University of Glasgow | School of Computing Science

Honours Individual Project Dissertation

Using Deep Learning to predict overall survival times for breast cancer from H&E stained whole slide biopsy images

Anirbit Ghosh

March 24, 2023

Abstract

Current pathological practice of breast cancer diagnosis involves the manual annotation of malignant tumours in the biopsied tissue slide by a trained pathologist. This process is highly resource intensive and is subject to interpretation variability between different pathologists leading to inconsistent outcomes. Therefore, a Deep Learning based automated cancer detection system which can identify the extent of malignancy from a digital whole slide image (WSI) would increase the efficiency of the process. The step after the initial cancer diagnosis requires vast amounts of patient data to be collected and processed in order to deliver a survival prognosis. The clinical data ranges from general health and quality of life measures to specific morphological features of the malignant tumour. These must be extracted manually by pathologists through extensive and time consuming tests that lead to significant delays to much needed patient treatment. On the contrary, a plethora of cellular and physiological information which represents the exact nature of malignancy in a patient is readily available from the WSI of biopsied tissue. However, trying to manually analyze whole slide images is extremely expensive in terms of time and storage due to the sheer size of these images. Thus, we have attempted to augment the Deep Learning based cancer detection system to automatically extract meaningful features from WSIs. Then we investigated the viability of using these image-based features in characterizing cancer severity and extrapolating a survival time estimate without any associated clinical data. In this project, we proposed a supervised learning approach to train a deep convolutional neural network on the annotated Camelyon 16 dataset, to learn how to separate malignant tumours from healthy tissue in a given WSI. Despite the lack of supporting literature validating the approach of training a network on breast metastatic *lymph node* tissue to detect malignancy in *breast tissue*, the results showed reasonably accurate outcomes and one of the extracted image-based features even had a statistically significant correlation to cancer severity. We concluded that knowledge of breast metastases captured by our model can be viably transferred and applied to other tissue types. Finally, we curated a dataset of 74 breast cancer WSIs from the The Cancer Genome Atlas (TCGA) repository along with each patient's associated survival duration. We proposed fitting univariate Cox hazard models using the features extracted by our network from each patient's WSI as model covariates. One of the covariates displayed statistical significance with disease hazard. With the fitted model, we have then regressed a median survival time for each patient based on the covariate value calculated from their associated WSI. This resulted in a high average root mean squared error and a standard deviation of 47 ± 29 months between the predicted and actual survival times. Despite the erroneous predictions, the model was able to accurately characterize the general trend in survival times across different patients based on the relative metastatic severity captured by our image-based features. Our work concludes that without associated clinical data WSIs alone are not viable in precisely predicting an exact survival time as the disease hazard of each patient is uniquely influenced by a combination of several external factors. This process can only produce a rough baseline to allow identification of high risk cases and prioritize patients for treatment.

Education Use Consent

I hereby grant my permission for this project to be stored, distributed and shown to other University of Glasgow students and staff for educational purposes. **Please note that you are under no obligation to sign this declaration, but doing so would help future students.**

Signature: Anirbit Ghosh Date: 24 March 2023

Contents

1 Introduction

1.1 Motivation

With a global estimate of 18 million cases (Sung et al. (2021)) of cancer diagnosed each year, as of 2020, female breast cancer has become the most common contributor making up over 12.5% of all cases of which 684,996 cases were fatal making it the fifth leading cause of death. However, with an average 75% 10-year survival rate across all stages of breast cancer, it shows the best prospects for generally being the most curable provided an early diagnosis is made, followed by appropriate and timely treatment. This constitutes the rationale for focusing this dissertation on improving the efficiency of breast cancer diagnosis and prognosis prediction.

A biopsy is the only definitive way of diagnosing breast cancer. Traditionally a trained pathologist would be required to manually inspect a biopsied tissue sample and identify potential areas of abnormal, malignant cell growth. Subsequently an oncologist would use this pathology report to determine the patient's prognosis in terms of likelihood of recovery and survival chance which guides the course of treatment. Classical prognosis prediction relies on population-level estimates by comparing a patient's case to previously documented similar cases based on cancer site, tumour grade, cancer stage and certain cellular markers. As every patient's cancer manifests uniquely, this technique is further supplemented by collecting the individual patient's clinical and quality of life information combined with several test results of multiple bio-markers. This complete diagnostic workflow is not only time consuming but also highly subjective and non-reproducible due to vast amount of data required to characterize each patient's condition and the manual effort required to produce and interpret that data.

If the delay between tissue collection, diagnosis and subsequent prognosis prediction could be minimized, it would allow patients to receive the appropriate treatment much quicker which can increase the likelihood of recovery in particularly time sensitive cases. Furthermore, reducing human involvement in the diagnostic procedure can minimize interpretation variability making diagnoses more consistent and robust.

1.2 General problem and our hypothesis

The biopsied sample of breast tissue used in the diagnostic process exactly characterizes a patient's individual cancer manifestation. The structural and cellular level information obtained from biopsies is very valuable in understanding the extent and severity of cancer. A digital representation of the biopsied tissue can be obtained in the form of a whole slide image (WSI), which is a high resolution, multi-level replication of the stained tissue as observed under a microscope at varying zoom levels. These digital slide images can be used in an automated cancer diagnostic system by treating the detection of malignant cells in the image as a classification problem.

The general problem we are attempting to solve is to utilize digital whole slide images of biopsied tissue to automate the diagnosis of breast cancer without requiring intensive, manual inspection of tissue slides by pathologists. Furthermore, we are attempting to estimate the patient's prognosis in terms of an overall survival time by utilizing information extracted from the whole slide image, thereby potentially eliminating the dependence on extensive clinical data collection and analysis.

This paper proposes a supervised learning model which will be trained with patches of breast cancer whole slide images to learn how to perform slide level classification of malignant and benign tissue. Subsequently, we develop and extract some image-based features from the segmented whole slide images to predict an estimate for the survival time of the given patient.

1.3 Aim

This project aims to investigate the viability of deep-learning based automated breast cancer diagnosis and corresponding survival time prediction only using digital slide images of the affected tissue. We are trying to minimize or potentially eliminate the requirement of manual human intervention and vast amounts of clinical data in the diagnostic process. The project will achieve this by performing the following tasks:

- elaborate the idea of automated cancer detection from whole slide images and describe the problem of inferring an estimated disease prognosis from a single slide image.
- prepare a training dataset with fixed size patches of whole slide images of breast tissue samples along with patch-level annotations for malignancy or benignity. The dataset must be standardized to eliminate any impact of human variability involved in the preparation of tissue slides collected from various sources.
- implement and train a supervised learning based deep convolutional neural network that can learn to distinguish between malignant and benign breast tissue. It will accept patches of a whole slide image and output a fully segmented image map showing areas of malignancy.
- develop and extract a feature metric from the segmented WSI to quantify and characterize the nature of breast cancer in a given tissue slide. Using this metric, generate a statistical model to predict a proportional survival time for any given case.
- evaluate the performance of the cancer detection network against clinically annotated ground truth samples. Measure accuracy of survival time prediction against available clinical data and infer validity of exclusively using whole slide images in delivering disease prognosis.

2 Background

2.1 Whole Slide Images

2.1.1 Slide preparation

Biopsied tissue blocks are sliced into thin sections and mounted on glass slides. For cancer diagnosis, these tissue slides are generally stained with a Hematoxylin and Eosin dye which has been proven (*Bancroft and Layton (2013)*) to exhibit selective affinity for specific cellular structures. As visible in the skin biopsy sample shown in figure 2.1, stained regions are more prone to absorbing light making them more prominently visible under illumination in a microscopy. In breast tissue, the Hematoxylin component stains the nuclei of cells in a blue colour as it binds to acidic structures, particularly RNA and DNA (*Chan (2014)*). Eosin on the contrary stains any connective tissue and cellular cytoplasm in shades of red (*Bancroft and Layton (2013)*), which in breast tissue samples makes up the majority of the surface due to the presence of muscles, fatty tissue and collagen. Collagen is the dominant component in tissue surrounding breast ducts which is the most common area of origin for breast cancer making this staining even more effective in distinguishing between normal and abnormal cellular growth. Therefore, H&E staining has become the histopathology standard when diagnosing breast cancer from biopsied tissue.

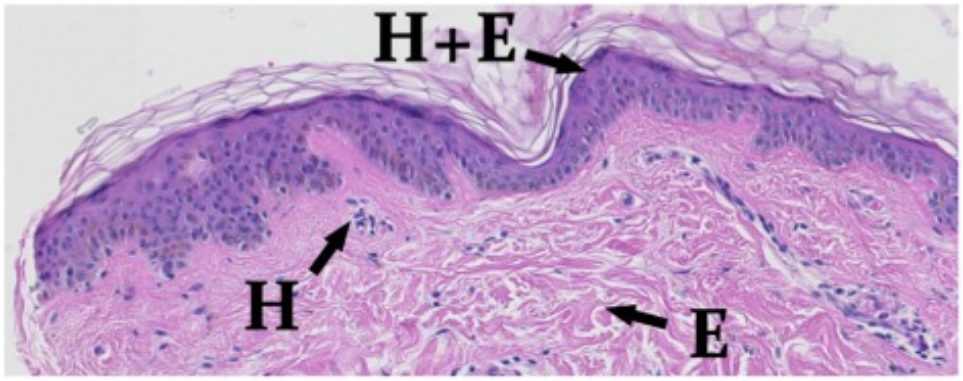


Figure 2.1: *Section of skin showing Hematoxylin(H) & Eosin(E) stain interactions.*

2.1.2 Slide digitization

Physical tissue slides are digitized using whole slide image scanners which contain an illumination system, an optical microscope and a focusing apparatus (*Ghaznavi et al. (2013)*). Scanners most commonly use Brightfield illumination (*Kino and Corle (1996)*) in which the entire tissue sample is uniformly illuminated. This works in conjunction with the H&E staining to display the tissue as a dark coloured image on a white background making it easier to distinguish cellular structure.

The final output of the slide scanner is a virtual rendering of the entire tissue slide, that can be viewed at resolutions of upto 0.25 μm corresponding to a 40x objective power of the scanning

microscope. In order to replicate a slide under a microscope, WSIs exhibit a pyramid structure consisting of images at multiple resolutions. The base level stores the image in diagnostic resolution of a 40x objective, often containing 100,000 x 100,000 pixels (*Wang et al. (2012)*). As displayed in figure 2.2, the resolution is down sampled by a fixed factor at each subsequent level. The Cancer Genome Atlas program's repository contains WSIs with 4 resolution levels down-sampled with factors of 1.0, 4.0, 16.0 and 64.0. This feature of WSIs makes these images extremely bulky with average sizes around 1GB per image.

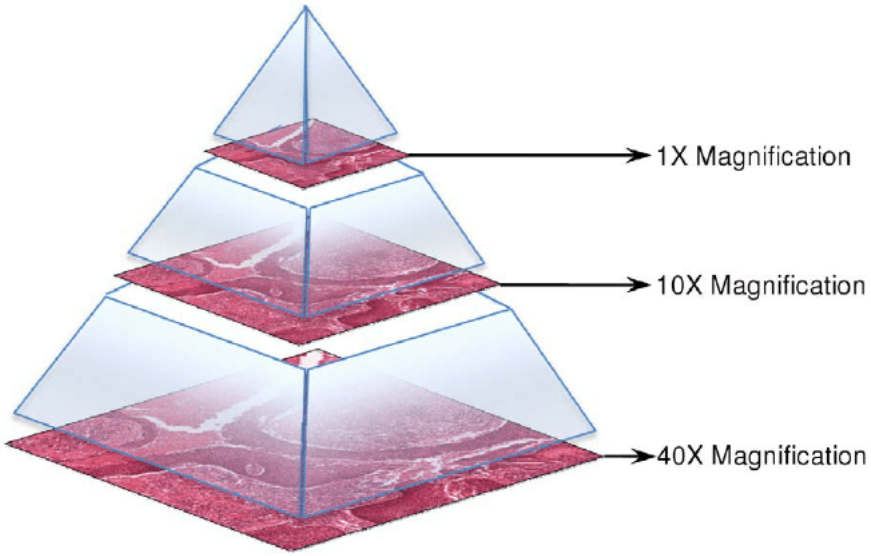


Figure 2.2: *Illustration of pyramid structure observed in whole slide images. Highest resolution at 40x magnification followed by 2 downsampled levels at 4.0 (10x) and 40.0 (1x).*

Viewing or analyzing WSIs in their entirety is infeasible as they would likely exhaust a standard computer's available memory and due to their massive pixel counts it would be very inefficient to perform any image-processing tasks on them. It was demonstrated by Wang et al. (2012) that sectioning the WSI into several smaller, fixed size tiles at its diagnostic resolution and performing image analysis tasks on each tile allows for more efficient processing compared to using the entire slide image at once. As shown by Aeffner et al. (2019), confining analysis efforts to smaller regions of interest (ROIs) rather than the entire WSI is often necessary to devise any accurate or computationally viable image analysis algorithms. The tile based results can be aggregated in order to generate masks that highlight all ROIs on lower resolution thumbnails of the original WSI in order to display the results of our image analysis tasks.

2.1.3 Slide normalization

The manual preparation of tissue slides by pathologists is a very sensitive process and despite having an established staining protocol, several factors can imbue a high degree of variability in the presentation of the resulting WSI. As demonstrated by Anghel et al. (2019), the lack of consistency in the quality and appearance of slide images is attributed to varying antigen or dye concentrations, temperatures at which the tissue is stored, physical conditions the slides are subjected to when being put through slide scanners and random human error involved in the process. **Figure 2.3** illustrates how despite having a standardized staining procedure like the Hematoxylin & Eosin staining, no two slides of the same tissue will exhibit identical color, stain intensity or physical structure.

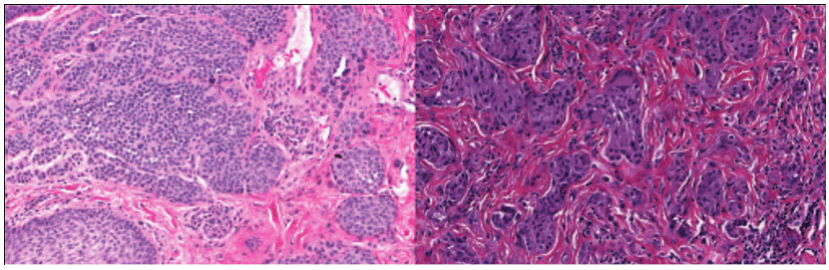


Figure 2.3: *Example of two H&E stained whole slide images of melanomas obtained from different laboratories indicating the drastic difference in color appearance.*

Li and Plataniotis (2015) establishes that performing any quantitative analysis on whole slide images requires numerical features obtained from an input image to be compared against features learned from prior training data. The disparities observed between the compared features help quantify any histopathological information conveyed by such images. If there exists additional variability in the WSI stemming from inconsistent slide preparation, then that corrupts the extracted features as they account for these non-histopathological differences and deviate from the true characterization of the sample. There is particularly high colour and intensity variation in slide images sourced from different pathology labs as each location has different conditions and personnel under which slides are produced. According to the paper on automated prostate cancer detection by Gorelick et al. (2013), information conveyed by color variation on tissue samples is of extreme significance in the quantitative analysis of histopathological images. Therefore, it is crucial to achieve greater color consistency by implementing color normalization algorithms when working with models that utilize pattern recognition and color processing algorithms to deliver a diagnosis from learned features.

2.1.4 Macenko method for color normalization

The Macenko stain normalization method (Macenko et al. (2009)) is considered the gold standard for performing color normalization on H&E stained whole slide images. This is evidenced by its application in a wide range of works such as, a paper on an end to end learning model for predicting tumour subtypes and genetic mutations from H&E stained slide images by van Treeck et al. (2021), predicting colorectal cancer from H&E stained slides using deep learning by Kather et al. (2019) and Anghel et al. (2019) paper on robust stain normalization system for whole slide images.

For slide images captured with standard RGB scanners, the stain absorbs three distinct wavelengths of light. Macenko et al. (2009) defines a *stain vector* as the proportion of each wavelength absorbed by the staining component. The overall idea behind this normalization algorithm is that the color of a RGB pixel in a stained slide image is a linear combination of two stain vectors, corresponding to Hematoxylin (H) and Eosin (E). Being able to estimate these stain vectors effectively allows us to project each pixel onto any color plane, thereby converting the WSI into a standardized color profile, as shown in figure 2.4. The Macenko stain normalization pipeline can be summarized as:

- Applying a singular value decomposition (SVD) operation, approximate the H&E stain vectors of all pixels inside the tissue sample (excluding white background pixels) in a given RGB slide image.
- Apply an intensity correction operation to compensate for irregularities in stain concentration, quality of stain and staining protocol used by the original pathologist.
- Project resulting image on a reference plane to ensure all normalized pixels exhibit a similar color profile.

Macenko Algorithm
Macenko et al. (2009) prescribes the SVD based stain normalization algorithm as follows:

INPUT : RGB slide image

- Transform RGB slide to Opitcal Density (OD) form where $OD = -\log_{10}(I)$, I being the RGB vector normalized to [0.1].
- Remove pixels with OD intensity less than a pre-determined parameter β. Macenko et al. (2009) identified $\beta = 0.15$ provides the most robust normalization.
- Perform singular value decomposition on the OD tuples. Extract the two largest values by magnitude and generate the OD plane in their corresponding direction.
- Project input image data onto the resulting OD plane and normalize to unit length.
- Calculate θ, which is the angle of each point with respect to the direction of the 1^{st} SVD value.
- Using pre-determined parameter α, calculate robust extremes of the angle which correspond to the α^{th} and $(100 - \alpha)^{th}$ percentiles. Macenko et al. (2009) empirically recommends a value of $\alpha = 1$ for the best normalization result.
- Project extreme values back to OD space.
- Build optical density matrix (ODM) using the obtained projection from previous step
- Inverse ODM to estimate the respective concentration of the Hematoxylin and Eosin stains (C_h and C_e).
- Determine the $(100 - \alpha^{th})$ percentile of C_h and C_e.
- Upon transforming the normalized concentrations of each individual stain component to OD space, revert it back to RGB using an H&E reference template.

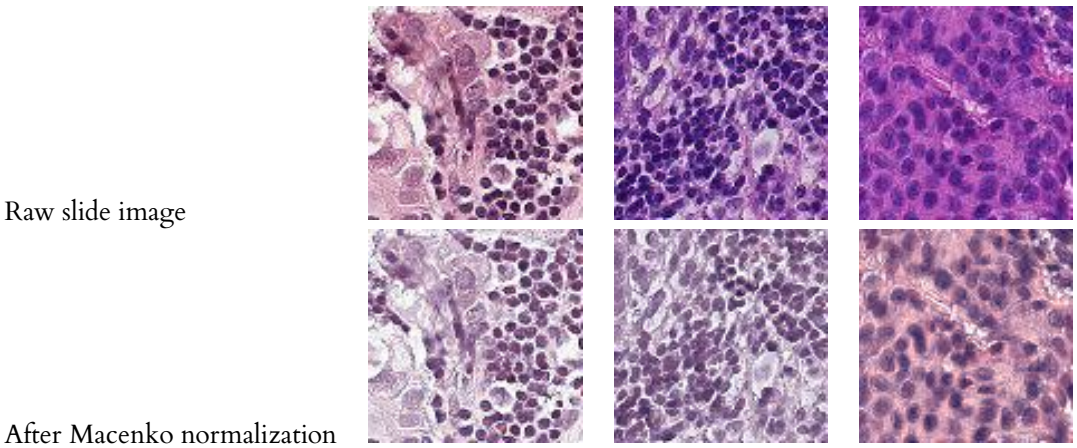


Figure 2.4: *Examples taken from our model training data - Illustrating effect of Macenko normalization on WSI patches of varying colour intensities. The normalization projects slide images with very different colour profiles to a standard color plane such that they become relatively closer in appearance.*

2.2 Deep Learning in histopathological analysis

2.2.1 Applications in WSI based diagnosis

As proposed by Gurcan et al. (2009), diagnosing diseases from histopathological images requires the identification of some histological features, for example, cancer nuclei, glands or lymphocytes that serve as indicators of disease severity. With growing popularity of computer assisted diagnosis systems, Deep Learning has been proven to be the preferred approach for such classification and segmentation tasks on whole slide images.

Kather et al. (2019) has proposed a supervised learning model that uses transfer learning to train a CNN that performs a patch based classification of nine tissue types observed in H&E stained WSIs of Colorectal cancer (CRC) samples. The study used 86 H&E slides of CRC tissue where regions of interest (ROI) were manually annotated at a slide level to identify relevant tissue classes. Labelled slides were used to create a training set of image patches, each consisting of uniformly distributed ROI annotations, without any associated clinical data. The trained CNN would take a WSI, split it into 224x224 px patches and produce a prediction for the type of tissue a given patch represents with a unique color. Therefore, tile-level predictions were obtained to fully segment the given WSI into its constituent tissues and subsequently extract a deep-stromal score from the multi-class segmented WSI to determine the severity of CRC.

A paper on detecting invasive ductal carcinoma (IDC) from whole slide images using a CNN was presented by Cruz-Roa et al. (2014). As shown in figure 2.5, the paper proposes a supervised learning model trained using WSI patches containing patch-level annotations to learn a hierarchical representation of malignant vs benign tissue. The CNN identifies tumour regions to determine tumour extent and severity and subsequently estimate disease prognosis. This deep learning based IDC region classification displayed a 84.23% accuracy when compared to ground-truth pathologist annotations. The paper presents a comparison of the deep learning approach to more traditional machine learning classifiers implementing a Random Forest method for tumour classification using hand-crafted features like tumour area, texture, nuclear structure, color etc. Using a RGB histogram was only 77.24% accurate while a fuzzy color histogram showed 78.74% accuracy, proving the effectiveness of deep learning approaches when performing histopathological analysis.

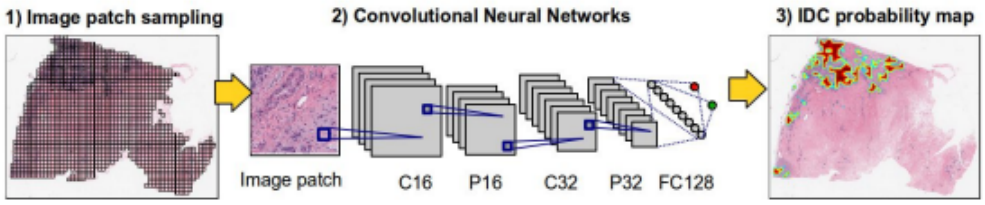


Figure 2.5: *IDC detection framework from H&E stained WSI - 1)Splitting WSI into tiles, 2)CNN outputs tile-level prediction for probability of being IDC positive, 3)Aggregate predicted tiles to show heatmap of most likely IDC regions with probability > 0.29.*

The use of a patch/tile-based approach to train and segment WSIs is observed to be the most common technique to overcome the memory logistic issues associated with the large size and resolution of WSIs. An alternative approach to processing large WSI data was demonstrated by Qin et al. (2018) using a feature pyramid method comprising of a ResNet50 pipeline. The ResNet50 architecture was used as an encoder while combining the feature pyramid blocks as decoders which produced a segmentation accuracy of 63% when tested on the CAMELYON16 challenge dataset.

In the context of segmenting malignant breast tissue in H&E stained WSIs, the paper by Priego-Torres et al. (2020) proposes a fast semantic segmentation pipeline. As observed in figure 2.6, it performs tile-level segmentation of malignant or benign tissue using a deep convolutional neural network (DCNN) along with a dilated convolutional encoder-decoder architecture. The paper prescribes using a pre-trained MobileNetV2 network as the backbone of the DCNN. The CNN's max pooling layer is replaced with the convolutional encoder to allow features to be extracted at any resolution, independent of the resolution of the input WSI. This makes it possible to accurately segment regions of the same tissue that may appear different due to scaling disparities in the input image compared to the training images. They applied a softmax function to the last layer of the CNN (as shown in figure 2.6) in order to convert the discrete prediction (malignant vs benign) into tile-level probability maps. These probability maps were merged using a Conditional Random Field (CRF) method that allowed overlapping regions of tiles to be captured more efficiently without any loss of information.

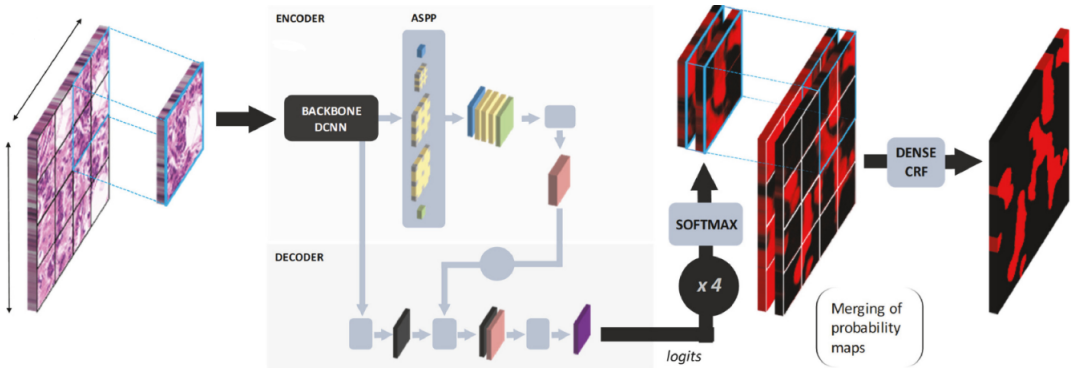


Figure 2.6: *Tile-based semantic segmentation using DCNN and convolutional encoder to obtain probability maps of segmented tissue.*

2.2.2 Methods to process WSI data for deep learning

Whole slide images are captured at a minimum 40x magnification resulting in images having resolutions exceeding $100,000 \times 100,000$ pixels with an average uncompressed size of 5GB (Bandi et al. (2018)). Thus, a single image often exceeds the available memory size in most computers making it inefficient to perform tasks like patch extraction by loading the entire WSI to memory. This necessitates the use of specialized software to access such images and perform deep learning tasks such as extracting patches for patch-based tissue segmentation.

Goode et al. (2013) have created Open Slide which is an open-source, vendor-neutral C library for viewing, processing and analyzing WSIs. The major problem in digital pathology is the lack of a standardized format for WSI. This causes WSI vendors to create their own proprietary format that requires their proprietary software to process. Open Slide aims to increase interoperability by facilitating transparent handling of various vendor formats of WSIs while supporting multiple programming language APIs. It offers a Python package to enable WSI processing through Python which makes it accessible for academic research.

Bankhead et al. (2017) proposes a Java based whole slide processing tool called QuPath. It can be accessed through its own native GUI to perform WSI analysis but also supports Groovy scripts to execute high level algorithms such as patch extraction from WSIs. QuPath also provides the ability to build low level machine learning classifiers like Support Vector Machines or Random Forests within its integrated GUI environment. Recent developments with deep learning are yet to be translated into QuPath as it does not support training CNNs natively and only allows predictions to be imported and viewed (Pedersen et al. (2021)). It is also very restrictive due to being a GUI

and not having any API support for other languages. The GUI itself has a steep learning curve and is not very intuitive for users who are not experts/researchers in histopathology.

2.2.3 Data availability for CNN training

As reported in the paper on Colorectal Cancer diagnosis from histology slides by Kather et al. (2019), all data used in training and validation of the CNN was manually annotated by an expert pathologist. They sectioned 86 CRC H&E whole slides from the NCT biobank and UMM pathology archive to create a training set of 100,000 tiles. Each tile had to be manually annotated at a tile level to describe the nine relevant tissue classes. Their validation set comprised of 862 H&E slides retrieved from 500 CRC patients on The Cancer Genome Atlas (TCGA) repository. Data from TCGA does not contain any pixel or slide level annotations and had to be manually reviewed to identify relevant histological structures.

Priego-Torres et al. (2020) further lay emphasis on the lack of publicly available H&E stained whole slide images with strong pixel level annotations being a major hindrance in the development of supervised learning models. Their solution to this lack of annotated data was to create an OpenSlide based, online WSI viewer that allowed pathologists from across the world to view and hand annotate tumour regions, thereby constructing a training dataset tailored to this project for breast cancer lesion segmentation.

The CAMELYON16 and CAMELYON17 challenges were identified as viable data sources for training deep learning models to detect breast cancer metastases. Khened et al. (2021) proposed a generalized framework for WSI segmentation and validated its performance in breast cancer segmentation using CAMELYON16 and CAMELYON17 data. As shown by Bejnordi et al. (2017) in their paper on deep learning algorithms for breast metastases detection, the CAMELYON16 data serves as a very robust dataset for training supervised learning models as it contains 399 WSI (159 malignant and 240 benign samples), sourced from two separate laboratories with pixel level annotations of breast metastases found in sentinel lymph nodes.

CAMELYON17 on the other hand was showcased by Bandi et al. (2018) in assessing potential deep learning techniques for breast cancer detection and prognosis assessment. This dataset was built upon CAMELYON16 to include 1399 WSI of breast metastases in sentinel lymph node sections, sourced from five different laboratories to more accurately represent staining variability and allow for more robust training. Instead of pixel level annotations, CAMELYON17 included slide level annotations to allow for a shift of focus on patient-level diagnosis followed by patient prognosis prediction rather than diagnosing metastases in individual WSI.

Why is CAMELYON data effective for diagnosing breast cancer?
The effectiveness of CAMELYON16 and CAMELYON17 data in diagnosing breast cancer despite containing lymph node biopsies is explained by Sobin et al. (2011) in their TNM staging system. In diagnosing breast cancer, clinicians consider the extent of tumour growth (T-stage), spreading of tumour cells to lymph nodes (N-stage) and the degree of metastases in other parts of the body (M-stage). Presence of breast metastatic cells in sentinel lymph nodes, is therefore a significant indicator of breast cancer and could potentially be used to learn general breast metastases characterization in deep learning models. Due to the lack annotated breast tissue data, we will attempt to transfer the learning obtained from breast metastases seen in lymph nodes to detect malignancy in breast tissue. As the metastatic cells display the same characteristic structure, we expect our model to learn how to identify those cells irrespective of the surrounding tissue environment.

2.3 Cancer prognosis estimation

2.3.1 Survival time analysis methods

As an extension to deep learning based CRC diagnosis, Kather et al. (2019) have proposed fitting a univariate Cox-proportional Hazard model with features extracted from each tissue class to model their respective influence on patient-level survival prediction. Utilizing a Youden index (sensitivity + specificity - 1), they determined an optimal Hazard Ratio (HR) cutoff value. All counts of tissue classes with HR > cutoff were weighted with their HR and combined to develop a deep stromal score quantifying slide-level disease severity. They propose a further step to fit a multivariate Cox proportional hazard model on the TCGA validation data, in order to adjust for clinical factors like tumour stage, sex and age when obtaining the Hazard Ratio. This multivariate HR showed higher confidence in prognosis predictions as it accounted for both slide-level tissue morphology and patient specific clinical factors.

Other instances of survival analysis on histopathology images, such as the paper by Wetstein et al. (2022) illustrates a Kaplan-Meier survival analysis using a univariate and a multivariate Cox proportional hazard model. The multivariate analysis was included to produce more realistic survival predictions that accounted for clinical characteristics of each patient, such as current treatment, tumour dimensions and lymphovascular state. All these factors are ignored by the univariate model which only aims to utilize histological features in its prediction. A similar methodology was also proposed by Liu and Kurc (2022) using univariable Cox proportional hazard models. They additionally demonstrated the use of Python's *Lifelines* package in streamlining the Cox model fitting process and KM plot generation to visualize survival events.

Majority of work in the field of automated survival estimation involves linear semi-parametric Cox models as described previously. The applicability of these methods are constrained by the requirement of strong, pixel-level annotations in facilitating supervised learning models to extract precise features to successfully quantify survival. Alternate approaches of survival analysis are still being developed but are not as prominently established. Katzman et al. (2018) proposed a feed-forward neural network called DeepSurv to quantify the influence of a patient's clinical and histological covariates on their hazard rate and subsequently support treatment recommendations. This method uses an alternate approach to previously described linear Cox regression by using a neural network to optimize the partial likelihood loss function of the Cox model using non-linear features. Wulczyn et al. (2020) illustrates another method of survival prediction using a weakly supervised learning approach. Uniformly sampled patches were drawn from the slides of a given patient, only knowing if the slide was malignant or benign. Image-based features extracted by the CNN from these patches were processed by a custom loss function - combination of cross-entropy and Cox partial likelihood loss - to output a probability distribution over discretized survival intervals for each patient.

2.3.2 Kaplan-Meier analysis

The Cox proportional hazard model only describes the hazard or probability of a certain survival event happening at specific points in time. It is useful in analyzing the risk of survival associated with given features. However, it cannot predict overall survival time as the Cox model assumes a baseline hazard which is only scaled proportionally with different covariates to get the relative hazard at each level. The hazard would remain constant over all time unless a covariate level changes causing the true survival risk of a patient to remain undefined which prevents a survival time inference. Additionally, hazard is inversely related to survival, with survival showing rapid decline at high hazard function values. As defined by Bewick et al. (2004), having survived until a given time t, the hazard function $\lambda(t)$ corresponds to the probability of dying at t. Whereas, the KM survival function $S(t)$, represents probability of surviving past time t. Schober and Vetter (2021) proposes using the Kaplan-Meier method for producing exact survival time predictions. As

shown in equation 2.1, Kaplan-Meier generates a non-parametric survival function that estimates the probability of patients surviving longer than a given time t.

$$S(t) = \prod_{t_i \leq t} (1 - \frac{d_i}{n_i}) \tag{2.1}$$

In the equation d_i represents the number of events (deaths in our case) that took place in each time t_i and n_i is the number of subjects that survived at that time. We multiply 1– probability of death at each time upto the given time t to give us the probability of survival past t. Showing overall survival time prediction using Kaplan-Meier graphs is the most common methodology as observed in all survival analysis papers such as Kather et al. (2019), Liu and Kurc (2022) and Wetstein et al. (2022).

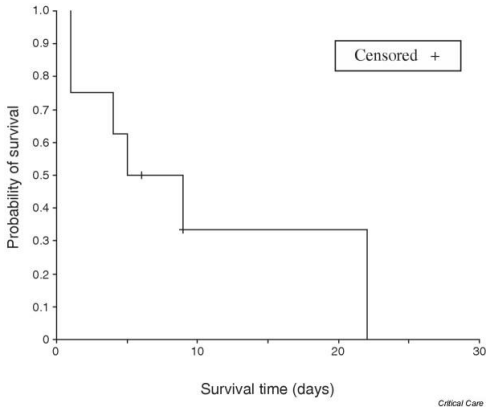


Figure 2.7: *Bewick et al. (2004) - Example Kaplan-Meier plot showing overall survival probability over time (in months).*

As observed in figure 2.7, a KM plot is a step-wise function plotting probability of survival on the y-axis and survival time in any discrete unit on the x-axis. KM plots can support multiple curves on the graph where each represent a unique covariate setting to see their influence on survival. The graph displays vertical drops in survival probability over time at points where an event of interest occurs, which in survival analysis is the death of a subject. The horizontal distance to these points on the curve represent the total interval of time after which the event occurred, thereby showing the probability a subject survives for that period of time. As proposed by Bewick et al. (2004), the median survival time is often a good prediction of overall survival as it is the point of 0.5 probability where half the subjects die and the other half survives at that point of time. This median survival time can be used to compare multiple KM plots in terms of how severely covariate influences survival.

2.3.3 Cox proportional hazard model

The original paper by Cox (1972), demonstrated a generalized regression method as an extension to the Kaplan-Meier survival analysis approach. The Cox proportional hazard model is a linear regression model defined in terms of a hazard function (λ) as follows

$$\lambda(t|\mathbf{x}) = \lambda_0(t) \exp(\beta_i \mathbf{x}_i) \tag{2.2}$$

where t represents time, x_i denotes a vector of each covariate being modelled such that $i = 1$ for univariable Cox modelling and $i > 1$ for multivariable models. β represents a vector of all regression coefficients in the model while $\lambda_0(t)$ denotes the baseline hazard under initial conditions of $\mathbf{x} = 0$.

This hazard model displays the relation between the time until some event occurs (in survival models the event is the death of a subject) to the given covariates. The proportional nature of the model implies a multiplicative increase in hazard rate with unit increase in the covariate(s) across all points of time. This can be demonstrated by considering a univariable model where x and β are both one element vectors representing single covariate regression. Following a unit increase in the covariate, such that we have $x + 1$ as our new covariate, the new hazard function is

$$\begin{aligned} \lambda(t|x+1) &= \lambda_0(t)\exp(\beta(x+1)) \\ &= \lambda_0(t)\exp(x\beta+\beta)) \\ &= (\lambda_0(t)\exp(x\beta)) + \exp(\beta) \\ &= \lambda(t|x)\exp(\beta) \end{aligned} \tag{2.3}$$

denoting a constant multiplicative scaling by $\exp(\beta)$ of our original hazard function upon a unit increase in covariate. Furthermore, upon changing sides of the new hazard function we get

$$\frac{\lambda(t|x+1)}{\lambda(t|x)} = \exp(\beta) \tag{2.4}$$

where the ratio $\exp(\beta)$ is a constant with no time dependence. This represents the "proportional" nature of the Cox hazard model where the relationship over time between original and subsequent hazard functions remains constant despite changing covariates. The Lifelines python package mentioned in section 2.3.1, combines the Kaplan-Meier method with the Cox-proportional hazard model to obtain an overall baseline hazard which is then proportionally scaled using the provided covariate to estimate the unknown hazard at that level. This is then used to work out a median survival time estimate for the patient exhibiting the provided covariate.

3 Analysis

3.1 General Problem

3.1.1 Cancer detection

The first step to delivering a cancer prognosis from histopathological images is to quantify disease severity and extent through image-based features. As illustrated in figure 3.1, there is a clear visual difference between malignant and healthy tissue samples. Firstly we observe in figure 3.1(b), that tumour cells are enclosed within distinct patches among the non-background pixels rather than being homogeneously dispersed throughout the body of the tissue. Secondly in figure 3.1(c) we observe a difference in stain intensity between the surrounding healthy cells and the tumour within the annotation lines. This color disparity is caused by abnormal cellular growth - displaying an increase in the size of Hematoxylin stained (blue) nuclei and reduced nuclei density leading to an increase in Eosin stained (pink) cytoplasm observed in tumour regions.

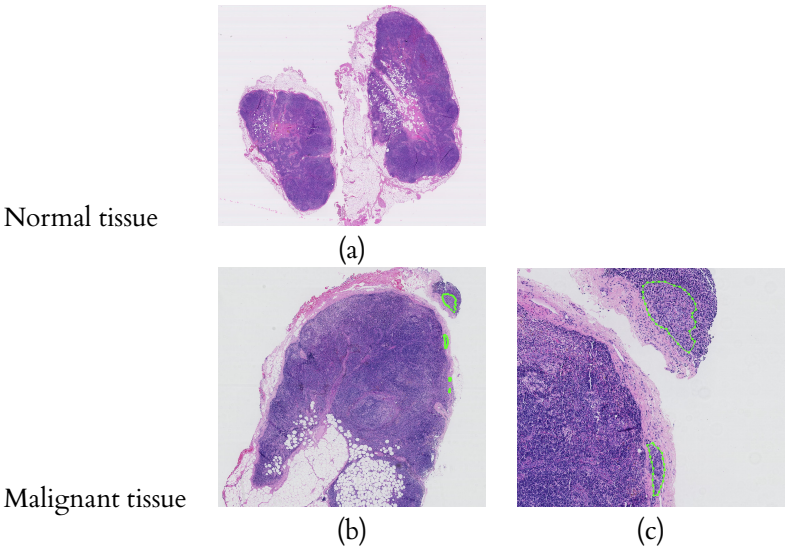


Figure 3.1: *Examples taken from CAMELYON16 training data - Normal tissue (a) exhibits healthy breast tissue with no cancerous tumour. Malignant tissue (b) represents annotated location of malignant tumour with respect to the overall slide and (c) displays a closer view of malignant cells inside annotated region.*

The general problem we are trying to solve at this stage is to identify whether a given tissue slide is cancerous and subsequently segment its malignant areas to extract meaningful features that convey disease severity and extent. As prescribed in multiple literature reviewed in section 2.2.1 we can perform a tile-based analysis of the whole slide image. Accounting for differences in cellular structure and texture observed in varying tissue samples, we can use the visual differences

observed in healthy and cancerous cells to predict tiles as malignant or benign based on their constituent cellular morphology. Tumour areas will be detected by high density of tiles predicted as malignant and thereby segmenting the WSI into regions of interest. Since we know a given WSI has a fixed pixel count at the sampling resolution, we can quantify the extent of a tumour in terms of the pixel area covered by malignant tiles. As inferred from figure 3.1, the difference in stain intensity of tumour cells compared to healthy cells can be used to quantify cancer severity in terms of an aggregated color intensity metric of malignant tile regions. Severely abnormal nuclear growth in tumour regions will produce much darker stain intensities due to large nuclei, whereas severe cytoplasmic expansion will display significantly lighter color intensities compared to healthy cellular regions.

3.1.2 Survival time estimation

As explained by the literature reviewed in section 2.3.1 we can utilize the image-based features extracted from the previous stage as covariates to fit Cox proportional hazard (CPH) models. This will allow us to obtain the hazard function for each feature and measure its influence as a risk factor affecting survival of patients. The CPH models for each covariate enables comparison between different features to identify their relative significance in prognosis estimation. The final problem of predicting the overall survival time can then be solved by generating Kaplan-Meier plots of the most significant covariates and obtaining the median survival time from the graph as shown in figure 2.7. These fitted models can be generalized to any WSI given as input, by first processing the WSI as tiles through our CNN to extract the relevant features. These features are mapped to our fitted regression models to generate its corresponding KM plot and deliver the final disease prognosis in terms of a median survival time.

3.2 Whole slide images

3.2.1 Data availability

As discussed in section 2.2.3, the most widely used approach for automated cancer detection from histopathological images requires large datasets with strong, pixel-level annotations to train a robust supervised learning model. Creating our own custom dataset to train and validate our model would involve sourcing sufficiently diverse WSIs and having an expert pathologist manually annotate each WSI for our tile-based approach, which is extremely resource intensive and infeasible for the scope of this project. Therefore, this paper uses CAMELYON16 data as our training dataset since it has been annotated at a pixel level by expert pathologists.

3.2.2 PCAM dataset for model training

We have utilized the PatchCAMELYON (PCAM) version (Veeling et al. (2018)) which is a pre-tiled, publicly available dataset containing over 327,000 tiles sourced from 399 WSIs included in the CAMELYON16 data. The PCAM data comprises of 96x96px tiles with binary tile-level annotations indicating whether a tile is malignant or benign based on the presence of cancer cells in the center 32x32px patch of each tile. The CAMELYON16 WSIs are provided a 40x objective but PCAM tiles are downsampled by 10x to increase field of view. The PCAM dataset replicates the same class distribution in each train-test split as the CAMELYON16 dataset, including 50/50 balance between malignant and benign tiles with no overlap to allow unbiased model training. PCAM also holds out 20% of training tiles to create a validation set to allow models to be cross validated. The tile annotations are provided as CSV files in the following format:

id	coord_y	coord_x	tumor_patch	center_tumor_patch	wsi
1	148544	74048	1	1	camelyon16_train_tumor_003

The X and Y coordinates represent the pixel coordinate of the top-left vertex of the 96x96px tile taken from the whole slide image specified in the wsi column. A binary (0 or 1) annotation is provided for each tile. A value of 1 for tumor_patch indicates the presence of cancer cells located anywhere within the 96x96px tile. Whereas, an annotation of 1 for center_tumor_patch indicates that malignant tissue is present in the center 32x32px patch of the tile.

3.3 Processing

3.3.1 Supervised learning and survival prediction

After reviewing the methods employed by other research papers in section 2.2, we will be utilizing a supervised learning model to solve the first stage of our general problem. This model will employ a tile-based approach to learn how to segment WSIs into malignant and benign regions. We will employ the strongly annotated PCAM data as the training dataset for our CNN. This will be an end-to-end learning model that learns how to characterize malignant and benign tissue and output a fully segmented representation of the input WSI with clearly demarcated tumour regions. The segmented image will then be used to extract features that can meaningfully quantify disease severity in a given histopathological sample.

For the second stage of our general problem, we aim to develop a model that can produce overall survival time predictions based on certain quantifiable features extracted from a given WSI by our trained CNN. To arrive at a generalizable survival model we will need to curate a model training dataset containing these image-based features along with clinical labels of associated survival time prognosis for each WSI. Based on our previous discussion in section 2.2.3, TCGA data has been proven to be a credible and high-quality data source for clinically relevant histopathological images. Another advantage of TCGA is that each sample also includes diagnostic data, such as the actual survival time recorded for each patient from the point of diagnosis to the point of death. We will apply our trained CNN to a collection of breast cancer WSIs sourced from TCGA to extract image-based features. These features will be used as covariates along with associated ground-truth survival duration labels to fit linear Cox regression models to estimate an overall survival time influenced by that covariate.

3.3.2 Image pre-processing

The PCAM dataset used to train our CNN to solve the first stage of our general problem is a derivative of the CAMELYON16 database, which is a vast and diverse collection of histopathological samples sourced from multiple pathology laboratories. As discussed in section 2.1.3, variability in preparation conditions introduces inconsistencies to the quality and appearance of WSIs. This makes it necessary to apply Macenko normalization to our PCAM data before using it to train our model. This ensures that our model learns to characterize the true morphological difference between malignant and benign breast tissue samples rather than learning the random noise and calibration bias introduced during WSI preparation.

When applying this model to TCGA breast cancer slides to extract image-based features, we must split the WSI into tiles and eliminate background dominant tiles. Furthermore, we perform Macenko normalization on each tile to make our input data consistent relative to our model's trained features and ensure accurate segmentation of malignant regions.

3.3.3 Ground truth validation

For the purpose of training a deep learning model to detect breast cancer we require histopathological data with clinically valid labels to actually learn how to characterize the morphology of breast metastases. The Camelyon16 dataset has been manually annotated by expert pathologists

and will serve as the basis of training for our cancer detection model. We will then optimize the trained model hyperparameters and measure its performance in segmenting malignant regions of histopathological images against the annotated validation set of the PCAM data.

As described previously, the second stage of our problem aims to develop a regression model that can be generalized to predict an overall survival prognosis given breast cancer biopsy images as input. Fitting this model requires pathologically valid labels of associated survival time predictions of each WSI used in populating the relevant feature dataset. These ground truth labels for each WSI sourced from TCGA can be accessed through cBioPortal, which is a large scale genomic dataset containing the accompanying diagnostic data for each histopathological sample image hosted on TCGA. The performance of the fitted survival model will then be measured on unseen WSI samples taken from TCGA and the corresponding survival prediction will be validated against available diagnostic data on cBioPortal [1].

[1] https://www.cbioportal.org/

4 Design

4.1 Cancer detection system overview

We implement a neural network to solve the first stage of our overall problem. Figure 4.1 illustrates a logical workflow of how we go from an input breast tissue slide to a segmented malignancy feature map. As seen in the upper half of the figure, the network is first trained with batches of WSIs, each sectioned into labelled, colour-normalized tiles. During the training process, every RGB tile undergoes a series of convolutions as it passes through each layer of our network. Neurons learn a weight based on the feature representation extracted by kernel convolutions in each layer. The final layer's output is a probability prediction for each relevant class. This is compared to ground truth label and a loss is calculated. Our goal is to minimize this loss value. This is achieved by back-propagating and using an optimizer to update the weights of each neuron which improves the learned feature representation and allows for more precise predictions.

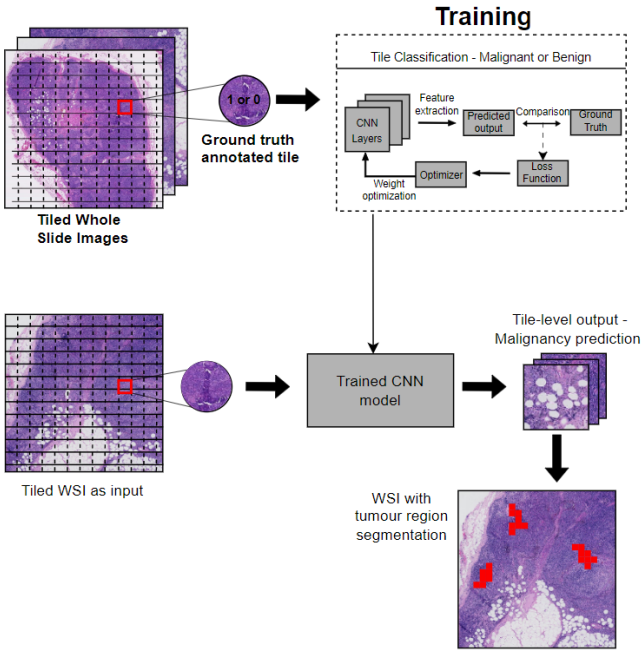


Figure 4.1: *Workflow of cancer detection system design. The upper half of the figure demonstrates the training procedure of the neural network. It uses annotated WSI tiles to learn feature representations of breast metastatic tissue samples. The training will use the ground-truth annotations of each tile to optimize the neuron weights and minimize training loss. In the lower half of the image, the trained model is used to produce tile-level, binary predictions for a given WSI. The predicted tiles are used to generate a segmented feature map of the original WSI*

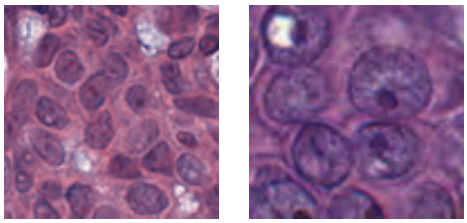
The trained model will process an input WSI on a tile-by-tile basis to estimate the presence of cancer cells in each tile. Using the learned feature representations of breast cancer morphology, the neural network performs a binary classification task to output the most likely class (malignant (1) or benign (0)) for each tile. The tiled predictions will be overlaid on a thumbnail of the WSI to produce a segmented feature map indicating potential tumour regions, as illustrated in figure 4.1. We repeat this process for a series of WSIs taken from the TCGA repository to generate a collection of segmented feature maps. For the next stage of our problem, we will manipulate this feature dataset to extract covariates that can effectively model and estimate survival time.

4.1.1 Tiling and pre-processing

Since the Camelyon16 data is sourced from multiple pathological laboratories, we will be applying the Macenko normalization alogrithm (section 2.1.4) on each tile of the PCAM dataset before training our model. This normalization will allow the model to learn features of cellular morphology in tiles, based on their actual structure rather than accounting for differences in colour appearance and treating them as different features. This introduces colour invariance to allow the model to more accurately identify similar tissue structures across different WSIs.

The WSIs processed by our CNN to generate segmented tumour maps for the survival analysis, must be tiled individually as TCGA only contains whole histopathological slides. To maintain consistency between training and feature extraction tiles, there are two main points to consider - tile dimension and tile resolution. We utilize the OpenSlide library described in section 2.2.2 to convert a WSI into a stream of tile objects of specified dimension at a chosen resolution level. Therefore, we extract 96×96px tiles downsampled by a factor of 10 to a 4x objective to match the PCAM tile dimensions.

Figure 4.2 illustrates that a tile sampled at 4x magnification (10x downsampling) shows a much wider field of view compared to a 40x objective tile. As explained in section 3.1.1, detecting tumour regions is based on identifying abnormally large nucleus or increased cytoplasm dispersion. The lower resolution exposes spatial information that allows our model to identify regions of abnormal nuclear growth relative to surrounding cellular structure. The 40x magnification however, zooms in on individual nuclei to the point where any information on spatial locality is lost and our model cannot characterize nucleus size or distribution.



(a) (b)

Figure 4.2: *Illustrates two tiled sections sampled from a TCGA WSI, used in feature dataset generation - (a)* 96×96*px tile sampled at 4x objective shows greater field of view and less granular detail (b)* 96×96*px tile sampled at 40x objective has smaller field of view and shows individual nuclei at greater detail.*

We must also remove any blank tiles that are predominantly background. This reduces the computation time of our cancer detection pipeline by only processing tiles containing relevant tissue structures. The tile filtering will be done by choosing an intensity threshold representing the average intensity of Macenko normalized H&E stain and eliminating all tiles with higher average intensity.

4.1.2 Tumour prediction - classification and segmentation

Tiling is equivalent to dividing a given image into a grid of 96×96px squares. Each tile extracted from a WSI will be stored to disk along with an X-Y coordinate representing its location on the grid. We will generate two types of predictions from the CNN - a discrete binary class prediction and a continuous prediction of the probability of being malignant. The results from the CNN will be stored against each tile's coordinate in a dataset format. This dataset will serve as the feature dataset for survival analysis.

In order to visualize the tumour segmentation, we first generate a thumbnail of the WSI down-sampled to a resolution that allows viewing the entire image without causing memory issues. For the binary prediction, we map a distinct colour to each of the two classes. For each tile, we overlay the associated colour to its corresponding coordinate on the thumbnail. This creates a sharp segmentation between normal and potential tumour regions of the tissue. Using the continuous predictions, we can generate a complete heatmap showing the probability of regions being malignant on the tissue surface. The predictions will be a value between 0 and 1, which we map to a continuous colour scale thresholded to the same range and overlay the associated colour on each tile coordinate.

4.2 Survival estimation - regression

4.2.1 Model covariates

Malignancy spread score
Using the binary predictions obtained from our cancer detection model, we develop a scoring system to quantify the extent of malignancy in a given WSI. We estimate the degree of tumour spread in terms of pixel area covered by malignant tiles. This is equivalent to the number of tiles assigned a malignant class (=1) prediction by our CNN. Not all WSIs are the same size, with some including multiple or much larger tissue sections which would naturally consist of a higher number of non-background tiles. To prevent this metric from showing disproportionately high tumour spread for larger WSIs, we normalize the malignant tile count by the total area covered by non-background tiles. This produces a ratio representing the size of metastatic tumours as a percentage of overall tissue surface in a WSI. The score can then be used as a covariate that quantifies cancer severity in modelling overall survival time prognoses.

Mean malignant intensity
The continuous predictions generated by our model are in the form of a probability value between 0 and 1 showing the probability of a given tissue patch being malignant. This represents the confidence associated with each prediction that can be inferred as intensity of malignancy. A high probability corresponds to a tile with strong malignant features that are distinctly identifiable. Whereas, low probability indicates weakly malignant or benign morphology observed in the tile. We calculate the mean malignant intensity of a WSI by taking the average probability across all its tiles. A high mean intensity score (closer to 1) is likely to represent a more severe metastatic characterization caused by the presence of dominant regions of strong malignant predictions. A lower mean intensity (closer to 0), will correspond to majority low probability tiles (benign if <0.5) indicative of less severe metastasis.

4.2.2 Survival time prediction

Regression training data
In order to fit a generalized model that can estimate overall survival time for a given breast cancer slide, we first build a model training dataset. This will be developed using a collection of 74 WSIs sourced from TCGA. We restrict the WSI selection process to deceased patients

with available diagnostic data only. For each WSI, we retrieve the ground truth label of the patient's actual survival time (in months) - the duration from point of diagnosis to point of death - available on cBioPortal against the patient's TCGA case ID. Using survival times of *deceased* patients as the basis of our model allows us to estimate how long patients are expected to survive with different levels of cancer severity. Every WSI will be processed through our trained CNN to generate segmented tumour maps. The severity of each WSI will then be quantified by extracting the feature metrics described in the previous section. The final dataset will consist of an overall survival duration and a value for the covariates against each WSI to fit the regression model.

Survival model

As described in section 2.3.3, we first fit a Cox proportional hazard model using the training dataset described above to generate hazard functions for each covariate. Besides estimating the proportional hazard posed by the given covariate level as the hazard ratio, this allows us to determine each covariate's significance and validity as a risk factor impacting survival times of patients. The hazard ratio provides the probability of subjects dying at a given time which can be used as the $\frac{d_i}{n_i}$ component in equation 2.1 to get the survival probability at each time. As described in section 2.3.2, we then predict a median survival time estimate for different calculated covariate levels using the Kaplan-Meier method.

The final pipeline to obtain a survival time estimate for a WSI is as follows:

- Section WSI into 96x96px tiles at a 4x objective.
- Macenko normalize each tile.
- Generate tile level malignancy predictions using trained CNN.
- Calculate Mean Malignant Intensity and Malignancy Spread Score for the WSI.
- Query the fitted Cox models using the calcualted covariate scores to obtain median survival time predictions and generate survival probability plots for that covariate level.

5 Implementation

5.1 System overview

We have implemented a separate system corresponding to each of the two main stages of breast cancer analysis - diagnosis and prognosis. The diagnostic system generates both continuous and discrete predictions of malignant regions present in a breast tissue WSI. This system is trained using clinically annotated breast cancer slides as ground truth labels. The resulting predictions are used to extract features that quantify disease severity in terms of metastatic spread. Finally, the prognostic system fits a regression model using the diagnostic features to predict an overall survival time for each patient. We essentially perform 4 distinct high-level tasks: training, diagnostic predictions, model fitting and survival predictions.

5.1.1 Training overview

Due to the lack of a trained pathologist and an extremely resource intensive process of strongly annotating cancer WSI tiles by hand, we have used the PCAM dataset for training our neural network to perform cancer detection. The dataset contains tiled breast metastases WSIs sourced from the CAMELYON16 database along with associated tile-level binary annotations indicating if a tile is malignant (1) or benign(0), which is used as the ground truth labels.

We have implemented a pre-processing pipeline (as shown in figure 5.1), to prepare any input data into a format that can be processed by our neural network. We first feed our training data through this pipeline, skipping the tiling step as the PCAM dataset is already tiled into 96×96px patches. We subsequently normalize the colour of each tile using the Macenko algorithm to allow our model to achieve colour invariant. In the final stage of pre-processing we perform data augmentation to synthetically expand our training space with random variants of each tile, allowing our model to be more robust by learning rotational and translational invariance.

5.1.2 Malignancy predictions overview

As discussed in section 4.1, we have curated our own feature extraction dataset consisting of 74 WSIs sourced from TCGA. Using our trained neural network, we perform a tile-based binary classification task on each WSI to generate fully segmented feature maps indicating potential tumour regions. This implies that each WSI will be fed through the pre-processing pipeline (as shown in figure 5.1), without the data augmentation step, to generate colour normalized tiles which our trained model will then generate predictions for. The final predictions will be used to extract a malignancy spread score and mean malignant intensity as described in section 4.2.1.

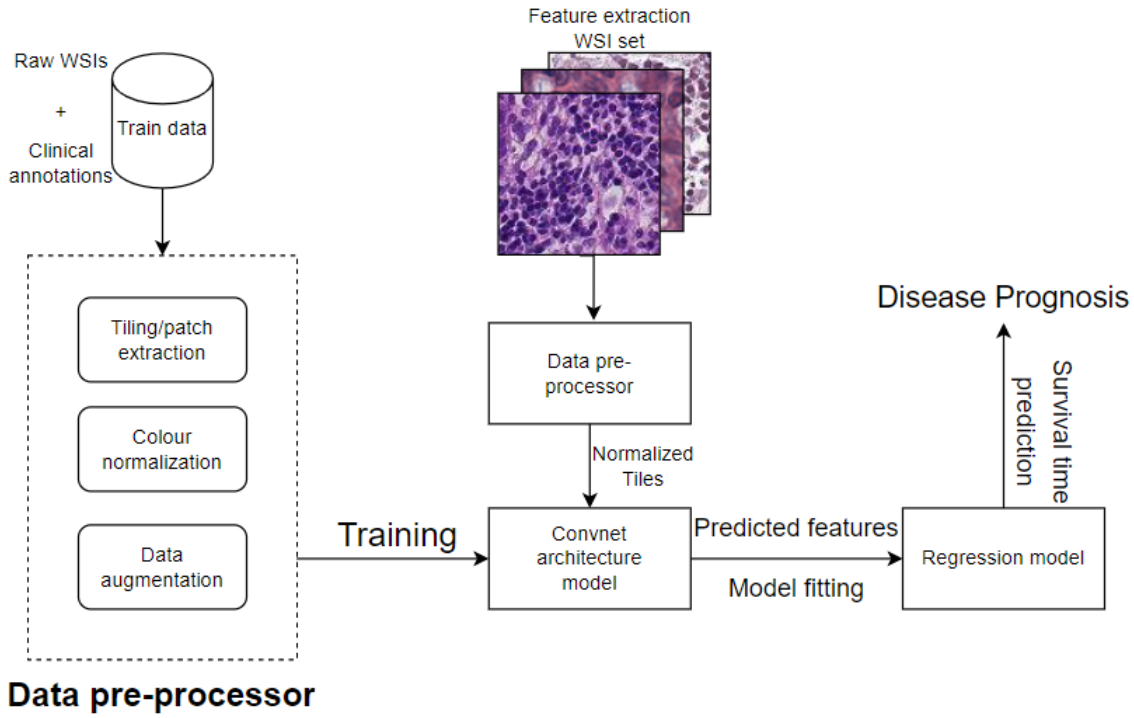


Figure 5.1: *Complete overview of our entire system implementation. PCAM data containing raw WSI tiles and ground truth annotations are used as training data. The tiled dataset is colour normalized and augmented using our pre-processing pipeline before being used to train our convnet architecture based neural network. We create a feature extraction dataset containing WSIs from TCGA. Each WSI is pre-processed into colour normalized tiles and fed through our trained model to generate feature maps indicating malignant regions. Predicted features are then used to fit a semi-parametric Cox regression model which uses Kaplan-Meier method to generate a median survival time prediction.*

5.1.3 Survival time predictions overview

Our neural network will generate discrete and continuous predictions that will be used to extract two scores that characterize the disease severity in terms of tumour spread of the given WSI. For all 74 WSIs in our feature extraction dataset, we also obtain their clinically determined survival times to serve as ground truth labels for our survival model. Finally, we use the predicted scores along with survival time annotations to fit a semi-parametric Cox regression models for each metric. Each model generates a hazard ratio and a negative log-likelihood indicating the associated risk factor of the covariate and the significance of impact on survival respectively. We then produce a median survival time estimate from the models to compare their performance with respect to the corresponding ground truth survival time and evaluate their effectiveness in survival prognosis estimation.

5.2 Machine Learning - Cancer diagnosis

5.2.1 Datasets

Training data
For the purpose of training our neural network model to make breast cancer predictions, we

have utilized the tiled and annotated version of the CAMELYON16 challenge data called the Patch CAMELYON (PCAM) dataset. It contains 399 WSIs, tiled into over 300,000 total tiles. Each tile is annotated with a binary annotation indicating malignancy nature as described in section 3.2.2. Tiles are 96×96px in dimension with their resolution downsampled by 10x to a 4x objective rather than using their 40x native objective.

The PCAM data contains separate datasets for training, validation and testing, each with an equal balance between malignant positive and healthy (malignant negative) tiles. Each set is stored in a Hierarchical Data Format version 5 (HDF5) file. This format stores extremely large (in this case 200,000 images) data in a structured format resembling an entire file system hierarchy as a single file. It allows individual data points to be accessed in an efficient manner without needing to transfer entire datasets to memory and as a result overloading available memory.

In an HDF5 file data is organized into groups representing folders. These groups contain other groups or the datasets which resemble files. As seen in listing B.1, each PCAM dataset contains two main groups labeled 'x' and 'y' under the root. The 'x' group holds the WSI tiles, represented as $96 \times 96 \times 3$ numpy arrays of 8bit unsigned integers (0 to 255) under groups labeled with a unique id for each tile. The 'y' group holds the malignancy annotation of each tile. The annotation is a binary label that is stored under the group with id corresponding to the tile it labels.

We iterate over the HDF5 datasets and save the numpy array of each tile as a `.jpeg` image into a `/tiles` directory within each of the three dataset directories. We extract 100,000 training tiles and 20,000 validation tiles, both equally balanced between the two classes, for example, the training tiles consist of 49,977 malignant (1) and 50,023 benign (0) tiles. We also extract the corresponding annotations from HDF5 format into a `.csv` file, indexed by the tile id as described in section 3.2.2. This allows for more efficient data handling through the CNN using dataloader objects that can readily associate batch of images to their corresponding labels during the training process.

Feature extraction data
We also created a second dataset consisting of raw WSIs to serve as the basis of extracting malignancy features required to develop a survival model for making survival time predictions. This dataset consisted of 74 WSIs sourced from the TCGA breast cancer repository. We cross checked each WSI's TCGA ID against the clinical data on cBioPortal to only select WSIs which had associated clinical data available showing overall survival time and belonged to deceased patients. This allowed us to retrieve the total survival duration for each patient to serve as ground truth labels for our survival model. All TCGA WSIs have a native 40.0 magnification at the highest resolution level and an average file size of 2GB due to their large pixel count. Images were downloaded and stored in disk in a `.svs` format which supports multi-level resolution sampling.

5.2.2 Data pre-processing

We store all relevant data in their raw format before being used with our machine learning model. As explained previously, histopathological slide images are subject to high levels of irregularities caused by conditions under which they were produced. Furthermore, WSIs are extremely massive images, which in their raw format simply cannot be processed in data batches like standard machine learning problems. Therefore, we have developed a general pre-processing pipeline consisting of operations that convert the raw image data into a consistent format that can be processed more efficiently by our neural network. This pipeline performs three main operations - tiling, colour normalization and data augmentation.

WSI Tiling
The PCAM training data consists of pre-tiled WSIs with each tile having dimensions 96×96px

and a 10x downsampled resolution. As explained in section 4.1.1, we have to ensure we tile each WSI in the feature extraction dataset using the same dimensions so that our model can make meaningful predictions based on the model weights learned from PCAM tiles. We utilize the OpenSlide package to tile the WSIs without needing to load the entire image into memory.

We fetch each WSI's `.svs` file from disk and open it as an `open_slide` object as shown in listing B.2. This allows us to inspect the slide's metadata using `slide.properties` attribute and choose which level number corresponding to our desired sampling resolution. We then use the `DeepZoomGenerator` object which takes as parameter the WSI as an `open_slide` object, the tiling dimension and an overlap percentage to convert the slide image into a stream of tile references.

We set the desired resolution level - it is the 2nd highest level for TCGA WSIs that exhibit a 10x downsampling to 4x objective. We iterate over each tile reference and obtain the individual tile image as a Numpy array at the specified resolution. We must also filter out any tile that is majority blank background in order to reduce our overall processing space and prevent unnecessary computation. We completely tiled a sample WSI to analyze a collection of its tiles. On average undesirable tiles with majority white background had a mean pixel intensity of >230. Whereas, useful tiles with significant tissue content had an average pixel std deviation of >15. Therefore, a mean intensity lower than 230 indicates the presence of darker tissue regions on the tile. Furthermore, a std deviation of greater than 15 indicates greater variance in pixel intensity values, implying that the darker regions are significant enough for the tile to not be majority white space. As shown in the above code, we filtered out all tiles that did not meet the mean and std deviation requirements before saving each tile as a `.tiff` file.

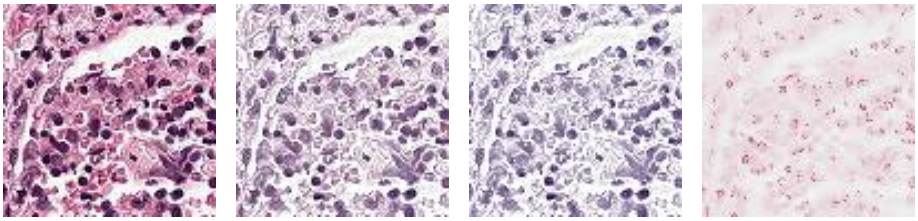
Macenko normalization

As described in section 2.1.3, we follow the standard algorithm prescribed by Macenko to implement a function that accepts as input a tile image and returns a colour normalized tile.

We implemented a `macenko_norm_H&E()` function that takes as parameters the input image and normalization factors. We use the recommended parameter values of $\alpha = 1$ and $\beta = 0.15$. Additionally we also provide a normalizing factor of $IO = 240$, taken from the MATLAB implementation [1] of the algorithm. Our implementation calculates the optical density (OD) of our input image and eliminates transparent pixels below the β threshold. We perform an SVD operation using `np.linalg.eigh` on the OD matrix and project it onto the plane represented by the largest eigenvalues using a dot product with the corresponding eigenvectors. Computing the angles in the direction of the first SVD we find its robust extremes. These angles determine the individual stain vectors corresponding to Hematoxylin and Eosin components. Using the obtained HE matrix, we determine the least square solution to the equation `HE @ X = Y` using `np.linalg.lstsq(HE, Y)` where Y represents the OD matrix of the input image. The computed solution is the concentration factor that scales the HE matrix to obtain the given image's stain appearance. We finally normalize the obtained concentration by dividing it using maximum H&E concentration values and recreate the input image in the normalized colour space. This is finally projected back to RGB space and returned as the final Macenko normalized image along with the H-image and E-image as shown in figure 5.2.

As shown in listing B.2, we extended the tiling pipeline by applying a normalization function on each tile's Numpy array and saving the normalized version. There is code to catch any convergence errors thrown by Numpy's `linalg.eigh` function that is often quite unstable and can fail when values are too high. This is the case with external and tissue edge tiles that contain large areas of white background but pass our mean intensity filter due to having high std deviation caused by presence of debris.

[1] https://github.com/mitkovetta/staining-normalization



(a) Default tile *(b) Normalized Tile* *(c) H-component tile* *(d) E-component tile*

Figure 5.2: *Illustrating effect of Macenko normalization on the base H&E stained tile. Additionally shows the H-component tile with prominently highlighted nuclei and the E-component highlighting cytoplasmic distribution*

Data augmentation
Following good scientific practice when working with limited training data, we have implemented a series of data augmentation steps as a method of synthetically expanding our training data space by including randomized variations of existing images. We utilize *torchvision.transforms.Compose()* to combine a series of random translational transforms, each with occurrence probability of p = 0.5, and a single 45 degree random rotational transform. The final step is a `.ToTensor()` transform that converts the image variant into a format that the CNN can readily process. The custom dataset class (explained further in section 5.2.4) applies these data transformations to each image that is fetched from the dataset during the model training loop. Data augmentation also serves to give our model rotational and translational invariance allowing it to make more robust predictions irrespective of the image orientation.

5.2.3 Neural network architecture

Following the most common approach to solving image analysis tasks using deep learning model, we have implemented a convolutional neural network based on the state-of-the-art ConvNet architecture. Our implementation utilizes a series of standard convolution layers followed by dense layers to produce the final class prediction. We have implemented our CNN using the PyTorch [2] library for Python.

As illustrated in figure 5.3, we provide tensors of RGB tiles having dimension $96 \times 96 \times 3$ as input to our CNN. The input is received as image batches which we have set to a batch size of 64. Next, the images are passed through 4 hidden convolution layers, each using a 3×3 kernel with default values for padding = 0 and stride = 1. The first layer convolves the (3, 96, 96) shaped image tensor using 8 total filters to produce an output of dimension (8, 94, 94). At every subsequent hidden layer, the number of filters is increased by a factor of 2 ($8 \rightarrow 16 \rightarrow 32 \rightarrow 64$) which in turn progressively doubles the channel dimension while reducing the image size. The output from each hidden layer is passed to a Max Pooling layer with a kernel size and stride of 2. This downsamples the image dimensions exactly by a factor of 2. Max Pooling allows the neurons to retain the values of the most prominent features from the convolved feature map which allows the model to identify any abnormalities in cellular morphology that can indicate malignancy. We also apply a tanh activation function to all hidden layers. After 4 layers of convolutions, pooling and activation, the output feature maps has dimension (64, 4, 4) which is then passed to the classification layers.

Classification unit
The fully convolved batch of $4 \times 4 \times 64$ feature maps are each flattened into a 1D tensor of dimensions 1×1024 and then fed to the classification unit composed of 2 fully connected layers.

[2] https://pytorch.org/

Only the first layer is given a tanh activation. We tried using a ReLU activation for the hidden layers and the first fully connected layer, which is the most standard activation function used nowadays. However, a ReLU activation produced outputs that were all biased towards a single class so we chose the tanh activation instead. We then apply a dropout layer with probability of 0.75 over the default value of 0.5. This is a regularisation step aimed to reduce over-fitting by randomly dropping 75% of the neuron connections from the previous layer. This introduces noise to the training process and forces the current layer to be more robust by making predictions from sparse input. The output from the second FC layer represents a 2 element tensor and consists of the respective logit scores for the two prediction classes. Since our model uses a negative log likelihood loss function during training, we applied a log_softmax activation to obtain the final output of our model as log probability of the two relevant classes.

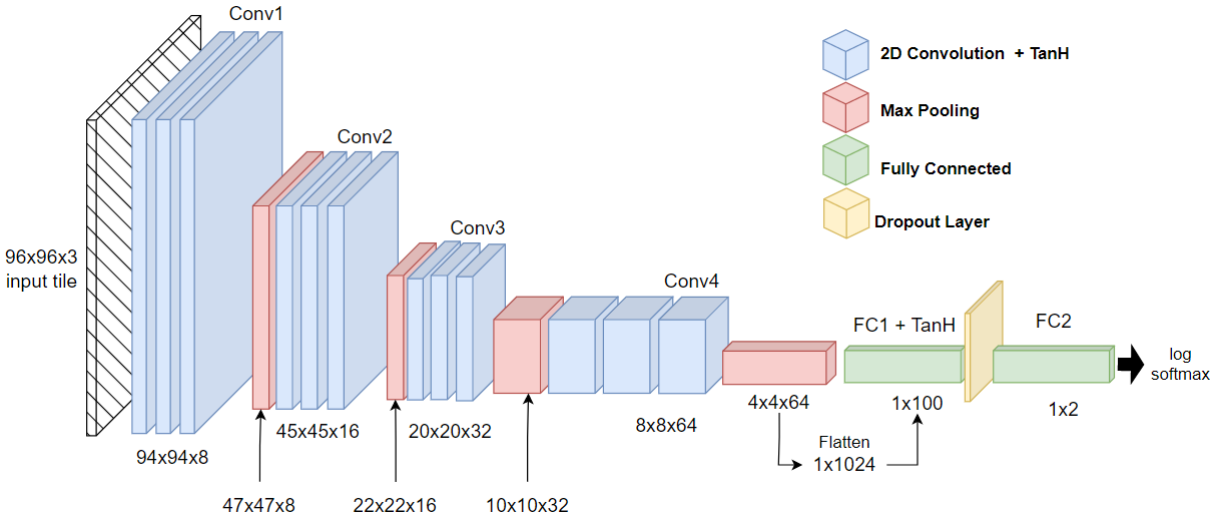


Figure 5.3: *Our implemented model follows a standard ConvNet architecture for image classification. The first part of the network involves 4 × 2D convolution layers with a TanH activation on each hidden layer. Each hidden layer is connected to a Max Pooling layer that extracts the most prominent feature from the previously generated feature map. The final pooling layer's output is flattened into a 1x1024 element tensor which is used to generate logit scores for the two classes by 2 fully connected layers. The final FC layer's output is log-softmaxed to obtain log probabilities corresponding to the two prediction classes.*

5.2.4 Training

Data loading

The HDF5 format is usually very memory efficient when dealing with large datasets allowing us to load relevant slices of data into memory. However, we are required to perform pre-processing tasks on individual images and then feed the resulting images through the neural network. It was highly memory inefficient to extract images from the HDF5 dataset, pre-process them and then hold all images in memory for further processing. Therefore, we read raw images from the disk, feed them through the pre-processing described above and write processed images back to disk. Listing B.3, displays the solution we implemented for continuous and efficient data fetching from disk for model training. We utilize a dedicated dataset component, to fetch processed images from disk and create a stream of image references and their labels which we pass to PyTorch dataloaders to seamlessly extract image batches and feed them to our CNN for training and validation.

Model training and validation
As explained previously, we used the PCAM dataset's training data consisting of 100,000 histological tiles to train our model to detect malignant tissue. The training set was balanced with equal distribution between the two predictable classes. The PCAM dataset also provides a dedicated validation set which we limited to 25,000 images to use for model validation during training.

We create PyTorch dataloaders for each dataset by fetching images from disk using our custom dataset class. We process training data in batches of 64 images. As explained previously, we use random transformations on training data to augment the dataset. However, we don't use data augmentation on the validation set and only transform each validation image into a PyTorch tensor. To train our model, using a **Negative Log Likelihood** loss function was appropriate because we have a single-label, binary classification problem. The model also uses an **Adam** optimizer with an initial learning rate of 0.0005. This was chosen over the standard SGD method because SGD has a single, fixed learning rate for all weights. Adam maintains separate learning rates for each neuron weight that get updated independently as learning unfolds allowing for more robust feature learning. Consequently, we use a **ReduceLROnPlateau** learning rate scheduler which allows the model's learning rate to be reduced whenever learning stagnates in an attempt to improve model learning. During training process, we simultaneously calculate NLLLoss for both training and validation batches. While training loss is used to backpropagate and update neuron weights using our optimizer, the validation loss is used with the LR scheduler to reduce the model's learning rate by a factor of 0.5 if the loss does not show any improvement for 20 consecutive epochs.

Training time
Our CNN model was trained using pytorch's CUDA package to take advantage of GPU accelerated processing, thereby reducing training time significantly. The model was trained on a system with current state-of-the-art NVIDIA RTX 3070 GPU. The training time for our model was a bottleneck as we are required to train on 100,000 images and validate against 25,000 images every epoch and we used 100 epochs for our model training. This total training time ranged between 2 to 5 hours based on our batch sizes. The prediction stage using our trained model proved to be even more significant as a computational bottleneck. Generating tumour maps of 74 WSIs involved predicting over 2 million tiles and processing 800GB of total image data through our model. The entire feature dataset took over 8 hours to be generated each time making repeated feature extractions infeasible.

Hyperparameter tuning
We have undertaken a parameter space exploration task to identify the most optimal configuration for our model, particularly the activation function, learning rate, batch size and dropout rate. We have utilized PyTorch's `Ray[Tune]` package to automatically search through combinations of parameter settings and find the most optimal configuration. We avoided exhaustively testing every possible combination of parameter values as that was causing a "combinatorial explosion", crashing the entire process due to overloaded GPU memory. Therefore, we chose to test 20 models with randomly sampled parameter values from the given search space. Using each model's validation loss as the chosen performance metric, we have identified the most optimal configuration for our final model that we illustrated in section 5.2.3. Given the scale of training and validation data we used, each tuning model took around 8 to 10 minutes to complete every training epoch which added up to over 13 hours for a single model to be trained and evaluated over 100 epochs. Thus, we trained our parameter tuning prototype models on a 10,000 image subset of our training data (still maintaining a 50-50 class balance). We also reduced the training epochs to 50 as that was sufficient to get a general idea of which parameter settings were trending in the optimal direction. Furthermore, `Ray[Tune]` allows setting a `grace_period=10` parameter to the tuning process which automatically stops training and discards a model if it does not show improvement in

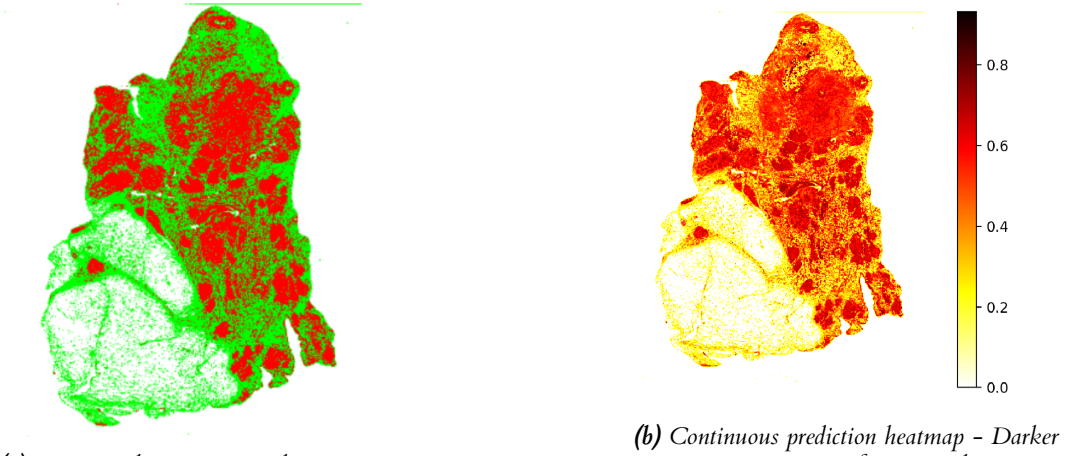
performance metrics for over 10 epochs.

5.3 Regression - Survival time prediction

5.3.1 Feature extraction and visualization

As explained previously we have curated a dataset of 74 breast cancer WSIs from TCGA to extract relevant features characterizing their malignancy severity and build a survival time model around those features. We feed each WSI through our pre-processor and subsequently generate a prediction for each individual tile. All predictions for a single WSI are stored in a `.csv` file as labels against grid coordinates of its tiles - binary predictions display a 0 or 1 label and continuous predictions display a probability value of belonging to class 1 (malignant). We have then used the `Pandas` library to process this structured prediction data and calculate, for each WSI, its Malignancy Spread Score and Mean Malignant Intensity as explained in section 4.2.1.

As shown in figure 5.4 and further in appendix A, we also implemented a visualization pipeline using `OpenCV` that displays classification rresults as a tumour map indicating predicted regions of malignancy for visual confirmation and validation. Due to large WSI resolutions, we displayed the predictions as a mask overlaid on top of a downsampled thumbnail of the slide. We downsampled our image to a resolution where the number of image pixels had a 1:1 ratio with the number of tiles, such that a tile at grid coordinate (X, Y) represents the pixel at that coordinate on the thumbnail. Based on each tile's predicted value, we mapped the corresponding pixel to a colour. We used a discrete (green - benign, red - malignant) map for binary predictions and a reversed continuous map for probability predictions (higher malignant intensity maps to darker colours).



(a) *Binary prediction map - red areas represent potential malignancy and green areas represent healthy tissue*

(b) *Continuous prediction heatmap - Darker areas represent regions of strong malignant intensity while lighter regions show lower malignant intensity/healthy tissue*

Figure 5.4: *The two types of feature maps generated during the prediction stage by our trained model.*

5.3.2 Regression model fitting

As mentioned in section 2.3.1, we use the Lifelines Python package to develop our survival models. We first calculated the covariate scores, for each WSI in the feature extraction dataset. The calculated scores for each WSI were paired with the ground truth survival duration of the patient as retrieved from the associated clinical data available on CBioPortal [3].

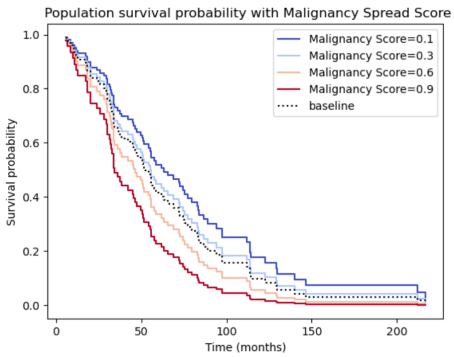
[3] https://www.cbioportal.org/study/clinicalData?id=brca_tcga_pub2015%2Cbrca_tcga%2Cbrca_tcga_pub%2Cbrca_tcga_pan_can_atlas_2018

Cox Proportional Hazards Model

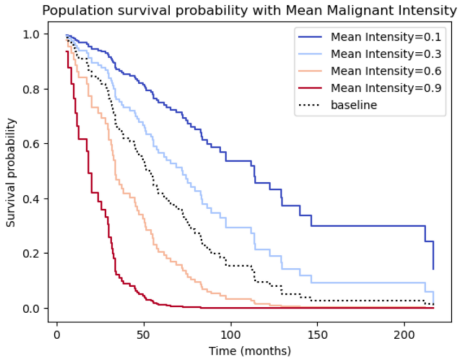
We use the `CoxPHFitter` class from the Lifelines package to train a semi-parametric survival model using a dataset containing malignancy scores, survival duration and survival status of each patient. The Cox model uses the survival time of each patient as the duration parameter, the survival status as the event parameter and any other columns provided are used as covariates influencing survival hazard. Based on the input parameters, the Cox model develops a baseline hazard function. This is inferred as a change in population-level hazard over time given the occurrence of an event, i.e. the death of a subject in our case, at different points in time. Any individual patient's corresponding Log-hazard is then considered to be a linear function of their corresponding covariate value and this baseline generated by the Cox model as shown in equation 2.2.

Survival time prediction

We obtain two univariate Cox proportional hazard functions, each using one of the two malignancy features as hazard covariate in order to infer their individual impacts as a risk factor on patient survival. Finally, we used each model's `CoxPHFitter.predict_median(covariate)` function, with computed value(s) of the corresponding covariate provided as a parameter. This generated a median survival time prediction for the given score(s), in the same units as the training data, giving us an overall survival prognoses estimate for each patient.



(a) KM Plot of Malignancy Spread Score showing survival probability of patients with a score of 0.1, 0.3, 0.6 and 0.9 compared to the baseline survival.



(b) KM Plot of Mean Malignant Intensity showing survival probability of patients with a score of 0.1, 0.3, 0.6 and 0.9 compared to the baseline survival.

Figure 5.5: *Effect of the two covariates - MSS and MMI - on survival probability over time with each curve corresponding to a different covariate level. Higher covariate values shift the graph further below the baseline survival indicating lower survival. Similarly, lower scores implying relatively less severe malignancy display a graph above the baseline indicating reduced risk and better survival.*

As shown in figure 5.5, we have also generalized the model to show population level survival trends over time for a given covariate by using the `plot_partial_effects_on_outcome()` function. It generates Kaplan-Meier plots corresponding to each value provided, showing the covariate's exclusive effect on the survival probability of a cohort possessing that malignancy level, assuming all other factors remain constant. This probability decreases over time for all covariates indicating lower survival at later stages as metastases progresses accounting for no external factors like treatment. Furthermore, The different covariate levels are plotted as graphs shifted around the baseline owing to the proportional nature of the Cox model as discussed in 2.3.3.

6 Evaluation

6.1 Classification stage - Cancer detection

6.1.1 Classifier hyperparameter optimization

The best model described in section 5.2.3, was developed with parameters determined by the results of hyperparameter tuning. We have investigated various configurations from the following hyperparameter space to identify the optimal model setting :

- activation function : tanh, relu, leaky relu
- dropout rate : $p \in [0, 1)$
- batch size : 32, 64, 128
- learning rate : $lr \in$ [1e-5, 1e-1)

As described in section 5.2.4, we first iteratively trained 20 random models sampled from the overall parameter space described above. Using loss and accuracy obtained from the validation dataset as our metric of choice, we identified the best model as highlighted in figure 6.1. The best model comprised of : TanH activation, 0.5 dropout, 0.001 learning rate and 64 training batch size. The results indicated that models using TanH activation generally displayed the best performance overall, outperforming ReLU and Leaky ReLU by a large margin. The batch sizes yielded the standard, expected value of 64 and 32 for an image classification problem implying they are not very consequential on the final performance.

Trial name	status	loc	activation_func	dropout_rate	lr	train_batch	val_batch	loss	accuracy
train_cifar_5355e_00000	TERMINATED	127.0.0.1:9832	leaky_relu	0.0484118	0.000495557	32	128	55.2716	0.7997
train_cifar_5355e_00001	TERMINATED	127.0.0.1:16712	tanh	0.738424	0.0114386	64	128	93.7074	0.5064
train_cifar_5355e_00002	TERMINATED	127.0.0.1:16456	tanh	0.858592	0.0095901	128	32	24.3431	0.4994
train_cifar_5355e_00003	TERMINATED	127.0.0.1:16040	tanh	0.697803	0.00376963	32	128	90.0466	0.5047
train_cifar_5355e_00004	TERMINATED	127.0.0.1:3540	relu	0.294141	0.0102235	128	64	44.2055	0.4972
train_cifar_5355e_00005	TERMINATED	127.0.0.1:8780	tanh	0.137739	0.0361084	128	32	22.481	0.5039
train_cifar_5355e_00006	TERMINATED	127.0.0.1:25232	leaky_relu	0.850451	0.0225711	32	32	401.069	0.5053
train_cifar_5355e_00007	TERMINATED	127.0.0.1:7972	tanh	0.510485	0.00109814	64	32	13.4643	0.817
train_cifar_5355e_00008	TERMINATED	127.0.0.1:28560	tanh	0.602552	0.034952	32	64	72.7751	0.5026
train_cifar_5355e_00009	TERMINATED	127.0.0.1:13660	leaky_relu	0.553799	0.000355438	32	128	53.3818	0.8039
train_cifar_5355e_00010	TERMINATED	127.0.0.1:26216	tanh	0.980643	0.000342647	64	32	16.2811	0.7568
train_cifar_5355e_00011	TERMINATED	127.0.0.1:4484	relu	0.450433	0.0415629	128	32	22.1587	0.5028
train_cifar_5355e_00012	TERMINATED	127.0.0.1:25084	relu	0.802768	0.0298093	128	32	22.1687	0.4972
train_cifar_5355e_00013	TERMINATED	127.0.0.1:28156	relu	0.400066	0.00456903	64	128	87.7444	0.4972
train_cifar_5355e_00014	TERMINATED	127.0.0.1:17508	relu	0.102866	0.00751569	64	128	87.8616	0.4972
train_cifar_5355e_00015	TERMINATED	127.0.0.1:22556	relu	0.0384394	0.000638538	128	128	61.3176	0.7736
train_cifar_5355e_00016	TERMINATED	127.0.0.1:9528	tanh	0.704318	0.00214322	64	128	63.7932	0.7773
train_cifar_5355e_00017	TERMINATED	127.0.0.1:21824	leaky_relu	0.330829	0.0832476	32	64	4116.53	0.4982
train_cifar_5355e_00018	TERMINATED	127.0.0.1:17300	leaky_relu	0.322477	0.0345836	128	128	96.3564	0.5796
train_cifar_5355e_00019	TERMINATED	127.0.0.1:27732	relu	0.201401	0.00520603	64	128	59.1781	0.7811

Figure 6.1: *Stage 1 Hyperparameter tuning results - Showing validation loss and accuracy values for 20 trained models with randomly sampled parameter settings. The best model with highest accuracy and lowest validation loss is highlighted in red.*

However, randomly sampling only 20 models did not allow us to fully explore the range of values chosen for learning rate and dropout. Due to us varying other parameters, there was some repetition observed in the values sampled for LR and Dropout across the 20 models causing a large range of values to remain unexplored. So we ran another tuning iteration with 10 models, this time keeping activation function and batch sizes fixed at the best setting obtained from the previous run and only sampling values for LR and Dropout. This allowed us to sample more

Trial name	status	loc	activation_func	dropout_rate	lr	train_batch	val_batch	loss	accuracy
train_cifar_1a50a_00000	TERMINATED	127.0.0.1:20264	tanh	0.749357	0.000515635	64	32	13.4791	0.8084
train_cifar_1a50a_00001	TERMINATED	127.0.0.1:17652	tanh	0.0432854	0.000845309	64	32	14.6111	0.7967
train_cifar_1a50a_00002	TERMINATED	127.0.0.1:21508	tanh	0.549149	0.0510212	64	32	29.3198	0.4908
train_cifar_1a50a_00003	TERMINATED	127.0.0.1:16808	tanh	0.822937	0.00112886	64	32	16.2125	0.7807
train_cifar_1a50a_00004	TERMINATED	127.0.0.1:16772	tanh	0.772442	0.000820846	64	32	17.6956	0.7353
train_cifar_1a50a_00005	TERMINATED	127.0.0.1:12320	tanh	0.617917	1.03972e-05	64	32	20.6129	0.6576
train_cifar_1a50a_00006	TERMINATED	127.0.0.1:1748	tanh	0.447581	0.00319908	64	32	18.6716	0.695
train_cifar_1a50a_00007	TERMINATED	127.0.0.1:7696	tanh	0.368116	0.00281104	64	32	16.4534	0.7564
train_cifar_1a50a_00008	TERMINATED	127.0.0.1:10948	tanh	0.5964	0.0071343	64	32	23.9339	0.5041
train_cifar_1a50a_00009	TERMINATED	127.0.0.1:2900	tanh	0.679444	0.0437231	64	32	27.8332	0.5039

Figure 6.2: *Stage 2 Hyperparameter tuning results - Showing validation loss and accuracy values for 10 trained models with randomly sampled values for Learning Rate and Dropout while keeping Activation function and Batch sizes fixed to the value obtained in stage 1.*

unique values uniformly across the entire range of values specified above, yielding the final results as shown in figure 6.2

As highlighted in figure 6.2, during the second iteration of hyperparameter tuning, a 0.75 Dropout and 0.0005 learning rate produced the best accuracy and lowest loss values when combined with the other parameters obtained in stage 1. The resulting model - TanH activation, 0.75 Dropout, 0.0005 LR and 64 training batch size was chosen as the final architecture, as described in section 5.2.3, for the classification task.

6.1.2 Metrics used for evaluating classification

Our classification task predicts binary labels - malignant or benign - for each WSI tile. In order to measure the performance of our trained model we have chosen to use the standard metrics used in single-label classification - accuracy, precision and recall. We have processed a collection of 20,000 test images obtained from the PCAM dataset through our prediction pipeline to calculate these metrics for our model and also plot a 2×2 confusion matrix to display the correctness of our prediction results. Using the concepts of true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN) we calculated the metrics as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{6.2}$$

$$Precision = \frac{TP}{TP + FP} \tag{6.3}$$

6.1.3 How well does our deep learning model detect the presence of breast cancer from histopathology images?

In figure 6.3, we have illustrated our model's overall prediction results on the test dataset as a binary-class confusion matrix. This illustrates, that utilizing the model with the best parameter settings obtained after hyperparameter optimization, achieves much higher overall performance. This is further evidenced in table 6.1, displaying the values obtained for our chosen metrics. Our model can distinguish between malignant and benign tiles with a 67% accuracy indicating it performs better than a random model. However, it exhibits a 66% precision in predicting malignant tiles. Given that our dataset is equally balanced, a completely random classifier would have displayed 50% precision. Although this indicates that our model has actually learned something meaningful, it doesn't exhibit very high confidence in its malignant predictions. Overall, the majority of malignant predictions are likely to be correct with a relatively smaller subset of those predictions being falsely labeled as malignant.

Accuracy	Recall	Precision
0.67	0.74	0.66

Table 6.1: *Computed scores for performance metrics - Accuracy, Recall and Precision - obtained from predictions generated on 20,000 PCAM test set images by the model with best parameters obtained from hyperparameter tuning.*

We observe a recall score of 0.74 indicating our model is successfully detecting majority of all possible malignant tiles. When translated to the context of cancer characterization in a WSI, this would imply a slight underestimation of metastases severity as 26% of actually malignant tiles are predicted as benign. As discussed in section 3.1.1, a combination of abnormal nucleus growth and low cytoplasmic density indicate the presence of cancer. This makes detecting malignant tissue a lot more nuanced than detecting benign tissue as a combination of both features are required for a section of tissue to be malignant. Our model often encounters cases where one or both of these metastases markers are present but likely not visually dominant in the tile causing the prediction to become biased towards the benign class as it fails to meet the feature requirements of malignancy. Considering random errors and variability in the quality of tissue slides, our model demonstrates good success in being able to capture the overall extent of cancer in a given collection of tissue sections.

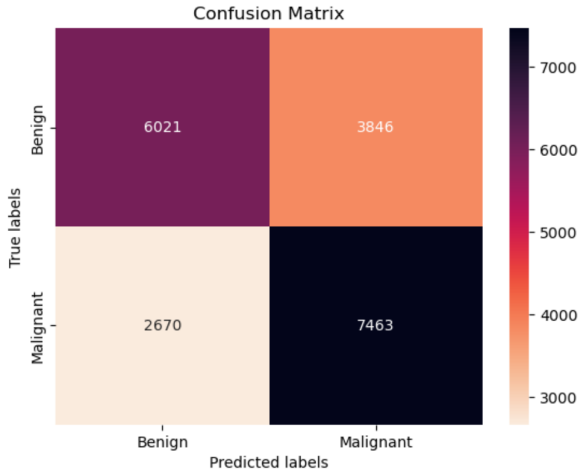


Figure 6.3: *Confusion matrix showing overall prediction performance of our best model when predicting classes for the PCAM test set images. Out of 20,000 query images - we have 10,133 malignant and 9867 benign tiles from which the model correctly predicts 7463 malignant and 6021 benign tiles.*

6.2 Regression stage - Survival prognosis prediction

6.2.1 Metrics used for evaluating regression

Evaluating goodness of fit of survival models

We use each of the two scores, mean malignant intensity (MMI) and the malignancy spread score (MSS) calculated from the fully segmented WSIs in the feature extraction dataset, as covariates to fit separate univariable Cox Hazard models. Before making survival time predictions, we must evaluate the effectiveness of our fitted models to deem if the predictions generated are meaningful. In order to understand the significance of correlation, if any exists at all, between our chosen

image-based feature scores and the survival prognosis of a given a sample, we have utilized the following statistical metrics:

- **Regression coefficients**: The regression coefficients, obtained as the β vector from our Cox hazard function (shown in eq 2.2), provide a measure of the direction of hazard for the associated covariate. A positive β coefficient implies our covariate represents a risk factor such that a higher value will contribute to a higher risk of death translating to lower survival duration. Whereas, a negative coefficient implies a protective factor which reduces the risk of death at higher values thereby improving survival prospects.
- **Wald coefficient**: This metric, also called the z value, gives us a measure of how statistically different from 0 is the β regression coefficient of our hazard function (shown in equation 2.2). Higher the value the stronger is the effect of the associated covariate on the survival of a patient. We calculated the Wald statistic value using equation 6.4, where $\hat{\theta}$ is the regression coefficient of our Cox hazard model and $se(\hat{\theta})$ is the standard error in the estimated parameter.

$$W = \frac{\hat{\theta}}{se(\hat{\theta})} \tag{6.4}$$

- **Hazard ratios**: We calculate the hazard ratio (HR) for a given covariate using equation 6.5, where β is the regression coefficient of the model. Therefore, A $HR > 1$ signifies a covariate being associated with greater hazard negatively impacting survival times. A $HR < 1$ is associated with covariates that reduce the hazard while $HR = 1$ has no effect on survival time. We then generate the 95% confidence intervals (CI) of our model's hazard ratio. If the model does not include HR=1 (the null hypothesis) in its CI, then the associated covariate is a significant risk factor to survival. By computing the overall statistical significance (p-value) of our model using a log-rank test we further substantiate the validity and the significance of our model covariate as a potential risk factor to patient survival.

$$HR = exp(\beta) \tag{6.5}$$

Evaluating prediction performance of models

We have chosen to use root mean square error (RMSE) as the primary metric to analyze the predictive performance of our survival models. In linear regression problems, RMSE is the most common choice for evaluating model performance as it provides a way to quantify exactly how far our model's predictions are from the ground-truth values. Furthermore, using RMSE over standard MSE has the advantage of producing the deviation in the same unit as our predicted variable. In our case, this will tell us, in terms of months, how far off our survival time estimates are from the expected clinical prognosis.

We computed the RMSE for survival time predictions generated by each model across a collection of WSIs using equation 6.6. We essentially calculate the squared difference between the predicted value, t_i, and the actual clinically recorded survival duration, $\hat{t}_i$, for each patient. Finally we take the mean error across all N WSIs and square root it to get the average error in months.

$$RMSE = \sqrt{\frac{\sum_{n=1}^{N}(t_i - \hat{t}_i)^2}{N}} \tag{6.6}$$

We have computed the average RMSE values across multiple sets of WSIs to give us a better estimate of the error that should be expected when predicting the survival time from a random WSI. Additionally we also compute the standard deviation (SD) of these average RMSE values to quantify the distribution of errors. The SD gives us a measure of the model's uncertainty range to predict how good or bad a prediction is likely to be. This SD value is calculated using equation

6.7 where t_i is the RMSE of each WSI set and $\bar{t_i}$ is the mean RMSE across all WSIs.

$$StdDev = \sqrt{\frac{\sum_{n=1}^{N}(t_i - \bar{t_i})^2}{N-1}} \tag{6.7}$$

6.2.2 How effective were our chosen image-based features as survival model covariates?

We calculated the mean malignant intensity (MMI) and malignancy spread score (MSS) from each WSI's predictions generated by our model. We used the `CoxPHFitter` class from the Lifelines package (described in section 2.3.1) to generate a univariable Cox hazard function for each covariate using the ground truth survival durations and the computed malignancy feature scores. From each fitted model, using the `print_summary()` function, we obtained the model regression coefficient (β), hazard ratio ($\exp(\beta)$), standard error in the regression coefficient ($se(\beta)$) and calculated the Wald statistic (z-value) using equation 6.4. The final metric values for each covariate are displayed in table 6.2.

Covariate	β	$\exp(\beta)$	$se(\beta)$	z	p
Malignancy Spread Score	1.02	2.78	0.58	1.76	0.08
Mean Malignant Intensity	3.41	30.20	1.51	2.25	0.02

Table 6.2: *The values obtained for the regression coefficient, hazard ratio, standard error in regression, Wald statistic (z-value) and overall statistical significance (p-value) of each covariate's Cox hazard model.*

As noted in table 6.2, we obtained Cox models having a positive regression coefficient of 1.02 and 3.41 respectively for each covariate. A value of $\beta > 0$ for the hazard functions signifies that both covariates represent a non-protective risk factor, negatively influencing patient survival at higher values. This implies that our chosen image-based features were able to independently capture an approximate representation of the metastases severity in a sample to an extent where the extracted feature score exhibited a positive correlation with the associated hazard of death.

The Cox model with the MSS covariate displayed a hazard ratio of $exp(1.02) = 2.78$ while the MMI model had a HR of $exp(3.41) = 30.20$. With HR > 1, both models are associated with having a negative effect on survival times at higher covariate levels. However, the MMI model has a much larger HR than MSS, implying that it has a much stronger influence on the associated metastatic hazard than the MSS covariate. Due to the proportional nature of the Cox model, every unit increase in the MMI value assumes an increase in hazard and a proportional decrease in survival time by a factor of 30.2 compared to 2.78 for the MSS model. Thus, the MMI covariate allowed the hazard function to span over a large range of values and significantly model differences in survival time associated with very small changes in the covariate level. Whereas, with the MSS covariate, the lower HR value restricted the hazard function to a smaller range making differences in survival time at varying covariate levels less pronounced and more likely to be obscured due to values being condensed to a small range.

In order to test for statistical significance of our models, we performed a null hypothesis test by plotting the 95% confidence intervals of each model's log-hazard ratio ($log(HR) = log(\exp\beta) = \beta$), as shown in figure 6.4. The upper and lower bounds of the 95% confidence intervals are listed in table 6.3. Due to the large difference between the upper and lower bound HR values of the two covariates, we chose $\log(HR)$ in order to scale down HR values to a similar range. We observe that the CI of the MMI covariate does not include the null value of $HR = 1 \implies \log(HR) = 0$ (shown as the blue line in figure 6.4). This validates the MMI covariate as a significant risk factor affecting patient survival. However, the MSS covariate includes the null value in its confidence interval making it less statistically significant as a risk factor to patient survival as the true value of corresponding hazard can take the value of 1.

Covariate	log(HR) lower 95%	log(HR) upper 95%	HR lower 95%	HR upper 95%
MSS	-0.12	2.17	0.89	8.74
MMI	0.44	6.37	1.56	586.49

Table 6.3: *The upper and lower hazard ratio bounds for 95% confidence intervals. Also displaying the upper and lower bounds for log(HR) confidence intervals.*

It is worth noting that despite having a statistically significant effect on patient survival, the log(HR) of the MMI covariate displayed a large confidence interval. This made the model's predictions less accurate as the actual HR for a given covariate level can vary over a very large range of values (shown in table 6.3). The high uncertainty involved in determining the true hazard value using the MMI covariate propagated into the survival time estimates making the predictions very noisy and deviate significantly from the expected ground truth (discussed in the next section). On the contrary, the MSS covariate had a smaller CI for its log(HR), making the true HR lie within a much smaller range of values. This reduced the uncertainty involved in determining the true hazard value for a given covariate. However, the model fit of this covariate itself was not statistically significant.

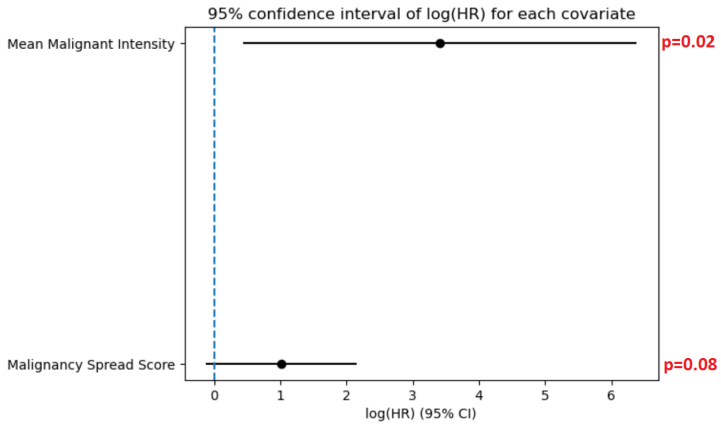


Figure 6.4: *We illustrate the 95% confidence intervals of the log(HR) of Mean Malignant Intensity and Malignancy Spread Score covariates. The null hypothesis of log(HR)=0 is highlighted. We observe the MSS covariate includes the null value within its CI, while the MMI covariate does not. We also display the corresponding p-values, calculated using the log-rank test, of each covariate to further substantiate their respective statistical significance.*

We calculated a Wald's z-value of 1.76 for the MSS model and 2.25 for the MMI model using equation 6.4. This supports our previous results by establishing the MMI covariate as having the stronger effect on patient survival. The MSS model failing to reject the null hypothesis according to the 95% confidence interval means it may have no effect on the survival risk. However, the computed z-value assigns a non-zero significance to the MSS covariate in being able to effect patient survival. This implies that for the given patient data, the MSS covariate is not likely to take up the null hazard value and is able to model some meaningful, albeit minor, effect on survival times. Our generated models also output the overall statistical significance of each covariate as a p-value (shown in figure 6.4). In the context of Cox models, a $p < 0.05$ is considered reasonably significant as it represents a 5% risk of our model not fitting the data. Consequently, the MMI covariate despite having higher uncertainty in its hazard ratio, displayed the more statistically significant ($p = 0.02$) model fit compared to the MSS covariate ($p = 0.08$).

6.2.3 How effectively does our model generate overall survival time predictions from WSI-based features?

Given the limited amount of data available for survival modeling, we undertook a 5-fold cross validation approach to measure our model's prediction performance. We split our dataset of 74 WSIs into 5 parts, using 4 parts to fit our Cox model and 1 part for validation at every iteration. We validated our model's performance by first generating a median survival time prediction for each WSI in the validation fold using the model fitted with the training folds. We calculated the RMSE values between predicted survival times and the ground truth corresponding to each validation set. Finally, as shown in table 6.4, we obtain the average RMSE across all 5 folds to get the overall deviation of our predictions from the ground truth values. We also calculated the associated standard deviation (SD) to understand the distribution of errors in our predictions.

Covariate	Mean RMSE (months)	+/− SD (months)
MMI	47.0748	29.0246
MSS	47.1201	29.0011

Table 6.4: *Average RMSE and SD values obtained after 5 cross fold validation of our survival model using MMI and MSS covariates. The RMSE (in months) was calculated for each validation fold and finally the mean RMSE across 5 folds was measured along with the standard deviation in the RMSE values.*

This gives us an idea of how far our survival model's predictions are from the ground truth, when using each covariate. We observe a mean RMSE ± SD of 47.075 ± 29.025 for predictions produced by the MMI model on our WSI dataset. Whereas, the MSS model yielded an overall error of 47.120 ± 29.001 in its predictions. This implies that both our models generate noisy survival time predictions with high average uncertainty. However, the error is also distributed over a large range as seen from the associated standard deviation values. As observed in the boxplot in figure 6.5, the models can generate predictions that are as precise as upto within 18 months of the ground-truth values. However, we also observe a large outlier of RMSE greater than 90 months which can severely skew any predictions. Both covariates showed similar performance - prediction errors being concentrated in the 35-50 month range with a similar overall distribution. Comparatively, the MMI model displayed marginally lower average errors than the MSS model which implied slightly better prediction accuracy. This is likely because the probability predictions more accurately quantified the actual extent of cancer compared to binary predictions. The binary model thresholds predictions at 0.5 probability, causing low confidence predictions (close to 0.5) to get sharply classified as malignant or benign. Using the probability value as malignant intensity accounts for such low confidence predictions, weighing them as less intense and giving us a more precise quantification of malignancy.

How good are the predicted survival times relative to the clinical values of overall patient survival duration?
In figure 6.6 we have plotted the median survival time predictions of our survival models against the clinically determined survival duration of all 74 patients. Figure 6.6a and 6.6b display the prediction results generated using the MMI and MSS covariates respectively.

We observe that there is no definite linear correlation between the predicted and true values as we would expect if predictions were aligned with their corresponding ground-truth. For both models, the predicted times are between 30 and 80 months and don't scale very well for samples with higher survival times. Similarly, samples with low survival times of <30 months are not predicted accurately. We used the time between diagnosis and point of death to fit our models. This makes our data left-censored as it does not account for the length of time a patient might have had cancer before the diagnosis. A late diagnosis would result in lower survival time recorded due to disease progression and possibly the lack of treatments options at later stages. This

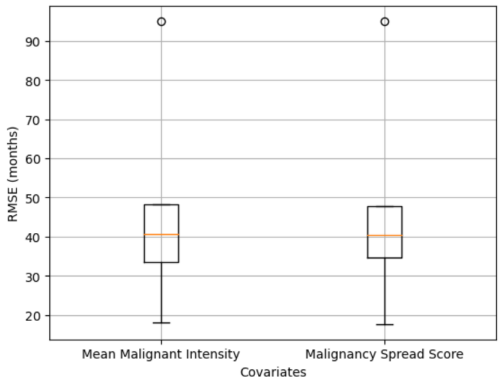
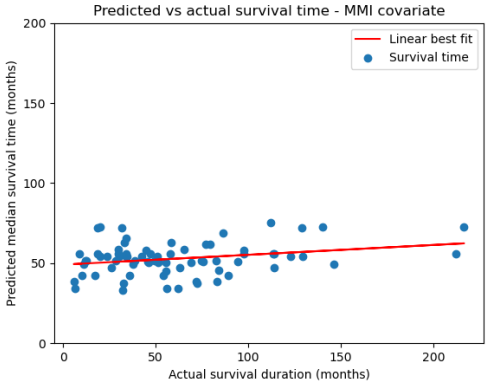
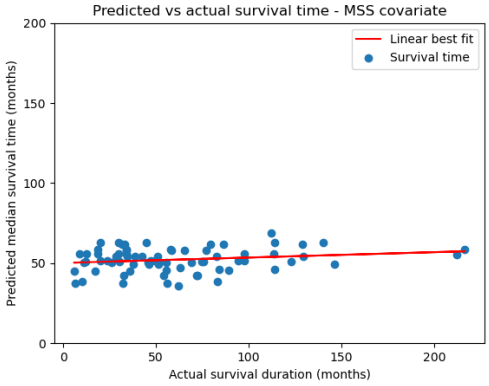


Figure 6.5: *Box plot showing the distribution of RMSE in each covariate's survival time predictions. Both covariates displayed similar average errors with a lower bound of 18 months and an upper outlier of 90 months.*

effectively raises the baseline hazard of the patient and since our models do not account for these effects, the proportional hazard obtained is likely to be underestimated leading to overestimated survival time predictions. Similarly, our models do not account for the reduction in hazard caused by treatment received post-diagnosis that has the effect of extending survival time of patients. The level of treatment, if any at all, is also variable across patients as it depends on factors like affordability or presence of health insurance. This underestimates the hazard by varying degrees when basing our model predictions on the physical extent of the tumour alone.



(a) *Mean malignant intensity covariate - median survival time predictions against clinically determined survival duration of each patient*



(b) *Malignancy spread score covariate - median survival time predictions against clinically determined survival duration of each patient*

Figure 6.6: *Plots showing prediction performance of our models against ground-truth survival times*

We do observe that our model predictions display a reasonable trend in relative survival time across samples. As we move further right in figure 6.6, with an increase in true survival time, our predicted values also show an upward trend. This implies that our model is not predicting random times and the covariates are able to capture the relative differences in metastases severity in terms of tumour spread, assigning higher survival times to less severe and low survival times to more severe cases. The predicted time, does not reflect real-life values because the computed univariate hazard does not have the explanative power to represent an individual's true hazard which is affected by several factors, causing predictions to not vary massively from the weighted

average of the model. For example, age is a very revealing factor when it comes to patient survival as an older patient is likely to have a shorter overall survival time from the point of diagnosis compared to a patient who was diagnosed at a younger age. Similarly, smoking status, underlying medical conditions, immune system health and several quality of life factors directly influence the malignancy hazard. As we are taking a univariable approach, computing the baseline hazard with a single covariate introduces noise from all the unknown factors into our model. This makes the hazard function less generalizable to multiple patients since each has their own unique set of external circumstances that influence overall survival besides the physical manifestation of cancer. This translates into the proportional hazard calculated for each covariate level having high uncertainties, making our predictions highly erroneous.

How does limited data availability impact prediction performance?
We were restricted to a dataset of 74 WSIs to use for survival modeling. In order to investigate the effect of increasing training data on regression performance, we first obtained a random 90-10 train-test split of our TCGA breast cancer dataset. As shown in table 6.5, we the split that 90% data into 4 subsets of specified sizes. With each subset, we fitted Cox models for both covariates and obtained the median survival time predictions of samples in the held out 10% data. Using the generated predictions and the ground-truth survival duration of each WSI we calculated the RMSE value, as shown in the table, for each covariate.

Dataset size	MMI RMSE	MSS RMSE
25%	41.77	54.11
50%	35.73	47.48
75%	34.49	43.38
100%	34.50	41.79

Table 6.5: *The calculated RMSE results (in months) for both, mean malignant intensity and malignancy spread score covariates, when the model is fitted with data subsets of the specified size.*

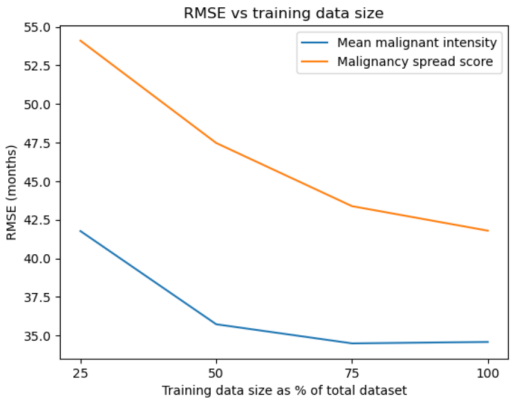


Figure 6.7: *Illustrating the effect of varying training dataset size on RMSE of survival time predictions. The RMSE decreases with increase in training data, expressed as a percentage of total data available.*

As illustrated in figure 6.7, we observe a decrease in the prediction RMSE for both covariates as dataset size increases. The difference between 25% and 100% data is only additional samples, which produces a large decrease in prediction RMSE with a relatively small increase in training data. Therefore, despite our model's predictions having large error values, the observed trend indicates that it could be caused by the lack of available clinical data - increasing the size of the training dataset is likely to improve our model's predictions. However, the RMSE reduction plateaus between 75% and 100% data indicating that increasing data alone is not going to infinitely improve predictions. This is because, our survival data had a large gap between 150 and 200 months (figure 6.6); the training samples did not contain a true survival duration falling in that range. As a result, our model failed to correctly characterize covariates belonging to the missing range of survival times, which led to poor generalization and higher error values. Therefore, increasing training data size is not likely to rectify and improve this regression performance

beyond our existing data. However, a larger sample size would be more likely to uniformly span a more complete range of ground-truth values allowing our model to make better predictions over a larger covariate range.

6.3 Discussion

The primary limitation with our classification system is that the model was trained using patches of breast metastatic lymph node tissue. The tissue composition in lymph nodes is very different from that observed in the TCGA breast tissue samples we have used for survival analysis. Therefore, when given a sample entirely composed of tissue previously unseen in the training data, our model is likely to fail in accurately identifying metastatic cells amidst the healthy tissue. However, we can argue that the learning obtained from breast metastases in lymph nodes does transfer to an extent when making predictions on breast tissue. This is evident from the segmentation maps generated by our model for the TCGA WSIs where the predictions were able to identify distinct regions of high malignant concentration instead of randomly predicting patches all over the tissue. This is likely because the individual malignant cells that metastasize from breast tissue to the lymph nodes retain the same characteristics such that they can be identified by their visual appearance. These results also support the success of our chosen ConvNet architecture as it was able to produce reasonably good results even with contextually different training data. Our model's performance would greatly improve if we directly trained it on breast tissue samples instead of transferring the knowledge from lymph samples. We resorted to using PCAM data, due to the limited availability of annotated breast cancer samples for our supervised learning approach. A weakly supervised approach would be more appropriate here as it would not require an infeasibly large tiled dataset to be manually annotated. It would instead utilize slide-level regions of interest annotations to learn relevant tissue structures.

The task of survival modeling was more complex, exhibiting high inaccuracies in the predicted overall survival times which was caused by several factors. The primary limitation was that our chosen covariates only accounted for the risk posed by tumour spread in a given sample. In reality, patient survival is influenced by factors beyond the physical extent of malignancy. These are likely to include age, smoking status, underlying medical conditions, cancer subtype, immune system response and treatment undergone by the patient. Choosing a univariate Cox model disregarded the influence of these factors besides the associated covariate when calculating the disease hazard of each patient. Consequently, the model linearly scaled an incomplete baseline hazard for each patient, failing to precisely characterize their individual survival risk as everyone has a unique combination of external factors affecting their prognosis. However, we observed that our predictions reflected the correct intra-sample survival trends, such that samples with longer survival durations were predicted a better prognosis than samples having lower survival. This implied that cancer severity quantified by our covariates had a limited, albeit meaningful, influence on estimating the associated disease hazard allowing it to make predictions in the appropriate directions. Our predictions could be improved by opting for a multivariable Cox approach using non-image based clinical data as additional covariates. This would estimate a more accurate hazard for each patient by accounting for the influence of external factors.

Overall, the correlation between histopathological image-based features and patient survival was not strong enough to predict survival times precisely. Even with more data to ensure samples were uniformly distributed over a complete range of survival times, the influence of external variables is so significant that survival times cannot be viably generalized using biopsy images alone. However, we did infer that the extent of malignancy observed in WSIs does display some bearing on patient survival as it characterized the general survival expectation correctly. Thus, our original goal to minimize the delay in prognosis delivery by eliminating extensive data collection tests can still be achieved by supplementing the biopsy image-based features with basic clinical information that do not require complex procedures to obtain.

7 Conclusion

We have explored the viability of using histopathological slide images as a quantifier of cancer severity and thereby an estimator of overall survival time in breast cancer patients. To perform this investigation, we have developed a two-part system corresponding to the diagnosis and prognosis stages. The former consists of a supervised learning based ConvNet model, trained on 100,000 images of breast metastases to learn how to identify malignant and healthy regions in a given tissue slide image. For the latter, we prepared a dataset of 74 WSIs of deceased breast cancer patients, with their associated survival duration measured from the point of diagnosis. The trained classification model was used on each of these WSIs to extract features quantifying disease severity as the extent of malignancy spread which were then used as covariates in survival modeling. We fitted independent univariate Cox hazard models with the extracted covariates and survival durations. Finally, from these models we regressed a median survival time for each patient based on their associated covariate level using a Kaplan-Meier method.

Our classification model relied on transferring the learning from metastases affected lymph node tissue to breast tissue. We measured our model's diagnostic performance on the lymph node data which performed reasonably well with accuracy of nearly 70%. However, we were unable to quantitatively evaluate how well it works, if at all, on TCGA breast tissue sections without pathologist verification or having some clinically annotated slides to validate our predictions against. We therefore relied on qualitatively evaluating the tumour maps produced from breast tissue samples. We found that the segmentation maps generated by our model displayed little to none randomly scattered noisy predictions and consistently predicted specific regions of high malignancy concentration which bears resemblance to somewhat realistic tumour growth.

On the other hand, we found that our survival prediction system was significantly more erroneous. Only one of our extracted image-based covariates, the Mean Malignant Intensity, proved to have a statistically significant effect on patient survival. Although, it showed high error values causing the predicted survival times to deviate from the expected ground truth, the model's standard deviation indicated that the covariate can predict survival times upto within 18 months of actual survival. Both models, however, signified that the covariates extracted from the WSIs can successfully distinguish different relative metastases severity to some extent as the survival time predictions displayed correct intra-sample trends; more severe samples with low survival duration were predicted lower survival times relative to less severe samples with higher survival.

The inaccuracies in survival time prediction were caused by the inherently complex nature of survival where it is influenced by a multitude of factors besides metastases severity alone. The data used to fit the regression models were left censored giving us no information on any delays incurred leading upto the diagnosis and we also do not factor in any treatment received post diagnosis. These unknown factors, are likely to have a significant influence on patient survival by extending or reducing the overall survival time respectively. Similarly, several other physiological and quality of life factors serve as risk factors to influence patient survival which our univariate models do not account for. Therefore, we concluded that using histopathological images in isolation is not viable in estimating anything more than a very rough median survival baseline as each patient's individual disease hazard is influenced by a unique set of external factors which cannot be characterized from biopsied tissue alone.

We can validate the overall outcome of our work by comparing it to a similar study done by Wetstein et al. (2022). They proposed a deep learning system to grade breast cancer tumours observed in biopsied WSIs and used the tumour grade as a covariate to fit the univariate Cox hazard model. Their model arrived at a very similar result when making overall survival time predictions. The covariate showed statistical significance ($p < 0.05$) on disease hazard but was only able to "distinguish[es] between low/intermediate and high grade tumors and find[s] a trend in the survival of the two predicted groups."

Overall, this study served as a proof of concept that histological images do hold sufficient information of metastatic morphology to valuably supplement the diagnostic and prognostic process for cancer patients. In effect, completely automated prognosis estimation from slide images alone is not likely to replace manual pathologist evaluation any time soon due to the sensitive nature of making such decisions. However, it can provide a very crude initial approximation of overall survival time to identify potentially high-risk cases and prioritize patients for more advanced treatment.

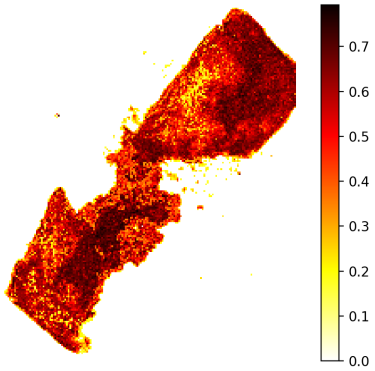
7.1 Future Work

A potential improvement to explore would be to use multivariable Cox models to incorporate additional covariates on top of the extracted image features. This is supported by the work of Kather et al. (2019), where they used "tumor, node, and metastases (TNM) stage, sex, and age as covariates" in addition to the deep stromal score to predict suvival times in colorectal cancer patients. Multiple covariates are likely to produce a more realistic hazard model for each patient that will more accurately characterize the individual risk that influence survival, often more significantly than the tumour spread observed in biopsied tissue.

Another major augmentation to improve survival time predictions would be to use a different survival model instead of the Cox proportional hazards model. Though it is the most popular approach in demonstrating the effect of varying covariates on relative disease hazard over all points in time, the major drawback associated with Cox models is that the proportional hazard assumption does not hold over time in real life. This is particularly the case when dealing with covariates whose effect becomes more diminished (or pronounced) over time and does not scale linearly with a fixed baseline such as patients developing natural resistance against some form of treatment causing its protective effect to weaken over time. As proposed by Klein et al. (2014), the Aalen additive hazards regression would be more suitable in modeling realistic survival times as it is a parametric model that allows the integration of time-varying covariates.

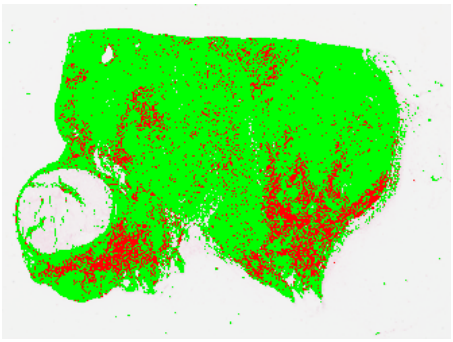
A Appendix - Cancer detection results and associated survival duration

(a) Binary tumour map - red areas imply malignant and green implies benign tissue

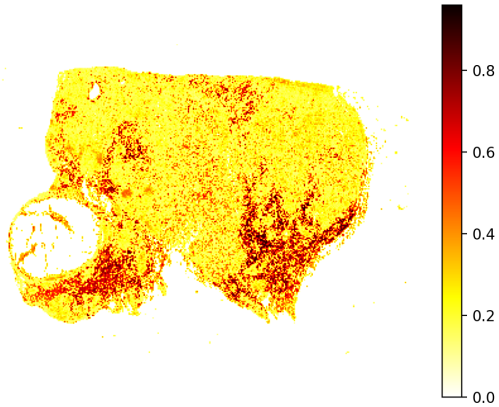


(b) Tumour probability heatmap - darker regions imply a stronger malignant signal

Figure A.1: *Predictions generated by our cancer detection model on breast tissue sample of patient TCGA-AC-A23H.* ***Recorded survival duration: 6.47 months.***

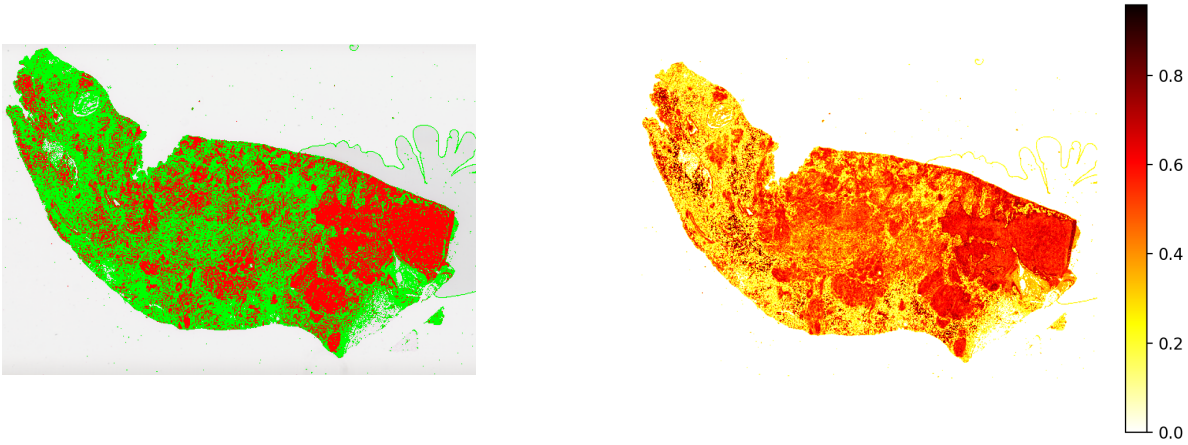


(a) Binary tumour map - red areas imply malignant and green implies benign tissue



(b) Tumour probability heatmap - darker regions imply a stronger malignant signal

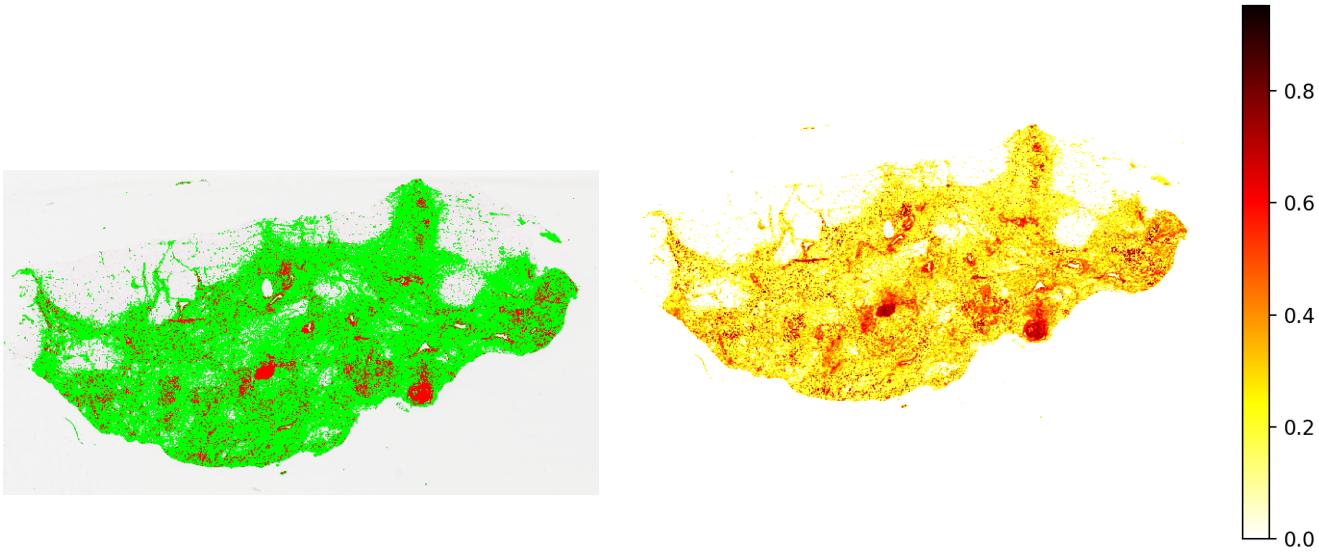
Figure A.2: *Predictions generated by our cancer detection model on breast tissue sample of patient TCGA-GM-A2DA.* ***Recorded survival duration: 216.59 months.***



(a) Binary tumour map - red areas imply malignant and green implies benign tissue

(b) Tumour probability heatmap - darker regions imply a stronger malignant signal

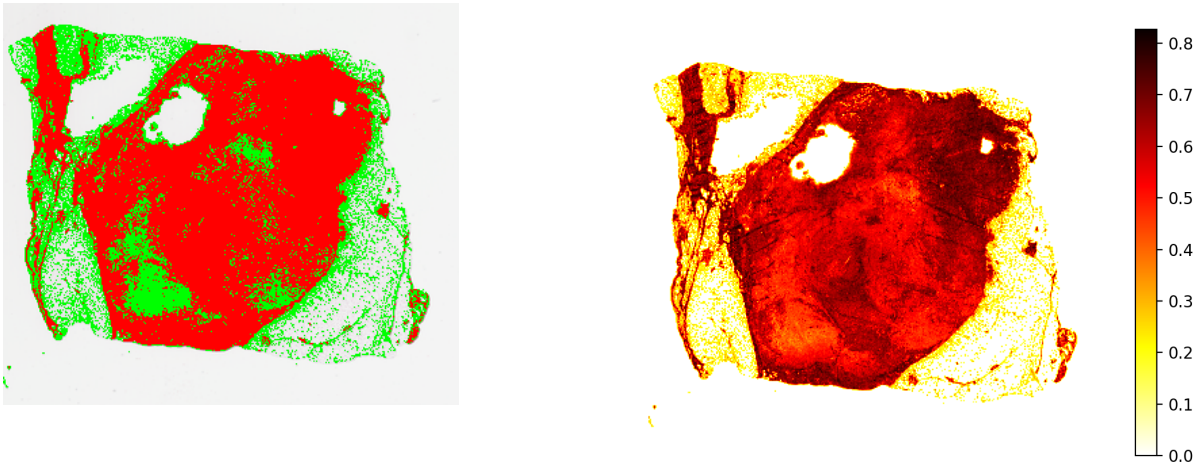
Figure A.3: *Predictions generated by our cancer detection model on breast tissue sample of patient TCGA-BH-A1EV.* ***Recorded survival duration: 11.99 months.***



(a) Binary tumour map - red areas imply malignant and green implies benign tissue

(b) Tumour probability heatmap - darker regions imply a stronger malignant signal

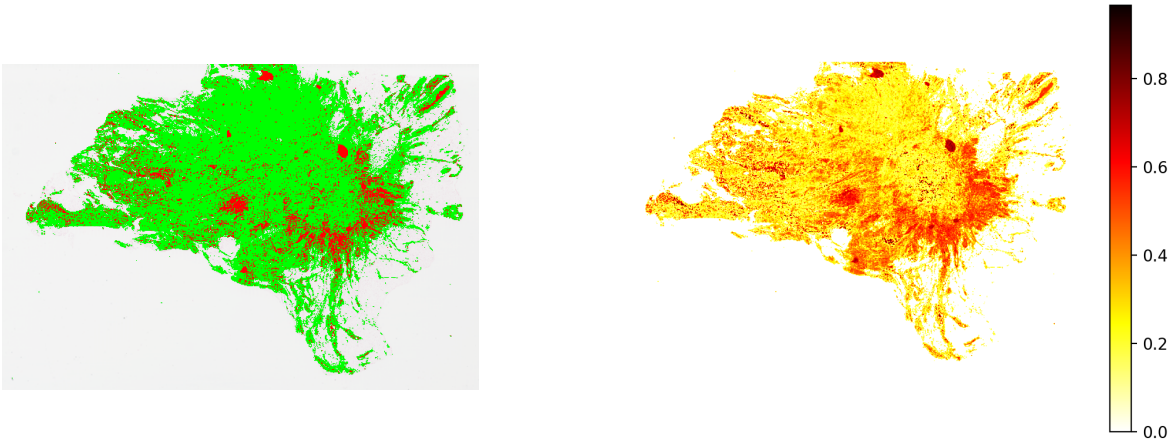
Figure A.4: *Predictions generated by our cancer detection model on breast tissue sample of patient TCGA-B6-A0RQ.* ***Recorded survival duration: 140.18 months.***



(a) Binary tumour map - red areas imply malignant and green implies benign tissue

(b) Tumour probability heatmap - darker regions imply a stronger malignant signal

Figure A.5: *Predictions generated by our cancer detection model on breast tissue sample of patient TCGA-AR-A1AR.* ***Recorded survival duration: 17.21 months.***



(a) Binary tumour map - red areas imply malignant and green implies benign tissue

(b) Tumour probability heatmap - darker regions imply a stronger malignant signal

Figure A.6: *Predictions generated by our cancer detection model on breast tissue sample of patient TCGA-B6-A0IP.* ***Recorded survival duration: 128.98 months.***

Figure	Sample ID	True survival duration (months)
A.1	TCGA-AC-A23H	6.47
A.2	TCGA-GM-A2DA	216.59
A.3	TCGA-BH-A1EV	11.99
A.4	TCGA-B6-A0RQ	140.18
A.5	TCGA-AR-A1AR	17.21
A.6	TCGA-B6-A0IP	128.98

Table A.1: *Clinical data of overall survival duration for each patient*

B Appendix - Code snippets

```
# root group
/
   # image data group
   x/
      # tile id group
      0/
         [96 x 96 x 3 - 8bit unsigned integer numpy array]
      1/
      2/
      ...
   # annotation group
   y/
      # tile id group
      0/
         0
      1/
         1
      2/
      3/
      ...
```

Listing B.1: *PCAM Data HDF5 schema - Each set has a group labeled 'x' - containing all the tiles as numpy arrays of 8-bit unsigned integers. Each array has a unique id (0, 1, 2...) and shape* $96 \times 96 \times 3$ *indicating a* 96×96*px RGB tile. The second group labeled 'y' contains the annotations for each tile. The annotations are shown as binary labels against each tile id.*

```
from openslide import open_slide
from openslide.deepzoom import DeepZoomGenerator
slide = open_slide(wsi_file_path)
tiles = DeepZoomGenerator(slide, tile_size=96, overlap=0, limit_bounds=False)
level_num = (tiles.level_count)-2
cols, rows = tiles.level_tiles[level_num]
for row in range(rows):
      for col in range(cols):
         temp_tile = np.array(tiles.get_tile(level_num, (col,
             row)).convert('RGB'))
         if temp_tile.std() > 15 and temp_tile.mean() < 230 and temp_tile.shape
             == (96,96,3):
            try:
               macenko_norm, h_norm, e_norm = macenko_norm_HnE(temp_tile)
               tiff.imsave(save_path, macenko_norm)
```

```
            except np.linalg.LinAlgError as LinAlgError:
               tiff.imsave(save_path, temp_tile_RGB_np)
```

Listing B.2: *Code excerpt showing how OpenSlide and DeepZoomGenerator are used to obtain tiles of specified dimension and resolution level from an input WSI image and saving them as a .tiff image. We also added a modification to normalize each tile (explained in the next section) before saving. It calls* `macenko_norm_HnE()` *on each tile's numpy array and saves the normalized image. We also catch any SVD convergence errors thrown by tiles that have large areas of white space. This is the case for external tiles containing debris outside the main body of tissue.*

```
class breastCancerDataset(Dataset):
def __init__(self, data_dir, transform, data_type='train'):
   # path to images
   data_path = os.path.join(data_dir, data_type+"/tiles")
   # get list of images
   fnames = os.listdir(data_path)
   self.full_fnames = [os.path.join(data_path, f) for f in fnames]
   # labels are in a csv file names train_labels.csv
   labels_path = os.path.join(data_dir, data_type+"/" + data_type + "_labels.csv")
   labels_df = pd.read_csv(labels_path)
   # set data frame index to id
   labels_df.set_index("id", inplace=True)
   # obtain labels from data frame
   self.labels = [labels_df.loc[int(f[:-5])].values[3] for f in fnames]
   self.transform = transform
def __getitem__(self, index):
   if isinstance(index, slice):
      start = 0 if index.start == None else index.start
      stop = -1 if index.stop == None else index.stop
      step = 1 if index.step == None else index.step
      images = []
      for idx in range(start, stop, step):
         img = Image.open(self.full_fnames[idx])
         img = self.transform(img)
         images.append(img)
      return images, self.labels[start:stop:step]
   else:
      # open image, apply transform and return with label
      image = Image.open(self.full_fnames[index])
      image = self.transform(image)
      return image, self.labels[index]
```

Listing B.3: *Custom* `breastCancerDataset` *class to fetch all training and validation images and ground truth labels from a given base directory. It creates objects associating each image's path in disk with its corresponding label to make a stream of data that pytorch's dataloaders can readily use to fetch batches of images without needing to initially load all images to memory*

7 Bibliography

Aeffner, F., Zarella, M. D., Buchbinder, N., Bui, M. M., Goodman, M. R., Hartman, D. J., Lujan, G. M., Molani, M. A., Parwani, A. V., Lillard, K. et al. (2019), 'Introduction to digital image analysis in whole-slide imaging: a white paper from the digital pathology association', *Journal of pathology informatics* **10**(1), 9.

Anghel, A., Stanisavljevic, M., Andani, S., Papandreou, N., Rüschoff, J. H., Wild, P., Gabrani, M. and Pozidis, H. (2019), 'A high-performance system for robust stain normalization of whole-slide images in histopathology', *Frontiers in medicine* **6**, 193.

Bancroft, J. D. and Layton, C. (2013), *The hematoxylins and eosin*, Elsevier.

Bandi, P., Geessink, O., Manson, Q., Van Dijk, M., Balkenhol, M., Hermsen, M., Bejnordi, B. E., Lee, B., Paeng, K., Zhong, A. et al. (2018), 'From detection of individual metastases to classification of lymph node status at the patient level: the camelyon17 challenge', *IEEE transactions on medical imaging* **38**(2), 550–560.

Bankhead, P., Loughrey, M. B., Fernández, J. A., Dombrowski, Y., McArt, D. G., Dunne, P. D., McQuaid, S., Gray, R. T., Murray, L. J., Coleman, H. G. et al. (2017), 'Qupath: Open source software for digital pathology image analysis', *Scientific reports* **7**(1), 1–7.

Bejnordi, B. E., Veta, M., Van Diest, P. J., Van Ginneken, B., Karssemeijer, N., Litjens, G., Van Der Laak, J. A., Hermsen, M., Manson, Q. F., Balkenhol, M. et al. (2017), 'Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer', *Jama* **318**(22), 2199–2210.

Bewick, V., Cheek, L. and Ball, J. (2004), 'Statistics review 12: survival analysis', *Critical care* **8**(5), 1–6.

Chan, J. K. (2014), 'The wonderful colors of the hematoxylin–eosin stain in diagnostic surgical pathology', *International journal of surgical pathology* **22**(1), 12–32.

Cox, D. R. (1972), 'Regression models and life-tables', *Journal of the Royal Statistical Society: Series B (Methodological)* **34**(2), 187–202.

Cruz-Roa, A., Basavanhally, A., González, F., Gilmore, H., Feldman, M., Ganesan, S., Shih, N., Tomaszewski, J. and Madabhushi, A. (2014), Automatic detection of invasive ductal carcinoma in whole slide images with convolutional neural networks, *in* 'Medical Imaging 2014: Digital Pathology', Vol. 9041, SPIE, p. 904103.

Ghaznavi, F., Evans, A., Madabhushi, A. and Feldman, M. (2013), 'Digital imaging in pathology: whole-slide imaging and beyond', *Annu Rev Pathol* **8**(1), 331–359.

Goode, A., Gilbert, B., Harkes, J., Jukic, D. and Satyanarayanan, M. (2013), 'Openslide: A vendor-neutral software foundation for digital pathology', *Journal of pathology informatics* **4**(1), 27.

Gorelick, L., Veksler, O., Gaed, M., Gómez, J. A., Moussa, M., Bauman, G., Fenster, A. and Ward, A. D. (2013), 'Prostate histopathology: Learning tissue component histograms for cancer detection and classification', *IEEE transactions on medical imaging* **32**(10), 1804–1818.

Gurcan, M. N., Boucheron, L. E., Can, A., Madabhushi, A., Rajpoot, N. M. and Yener, B. (2009), 'Histopathological image analysis: A review', *IEEE reviews in biomedical engineering* **2**, 147–171.

Kather, J. N., Krisam, J., Charoentong, P., Luedde, T., Herpel, E., Weis, C. A., Gaiser, T., Marx, A., Valous, N. A., Ferber, D., Jansen, L., Reyes-Aldasoro, C. C., Zörnig, I., Jäger, D., Brenner, H., Chang-Claude, J., Hoffmeister, M. and Halama, N. (2019), 'Predicting survival from colorectal cancer histology slides using deep learning: A retrospective multicenter study', *PLoS Medicine* **16**.

Katzman, J. L., Shaham, U., Cloninger, A., Bates, J., Jiang, T. and Kluger, Y. (2018), 'Deepsurv: personalized treatment recommender system using a cox proportional hazards deep neural network', *BMC medical research methodology* **18**(1), 1–12.

Khened, M., Kori, A., Rajkumar, H., Krishnamurthi, G. and Srinivasan, B. (2021), 'A generalized deep learning framework for whole-slide image segmentation and analysis', *Scientific Reports* **11**.

Kino, G. S. and Corle, T. R. (1996), *Confocal scanning optical microscopy and related imaging systems*, Academic Press, pp. 1–66.

Klein, J. P., Van Houwelingen, H. C., Ibrahim, J. G. and Scheike, T. H. (2014), *Handbook of survival analysis*, CRC Press Boca Raton, FL:.

Li, X. and Plataniotis, K. N. (2015), 'A complete color normalization approach to histopathology images using color cues computed from saturation-weighted statistics', *IEEE Transactions on Biomedical Engineering* **62**(7), 1862–1873.

Liu, H. and Kurc, T. (2022), 'Deep learning for survival analysis in breast cancer with whole slide image data', *Bioinformatics* **38**, 3629–3637.

Macenko, M., Niethammer, M., Marron, J. S., Borland, D., Woosley, J. T., Guan, X., Schmitt, C. and Thomas, N. E. (2009), A method for normalizing histology slides for quantitative analysis, *in* '2009 IEEE international symposium on biomedical imaging: from nano to macro', IEEE, pp. 1107–1110.

Pedersen, A., Valla, M., Bofin, A. M., De Frutos, J. P., Reinertsen, I. and Smistad, E. (2021), 'Fastpathology: an open-source platform for deep learning-based research and decision support in digital pathology', *Ieee Access* **9**, 58216–58229.

Priego-Torres, B. M., Sanchez-Morillo, D., Fernandez-Granero, M. A. and Garcia-Rojo, M. (2020), 'Automatic segmentation of whole-slide h&e stained breast histopathology images using a deep convolutional neural network architecture', *Expert Systems With Applications* **151**, 113387.

Qin, P., Chen, J., Zeng, J., Chai, R. and Wang, L. (2018), 'Large-scale tissue histopathology image segmentation based on feature pyramid', *EURASIP Journal on Image and Video Processing* **2018**(1), 1–9.

Schober, P. and Vetter, T. R. (2021), 'Kaplan-meier curves, log-rank tests, and cox regression for time-to-event data', *Anesthesia & Analgesia* **132**(4), 969–970.

Sobin, L. H., Gospodarowicz, M. K. and Wittekind, C. (2011), *TNM classification of malignant tumours*, John Wiley & Sons.

Sung, H., Ferlay, J., Siegel, R. L., Laversanne, M., Soerjomataram, I., Jemal, A. and Bray, F. (2021), 'Global cancer statistics 2020: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries', *CA: a cancer journal for clinicians* **71**(3), 209–249.

van Treeck, M., Cifci, D., Laleh, N. G., Saldanha, O. L., Loeffler, C. M., Hewitt, K. J., Muti, H. S., Echle, A., Seibel, T., Seraphin, T. P. et al. (2021), 'Deepmed: A unified, modular pipeline for end-to-end deep learning in computational pathology', *BioRxiv* .

Veeling, B. S., Linmans, J., Winkens, J., Cohen, T. and Welling, M. (2018), 'Rotation equivariant CNNs for digital pathology'.

Wang, F., Oh, T. W., Vergara-Niedermayr, C., Kurc, T. and Saltz, J. (2012), Managing and querying whole slide images, *in* 'Medical Imaging 2012: Advanced PACS-Based Imaging Informatics and Therapeutic Applications', Vol. 8319, SPIE, pp. 137–148.

Wetstein, S. C., de Jong, V. M., Stathonikos, N., Opdam, M., Dackus, G. M., Pluim, J. P., van Diest, P. J. and Veta, M. (2022), 'Deep learning-based breast cancer grading and survival analysis on whole-slide histopathology images', *Scientific Reports* **12**.

Wulczyn, E., Steiner, D. F., Xu, Z., Sadhwani, A., Wang, H., Flament-Auvigne, I., Mermel, C. H., Chen, P.-H. C., Liu, Y. and Stumpe, M. C. (2020), 'Deep learning-based survival prediction for multiple cancer types using histopathology images', *PloS one* **15**(6), e0233678.