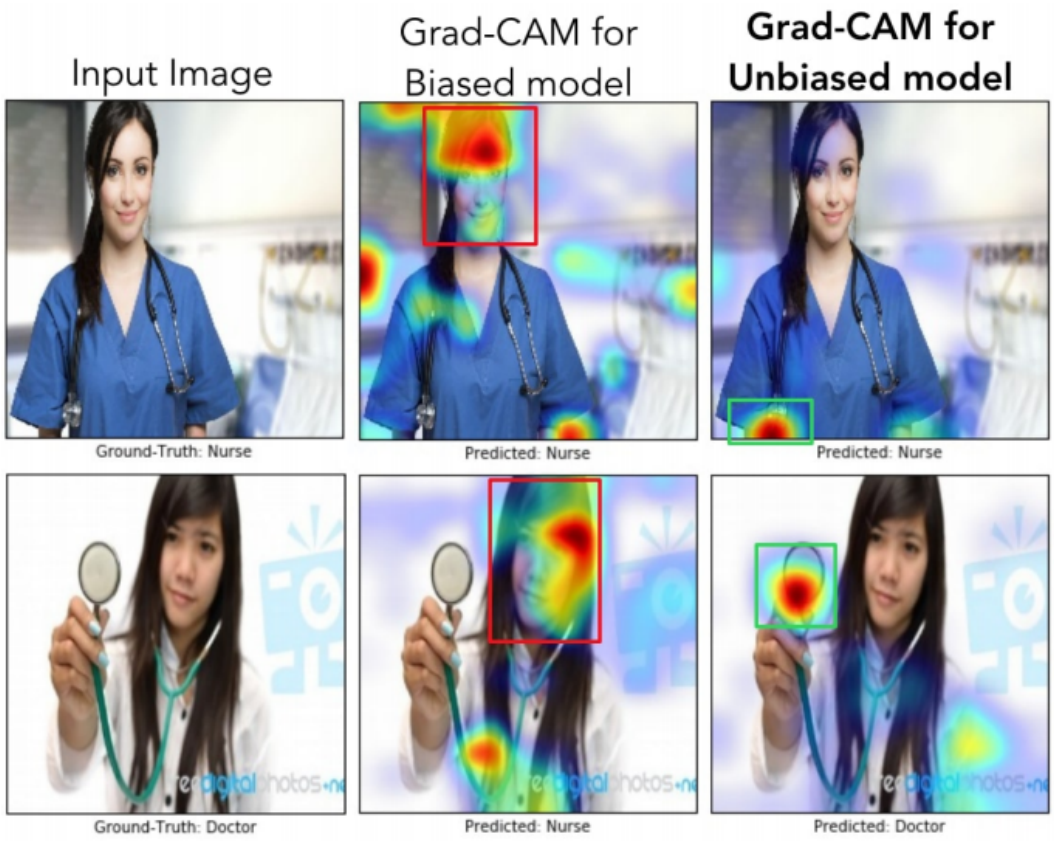


Figure 1.5: Comparison of Grad-CAM class-activations of gender-biased and unbiased models. The first column shows the input image with ground-truth label, and second and third columns the activations of predicted class from biased and unbiased models.[43]

For potentially critical applications such as patient diagnosis where an error may have fatal consequences, it is better to have a trained human pathologist in the loop. For instance, if the algorithm encounters samples out of training distribution, it may behave in unexpected ways. It has been studied that even though AI models are on par or exceed pathologist-level performance on narrow tasks, an expert pathologist who has the help of an AI model performs even better [44]. In the future, WSI viewing software may have more AI assistant features that highlight identified tumorous regions along with shape description and other statistic information for

human pathologists to analyze. Viewer software visual interface could look like in the Figure 1.6. These tools could merely indicate the regions that it found suspicious to a human expert, who would make the final decision. Human alertness and the ability to spot faint features in vast images can vary throughout the day and from expert to another. In contrast, an algorithm performs at constant level but can only do a narrow task and is not good at adapting to unseen cases. However, combining humans ability to reason and adapt past experience to new situations with algorithms tireless execution, we can have the best of both worlds.

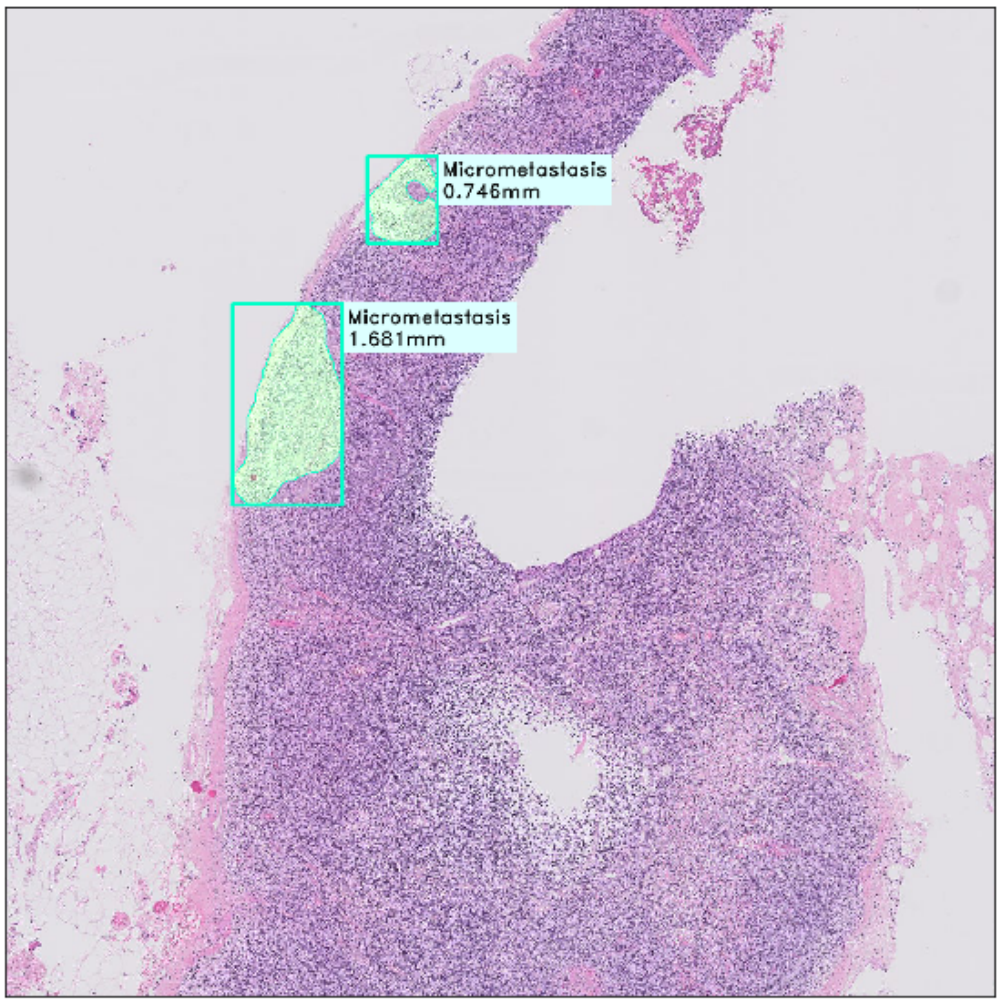


Figure 1.6: A hypothetical visualization of a WSI viewing software that highlights suspicious tumor regions along with shape and other statistics for a human pathologists to inspect.

1.3.4 Multilevel AI model

Since the introduction of WSI scanning technology, pathologists have been able to view digitized scans using software that allows panning and zooming the scanned tissue, similarly as with Google Earth satellite imagery. Before the digitized images, one could look at a tissue biopsy with different microscope magnification lenses to observe the sample in different scales before concluding on a diagnosis. However,

conventional CNN models are typically given a single input frame, and due to their fixed receptive fields, the scale range of observable features is limited. To cover a broader scale of features, CNN would need to have more convolutional kernels with varying receptive fields, and eventually, the input image size would limit the upper range [45]. The aim of this work is to investigate whether having a secondary context feature extraction branch would improve the classification performance of a WSI tile tumor classification model. The second branch adds the ability to see context-level features, which brings the classification setting closer to what human pathologists can observe.

1.4 Related work

The idea of having multiple input scales for a CNN model is not new. It has been tried for classification, segmentation and object detection in remote sensing and other large scale image tasks with success [46], [45], [47], [48]. The multi-scale design has been especially effective in segmenting larger features such as large constructions in satellite images and the approach has reduced holes in predicted masks significantly [47]. To combine multiple scales, there are several approaches in fusing information from different levels, and the simplest one is to concatenate pooled features from several single-scale encoders right before the classification layer. Another approach is to fuse before pooling to have spatial aware multi-scale features [46].

Multiple input scales have been also used in medical imaging in Optical Coherence Tomography and in WSI classification. [49], [50]. Liu et al. published a very similar multi-scale WSI tumor classification experiment in 2017, and reportedly, fusion of two magnification scales did not offer other benefits than smoother tumor probability maps. The model design for the multi scale feature fusion is show in Figure 1.7.

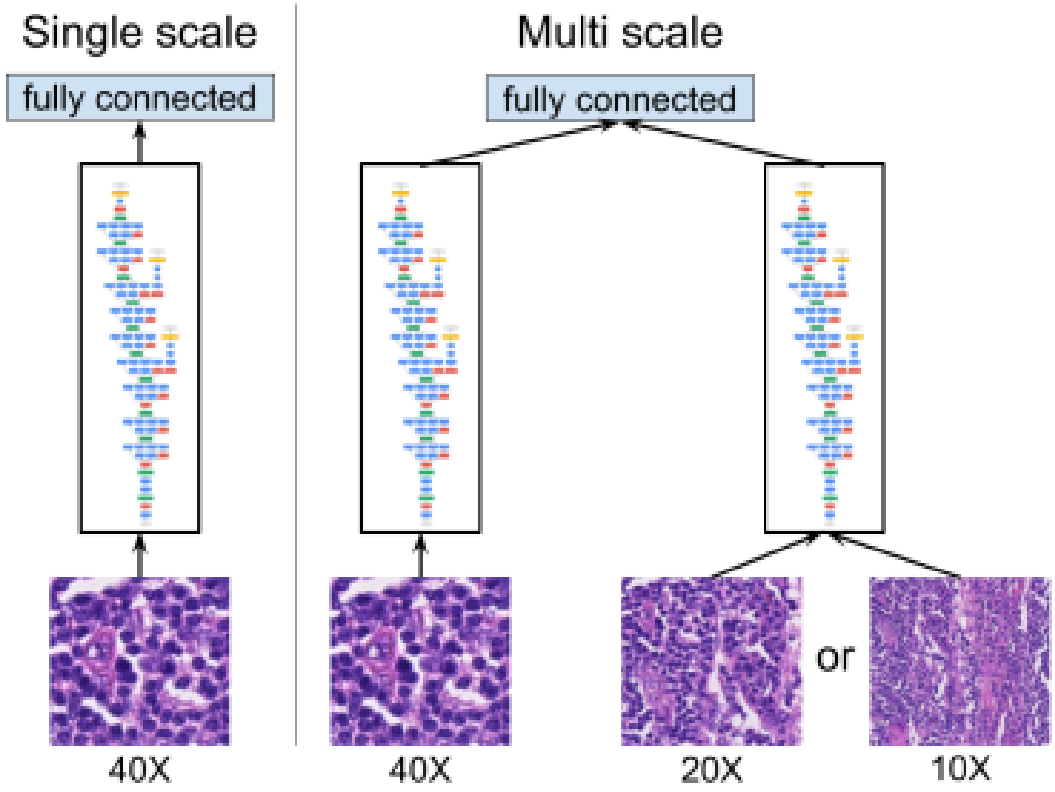


Figure 1.7: Tumor classification architecture designs for single scale and multi scale in the work of Liu et al. (2017). Patches of different magnifications are passing through Inception (V3) feature encoders to a fully connected classification layer. Features from different scales are concatenated for the classification. [50]

2 Methods and implementation

This chapter introduces the methods used for testing a hypothesis of including context-level CNN features to a tumor patch image classifier offers some benefit over just having local patch area features. The idea is related to a workflow of professional pathologists who inspect a WSI in multiple magnification levels. [51]

2.1 Image data set

CAMELYON17 hematoxylin & eosin (H&E)-stained lymph node section whole-slide image data set was used as the training and evaluation data. It is a larger and WSI-level annotated successor of the CAncer MEtastases in LYmph nOdes challeNge (CAMELYON16) data set that was created for a challenge, intended for improving the automated methods of detecting breast cancer metastasis. The data set has 1399 WSIs collected from five different medical centers. Centers 0,1 and 3 or Radboud University Medical Center (RUMC), Canisius-Wilhelmina Hospital (CWZ), and Rijnstate Hospital (RST) respectively, had similar 3DHistech Pannoramic Flash II 250 WSI scanners with pixel size of 0.24 μm. Center 2, University Medical Center Utrecht (UMCU) had Hamamatsu NanoZoomer-XR C12000-01 scanner with pixel size of 0.23 μm and center 4, Laboratory of Pathology East-Netherlands (LPON) had Philips IntelliSite Ultrafast Scanner with pixel size of 0.25 μm. The differences of color characteristics between centers are visualized in Figure 2.1 that shows ten randomly sampled image patches from each center. [11]

Center 4 LPON was used only for evaluation and centers 0,1,2 and 3 (RUMC, CWZ, UMCU, and RST) were used in CNN training in a leave-center-out cross-validation manner.

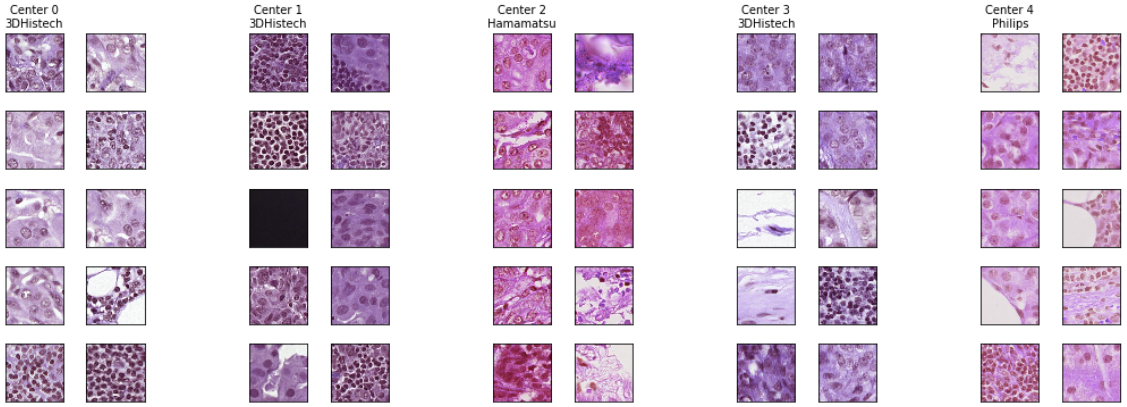
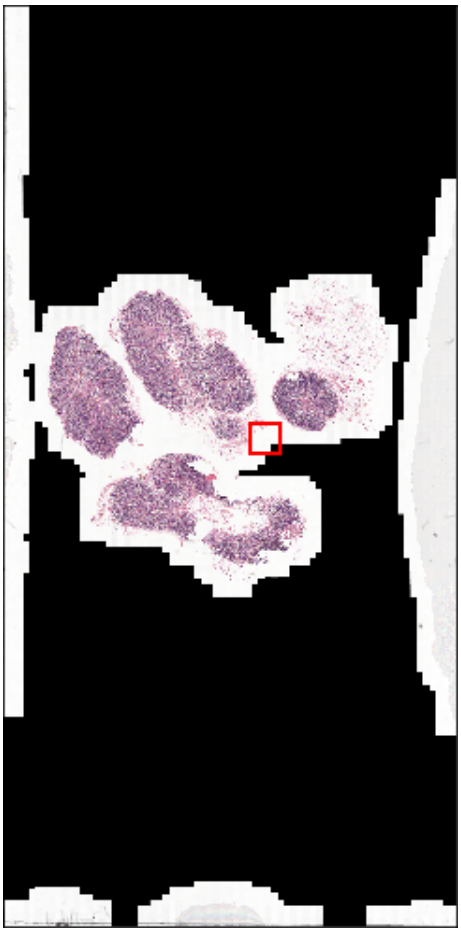


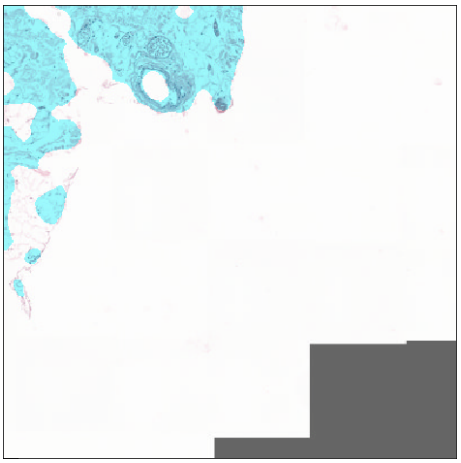
Figure 2.1: Random patch image samples from medical centers of CAMELYON17 showing the characteristic colors of different scanners and dyeing procedures.

CAMELYON17 WSIs had polygon annotations for metastatic regions, and these were used for evaluating the binary segmentation models. For a binary tumor versus normal tissue area classification data set, tissue areas were extracted from the gigapixel WSIs as shown in Figure 2.2. Figure 2.2a is a resized WSI with original colors and black areas that the scanner software has thresholded out. The red rectangle is the focus region of b and c figures. Tissue area mask was extracted from 16 times downsampled WSI by performing Otsu binarization for the saturation channel of hue, saturation and value color space (HSV) [52]. To fill small holes in the mask and to leave out small isolated tissue areas, morphological closing was applied two times with a kernel size of five, followed by median filtering with a kernel size of 15. Kernel sizes and the number of iterations were chosen by visually examining mask outputs of different combinations until satisfied with the segmentation result. The output tissue mask of a WSI crop region is shown as blue overlay color in Figure 2.2b.

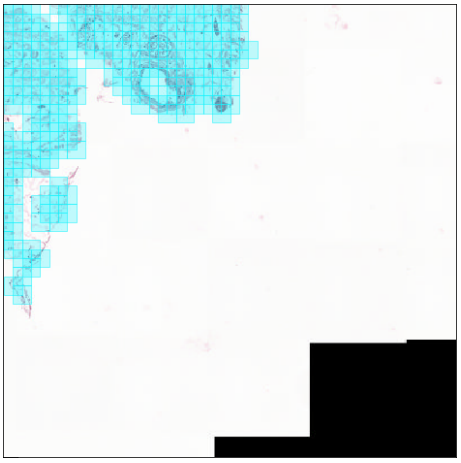
The tissue areas were further sampled into 256x256 pixel red, green and blue image channels (RGB) patches. The sampling was done in overlapping manner so that corners of one tile were the center coordinates of neighboring tiles. The resulting patches are visualized in Figure 2.2c where each blue tile is a sampled position. To collect tissue samples of different magnification levels, 256x256 sized images were sampled from each tile center coordinate with different downsampling factors of 16,8,4,2, and 1 (Figure 2.3). The label for binary classification was determined by the percentage of tumor area (shown in green in Figure 2.3) within the tile without downsampling. If tumor covered 75% or more of the area, the label was "tumor" and "normal" otherwise. The same label was used for all other downsampling variants.



(a) Complete original WSI. The red rectangle is the cropped area in Figures b and c. The scanner's imaging software is coloring empty areas as black to save in file size.



(b) WSI crop region where the segmented tissue areas are colored in blue.



(c) WSI crop region where the sampled tissue patches are shown as blue tiles.

Figure 2.2: Tissue area sampling from WSIs.

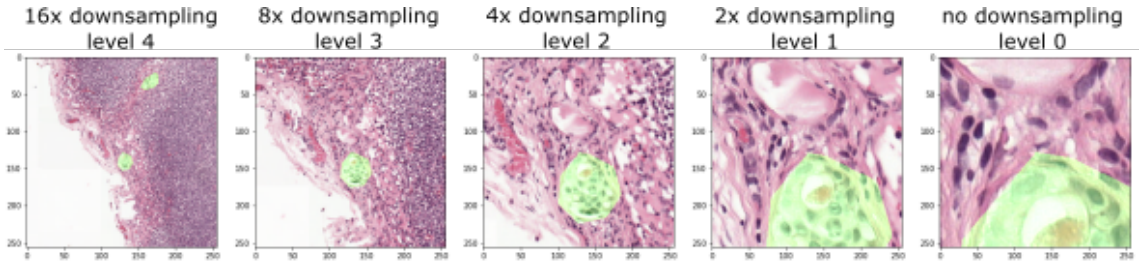


Figure 2.3: WSI crop patches of size 256x256 pixels from the same center coordinate and with different downsampling factors. The green masked areas are ground truth tumor area polygon annotation from the CAMELYON17 data set.

2.2 Normalization

To correct the color staining variation between medical centers and individual WSIs, some of the experiments were done with color stain normalized patch images. Normalization was done by separating the stain vector components of hematoxyling and eosin and normalizing their quantities [53]. The effect of stain color normalization can be seen in Figure 2.4.

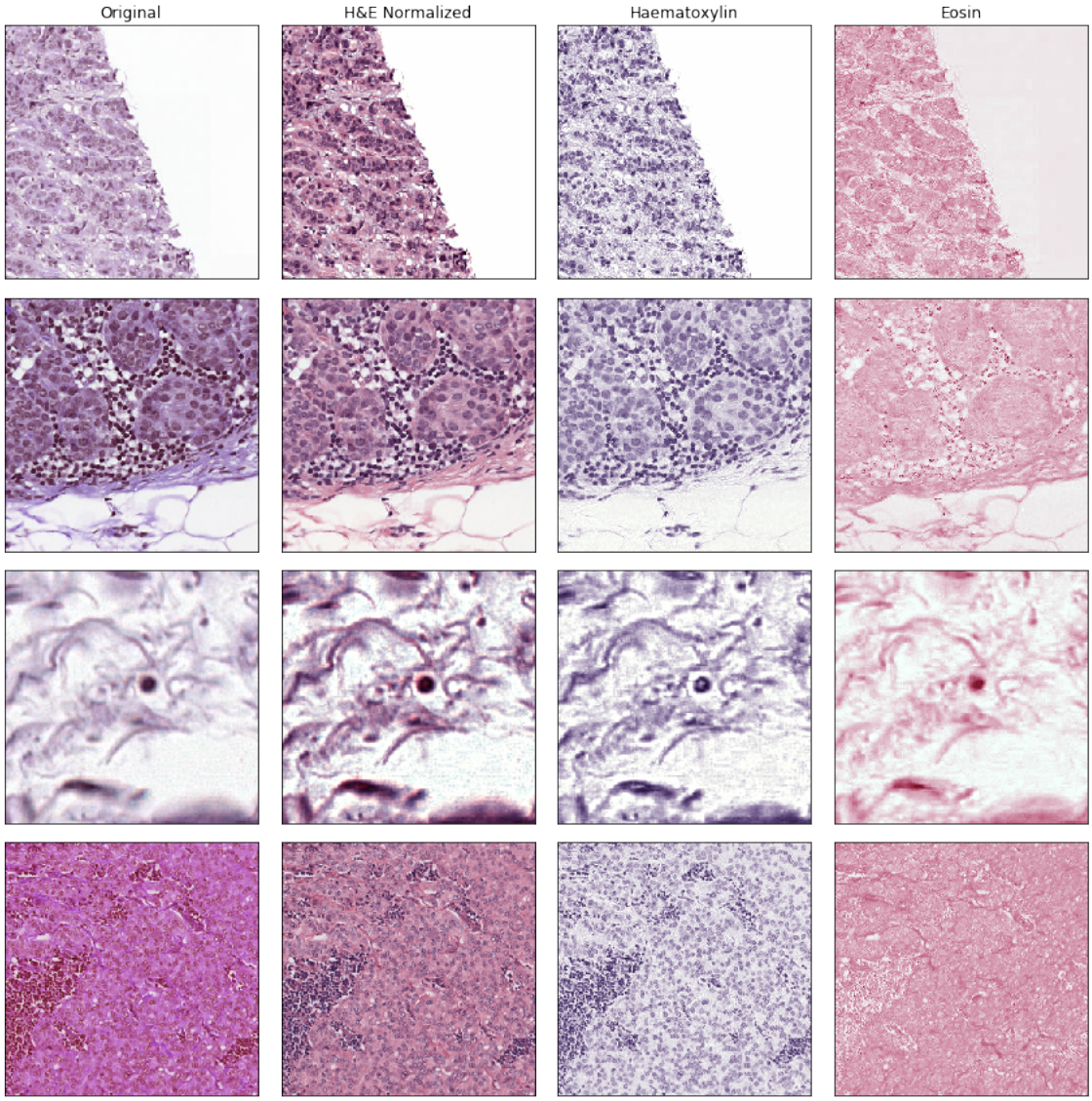


Figure 2.4: Stain normalization. The first column shows original samples, the second column stain normalized version, the third column only the Hematoxylin stain component and the fourth only the Eosin stain component.

2.3 Tasks and evaluation metrics

The work focused on two objectives, binary tumor classification of cropped tissue area tiles and WSI tumor area segmentation. For the binary tumor versus normal tile classification, the area under the receiver operating characteristic curve (ROC

curve) or AUC was used for the evaluation and model comparison.

For measuring the segmentation performance, intersection over union (IoU) and Dice similarity coefficient (DSC) were both used in comparing models segmentation accuracies. These metrics are positively correlated, so both metrics will yield similar rankings between models when comparing results of single inference tests. However, when taking an average of multiple tests, IoU penalizes outliers more than DSC.

2.4 Model

Models that were used in WSI region tumor binary classification consisted of backbone CNN feature extractor and a classification module. Feature extractor part took an RGB image as input downsized the width and height dimension to a small 7x7 while appending the channel dimension through a chain of convolutional and pooling layers. Finally, an adaptive average pooling layer was used to flatten activations to a vector of fixed length. This lost all spatial information and produced a feature vector representing averaged features of the whole input image area.

Backbone's features were fed to a classification head module that consisted of three blocks of 1D batch normalization layer [54], dropout layer [55], fully-connected linear layer, and ReLU activation. Dropout was kept high for regularization effect, so the final dropout layer's probability was 0.9, and the rest were 0.45. Softmax activation was used for the final activation, and it produced two probability values; for normal and tumor.

2.4.1 Multilevel CNN Model

The Multilevel model followed the same configurations of CNN backbone feature extractor and classification module head, but instead of a single backbone feature extractor, the model had two. Feature extractors were meant for two images of

the same region but in different magnifying levels. One with higher magnification was called a focus encoder, and the one with lower magnification was a context encoder. Both of them took the input in the same size and produced flattened feature vectors of fixed length. Before feeding these to a classification head, the vectors were concatenated so that the feature vector's length was doubled. This was followed by a classification head module similar to a regular one-level model except that the first fully-connected linear layer had two times more input features. The overview of the multilevel architecture is visualized in Figure 2.5.

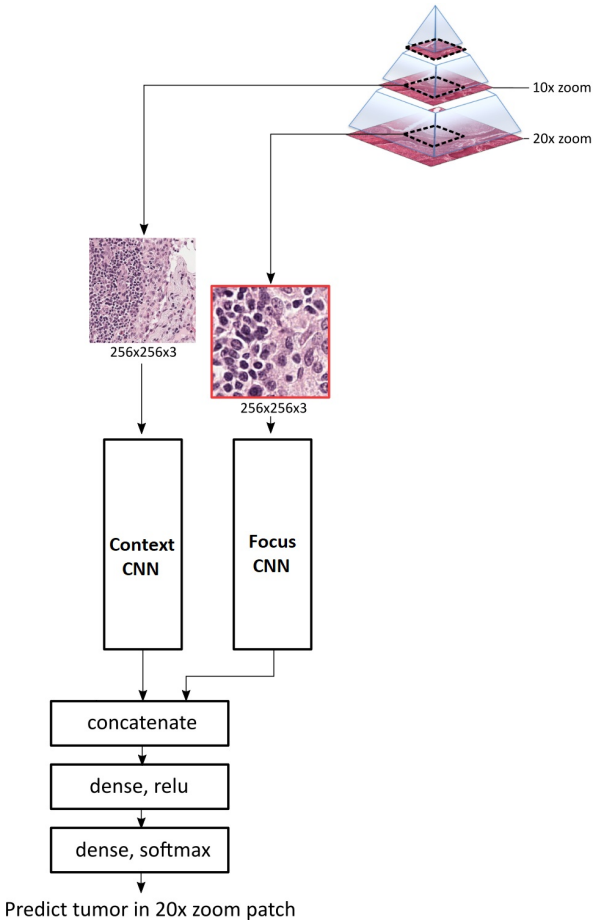


Figure 2.5: Model architecture overview. The model takes two inputs; context and focus images centered at the same slide region. Both inputs go through their own feature extracting encoders and these are concatenated to a classification head module consisting of two fully connected linear layers.

2.4.2 Backbone architectures

Different backbone base architectures were tested for feature extraction in the binary classification task. The architectures included ResNet, Densenet, Inception-ResNetV2, SENet and SE-ResNeXt base models [31], [56]–[58]. ImageNet top-1 and top-5 classification accuracies of their ImageNet pre-trained weights are given in Table 2.1. All base architectures were loaded with ImageNet weights. ResNet and Densnet weights were loaded from Torchvision's repository and the rest from Cadene's repository [59]–[61].

Table 2.1: Base architecture ImageNet accuracies

Architecture	Top-1 accuracy %	Top-5 accuracy %
SENet154	81.30	95.50
InceptionResNetV2	80.40	95.30
SE-ResNeXt101-32x4d	80.24	95.03
SE-ResNeXt50-32x4d	79.08	94.43
ResNet101	77.37	93.56
ResNet50	76.15	92.87
Densenet-169	76.00	93.00
Densenet-121	74.65	92.17
ResNet34	73.30	91.42
ResNet18	69.76	89.08

2.5 Training implementation

The base architectures were loaded with pre-trained Imagenet weights. It was assumed that these weights were already adjusted for extracting meaningful features out of images and thus giving a better starting point towards new tasks. It has

been shown that this type of transfer learning nearly always achieves better results compared to training from scratch [62]. Classification head module weights were initialized with Kaiming initialization, and to keep base architecture weights from dispersing while adjusting the head, all weights of base architecture were frozen during the first training phase called head training phase [63]. After head training, all weights were unfrozen, and the training was continued with a lower learning rate in the finetuning phase.

Training phases were performed with one cycle learning rate and momentum schedules, and a different number of epochs from one to ten were tried to find the minimum epochs for training convergence. Adam optimizer was used, and the maximum learning rate was determined for each model and training phase using a method called learning rate finder [28], [30]. In this method, the model is trained by gradually increasing the learning rate with every training batch. Training loss will decrease until the learning gets too high and the loss diverges. The optimal learning rate is selected from a point before the loss diverged where the learning rate is dropping the fastest.

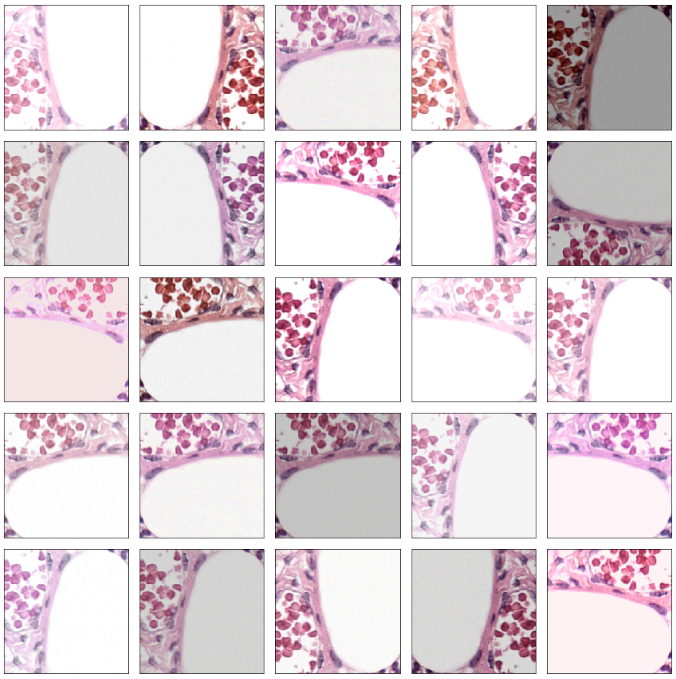
2.5.1 Augmentation

Image augmentations were applied randomly during the training as a way of regularization. The types of augmentations were chosen to mimic variations that could occur naturally due to differences in staining procedure or sample preparation. Albumentations image processing library [64] was used for random hue, saturation, and color value transforms, random Gaussian noise, random flips (horizontal and vertical), random 90-degree rotations, random brightness (limit 0.2), contrast (limit 0.2), and gamma (limit from 50 to 200) variations. The effect of these randomly applied variations is shown in figure 2.6 where the light set of augmentations had hue, saturation, and lightness shift limits as 15, 20, and 20 respectively. Strong color

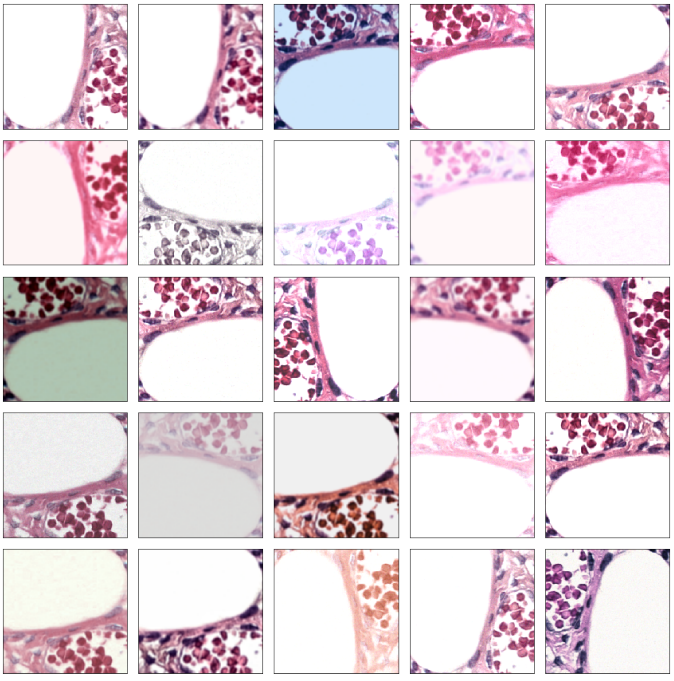
augmentations had 30, 30, and 20 respectively, but included also contrast limited adaptive histogram equalization (CLAHE) and RGB shift (red, green and blue shift limits of 30,15 and 30) augmentations.

In addition to color augmentations, a more sophisticated staining appearance variation method was applied. This method unmixes the hematoxylin and eosin components similarly as in normalization, but instead of normalizing their levels, stain component ratios is changed in a random manner and the resulting outputs is shown in 2.7.

Augmentations were not applied for validation or test sets. For multilevel models that receive two different magnifications images, the same augmentations were always applied for both of them.



(a) Light augmentation set.



(b) Strong color augmentation set.

Figure 2.6: Different augmentation applied to the same image.

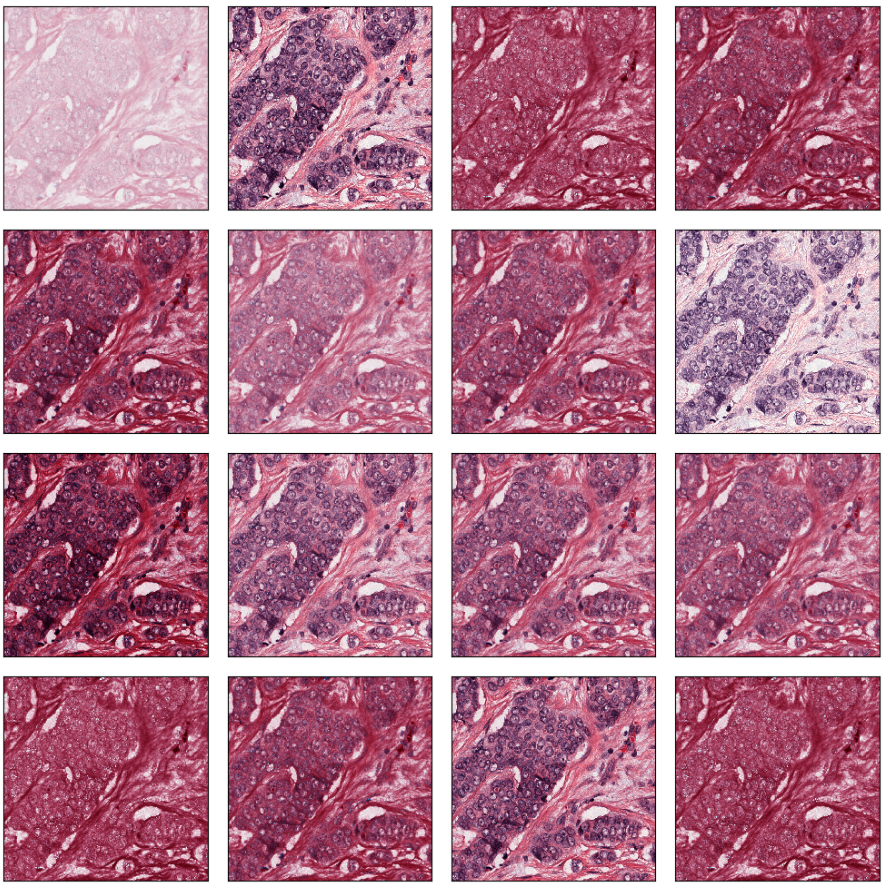


Figure 2.7: Stain appearance augmentation applied to the same image. The method unmixes hematoxylin and eosin color components and randomly alters their ratio.

2.5.2 Autoencoder pre-training

The effect of unsupervised pre-training was tested for the context backbone branch weights of the multilevel CNN model. An autoencoder model was constructed by stacking a dense encoding layer that compressed the output to 512 values and multiple upsampling blocks after the final pooling layer of backbone architecture. Upsampling block consisted of a bilinear upsampling, convolutional layer with a stride of one, ReLU activation, and batch normalization, except for the last block, which had sigmoid activation and no batch normalization. Four upsampling blocks were

used to get an output size same as the input size. The autoencoder model was trained with randomly augmented context-level WSI patch samples to produce an output similar to the input. The training was continued until the mean square error loss converged to levels of 1e-3.

Two different approaches were tried for using autoencoder training in the context branch. In the first one, only the backbone architecture part (Se-ResNeXt50) from the encoder part of the autoencoder was used for context branch. In the second, "bottleneck" layers were also included. The "bottleneck" layers were the convolutional blocks between the backbone and upsampling blocks. These reduced the output size of the context branch from 2048 to 64 and limited the amount of information that passes through the context branch. Hence, the term "bottleneck".

2.6 Patch classification

Only 0.2 to 0.5 percent of sampled tissue patches were labeled as tumors, as shown in Table 2.2. To balance the class distributions, an equal amount of normal patches as there were tumor patches were randomly selected. This was done separately for each medical center, so the total number of samples in each center was twice the number of tumor samples from that center. Re-sampling both training classes equal amounts skewed the training class distribution from the real world, but this was done to enforce equal training opportunities for both classes and to avoid the risk of model learning to predict only the majority class.

Patch classification models predicted tumor probability from which the AUC score was measured. Medical center cross-validation AUC scores were used in parameter tuning and model selection, and the test set was reserved for comparing models that were trained on all training medical centers. Model training with all training medical centers was repeated for five times with different random seeds to capture the training variance.

Table 2.2: Patch image statistics

Medical center	Tumor patches	Total patches
Center 0	12974	4157654
Center 1	6042	7173904
Center 2	18620	6759188
Center 3	29719	6558462
Center 4	4593	2143729

2.7 WSI segmentation

Segmentation was performed using the trained patch classification models by splitting the tissue area into smaller overlapping patches, similar to the ones used for patch classification. The classification model gave a tumor probability estimate for each tile center location, and these estimates were stitched back to a probability map and resized to original WSI dimensions using linear interpolation. Finally, the probability map was thresholded to a binary mask with a value selected based on training set fold cross-validation.

For testing, models were trained on all training set medical centers in a non-deterministic fashion, and the training was repeated five times with different random seeds to capture the training variance. Three of the test set slides had annotated tumor regions, and they were used in determining the segmentation performance. Non-tumor test slides were left out from testing since metrics such as DSC and IoU require the presence of two classes in the reference samples to have meaningful values. AUC was measured tile-wise before thresholding the predicted probabilities.

2.8 Statistical analysis

One-way ANOVA was used for testing the null hypothesis that models had similar classification or segmentation performances. The five training and testing rounds from each model gave a distribution of performance metrics that were compared between different models. ANOVA tested the hypothesis that the resulting scores of different models had the same mean of distribution. ANOVA was chosen because it was assumed when training the same model with different random seeds, the resulting test scores would approximately follow a normal distribution, and different models would have roughly the same variances.

2.9 Hardware and software

Models were trained in 64bit Ubuntu 16.04.6 LTS operating system inside Anaconda virtual environment using Python 3.6 programming language. PyTorch 1.1.0, Torchvision 0.3.0, and Fastai 1.0.52 were used for CNN model training and OpenSlide 3.4.1, ASAP 1.8, OpenCV 4.1.0 and Albumentations 0.2.3 were used for scanner tiff file reading, image processing, and image augmentations.

The utilized hardware had four Nvidia Tesla V100 (32 GB) graphics cards, but only one was used per training. The central processing unit was Intel Xeon Platinum 8160 (2.10 GHz), and the machine had 1510 GB of RAM and a 6.4 TB NVMe SSD.

3 Results

3.1 Autoencoder training

Figure 3.1 shows the reconstruction quality of the autoencoder model. The first column is the input samples, the second the autoencoder's output, and the third column shows the training target, which is the same as the input. All shown samples are taken from a validation fold that was unseen during model training.

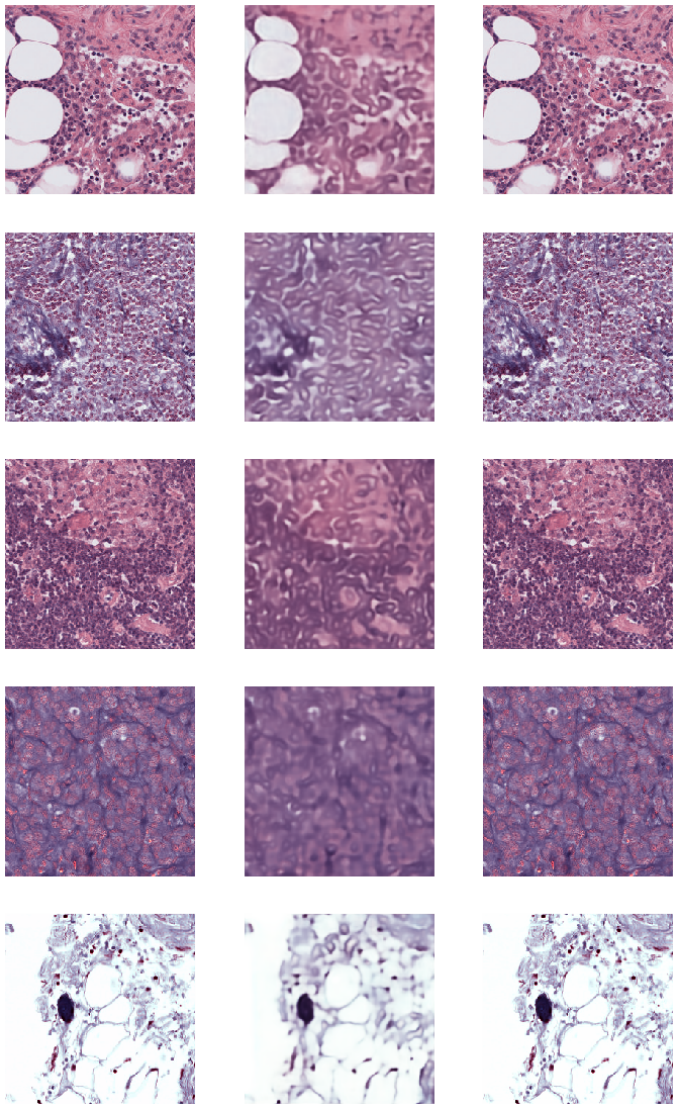


Figure 3.1: Autoencoder training results. The first column shows input samples, the second shows the autoencoder's output, and the third column shows the training target which is the same as the input.

3.2 Patch classification

Patch classification results include cross-validation scores from parameter and model optimization. Conventional CNN binary classification models that predict the class from a single input image are referred to as baseline models. Best baseline model

architectures were compared to multilevel alternatives.

3.2.1 Baseline model optimization

Baseline model tuning was performed using leave-medical-center-out cross-validation to find good base model architecture and a number of epochs to train classification head and base model, learning rate (LR), and whether to normalize the input stain components. Model selection and parameter tuning were made by changing one component at a time, and the average AUC score of folds of each test is shown in Table 3.1. Id value was given for tracking purposes, and suffixes N and A refer to input stain normalization and heavy color augmentations, respectively. Epoch and LR columns are separated to classification head only, and full model training values and the last three models were only trained from the classification head part.

Parameter tuning was started from a DenseNet121 backbone architecture as it is from the lighter end of models and relatively fast to train. The number of epoch search was started high and decreased gradually, and the best performance was achieved only with a single epoch of classification head training. Out of the tried backbone architectures, Se-ResNeXt101 32x4d and InceptionResNetv2 gave good scores. Se-ResNeXt101 32x4d was chosen for multilevel models. Adding heavy color augmentations gave better scores (average area under the ROC curve (AUC) of 97.356) compared to stain input normalization (average area under the ROC curve (AUC) of 96.045) in the final tests with Se-ResNeXt101 32x4d.

Table 3.1: Baseline parameter tuning results

Id	Model	Epochs	LR	Normalized	avg. AUC
01	DenseNet121	10/10	3e-3/1e-5	No	93.021
02	DenseNet121	8/4	3e-3/1e-5	No	94.560
03	DenseNet121	4/2	3e-3/5e-6	No	94.818
04	DenseNet121	4/2	3e-3/5e-6	Yes	94.886
05	DenseNet169	4/2	3e-3/5e-6	No	92.901
06	SENet154	4/2	3e-3/5e-6	No	95.734
07	InceptionResNetv2	4/2	3e-3/5e-6	No	96.305
07N	InceptionResNetv2	4/2	3e-3/5e-6	Yes	96.128
08	Se-ResNeXt101 32x4d	4/2	3e-3/5e-6	No	96.198
08N	Se-ResNeXt101 32x4d	4/2	3e-3/5e-6	Yes	96.302
10	Se-ResNeXt101 32x4d	1/-	1e-3/-	No	96.852
10N	Se-ResNeXt101 32x4d	1/-	1e-3/-	Yes	96.045
18A	Se-ResNeXt101 32x4d	1/-	1e-3/-	No	97.356

3.2.2 Multilevel model optimization

Compared to baseline models, multilevel models had two CNN feature extraction branches instead of one. The context and the focus branch, and different backbone model architectures were chosen for both of them. It was assumed that the context branch wouldn't need as heavy network architecture for feature extraction as the focus branch since it provided only supportive information about the surroundings.

Table 3.2 shows the results of multilevel model parameter and model selection. Only the classification head was trained for one epoch, and the backbones were kept frozen. All backbone models had pre-trained ImageNet weights, but three of the context (ctx.) branch models were autoencoder pre-trained in an unsupervised

manner. These are marked in the "Ctx. lvl./AE" column, where the first value tells the context input magnification level, and the second whether or not the context branch was autoencoder pre-trained. Id value was given for tracking purposes, and suffixes N tells if the inputs were stain normalized.

Table 3.2: Multilevel parameter tuning results

Id	Ctx. model	Focus model	LR	Ctx. lvl./AE	Norm.	avg. AUC
09	ResNet18	ResNet50	3e-3	3/No	No	96.575
11	ResNet34	ResNet101	3e-3	3/No	No	94.336
12N	ResNet18	ResNet50	1e-3	3/No	Yes	96.955
13N	Se-ResNeXt50	Se-ResNeXt101	1e-3	3/No	Yes	97.262
14N	Se-ResNeXt50	Se-ResNeXt101	2e-3	2/No	Yes	97.431
15N	Se-ResNeXt50	Se-ResNeXt101	2e-3	0/No	Yes	96.150
16N	Se-ResNeXt50	Se-ResNeXt101	2e-3	2/Yes	Yes	98.240
16	Se-ResNeXt50	Se-ResNeXt101	2e-3	2/Yes	No	98.878
17N	Se-ResNeXt50	Se-ResNeXt101	2e-3	2/Yes	Yes	98.125

3.2.3 Combined optimization results

AUC scores from all folds of all model optimization runs are shown in Figure 3.2. Se-ResNeXt101 backbone architecture was chosen to represent baseline performance based on its higher AUC compared to other architectures. Figure 3.3 is the same graph but with fold averages, and it gives a clearer picture of the cross-validation AUC between different training runs. The best baseline model (Id 18A) reaches AUC score of 97.356, which is highlighted as red dotted vertical line. The best multilevel model (ID 16) reaches AUC of 98.878, and these fold models were trained without normalization and had autoencoder pre-trained context branch without bottlenecks.

Fold 2, which uses the medical center 2 for validation, stands out with lower

AUC scores (Figure 3.4). Medical center 2 differs from other training folds by having a different scanner. A DenseNet121 baseline model ID 03 has the best AUC scores in fold-2 out of baseline models but still leaves behind the best multilevel model by a margin of over one.

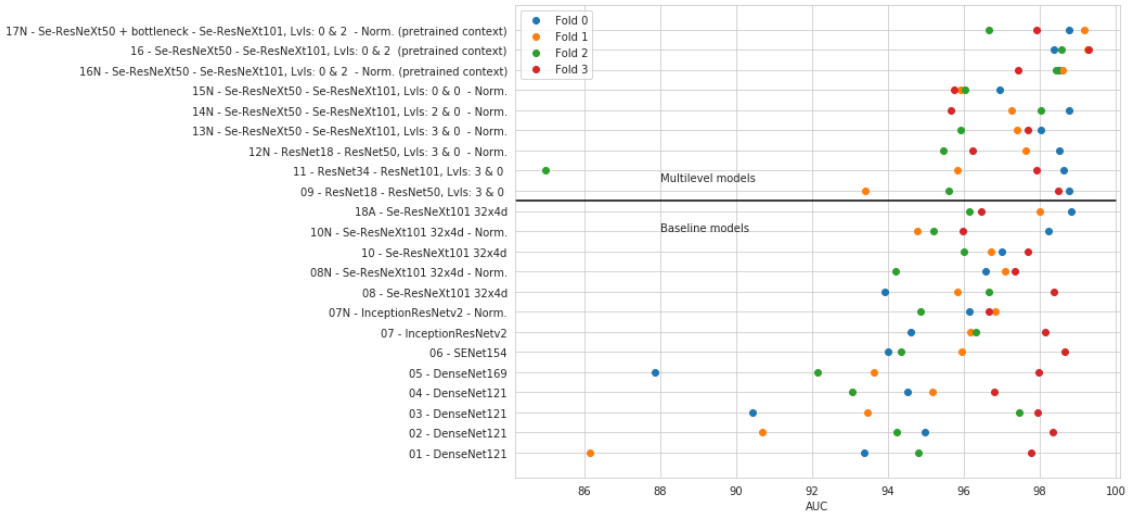


Figure 3.2: AUC scores of each training fold from all baseline and multilevel training runs. Model labels include the id, backbone information, input magnification level, and notes about stain normalization or autoencoder pre-training (pretrained context). Fold number tells which medical center was used for validation.

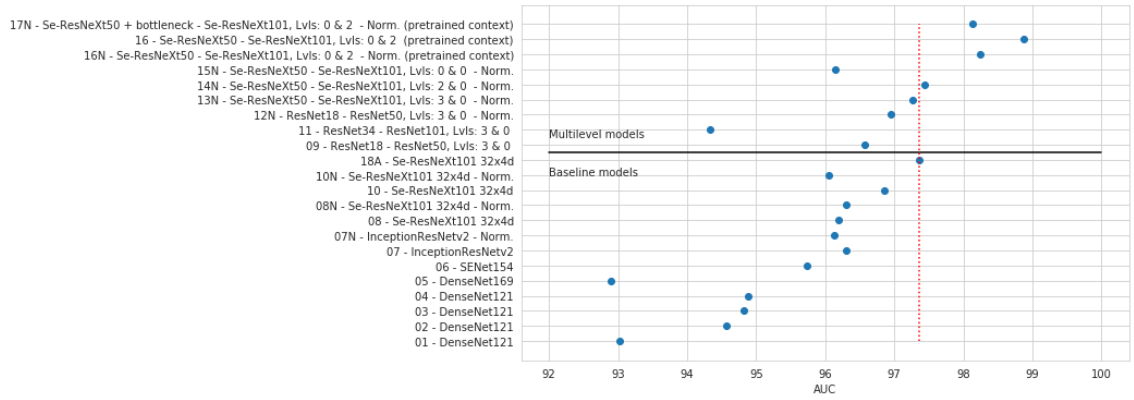


Figure 3.3: Average AUC scores from all folds. The red horizontal dotted line shows the best baseline performance.

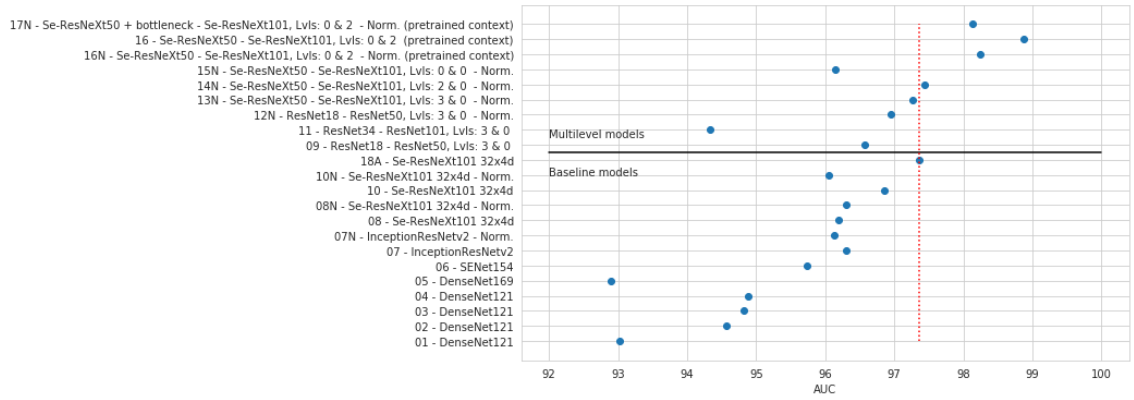


Figure 3.4: AUC scores where medical center 2 was used as the validation fold. This center had a different scanner than the other training fold centers. The red horizontal dotted line shows the best baseline AUC performance.

3.2.4 Grad-CAM visualizations

Class-specific Grad-CAM activations were visualized from the baseline model 3.5 and multilevel model to discover potential data biases and see what regions the models are focusing and basing their predictions. 3.6

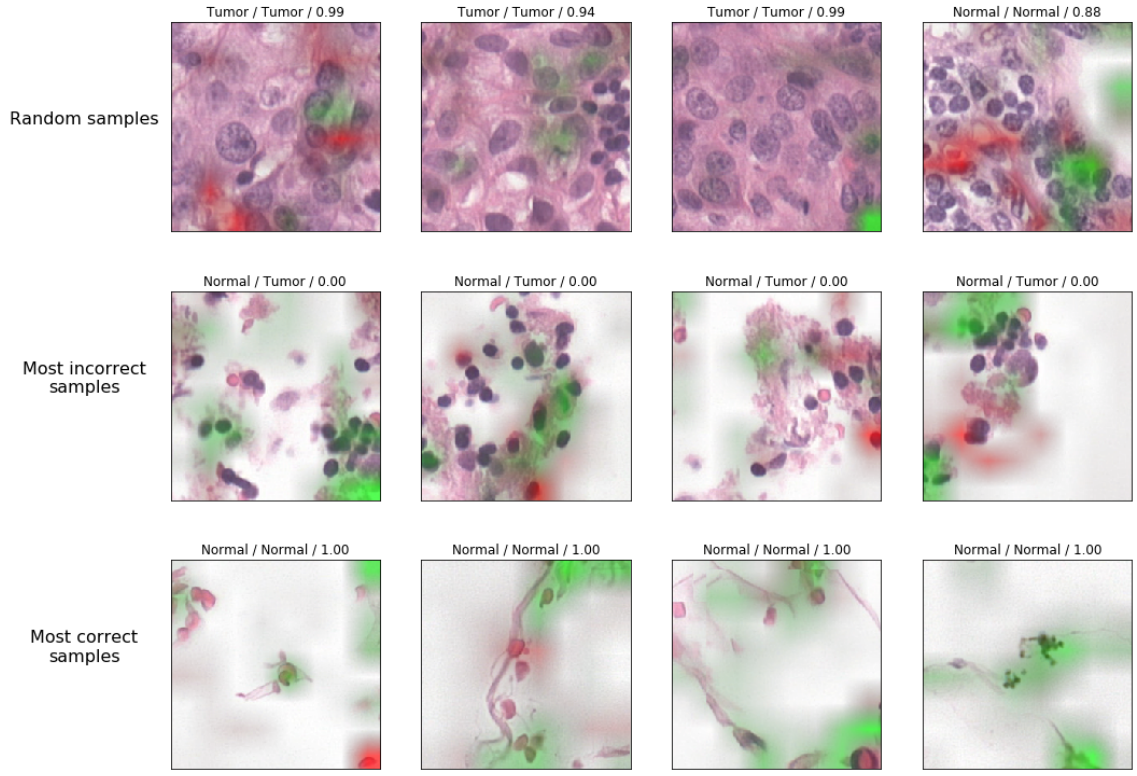


Figure 3.5: Baseline model's class activation maps of validation fold samples. Non-tumor activation regions are overlaid with green and tumor activation regions with red. Titles display (Predicted label/ Actual label/ Predicted tumor probability). First row shows predictions from random patches, second from patches with highest losses (most incorrect), and third from lowest patches with lowest losses (most correct).

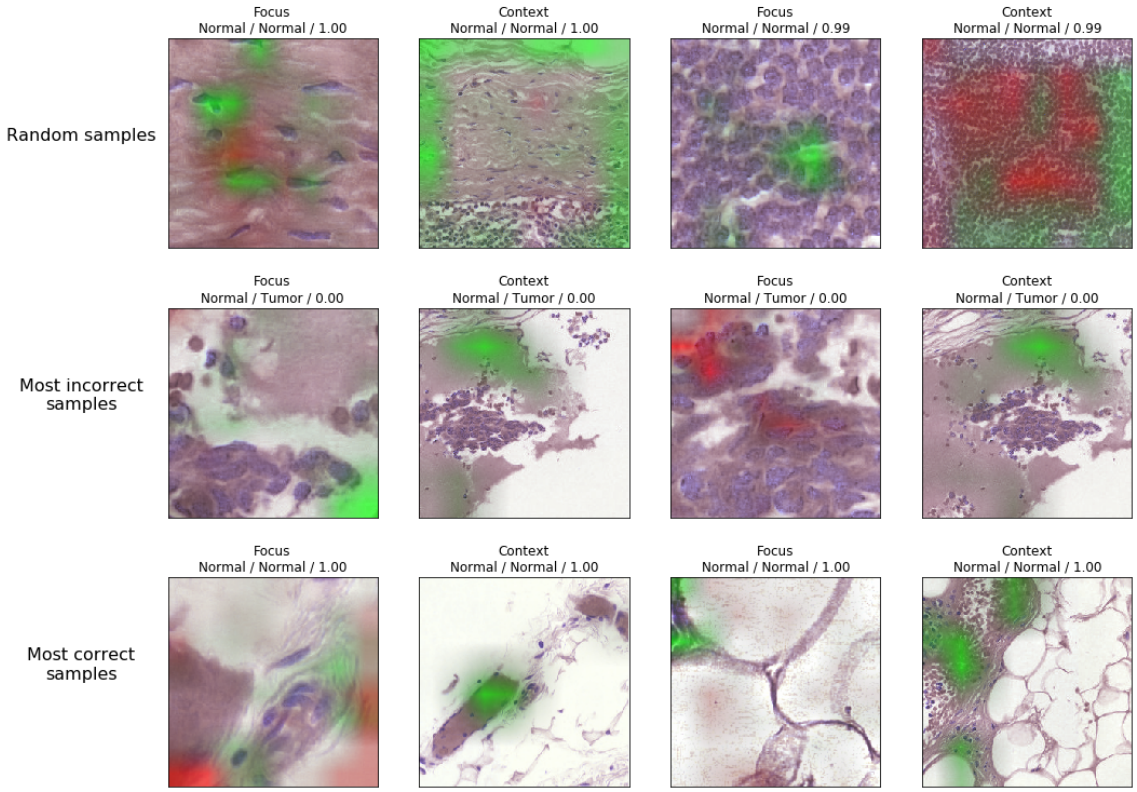


Figure 3.6: Multilevel model's class activation maps of validation fold samples. Non-tumor activation regions are overlaid with green and tumor activation regions with red. Titles display whether the sample is from focus or context branch and (Predicted label/ Actual label/ Predicted tumor probability). First row shows predictions from random patches, second from patches with highest losses (most incorrect), and third from lowest patches with lowest losses (most correct). The first two and last two columns of each row are from the same sample, and they display activations from focus and context branches separately.

3.2.5 Test set results

Table 3.3 shows the average AUC results of multilevel and baseline models. The average is taken from five non-deterministic training runs with different random seeds. Only the classification head was trained for one epoch, and the backbones were kept frozen. All backbone models had pre-trained ImageNet weights but some

of the context (ctx.) branch models were autoencoder pre-trained in an unsupervised manner. These are marked in the "AE" column. All baseline models had Se-ResNeXt101 backbones, and all multilevel models had Se-ResNeXt101 for focus and Se-ResNeXt50 for context branch. Id value was given for tracking purposes, and suffixes N tells if the inputs were stain normalized, and A if heavy color augmentations were used during training. Type column is either Multilevel or Baseline and shows the zoom levels for Multilevel models.

Table 3.3: Test set results

Id	Type	LR	AE	Norm.	avg. AUC
10	Baseline	1e-3	No	No	95.886
10N	Baseline	1e-3	No	Yes	95.527
18A	Baseline	1e-3	No	No	97.035
13	Multilevel (3&0)	1e-3	No	No	90.807
13N	Multilevel (3&0)	1e-3	No	Yes	94.103
14	Multilevel (2&0)	2e-3	No	No	95.745
14N	Multilevel (2&0)	2e-3	No	Yes	96.100
15	Multilevel (0&0)	2e-3	No	No	94.518
15N	Multilevel (0&0)	2e-3	No	Yes	96.261
16	Multilevel (2&0)	2e-3	Yes	No	95.396
16N	Multilevel (2&0)	2e-3	Yes	Yes	96.195
17	Multilevel (2&0)	2e-3	Yes	No	94.979
17N	Multilevel (2&0)	2e-3	Yes	Yes	94.679
19A	Multilevel (2&0)	2e-3	No	No	97.765
20A	Multilevel (0&0)	2e-3	No	No	96.478
21A	Multilevel (2&0)	2e-3	Yes	No	97.044

Figures 3.7 shows the test fold AUC results per each model training run and

Figure 3.8 shows the average value of replicate runs. Multilevel model id 19A outperforms the best baseline model id 18A.

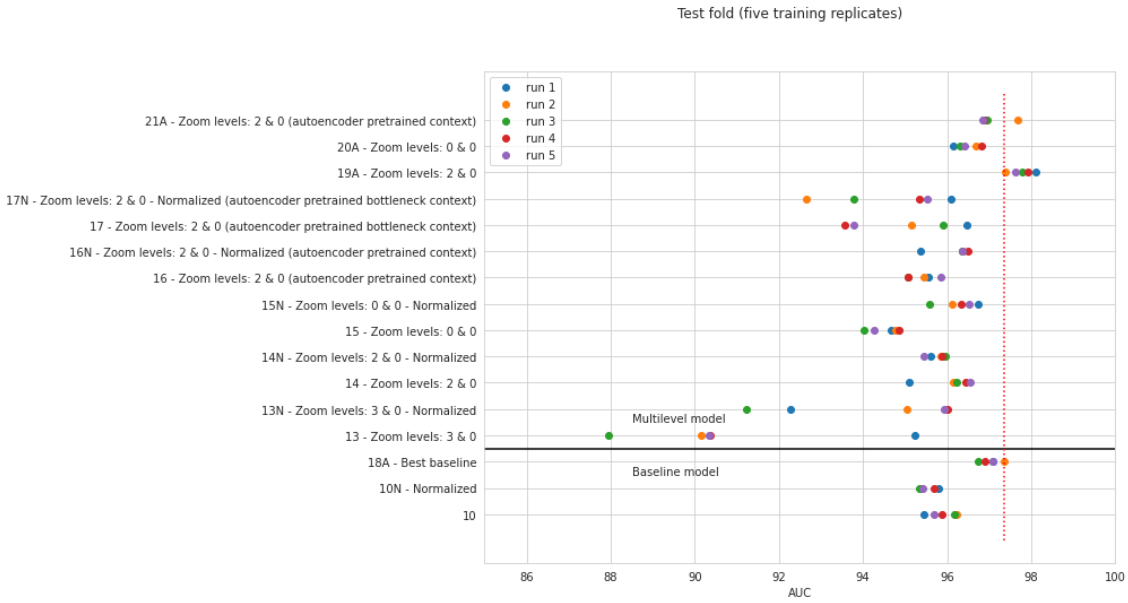


Figure 3.7: AUC scores for test fold from all baseline and multilevel training runs. Model labels include the id, backbone information, input magnification level, and notes about stain normalization or autoencoder pre-training (pretrained context). Each model was trained five times with different random seed and training run scores are shown in different colors. The red horizontal dotted line shows the best baseline run performance.

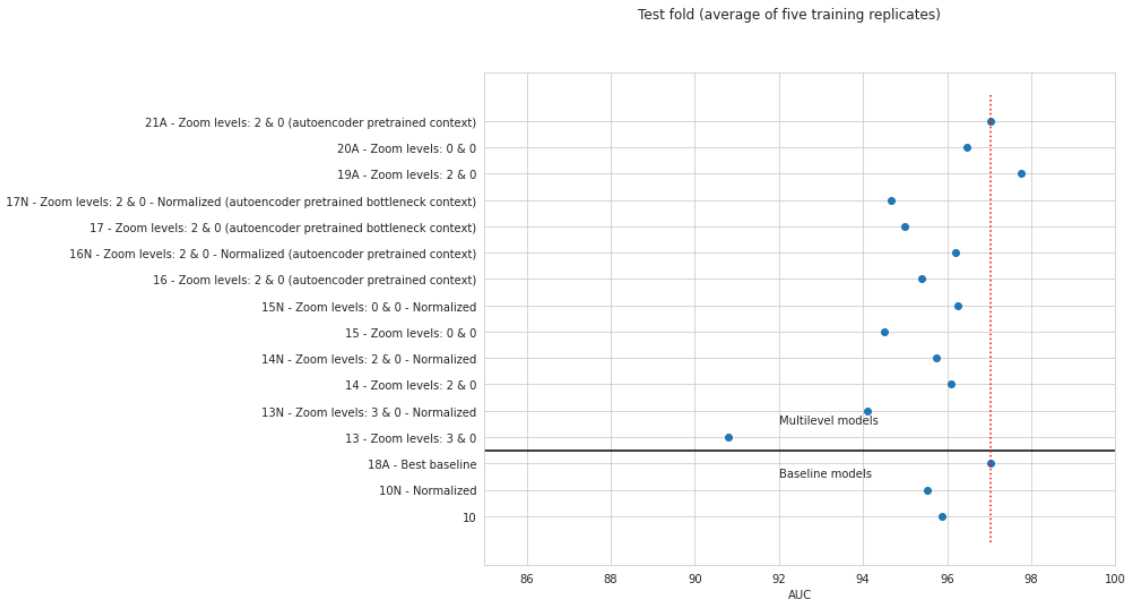


Figure 3.8: Average AUC scores for test fold from all baseline and multilevel training runs. Model labels include the id, backbone information, input magnification level, and notes about stain normalization or autoencoder pre-training (pretrained context). Each model was trained five times with different random seed and the average of runs is show in this plot. The red horizontal dotted line shows the best baseline performance.

The best baseline model ID 18A, best Multilevel model ID 19A, and Multilevel model without context information (ID 20A) were checked against the null hypothesis of all models score equally well. Model ID 18A had an average AUC of 97.035, ID 19A an average of 97.765, and ID 20A an average of 96.478. One-way ANOVA test gave uncorrected p-values of 0.0018 and 0.000068 for comparisons between 18A vs. 19A and 19A vs. 20A. Tukey-HSD corrected p-values were 0.0018 and 0.001, respectively. ANOVA requires that the values are normally distributed, the variances between the groups are equal. Shaphiro-Wilk's normality test showed that all groups were normally distributed with p-values between 0.7 and 0.9 (> 0.05). Levene test confirmed that the three groups had equal variances with a p-value 0.87

(> 0.05).

3.3 WSI segmentation

WSI-specific AUC, DSC and IoU scores are presented in Figures 3.9, 3.10 and 3.11.

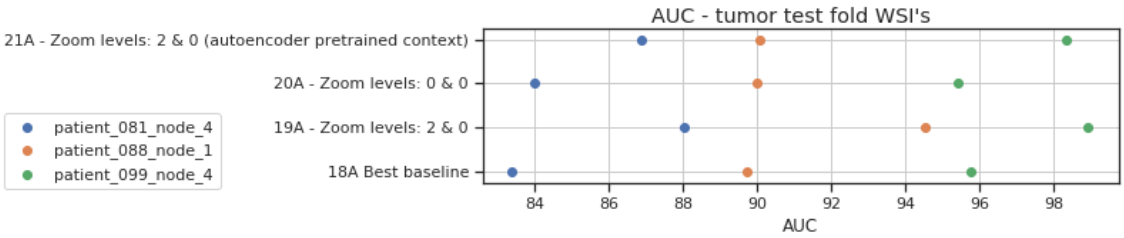


Figure 3.9: AUC scores for three test WSI tile predictions. WSI-specific AUC scores are shown in different colors.

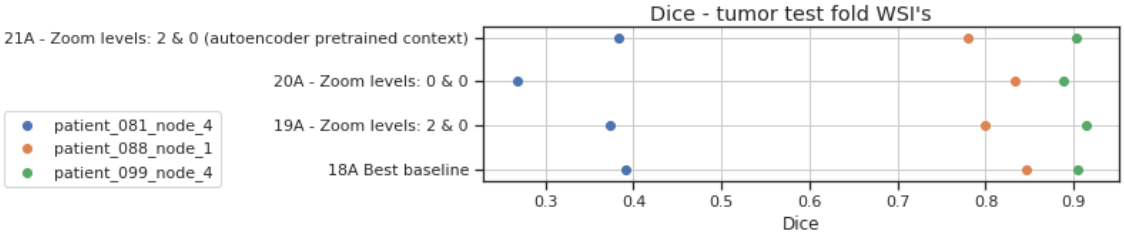


Figure 3.10: DCS metrics for the three test WSI segmentations. WSI-specific scores are shown in different colors.

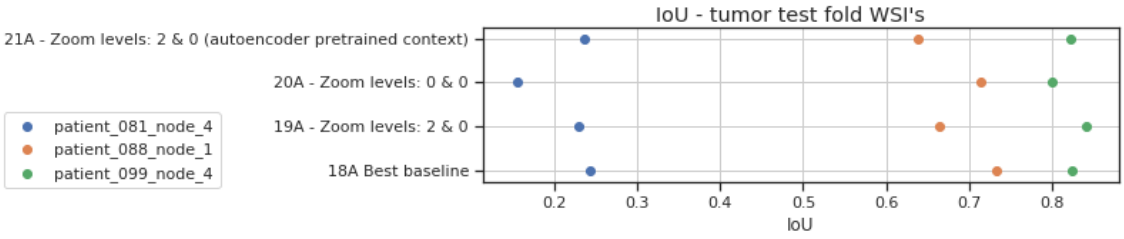


Figure 3.11: IoU metrics for the three test WSI segmentations. WSI-specific scores are shown in different colors.

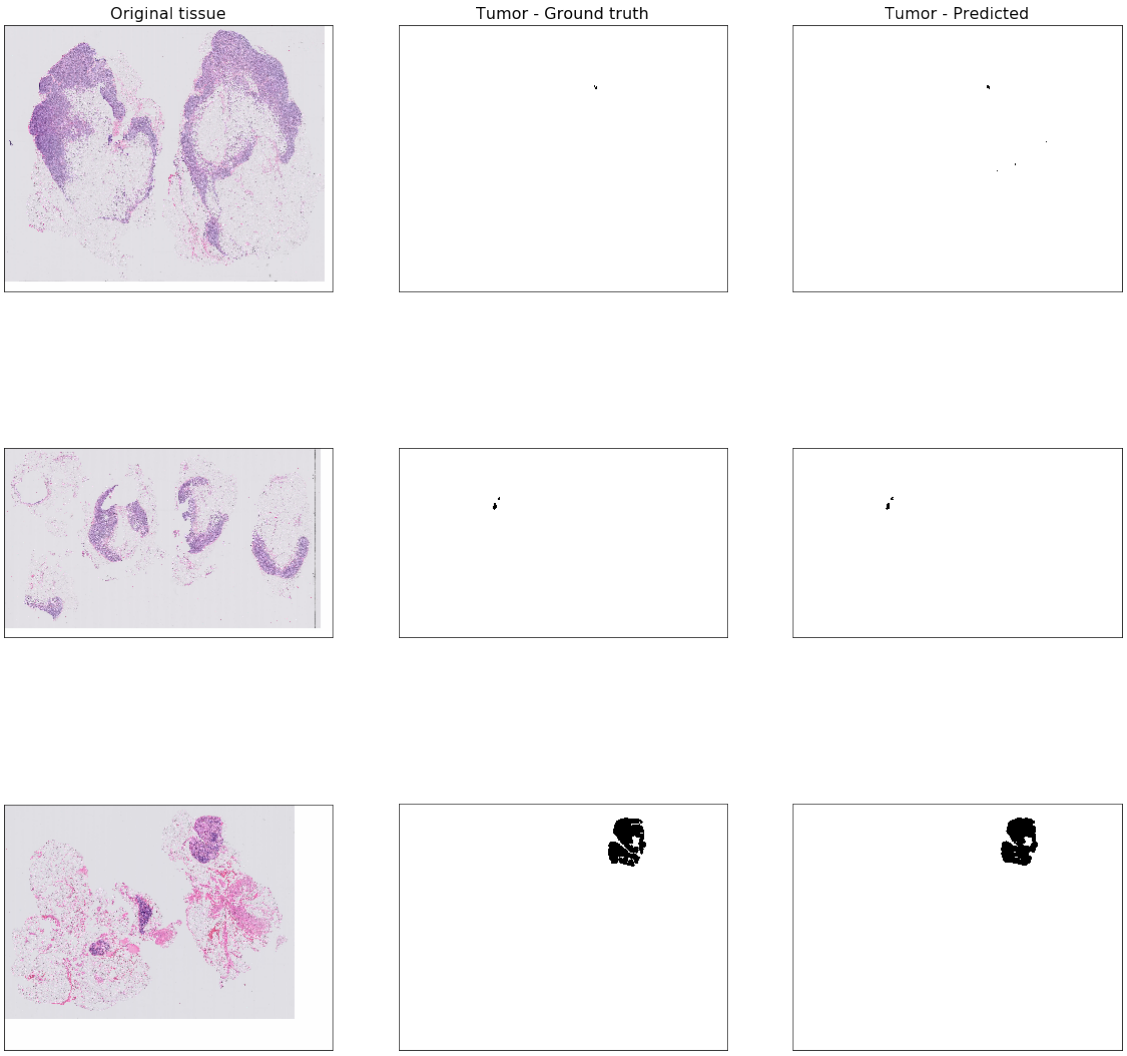


Figure 3.12: Segmentation results of the model 19A for three tumor test set WSI's. The first column shows the original tissue, second column the ground truth tumor annotations and third row the predicted tumor mask. Annotated and predicted tumor regions are colored in black. Rows are test fold WSI samples from top to down order: Patient-081-node-4, Patient-088-node-1 and Patient-099-node-4.

4 Discussion

4.1 Patch classification

The original hypothesis stated that including context-level CNN features together with patch-level CNN features (focus-level) in a patch image classifier would improve the tumor classification performance. When comparing three models; the best performing baseline model with only focus-level features, a multi-level model with focus and context-level features, and a multi-level model with two separate focus-level features, the focus and context-level model has the highest test set AUC (Figure 3.7), and it differs in scores from the other two statistically significantly.

A multi-level model with two separate focus-level feature branches performed the worst from the three models. This shows that the high score of the context and focus model was not caused by double the amount of features going to the classifier. On the contrary, doubling the number of classifier features seems to deteriorate the performance here. Even though the two focus-level branches had different feature encoders, Se-ResNeXt50 and Se-ResNeXt101, the two branches likely had many near-duplicate features, which did not increase the representative capacity. Furthermore, adding features that do not contribute much additional information would probably need more training iterations to reach a similar performance with more compressed feature representation.

The tumor regions marked by pathologists often covered greater than single

image patch-sized areas. This meant that for a tumor patch image, at least some of the neighboring patches in the context range had tumor as well. Thus, tumor signals would have been amplified when including context features, and this could be one reason why the context information helped. Another explanation could be availability of larger-scale features. Since context features were extracted from a less magnified scale and the input image size was kept the same, the CNN module had a higher receptive field in terms of micrometers and could potentially see patterns of larger scale.

The risk of including context-level features in patch-level classification increased ambiguity, especially in the border regions of a tumor. Class label is assigned only based on the focus region, but the CNN context features have no way of differentiating what part of the image the features were originating. All spatial information of the context branch is lost in the final pooling layer that compresses the width and height dimensions of features into channel averages. If the center of a context image were the only area free of tumor, the context features would have likely included tumor signals even if the focus patch would have been considered as non-tumor. This could have been addressed by making the context features spatially aware, for instance, by adding coordinate maps such as x- and y-gradient maps as new channels to the input. Similarly, this would have been achieved with using a CoordConv layer in place of one of the context branch convolutional layers [65].

The threshold of the tumor coverage percentage for labeling the patch image as tumor was 75%. Thus, using an average feature pooling instead of maximum pooling was justified to weight more on the whole set of features over any small region's high outliers. The label ambiguity increases when moving away from tumor mass center to tumor borders, and models that can see context will get mixed healthy and tumor signals on both sides of the boundary. Thus, it is possible that context information would impair classification accuracy near tumor borders.

Grad-CAM visualizations in Figures 3.5 and 3.6 show a possible data bias. Both models were the most certain in the border regions of tissue where the glass underneath the tissue was showing partially. Since tissue in these regions was not intact, the tumor annotations were not often present in these parts except for a few cases. Models probably learned to associate tissue border regions as non-tumor because of the class imbalance within similar samples, and thus, made very confident predictions for non-tumor. In the few samples that include a tumor in the tissue border, models made confident and wrong predictions. Grad-CAM visualization show that it is not the white glass part of the image that shows most non-tumor-specific activations but the contours of cells and tissue on the edges of white glass.

Adjusting context level feature encoder with unsupervised autoencoder training before training the classifier did not improve the classification accuracy. Another observation from the path classifier training was that using heavy color augmentations seemed to work better than normalizing stain colors. The point of color normalization was to fade out differences between scanners and different scanning parameters and decrease hardware bias. Stain normalization probably did not remove the bias and still left subtle differences in colorization that the CNN model was able to detect. Randomizing color hue, saturation and intensity was more effective in decreasing hardware bias and increasing the robustness against color differences. The downside of this is that any useful information that the color may hold becomes more difficult to learn. It is worth noting that this work did not explore normalization algorithms extensively. Thus, it is possible that the choice of the normalization method was not optimal.

4.2 WSI segmentation

The multi-level model improvements were not clear when examining WSI-level DSC and IoU scores but on rank-based AUC, multi-level had the best scores in all WSI

samples. The only sample where multi-level DSC and IoU were better than baseline was Patient-099-node-4 which had the largest tumor regions. The smaller the tumor regions were, the worse the DSC and IoU were. This would indicate that tumor prediction mask binarization thresholds that were chosen based on training set folds cross-validation were not generalizing to smaller tumor regions. This is likely since the majority of the patch samples in all folds came from larger tumor regions.

The reason why area-based metrics were worse in small tumor regions of the multi-level model compared to the baseline model is unclear. Smaller tumors have a higher boundary region to total tumor area ratio, so one explanation could be that context branch impairs boundary classification. However, higher AUC for multi-level in all samples, including the small tumors, confirms that the underlying reason is likely a poorly chosen threshold for multi-level models.

It is evident in Figure 3.12 that the multi-level model's segmentation predictions tend to label healthy regions that are covered by tumor masses as tumor. This could indicate that context is taken into account in class label prediction even though the focus patch would not have tumor.

4.3 Other applications for multilevel architecture

Multi-level CNN architecture is not only suitable for medical imaging but could be helpful to a plethora of applications such as remote sensing, defect detection, or whenever an image is reasonable to split into sub-regions for distributed processing. This architecture could suit small defect anomaly detection since the classification could be stated as "do focus features belong to context feature distribution?". Such a network could be trained in an unsupervised manner with the same and different focus-context image pairs similar to Siamese network [66].

The architecture is not limited to two dimensions either. The same approach would also work for 3D and for instance, this could be used for classifying 3D patches

in magnetic resonance imaging scans.

4.4 Future work

The main weakness in multi-level CNN architecture presented in this work was classification accuracy in the boundary regions of two classes. To fix this, one could try different pooling techniques in the final feature pooling of context encoder branch, or other methods that would retain spatial information of context features. The average pooling used here may not be optimal since it dilutes signals from a single spatial region. Mean and maximum concatenated pooling or generalized mean pooling could work better in cases where the context is not a uniform single-class region. Network architecture is not the only option for solving class border region ambiguity. Part of the issue may incur from poor data sampling. Thus, having more examples from borders or giving them more weight in loss calculations might help with border accuracy.

5 Conclusion

In the light of this work, adding a context zoom-level feature extraction branch to tissue classification improves the overall classification accuracy in certain conditions. Overall, the multilevel model reduced classification error by 24.6%; from the lowest baseline error of 2.97% to 2.24%. However, further analysis revealed that the method seems to increase ambiguity in border regions of tumor areas, whereas patches inside larger tumor areas are classified more accurately. Thus, multilevel architecture would be recommended only for WSI segmentation applications where an intact prediction mask is preferred over border accuracy. A conventional single-level approach is likely more suitable for the latter case or when segmenting smaller objects.

In conclusion, the multilevel model is a promising architecture for digital pathology examining large tissue regions. Its main shortcoming of border region ambiguity is possibly solvable by trivial changes in model architecture, such as switching the last pooling layer type to mean and maximum concatenated pooling.

References

[1] WHO, *WHO position paper on mammography screening*. Geneva, Switzerland: WHO Press, 2014.

[2] A. C. Society, *Survival rates for breast cancer*. [Online]. Available: https://www.cancer.org/cancer/breast-cancer/understanding-a-breast-cancer-diagnosis/breast-cancer-survival-rates.html.

[3] C. Carter, C. Allen, and D. Henson, "Relation of tumor size, lymph node status, and survival in 24,740 breast cancer cases", *cancer*, vol. 63, pp. 181–187, 1989.

[4] "Relevant impact of central pathology review on nodal classification in individual breast cancer patients", *The Annals of Oncology*, vol. 23, 2012. DOI: 10.1093/annonc/mds072.

[5] F. Greene, C. Compton, A. Fritz, J. Shas, and D. Winchester, *AJCC cancer staging atlas*. New York, United States of America: Springer, 2006.

[6] K. Sevensma and C. Lewis, *Axillary sentinel lymph node biopsy*. Treasure Island FL: StatPearls Publishing, 2020.

[7] M. Zarella, D. Bowman, F. Aeffner, N. Farahani, A. Xthona, S. Absar, A. Parwani, M. Bui, and D. Hartman, "A practical guide to whole slide imaging", *Archives of pathology & laboratory medicine*, vol. 143 (2), pp. 222–234, Feb. 2019.

[8] N. Atallah, M. Toss, C. Verill, M. Salto-Tellez, D. Snead, and E. Rakha, "Potential quality pitfalls of digitalized whole slide image of breast pathology in routine practice", *Modern Pathology*, Dec. 2021. DOI: 10.1038/s41379-021-01000-8.

[9] B. Lee and K. Paeng, "A robust and effective approach towards accurate metastasis detection and pn-stage classification in breast cancer", 2018. DOI: arXiv:1805.12067.

[10] B. Ehteshami, M. Veta, van Diest P, van Ginneken B, N. Karssemeijer, G. Litjens, V. der Laak J, M. Hermsen, Q. Mason, M. Balkenhol, O. Geessink, N. Stathonikos, V. D. M, P. Bult, F. Beca, A. Beck, D. Wang, A. Khosla, R. Gargeya, H. Irshad, A. Zhong, Q. Dou, Q. Li, H. Chen, H. Lin, P. Heng, C. Hab, E. Bruni, Q. Wong, U. Halici, M. Öner, R. Cetin-Atalay, M. Berseth, V. Khvatkov, A. Vylegzhanin, O. Kraus, M. Shaban, N. Rajpoot, R. Awan, K. Sirinukunwattana, T. Qaiser, Y. Tsang, D. Tellez, J. Annuscheit, P. Hufnagl, M. Valkonen, K. Kartasalo, L. Latonen, P. Ruusuvuori, K. Liimatainen, S. Albarqouni, B. Mungal, A. George, S. Demirci, N. Navab, S. Watanabe, S. Seno, Y. Takenaka, H. Matsuda, A. P. H, V. Kovalev, A. Kalinovsky, V. Liauchuk, G. Bueno, M. Fernendez-Carrobles, I. Serrano, O. Deniz, D. Racoceanu, and V. R, "Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer", *The Journal of the American Medical Association*, vol. 22, pp. 2199–2210, 2017. DOI: 10.1001/jama.2017.14585.

[11] P. Bandi, O. Geessink, Q. Manson, van Dijk M, M. Balkenhol, M. Hermsen, B. Bejnordi, B. Lee, K. Paeng, A. Zhong, Q. Li, F. Zanjani, S. Zinger, K. Fukuta, D. Komura, V. Ovtcharov, S. Cheng, S. Zeng, J. Thagaard, A. Dahl, H. Lin, H. Chen, L. Jacobsson, M. H. M, M. Cetin, E. Halici, H. Jackson, R. Chen, F. Both, and J. Franke, "From detection of individual metastases

to classification of lymph node status at the patient level: The camelyon17 challenge", *IEEE Transactions on Medical Imaging*, vol. 38, no. 2, pp. 550–560, 2018. DOI: 10.1109/TMI.2018.2867350.

[12] J. McCarthy, M. Minsky, N. Rochester, and C. Shannon, "A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955", *AI Mag*, vol. 27 (12), 2006.

[13] Y. Lecun, P. Haffner, and Y. Bengio, "Object recognition with gradient-based learning", Aug. 2000.

[14] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale", Oct. 2020.

[15] J. Yao and L. V. Wang, "Photoacoustic microscopy", *Laser & Photonics Reviews*, vol. 7, no. 5, pp. 758–778, 2013. DOI: https://doi.org/10.1002/lpor.201200060. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/lpor.201200060.

[16] A. Qorbani, F. Fereidouni, R. Levenson, S. Lahoubi, Z. Harmany, A. Todd, and M. Fung, "Muse (microscopy with uv surface excitation): A novel approach to real-time inexpensive slide free dermatopathology", *Journal of Cutaneous Pathology*, vol. 45, Aug. 2017. DOI: 10.1111/cup.13255.

[17] Food and D. Administartation, *Intellisite pathology solution.* [Online]. Available: https://www.fda.gov/drugs/resources-information-approved-drugs/intellisite-pathology-solution-pips-philips-medical-systems.

[18] Paige.AI, *Paige.ai.* [Online]. Available: https://www.paige.ai/technology.

[19] DeepLens, *Deeplens.* [Online]. Available: https://www.deeplens.ai/.

[20] Proscia, *Proscia.* [Online]. Available: https://proscia.com/.

[21] PathAI, *Pathai*. [Online]. Available: https://www.pathai.com/.

[22] Inspirata, *Inspirata*. [Online]. Available: https://www.inspirata.com/.

[23] DeePathology, *Deepathology*. [Online]. Available: https://deepathology.ai/.

[24] B. Kaustav, K. Schalper, D. Rimm, V. Velcheti, and A. Madabhushi, "Artificial intelligence in digital pathology — new tools for diagnosis and precision oncology.", *Nature reviews. Clinical oncology*, vol. 16 (11), pp. 703–715, Nov. 2019.

[25] S. Mukhopadhyay, M. Feldman, E. Abels, R. Ashfaq, S. Beltaifa, N. Cacciabeve, H. Cathro, L. Cheng, K. Cooper, G. Dickey, R. Gill, R. Heaton, R. Kerstens, G. L. R. Malhotra, J. Mandell, E. Manlucu, A. Mills, S. Mills, C. Moskaluk, and C. Taylor, "Whole slide imaging versus microscopy for primary diagnosis in surgical pathology: A multicenter blinded randomized noninferiority study of 1992 cases (pivotal study)", *Am. J. Surg. Pathol.*, vol. 42 (1), pp. 39–52, 2018.

[26] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning", *Nature*, vol. 521, pp. 436–444, 2015.

[27] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge", *International Journal of Computer Vision*, vol. 115, Sep. 2014. DOI: 10.1007/s11263-015-0816-y.

[28] D. Kingma and J. Ba, "Adam: A method for stochastic optimization", *International Conference on Learning Representations*, Dec. 2014.

[29] L. Smith, "Cyclical learning rates for training neural networks", Mar. 2017, pp. 464–472. DOI: 10.1109/WACV.2017.58.

[30] L. Smith, "A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay", Mar. 2018.

[31] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks", in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.

[32] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of tricks for image classification with convolutional neural networks", Jun. 2019, pp. 558–567. DOI: 10.1109/CVPR.2019.00065.

[33] B. E. Bejnordi, M. Veta, P. J. van Diest, B. van Ginneken, N. Karssemeijer, G. J. S. Litjens, J. A. van der Laak, M. Hermsen, Q. F. Manson, M. C. A. Balkenhol, O. G. F. Geessink, N. Stathonikos, M. C. van Dijk, P. Bult, F. Beca, A. H. Beck, D. Wang, A. Khosla, R. Gargeya, H. Irshad, A. Zhong, Q. Dou, Q. Li, H. Chen, H. Lin, P.-A. Heng, C. Hass, E. Bruni, Q. K.-S. Wong, U. Halici, M. Ü. Öner, R. Cetin-Atalay, M. Berseth, V. Khvatkov, A. I. Vylegzhanin, O. Z. Kraus, M. Shaban, N. M. Rajpoot, R. Awan, K. Sirinukunwattana, T. Qaiser, Y.-W. Tsang, D. Tellez, J. Annuscheit, P. Hufnagl, M. Valkonen, K. Kartasalo, L. Latonen, P. Ruusuvuori, K. Liimatainen, S. Albarqouni, B. Mungal, A. A. George, S. Demirci, N. Navab, S. Watanabe, S. Seno, Y. Takenaka, H. Matsuda, H. A. Phoulady, V. A. Kovalev, A. Kalinovsky, V. Liauchuk, G. Bueno, M. del Milagro Fernández-Carrobles, I. Serrano, O. Deniz, D. Racoceanu, and R. Venâncio, "Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer", *JAMA*, vol. 318, pp. 2199–2210, 2017.

[34] B. Lee and K. Paeng, "A robust and effective approach towards accurate metastasis detection and pn-stage classification in breast cancer", *ArXiv*, vol. abs/1805.120 2018.

[35] W. Bulten, K. Kartasalo, P.-H. Chen, P. Ström, H. Pinckaers, K. Nagpal, Y. Cai, D. Steiner, H. Boven, R. Vink, C. Hulsbergen-van de Kaa, J. van der Laak, M. Amin, A. Evans, T. Van der Kwast, R. Allan, P. Humphrey, H. Grönberg, H. Samaratunga, and J. Park, "Artificial intelligence for diagnosis and gleason grading of prostate cancer: The panda challenge", *Nature Medicine*, Jan. 2022. DOI: 10.1038/s41591-021-01620-2.

[36] T. A. Azevedo Tosta, P. R. de Faria, L. A. Neves, and M. Z. do Nascimento, "Computational normalization of h&e-stained histological images: Progress, challenges and future potential", *Artificial Intelligence in Medicine*, vol. 95, pp. 118–132, 2019, ISSN: 0933-3657. DOI: https://doi.org/10.1016/j.artmed.2018.10.004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S093336571830424X.

[37] M. Macenko, M. Niethammer, J. Marron, D. Borland, J. Woosley, X. Guan, C. Schmitt, and N. Thomas, "A method for normalizing histology slides for quantitative analysis.", vol. 9, Jun. 2009, pp. 1107–1110. DOI: 10.1109/ISBI.2009.5193250.

[38] D. Tellez, M. C. A. Balkenhol, I. Otte-Höller, R. van de Loo, R. Vogels, P. Bult, C. A. Wauters, W. Vreuls, S. J. J. Mol, N. Karssemeijer, G. J. S. Litjens, J. A. van der Laak, and F. Ciompi, "Whole-slide mitosis detection in h&e breast histology using phh3 as a reference to train distilled stain-invariant convolutional networks", *IEEE Transactions on Medical Imaging*, vol. 37, pp. 2126–2136, 2018.

[39] F. G. Zanjani, S. Zinger, B. E. Bejnordi, and J. van der Laak, "Histopathology stain-color normalization using deep generative models", 2018.

[40] V. Buhrmester, D. Muench, and M. Arens, "Analysis of explainers of black box deep neural networks for computer vision: A survey", Nov. 2019.

[41] R. Rs, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, "Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization", Oct. 2016.

[42] J. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net", Dec. 2014.

[43] R. Rs, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization", *International Journal of Computer Vision*, vol. 128, Feb. 2020. DOI: 10.1007/s11263-019-01228-7.

[44] D. Steiner, R. MacDonald, Y. Liu, P. Truszkowski, J. Hipp, C. Gammage, F. Thng, L. Peng, and M. Stumpe, "Impact of deep learning assistance on the histopathologic review of lymph nodes for metastatic breast cancer", *The American Journal of Surgical Pathology*, vol. 42, p. 1, Oct. 2018. DOI: 10.1097/PAS.0000000000001151.

[45] S. Genyun, H. Huang, A. Zhang, F. Li, H. Zhao, and H. Fu, "Fusion of multiscale convolutional neural networks for building extraction in very high-resolution images", *Remote Sensing*, vol. 11, p. 227, Jan. 2019. DOI: 10.3390/rs11030227.

[46] Y. Sun, L. Zhu, G. Wang, and F. Zhao, "Multi-input convolutional neural network for flower grading", *Journal of Electrical and Computer Engineering*, vol. 2017, pp. 1–8, Aug. 2017. DOI: 10.1155/2017/9240407.

[47] M. Längkvist, A. Kiselev, M. Alirezaie, and A. Loutfi, "Classification and segmentation of satellite orthoimagery using convolutional neural networks", *Remote Sensing*, vol. 8, p. 329, Apr. 2016. DOI: 10.3390/rs8040329.

[48] T. Panboonyuen, K. Jitkajornwanich, S. Lawawirojwong, P. Srestasathiern, and P. Vateekul, "Semantic segmentation on remotely sensed images using

an enhanced global convolutional network with channel attention and domain specific transfer learning", *Remote. Sens.*, vol. 11, p. 83, 2019.

[49] S. Li, Y. Liu, X. Sui, C. Chen, G. Tjio, D. Ting, and R. Goh, "Multi-instance multi-scale cnn for medical image classification", in. Oct. 2019, pp. 531–539, ISBN: 978-3-030-32250-2. DOI: 10.1007/978-3-030-32251-9_58.

[50] Y. Liu, K. Gadepalli, M. Norouzi, G. E. Dahl, T. Kohlberger, A. Boyko, S. Venugopalan, A. Timofeev, P. Q. Nelson, G. S. Corrado, J. D. Hipp, L. H. Peng, and M. C. Stumpe, "Detecting cancer metastases on gigapixel pathology images", *ArXiv*, vol. abs/1703.02442, 2017.

[51] Y. Liu, K. Gadepalli, M. Norouzzi, T. Dahl, T. Kohlberger, A. Boyko, S. Venugopalan, A. Timofeev, P. Nelson, G. Corrado, J. Hipp, L. Peng, and M. Stumpe, "Detecting cancer metastases on gigapixel pathology images", 2017. DOI: arXiv:1703.02442.

[52] 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.

[53] M. Macenko, M. Niethammer, J. S. Marron, D. Borland, J. T. Woosley, Xiaojun Guan, C. Schmitt, and N. E. Thomas, "A method for normalizing histology slides for quantitative analysis", in *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2009, pp. 1107–1110.

[54] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift", Feb. 2015.

[55] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting", *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, Jun. 2014.

[56] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[57] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks", in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2261–2269.

[58] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning", in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, ser. AAAI'17, San Francisco, California, USA: AAAI Press, 2017, pp. 4278–4284.

[59] S. Marcel and Y. Rodriguez, "Torchvision the machine-vision package of torch", in *Proceedings of the 18th ACM International Conference on Multimedia*, ser. MM '10, Firenze, Italy: Association for Computing Machinery, 2010, pp. 1485–1488, ISBN: 9781605589336. DOI: 10.1145/1873951.1874254. [Online]. Available: https://doi.org/10.1145/1873951.1874254.

[60] T. contributors, *Pytorch.vision*, version 0.3.0, Aug. 2019. [Online]. Available: https://github.com/pytorch/vision.

[61] pretrainedmodels contributors, *Cadane pretrained models, pytorch*, version 0.7.4, Aug. 2019. [Online]. Available: https://github.com/Cadene/pretrained-models.pytorch.

[62] S. Kornblith, J. Shlens, and Q. V. Le, "Do better imagenet models transfer better?", 2019. [Online]. Available: https://arxiv.org/pdf/1805.08974.pdf.

[63] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification", in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1026–1034.

[64] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations", *Information*, vol. 11, no. 2, 2020, ISSN: 2078-2489. DOI: 10.3390/info11020125. [Online]. Available: https://www.mdpi.com/2078-2489/11/2/125.

[65] R. Liu, J. Lehman, P. Molino, F. Such, E. Frank, A. Sergeev, and J. Yosinski, "An intriguing failing of convolutional neural networks and the coordconv solution", Jul. 2018.

[66] I. Melekhov, J. Kannala, and E. Rahtu, "Siamese network features for image matching", Dec. 2016, pp. 378–383. DOI: 10.1109/ICPR.2016.7899663.

Appendix A Code repository

The source code repository of the multilevel model work including additional documentation.

https://github.com/jpjuvo/camelyon17-multilevel

Appendix B Stain augmentation repository

The source code repository of the stain augmentation method.

https://github.com/jpjuvo/HEnorm_python