Deep learning with sub-optimal ground truth: rectal cancer segmentation on whole slide images

Rebecca S. Stone
University of Leeds, School of Computing
MSc Advanced Computer Science
2016-2017
sc16rsmy@leeds.ac.uk

Plagiarism declaration

The candidate confirms that the work submitted is their own and the appropriate credit has been given where reference has been made to the work of others.

I understand that failure to attribute material which is obtained from another source may be considered as plagiarism.

Signed: _____

Project summary

A convolutional neural network is used to detect cancerous regions of rectal tissue on a dataset of whole slide images. The annotations available as ground truth are rough and sub-optimal, so the final evaluation takes this into account and explores the capability of deep learning to surmount data quality issues.

Acknowledgements

Firstly, a heartfelt thank you to all my supervisors; Dr. Andy Bulpitt for giving me the opportunity and encouragement to work in this research area, Prof. David Hogg for providing computer vision expertise and assuring me that my neural network *would* learn; Dr. Darren Treanor for diving into the project full-heartedly. All three of them have made themselves available and have been invested, supportive, and extremely helpful during this project. I look forward to navigating through a PhD with them the next four years.

Additional thanks go to Alex Wright, Leo Pauly, and Hanh Tran for generously sharing their knowledge in related domains and saving me valuable time; to Martin Callaghan at the High-Performance Computing Centre at Leeds for his patient, helpful responses to all my questions while swimming through the shark-infested sea of Docker and Singularity and containers; to Nick West for providing the Eindhoven dataset, annotations, and support; and Emily Clarke for taking the time to take me on a fascinating tour of the pathology labs at the Leeds Teaching Hospitals.

And, a thank you to Nathanael, my first and always supporter.

Deliverables

The deliverables in Table 1 have been submitted in accordance with the requirements of this program. All code deliverables are located in a private Bitbucket repository within the username **@ysbecca** under the project labeled *digital-pathology*. The Eindhoven dataset itself is not included. To request access to the repository, please contact Rebecca Stone. The repository Wiki is available to the public at `https://bitbucket.org/ysbecca/digital-pathology/wiki/Home` and includes a detailed description of the repository contents.

	Description	Submission method
1	Discussion of the problem background and the solutions attempted up until the present; an analysis of their strengths and weaknesses, and a look into the future challenges and opportunities for machine learning algorithms in digital pathology	Introduction and Conclusion chapters
2	The scripts which extract the experiment data from the university server	The */scripts* sub-directory of repository
3	The scripts which perform all preprocessing needed, such as colour normalisation, extraction, etc. as well as the division of data into training, validation, and test sets	The */scripts* sub-directory of repository
4	The trained convolutional neural network and code which uses the TensorFlow libraries	The .ipynb files within the */scripts* sub-directory of repository
5	Evaluation of results, including relevant diagrams and a comparative study using work by other researchers on similar problems	Introduction and Evaluation chapters
6	Dissertation	This complete text

Table 1: MSc deliverables as outlined in the planning phase.

Contents

Chapter 1

Introduction

1.1 Overview

This chapter aims to introduce the project and provide an overview to digital pathology, deep learning, and the usage of deep learning algorithms particularly convolutional neural networks (CNN) in the medical imaging domain. It will conclude with a discussion of the current challenges to deep learning in pathology tasks.

1.2 Introduction to the problem and methodology

1.2.1 Project description

This project involves performing tumour segmentation on a collection of whole slide images featuring post-treatment rectal cancer, using a pure CNN approach. Whole slide images are digitised tissue slides, scanned using a whole slide scanner at high magnification and resolution. Segmentation involves distinguishing between the normal and cancerous tissue on a slide. The network is trained on small patches sampled from images designated as the training set, and then evaluated on the rest of the images, the test set. A pure CNN approach has been selected for this project due to its outstanding results in performing tumour segmentation on breast and brain whole slide images, discussed more thoroughly in this chapter. While the dataset available for this work is sufficiently large, it has sub-optimal annotations which are only rough estimates of the tumour area and contain significant incorrect ground truth. Thus, the purpose of this project is not only to demonstrate the ability of a CNN to perform segmentation on a unique dataset of rectal cancer whole slide images, but also to gain additional insight into what the network can learn from sub-optimal ground truth.

1.2.2 Pre-project planning

In the scoping and planning phase before the start of the project, the approximately 14 weeks of the project span were divided into several phases, shown in Figure 1.1.

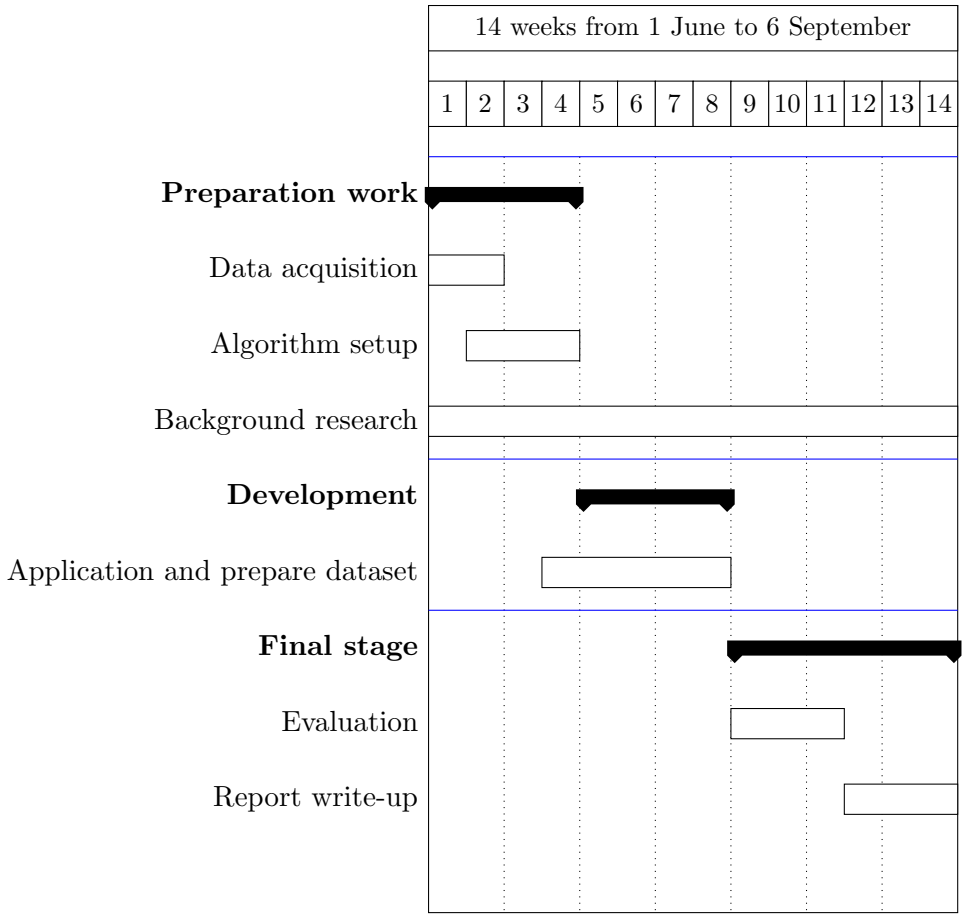


Figure 1.1: Gantt chart illustrating methodology.

While the major phases in bold were all completed according to schedule, several of the sub-tasks took longer or shorter than expected. In particular, the development period took longer, involving all the scripts needed to automatically read in the test images, load the annotations, sample patches and meta data and save them. The whole process is detailed in chapter 3. The developing, adjusting, and training of the CNN also took considerable time; each experiment took approximately 5 to 6 hours to run even with high-performance computing resources. Each step was time-consuming due to the sheer magnitude of the images and samples used. The report writing and background research continued progressively throughout the entirety of the project, so the final phase of the project required less time than anticipated. Chapter 7 includes a more detailed project reflection discussing the factors which slowed down certain tasks, and in retrospect, what could have been done to expedite or improve the work.

1.3 Digital pathology and colorectal cancer

1.3.1 Introduction to colorectal cancer

Colorectal cancer (also known as bowel cancer) is cancer which develops in either the colon or the rectum, when some of the epithelial cells lining the bowel mutate and grow abnormally and invasively. The World Cancer Report [34] identified colorectal cancer as the fourth most common cause of death worldwide, with more than 740,000 men and 610,000 women diagnosed in 2012. In

the same year, colorectal cancer constituted 10% of all global cancer cases. 65% of cases occur in more developed countries, which corresponds with research by the World Cancer Research Fund indicating increased risk with higher consumption of red and processed meats, alcoholic beverages, and presence of more abdominal fat [38].

In the colon and rectum, most tumours are adenocarcinomas, originating from gland cells in the epithelial tissue. A group of healthy gland cells are shown in Figure 1.2, with a single gland outlined. Mutated gland cells can adopt a large range of abnormal morphologies, making the carcinoma sometimes difficult to identify and grade even for expert pathologists.

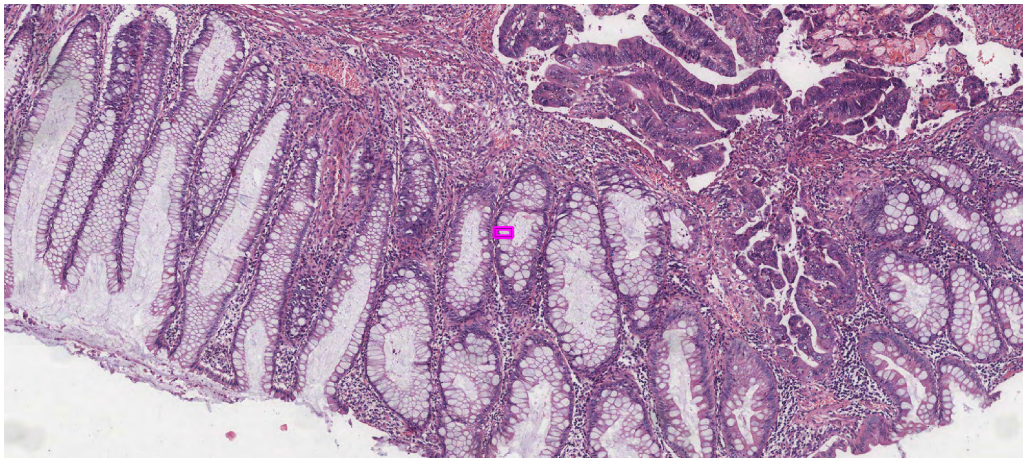


Figure 1.2: A portion of a whole slide image with a single normal gland cell in the epithelium tissue outlined in pink.

1.3.2 Brief history of digital pathology

Pathology is the study of disease through analysis of bodily fluid and tissue samples. A direct correlation exists between disease progression and tissue structure and changes in nuclei morphology, and specimen slides have long been used by pathologists and physicians for cancer analysis and diagnosis. However, the glass slides typically used for tissue and cell samples quickly accumulate, are easily damaged, and are difficult to store and transport. Furthermore, many crucial tasks performed using the slides are quantitative [14], involving accurate counting or segmenting of features, which is both labour-intensive and vulnerable to subjectivity and human error. Faulty diagnostic (disease presence) prediction is not only dangerous for patients with serious conditions [17]; treatments such as surgery, radiation, and chemotherapy can be fatal for patients with lower-grade, otherwise survivable cancers.

1.3.3 The typical clinical workflow

The Leeds Teaching Hospitals Pathology Services provides service to both the Leeds Teaching Hospitals and the local practices in West Yorkshire, and is representative of similar services worldwide. Tissue samples are typically taken from patients upon physician recommendation, and then pass through a series of stations at the pathology labs. Samples are cut and the parts of interest stored in plastic blocks, which are usually frozen or secured by adding a histological wax similar to paraffin. The prepared blocks are then placed into a vibrotome or similar slicing device which cuts the tissue into very thin slices. Each slice is positioned onto a glass slide, and undergoes a melting process which removes the wax. A fixed slide cover protects the front surface of the tissue.

Once the slides are prepared, they are brought to pathologists for examination. The pathologist reads the key points of the patient's record, views the slide under high magnification with a microscope, then makes a diagnosis which is then passed on to the physician. In labs moving towards digitization, the slides are digitised by passing through a whole slide scanner which processes several hundred slides at a time. Each whole slide image (WSI) is gigabytes in size.

Depending on the policies of the lab, glass slides from each patient are stored for a fixed number of years. They accumulate rapidly, and large, secured areas must be designated for their storage. At the Leeds Teaching Hospitals, the slides are stored on the basement floor, as their combined weight is too large for the building to support on any other floor.

1.3.4 The colour variation problem

A common problem plaguing digitised pathology slides is large variations in colour contrast and intensity [11]. Variations can come from the length of time the slide was held in the staining chemicals, the companies producing the stains, the image processing algorithm of the scanner, and the different scanner manufacturers. Many lab workflows include instruments which stain slides in bulk, reducing the variation in timing. However, studies have shown that even slides stained the same way on instruments from a single manufacturer contain large variations in colour [10].

While histochemical stains have long been used to help visualisation of tissue features, the human visual system is capable of adapting to large variation in colours, making it is difficult to quantify the effect of colour variation on pathologist's performance [3][6]. Converting the slides to whole slide images adds additional components to the problem. Both the scanners used to digitise slides and the displays and software used to view the images add colour variation. Significant research has examined ways to calibrate scanners, including different types of colour calibration slides similar to those used in digital cameras.

In the case of digital pathology, colour can also be adjusted after the scanning process. The Reinhard colour transfer is is a normalisation method published in 2001 [28] which uses statistics to transfer the colours of one image, the source image, to another, the target image. The images are first converted to the LAB colour space proposed by Ruderman et al. [30] which minimises correlation between channels, then each value in each channel separately is normalised to match the source image. This is performed by subtracting the mean from each colour channel from each value, then scaling the values based on the relative standard deviations of target and source images. The Reinhard method has been applied to histopathology image normalisation by Wang et al. [37], Magee et al. [22] and others with positive results. However, neither of these papers evaluate the normalisation success with respect to training a CNN.

Researchers have observed that Reinhard performs better on patches where one texture or colouring dominates, and poorly when there are multiple textures or colourings [22].This seems to indicate that Reinhard may not be the best normalisation algorithm for patches sampled from entire whole slide images, as there is considerable variation in texture even within a single patch.

In work by Cruz-Roa et al. [7] on breast cancer whole slide images, a simpler, domain-agnostic approach was adopted. Sample patches were converted from RGB into YUV colour space, and normalised to a variance of one and mean of zero. The goal was to simply emphasise differences between features and remove the correlations of the raw pixel values. While the overall results of applying a CNN to these pre-processed patches were good compared to using handcrafted features, the effect of the colour normalisation method itself was not evaluated.

Some researchers have questioned the importance altogether of colour variation to pathologists'clinical accuracy, as the human visual cortex has an astounding ability to adapt to colour variation. However, other literature by researchers at Leeds argues that colour plays a large role in

the diagnostic process and presents various calibration methods for whole slide scanners [6]. While not formally proven, it is widely accepted that the presence of colour, despite variations between slides, greatly assists pathologists in detecting tissue structures, nuclei, and other key features. For a CNN which has no domain knowledge and learns purely from training exemplars, however, it is not definitively known how much colour in whole slide images benefits or hinders learning.

Other works have combatted the variation with a different approach. Instead of trying to reduce variation, some researchers artificially generate additional colour variation so that the deep learning algorithms learn every possible variant [32]. In cases where there are larger datasets available, variation can be naturally obtained by taking large numbers of samples for training. While several colour normalisation methods are tested, the large sampling approach is used by default in this work.

1.3.5 Digital pathology today

In the past decade, the introduction of digital scanners has opened up a world of possibilities. Digital whole slide imaging, or the digitisation of traditional tissue slides, allows health practitioners to easily share images for second opinions, as well as provides researchers and medical residents with a wealth of new resources [21]. The availability of digital slides has also paved the way for the development of computer vision, machine learning, and other image analysis algorithms for diagnostic, prognostic (outcome), and theragnostic (choice of therapy) predictions [17]. Solutions have arisen for difficult tasks such as tissue and nuclei classification, segmentation and detection [14], yet have not generally been adopted for clinical use.

The storage of whole slide images, while much more compact than the storage of traditional glass slides, is an additional resource required for digitisation. The 355,966 slides stored on the Virtual Pathology Library server at Leeds use over 114 terabytes of memory, and as the lab increasingly commits to digitising more of their slides, storage is a necessary consideration.

1.4 Deep learning

1.4.1 A brief history

Deep learning is a branch of machine learning involving a layered architecture, each layer consisting of neurons which perform a non-linear transformation on inputs and produce an output. The origin of neural networks dates back to the single-neuron perceptron, proposed in 1943 by McCulloch and Pitts [24]. While an extremely simplistic modelling of a biological neuron in the brain, the perceptron was discovered to be capable of learning the separation boundary for linearly separable datasets in any dimension. Neurons can be connected and combined to form a multi-layer perceptron, theoretically proven to be trainable using error backpropagation. Researchers have proved that in fact, the trained multi-layer perceptron is capable of approximating the posteriori probability function of any input data [29]. Despite the proven strengths of multi-layer perceptrons, their success on actual datasets was elusive for decades. Researchers did not yet realise how much data was actually required for training. Large labeled datasets were not yet generally available. In addition, researchers initialised network weights incorrectly, introduced the wrong kind of non-linearity into the layers, and most importantly, did not have the compute resources to make training realistically feasible.

In the past decade, the development of supercomputers and GPUs and the rise of big data drastically changed the prospects for deep learning algorithms [33][27]. The weighted sums in each network layer are matrix operations which are well suited for GPUs. Researchers showed that not

only were deep networks capable of being trained and learning key tasks such as speech recognition, content filtering, board and video games, machine translation, and medical diagnosis, but many variations on deep networks also could be successful.

Recurrent neural networks (RNNs) are among the various successful deep learning algorithms at the forefront of research today. Deep RNNs achieved the best recorded accuracy in a standard speech recognition task with a combination of the layered learning from a deep network and the long-term memory achieved via reinforcement learning [9]. In addition, deep learning blended with traditional reinforcement learning has been shown to speed up and improve learning [25]. The groundbreaking success of AlphaGo used a novel combination of deep learning, reinforcement learning, and tree search [31]. Autoencoder neural networks are another deep learning algorithm which learn a function based on unlabelled data, thus making it an unsupervised learning algorithm which performs well in natural language tasks [19] .

1.4.2 Convolutional neural networks

History

Today, a class of deep neural networks called convolutional neural networks are one of the most widely applied types of deep networks. In 1968, a paper by Hubel and Wiesel showed a direct correlation between neurons in the visual cortexes of cats and monkeys and regions of the visual field [13]. The animals seeing certain distinct simple patterns always triggered certain neurons in their brains to fire. Based on this research, it is now known that the neurons which fire for a region of the visual field form a receptive field in the cortex, and cumulatively, all the receptive fields cover the entire cortex. Just as the multi-layer perceptron was initially developed modelling connected neurons in the brain, convolutional neural networks model the vision system. Unlike the fully-connected architecture of a deep neural network where each input value is related to every other input value in the same way, CNNs incorporate the concept of spatial structure, or receptive fields. Input pixels close to each other effectively form a receptive field, via shared weights and pooling. CNN architecture is discussed in more detail in chapter 4.

LeCun et al. in 1998 successfully used a CNN dubbed LeNet-5 to perform highly accurate digit recognition on the MNIST dataset of digits [16], examples of which are shown in Figure 1.3. Despite breakthrough results, the approach was still severely limited to small (32x32 pixel), low-resolution images due to the number of weights increasing proportionally to the number of layers. The LeNet-5 model expanded for larger or higher resolution images was simply not feasible using the compute resources available at the time.

Figure 1.3: Some size-normalised digits of the MNIST dataset from the work by LeCun et al. [16] proving that a CNN can successfully perform classification on images.

Early applications of CNNs to medical imaging mostly involved grayscale, lower-resolution images such as photomicrographs of the human corneal endothelium [42] or digital mammograms [41]. While successful, these applications were few until researchers discovered that GPUs could be

used for massive speedup of machine learning algorithms, demonstrating a more than 3x speedup on a fully-connected 2-layer neural network [33]. This research was quickly applied to CNNs as well, and opened the door for many applications on larger, complex images.

In the past decade, convolutional neural networks have been applied to computer vision problems such as object detection and classification (face recognition, etc.) [20]. Face recognition is a standard, challenging problem in computer vision as the images of interest are complex, multi-dimensional, and inconsistent. Before convolutional neural networks, many researchers investigated feature-based recognition algorithms which measured ratios and distances between facial features, or used template or graph matching [15]. The advent of neural networks saw the development of many valid solutions to computer vision tasks.

Design

CNNs use layers of neurons in a network and multiple processing stages to learn complex features from given data. These algorithms are well-suited to multi-faceted problems such as those presented by digital pathology because their success does not depend on prior domain knowledge or any real understanding of the problem. Rather, convolutional neural nets learn features directly from the training images themselves.

Training a CNN on images typically involves multiple stages. First, samples are generally taken from the images. The number and size of samples may vary depending on the task and image complexity. Each subsequent layer of the network then performs one of several operations. Convolution layers apply a filtering transformation to small windows of the image; ReLU (rectified linear unit) layers normalise all values; pooling layers typically reduce the dimensionality of the images while preserving the most critical features; and the final layer is fully-connected to produce the output. At different stages a certain percentage of units may be dropped out to prevent overfitting (dropout operations). Ordering and number of layers can also vary depending on the task. As with normal deep neural networks, the network weights are updated by gradient descent and back-propagation.

A strength of convolutional networks for images is that they learn features in hierarchical order, similar to the way an actual image can be examined on multiple levels. Earlier layers in the network pick up larger, general features, whereas subsequent deeper layers learn finer features. This approach allows the network to learn characteristic features from even complex images.

Despite their widespread success in many domains, neural networks for image analysis are plagued by two main disadvantages. The first is that a considerable amount of training data is required in order to train and test the networks. The algorithms "learn" based on what they have already seen, so more data generally can improve results. Only requiring data to learn can be considered an advantage; but when data is not be readily available, the algorithms do not perform well. This common problem is often addressed by generating additional data by performing small random transformations and distortions on the existing data. Secondly, the algorithms are computationally expensive especially when the data is high-resolution images. High-performance computing resources are usually necessary.

1.5 A general review of deep learning and pathology

The digitised images in histopathology are information-rich, dense, and extremely high-resolution. Many prior image analysis algorithms could not handle the scale of data involved in pathology, but modern deep learning algorithms have enormous potential [20]. Deep networks have been able to outperform traditional methods for several key problems in digital pathology, including mitosis

detection, epithelium-stroma classification, and various types of tumour segmentation. In the past decade, these tasks have been typically first tackled by gathering or generating as much training data as possible, and training a neural network. These brute-force methods outperform all the traditional methods. However, researchers realise that the data-driven approach is expensive both in adequate data and resources. In the past few years, solutions have evolved which employ a combination of a CNN and other techniques, with comparable results. Research is increasingly driven in the direction of finding a state-of-the-art solution which is feasible for clinical application.

1.5.1 Mitosis detection

Mitosis is the reproductive phase of a cell life cycle, which has a strong correlation to cancer invasiveness. Mitotic count is key in cancer grading, and is a complex task due to the variation in size and morphology of different cell types. Cirecsan et al. [5] use deep neural networks for mitosis detection. Expert annotations on whole slide images are taken as ground truth, and a deep neural network is trained to perform pixel classification where a pixel is either mitosis (when a fixed number of pixels away from the centre of mitosis) or non-mitosis. Additional training data is generated by adding rotations and transformations to the existing data. The problem is easily evaluated as a classification problem, and the best F-measure (F1) score (see chapter 5 for a detailed discussion of accuracy metrics) achieved is 0.782.

Several works have successfully produced comparable results on the same task, while reducing the computational demand of methods using deep learning. Malon et al. in [23] combine hand-crafted features based on nuclear morphology and the learned features of a CNN to attain an F-measure of 0.659. A year later in 2014, Wang et al. in [36] used the same approach, but with a more intelligent combination, attaining an F-measure of 0.734.

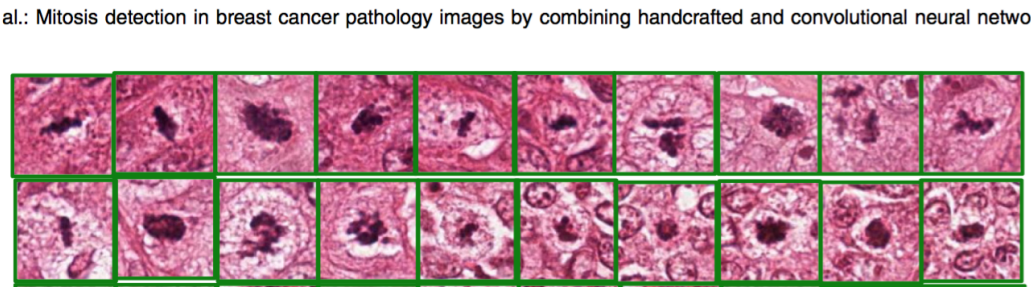


Figure 1.4: Examples of mitosis correctly identified in [36] using a combination of hand-crafted features and a CNN.

More recently in 2016, Chen et al. [4] reached an F-measure score of 0.788 on the same dataset, improving the learning process by cascading two convolutional neural networks together. The first network is a coarse retrieval model, which efficiently retrieves mitosis candidates from the images. The second is a finer discriminant model, distinguishing from among the candidates the true mitosis from the non-mitosis which appears similar. In addition to giving better results, having two networks, each of which performs a simpler task, improves the overall detection efficiency. These deep learning approaches far outperform traditional hand-crafted feature methods.

1.5.2 Epithelium-stroma classification

In breast and rectal cancer, another key feature for aiding diagnosis and prognosis is the epithelium-stroma ratio within the tumour tissue. Manual labelling is tedious and time-consuming, making computerised epithelium-stroma classification methods highly desirable. Hand-crafted feature methods can segment with reasonable accuracy, and are feasible to implement as stroma and epithelium tissue can be easily distinguished by texture and colour differences [2]. However, deep learning approaches far outperform those methods. A patch-based approach is presented by Xu et al. [39], with a deep convolutional neural network (see Figure 1.5) classifying small patches with a single label based on tissue type. The same problem can also be modeled as a segmentation task, by first identifying regions of interest, and then labelling them. The approach yielded a perfect result (F-score of 1.0 on test set) for both a breast cancer dataset and hemotoxylin and eosin stained colorectal cancer dataset.

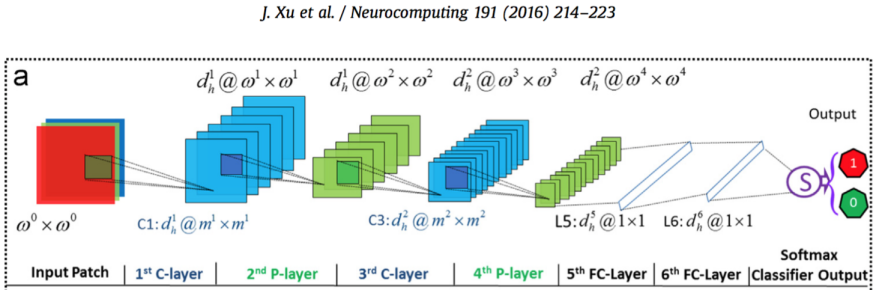


Figure 1.5: The CNN used by Xu et al. [39] in their patch-based approach for epithelium-stroma segmentation.

Huang et al. [12] take the success of deep learning in this task one step further, aiming to reduce the resources necessary for training convolutional networks. Their method employs transfer learning to eliminate the need for large well-annotated datasets. A CNN pre-trained on the public ImageNet dataset is used to extract features from the training set, and the learned distribution of the training set is then transferred to a target domain where the images are completely unlabelled. The final classification is comparable or better than the brute-force CNN methods and is argued to be much more applicable to clinical usage as well-labeled ground truth is required only for the source domain.

1.5.3 Tumour segmentation

A fundamental task of practicing pathologists is tumour detection. When a tumour is present in a tissue sample, it is closely examined to determine the invasiveness and extent of carcinoma. The largest tumour diameter is one of the measurements often used for cancer grading. Tumour detection and segmentation tasks can range from easy from a pathologist's perspective, to very difficult. This is because a single tumour type can have many variations in appearance as the cells evolve and reproduce. Within the tumour itself, there are different types of tissue. While some segmentations can be performed by a pathologist at relatively low magnification, others require a higher level of magnification.

The Medical Image Computing and Computer Assisted Interventions hosted a tumour segmentation on whole slide images as their yearly challenge in 2014. The challenge provided whole slide images from brain cancer patients. Researchers in [40] sampled patches from the dataset and used a CNN pre-trained on the ImageNet public dataset, and discriminative feature vectors retrieved

from the output. A support vector machine (SVM) was then used to distinguish between positive and negative samples. On the test set, this resulted in an accuracy of 0.84.

Several researchers have employed similar deep learning techniques to segment breast tumours. In [7], researchers compare the results of CNN learning with the traditional hand-crafted features methods. The best features method returns 0.78 accuracy, compared to the improved 0.84 achieved by the CNN. The CNN approach samples patches from the whole slide images, excluding borders and background slide, then trains a 3-layer CNN to perform binary classification on the training set. The test images are then classified by being subdivided into patches of the same size as the training samples and passed through the network. Heatmaps were used for visualising the results, as seen in Figure 1.6.

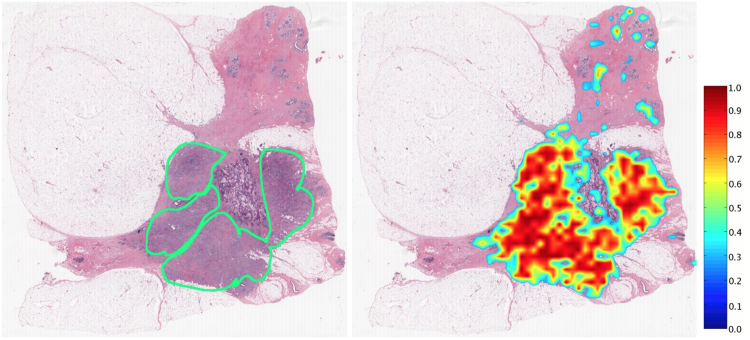


Figure 1.6: The resulting probability heatmap showing probabilities learned by the CNN on a breast cancer whole slide image in [7].

The International Symposium on Biomedical Imaging also held a challenge (Camelyon) in 2016 featuring breast cancer identification on whole slide images. Instead of combining different features or trying to improve computational efficiency, researchers for the winning approach [35] used a pure CNN approach, using four well-known deep learning algorithms for training: GoogLeNet, AlexNet, VGG16, and FaceNet. GoogLeNet performed the best with a patch classification accuracy of 0.984. They also thoroughly experimented with different zooms, resolutions, input patch sizes, and re-training the CNN with additional patches from difficult negative patches which closely resembled true positives. While arguably too computationally expensive for present clinical use, this brute-force method clearly highlights the ability of a CNN with adequate training data and ground truth to give outstanding results.

1.5.4 Conclusion

While deep neural networks have given promising results in many medical imaging tasks, including grading cells of cancerous liver tissue [18], identifying tissue components in prostrate slides [8], classifying eye disease from retinal images [1] and more, they have not yet proved generally feasible in clinical environments due to requiring powerful computing resources and massive amounts of training data [21] [4], as well as various other valid concerns raised by both computer scientists and clinicians. The whole slide images themselves, often between 20x to 40x magnification, are significantly larger than traditional images analysed in computer vision and thus contribute to the resource requirements. Thus, deep learning with whole slide images remains an open, challenging area full of potential.

The other weakness of deep learning algorithms for whole slide image analysis is the heavy reliance on good quality training data. Due to the magnification and size of whole slide images,

obtaining a large collection of annotations up to a high enough standard for machine learning (minimal false annotations even at high magnification) can be difficult. Producing the annotations is extremely manual and work-intensive for a pathologist. In the past few years, researchers have begun exploring variations of deep learning algorithms including combinations with genetic algorithms [26], self-organising neural networks [1], adding handcrafted features to speed up the training process [36], and transfer learning across domains [12]. These approaches attempt to address the issue of deep learning's data-heavy approach.

Chapter 2

The Eindhoven dataset

2.1 The whole slide images

2.1.1 Overview

The Eindhoven dataset, named for the city where the images were gathered in 2014, contains 202 separate cases each consisting of multiple whole slide images for a total of 3047 images. The clinical scenarios for each case are various regimens of post-treatment rectal cancer. Each slide is hemotoxylin and eosin (HE) stained. Each whole slide image is gigabytes in size (tens of thousands of pixels in width and height) as the slides were scanned by a whole slide scanner at 20x magnification.

Out of these images, a subset of 295 were manually delineated by pathologists using the image viewing software ImageScope by Aperio. The annotations are intended to assess the effectiveness of radiotherapy on the tumours, so include not only the remaining viable carcinoma, but also the associated scar tissue, fibrosis. As a result, many of the image annotations include large amounts of non-carcinoma, the area where the tumour has regressed. Regression often appears similar to normal tissue. This is an additional challenge for the neural network, which is discussed in detail in chapter 6. A pre-treatment dataset without regression, or new annotations of the complete Eindhoven set including only carcinoma would have been ideal, but due to time restraints, this work uses this pre-existing dataset. Sub-optimal datasets are common, and working with such data provides opportunities to explore the capabilities of deep learning. It is anticipated, however, that further work in the domain will be performed on more suitable datasets.

Some example images from the Eindhoven dataset (detailed discussion of data quality issues such as pen markings in bottom right image in subsequent section) are shown in Figure 2.1.

The images in this dataset were only available on a University of Leeds research server via a simple server API which allows access to sub-portions from each image of maximum 2000 pixels width and height at a time. Customisable parameters include the X and Y position of the block to be downloaded and its quality, zoom, height and width.

Direct access to the image files had been anticipated, however this was not possible during the project timeline, so several methods were taken to reduce the impact of latency related to API-only access. This significantly complicated the patch sampling algorithms discussed in chapter 3.

2.1.2 Examples

The more success the radiotherapy treatment had in a certain case, the more regression present within the annotation of the corresponding whole slide image. Conversely, an unsuccessful radio-

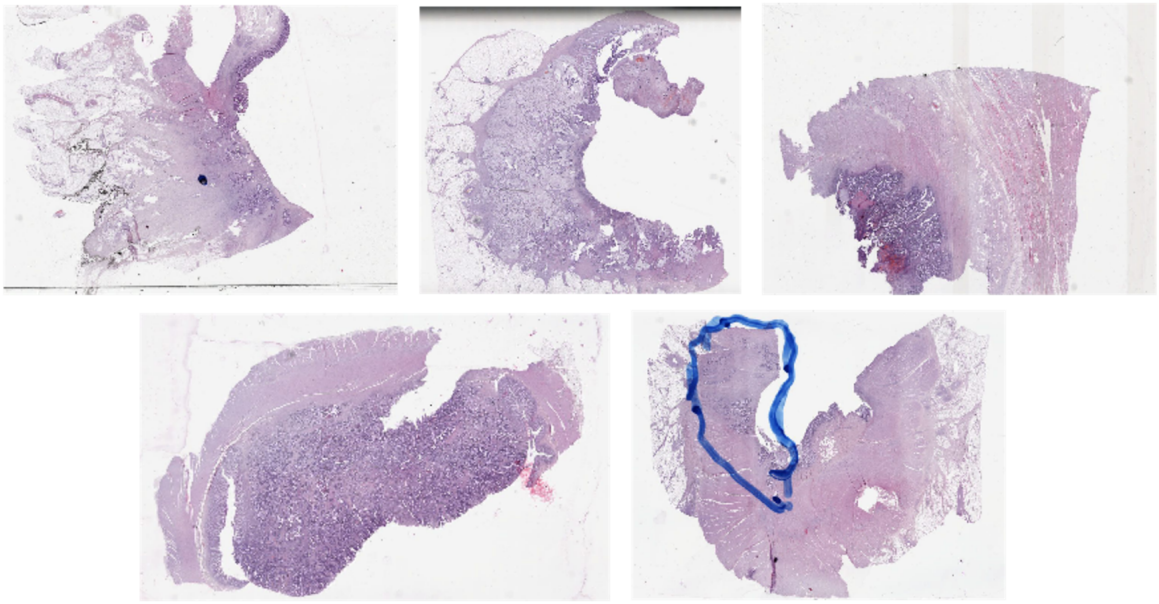


Figure 2.1: Five example whole slide images from the Eindhoven dataset.

therapy treatment implies that the annotation quite accurately encircles only the carcinoma on the slide. It is difficult to quantify the annotation quality of the whole dataset, but the majority of the images include significant amounts of regression. It is thus expected that learning to detect carcinoma will be difficult. From the perspective of deep learning, the best annotations will provide correct ground truth. Figure 2.2, 2.3, and 2.4 are examples of excellent, partial, and mostly inaccurate annotations.

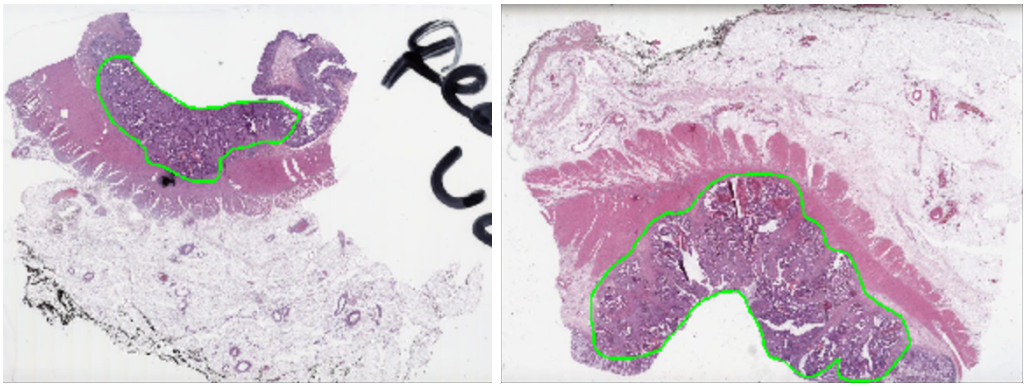


Figure 2.2: These annotations encircle the cancerous regions only.

2.2 XML annotations

Pathologists drew a polygon encircling the original tumour area on each whole slide image. The annotations are saved in XML format with each annotated region represented as a list of XY vertices, the pixel locations of the vertices in the image.

In order to establish ground truth for both training, validation, and testing, the XML files were converted to binary image masks as seen in Figure 2.5 with the same dimensions as the original

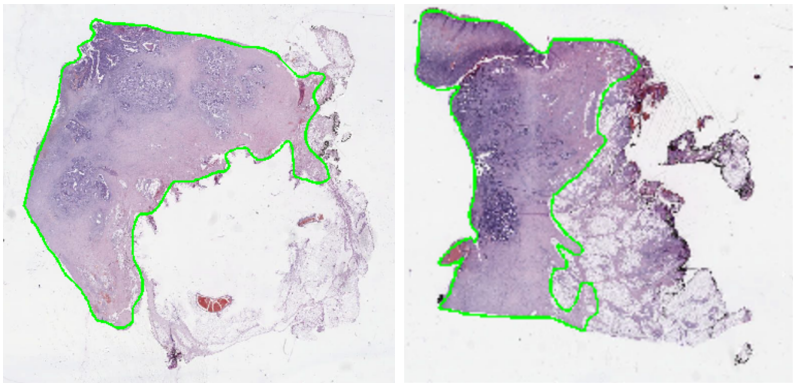


Figure 2.3: These annotations contain a mix of both carcinoma and regression.

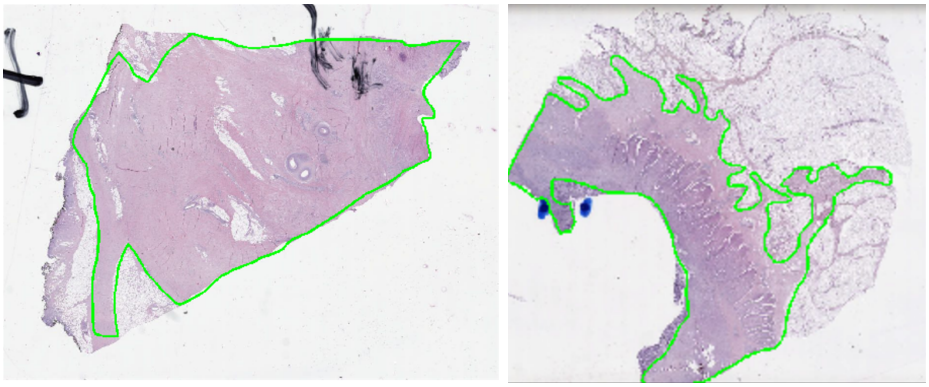


Figure 2.4: Neither of these annotations contain significant carcinoma; both are mostly fibrosis/regression.

image. They could also be imposed as annotations overlaid on the original images as in Figure 2.6. A simple, customised visualisation library was developed using the Python matplotlib module.

2.3 Data quality issues

A close examination of the dataset revealed five issues in the images themselves and the annotations. This examination is from the perspective of patch sampling for deep learning.

1. Firstly, it is clear from the accuracy level of the annotations (Figure 2.7) that the markings were not made while viewing at the highest level of magnification. This would have been incredibly labour-intensive for the pathologists, so instead they delineated from a zoomed out perspective, probably around a 2x or 4x magnification.

2. The annotations often enclose lumen, or large areas of background slide which should be labeled as normal not cancerous. Figure 2.8 shows one such case. While small white areas within cancerous regions may be legitimate characteristics of adenocarcinomas due to distortion of the gland cells, large areas are usually abnormal, due to large enclosed lumen or tears or cuts in the tissue. For the purpose of general dataset analysis, background areas of width or height larger than 200 pixels at full magnification are not considered features to learn.

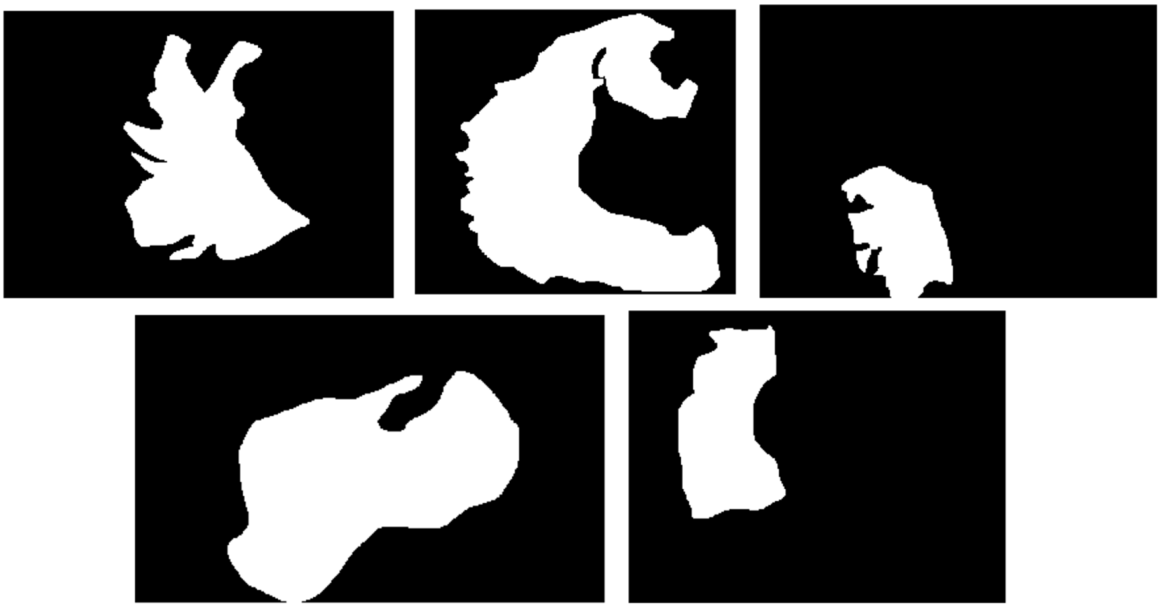


Figure 2.5: The binary masks of the example images in Figure 2.1, generated from XML vertices.

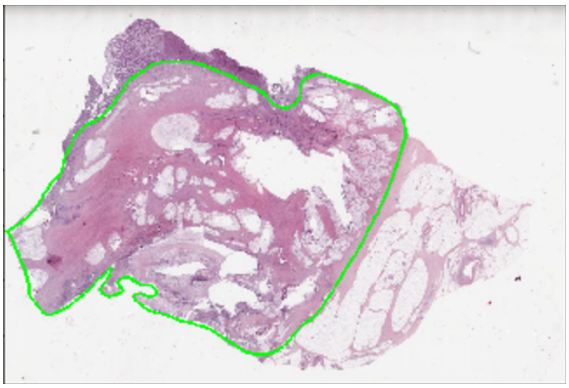


Figure 2.8: Issue 2: portion of tissue within a cancerous region, showing blank slide also enclosed and thus incorrectly labeled.

3. Occasionally a single slide includes two samples, neighbouring slices of a block placed on the same slide, which are nearly identical due to their proximity. It has not been confirmed, but it appears that pathologists were only instructed to annotate one cancerous portion even in these cases. In the case of multiple tissue samples, this means that there are large cancerous portions which are not included in the annotated region (see Figure 2.9). A large amount of cancerous tissue is thus incorrectly labeled as normal.

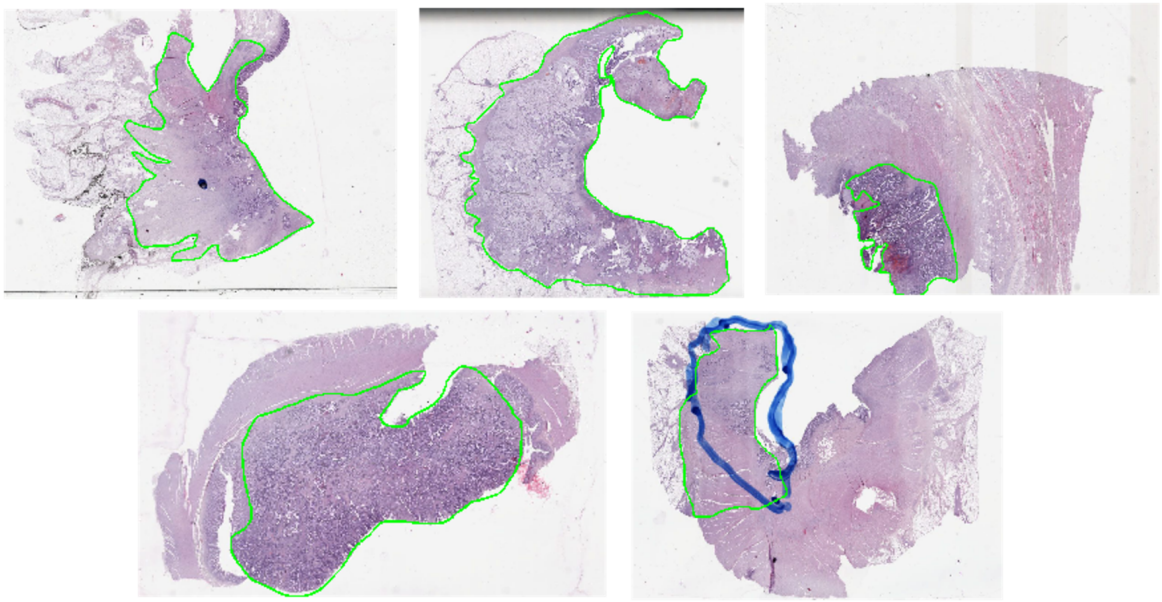


Figure 2.6: The annotations for the images in Figure 2.1.

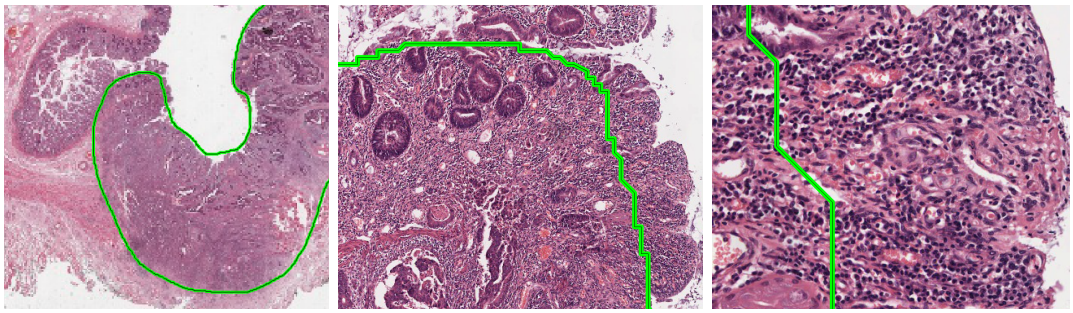


Figure 2.7: Issue 1: from left to right, decreasing accuracy corresponding with magnification (portion to the right of the annotation should also be included).

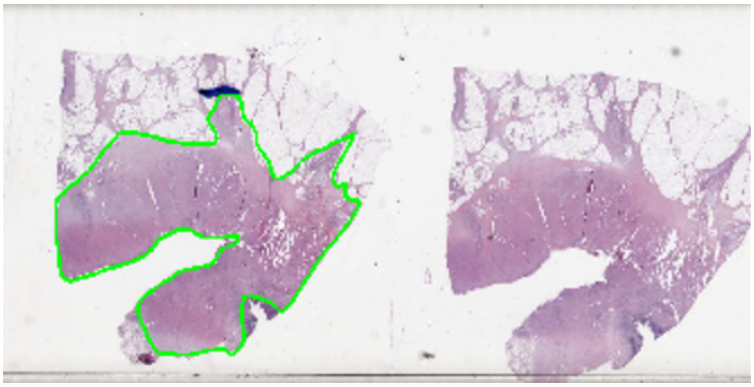


Figure 2.9: Issue 3: binary mask (right) of a single slide with multiple samples (left), both of which contain cancer, but only one of which is annotated.

4. Extra markings are present in a minority of the slides, which appear to be the edge of the slide cover or markings from a dark coloured pen on the slide cover itself before scanning.

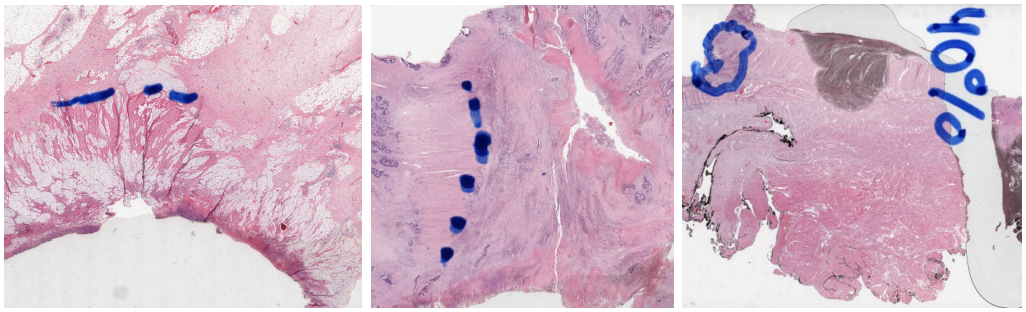


Figure 2.10: Issue 4: markings directly made on slide covers.

5. Some slides also contain more than one main cancerous region (Figure 2.11), and in an attempt to delineate both, the pathologist has connected them. The small region in the connecting portion of the image is likely normal, but according to the annotation would be labeled incorrectly as cancerous. Fortunately, the connecting regions are usually narrow, meaning that few patch samples will be sampled within them.

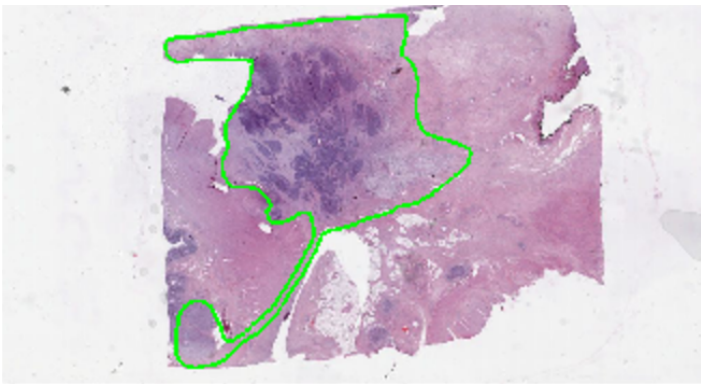


Figure 2.11: Issue 5: two connected regions.

Various approaches to reduce the impact of the four latter observations are discussed in the following section. These issues do not include the typical artefacts of the tissue samples themselves, including staining variations, folded-over tissue, and torn tissue. Table 2.1 shows the issues enumerated.

	Large enclosed background or lumen (greater than 200 pixels width or height)	Multiple samples/one annotation	Extra markings on slide cover	Connected portions
Cases	22	32	73	7

Table 2.1: Quantification of observations in whole slide images, out of 295 total images.

Chapter 3

Patch sampling

3.1 Patch sampling for training

In order to train the algorithm to learn characteristic features of cancerous and normal tissue, labeled training samples are needed. Based on the tested approaches of similar researchers with ductal carcinoma [7], epithelial and stromal segmentation [39] and mitosis detection [36], square patches of 100x100 pixels are used as training samples. In native resolution, the patches are 500x500 pixels, 0.5 microns per pixel. As the size of a nucleus is 10 microns, patches of this size display 250x250 microns, which insures that enough cell features are present in each sample. While ideally the ideal magnification and patch sample size would be systematically determined, due to time restraints the best parameters used by other researchers are adopted for this work. In addition, two practicing pathologists have roughly estimated the zoom and patch size necessary for a human to correctly label a sample as cancerous or normal, without seeing its larger context.

3.2 Patch sampling methods

Selecting patches from extremely large images and labelling them using a ground truth binary mask is not a trivial task. Several methods were developed and tested, evaluated in terms of efficiency and effectiveness.

3.2.1 Random sampling method

Researchers in [35] use a random sampling method to obtain millions of positive and negative samples. The first patch sampling method attempted for this project models that method. The algorithm chooses random locations within the image, checks with the mask to label the patch either inside or outside the mask, and continues iteratively until the desired number of patches with each label have been found. An attempt is made to select an equal number of each sample type from each image; however, due to the cancerous portions in some images being quite small, the algorithm allows a fixed number of attempts before moving on to another image. This method gives a relatively uniform patch sampling, but was in practice only efficient when sampling a small percentage of the total patches available in an image.

3.2.2 Binary search method

The second sampling algorithm more intelligently performs a binary search on the image by recursively dividing the binary mask into blocks which are identified as either purely inside or

outside the mask. Patches are sampled randomly from within those blocks, in the real image. Some experimentation was done to determine, for the image sizes and average complexity of the masks, how many blocks to find before ending the binary search early. While in practice more computationally expensive than the random approach, the binary search method has the added advantage of being able to choose patches near the boundary line of the mask as seen in Figure 3.1, by calculating block sizes and then selecting patches from the smaller blocks. It also can return a larger number of patches at little extra cost.

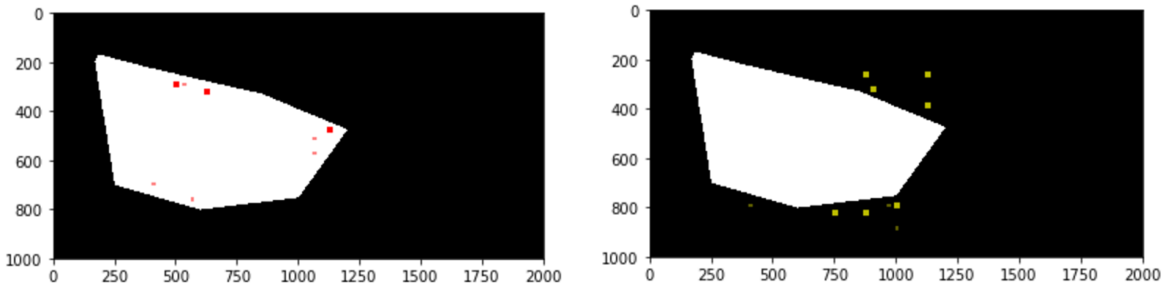


Figure 3.1: Example of selected patches from the binary search method on a test mask, choosing near boundary line.

While the binary search method proved more capable of choosing intelligent samples, it was computationally quite expensive due to the image sizes, and as a result, the search tree had to be pruned by early stopping. In practice, the random search provided a more uniform sampling distribution across the images than the binary search.

3.2.3 Grid sampling method

The final sampling method simply divides the entire whole slide image into a grid of non-overlapping patches as seen in Figure 3.2, and takes patches from every possible position. As each image can only be accessed in portions of maximum 2000 pixels across via the API, the entire image is scanned in rectangles 2000 pixels across and 100 pixels down, each rectangle subdivided into square patches, and each patch either accepted or rejected by thresholding functions.

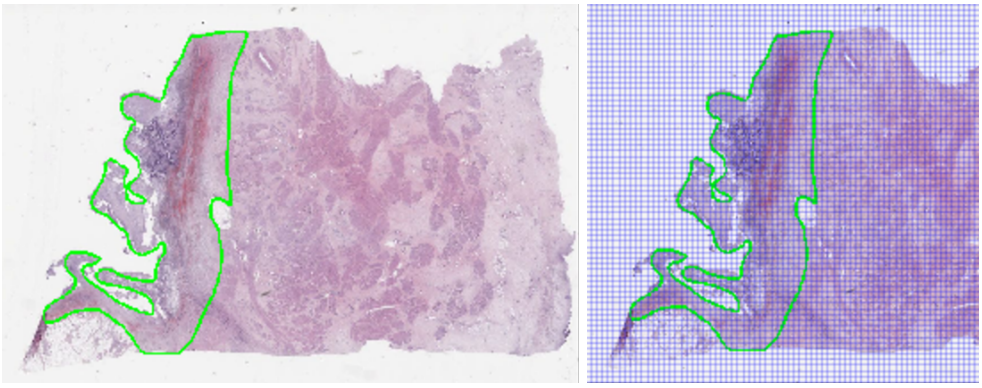


Figure 3.2: The original image with annotation and the sampling grid overlaid.

Thresholding and filtering functions are used to avoid taking patches from areas of little interest, such as ones featuring background slide and clearly non-cancerous areas such as those with low standard deviation across all colour channels (indicating very pale or sparse tissue which are not

likely to be cancerous). Figure 3.3 shows the modified sampling grid after the threshold functions have been applied.

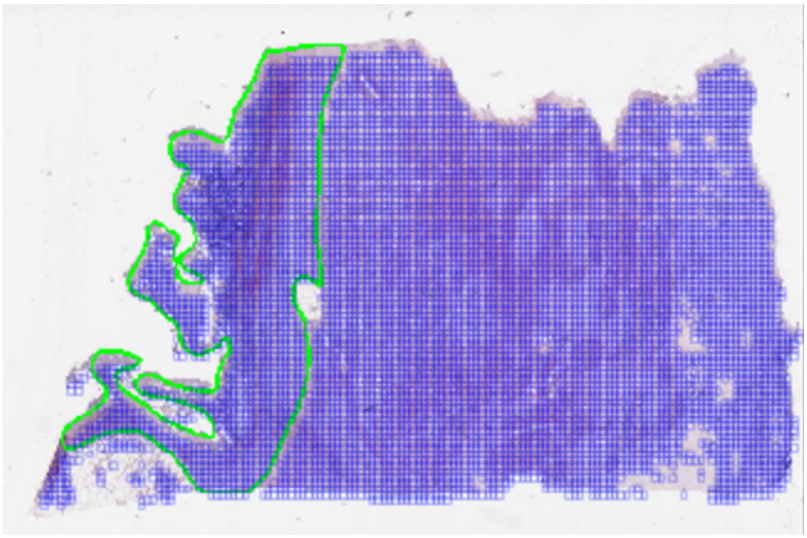


Figure 3.3: Image with the modified sampling grid after threshold functions eliminate background and patches with little variation.

It was initially thought that fewer sample patches would be required, but test runs of the CNN overfit the training set when the set size was too small (around 10,000 patches). Eventually, the grid sampling method was chosen for this work, due to the large number of patch samples needed for training. Other works, including [7] use a similar sampling method, but only sample a subset of grid positions. Xu et al. in [39] use a similar method with a step size of 40 pixels between every patch. In an attempt to provide more data for training and include as much colour variation as possible, no stride was used in this work and patches were sampled from every non-overlapping grid position not filtered out by the thresholding functions. Figure 3.4 shows patches in their original sampling order using the grid sampling method. In total, this method resulted in 1,264,499 patches.

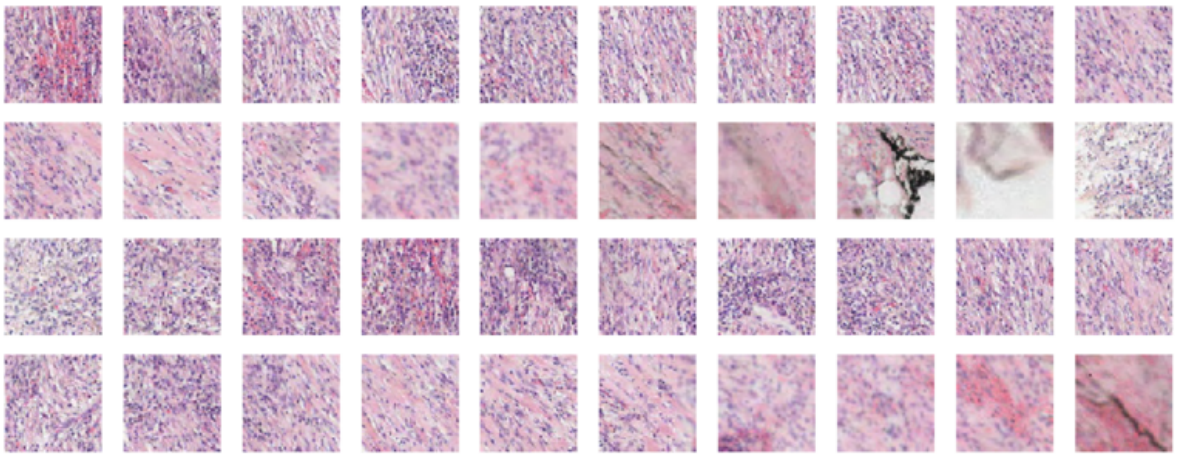


Figure 3.4: Example patches, in original ordering from left to right, sampled from a whole slide image.

3.3 Further preparation

3.3.1 Reducing the impact of issues with images and annotations

To reduce the impact of the first and fourth issues, that of general imprecision of the annotations and connected portions, patches are sampled only from areas which are entirely within or outside of the mask. This is different from the approach of Cruz-Roa et. al. [7] for breast cancer segmentation, where patches are labeled if they are 80% within or without the region of interest. This approach only takes "pure" patches, to try to reduce the noise of the training samples. In retrospect, this approach has little impact, as the tissue surrounding the carcinoma is often fibrosis or regression, so is already incorrectly labeled.

The second issue noted in the previous section is that of background slide enclosed in the cancerous region. Since all the background slide portions of each whole slide image are not cancerous, these can be ignored and not even used for training. To avoid sampling these areas, a simple threshold function is used which calculates the percentage of each patch which has RGB values greater than 210, and decides if the patch contains enough meaningful content. Whole slide scanners are generally calibrated to set the background slide values to 250, so the threshold used in this work generously allows for some colour distortion and shadows, for example along the slide cover edge.

There is no way to easily resolve the third issue automatically, where two tissue samples are mounted on the same slide but only one is annotated. The incorrectly labeled patch samples would definitely reduce accuracy if used for training. The 32 images with this issue (Figure 2.1) need to be discarded and not used for training or testing.

The fourth issue, that of pen markings directly on the slide covers, can be partially alleviated by colour normalisation and pre-processing techniques, but only when the patch is entirely covered by the marking as in some patches in Figure 3.5. As this is usually not the case, some noise is expected to be introduced into training due to this issue.

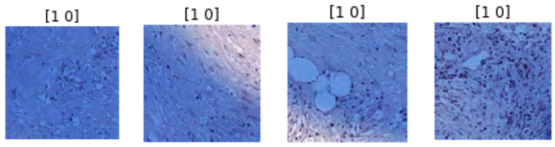


Figure 3.5: Patches sampled from marked regions, showing that colour is distorted but features are generally still detectable beneath the marking.

3.3.2 Dimension reduction

To improve the efficiency of the sampling methods, dimension reduction is performed on the binary mask by a fixed stride, which allows for faster checking when labelling sample patches. Coordinates from the reduced dimension are then converted back to the original dimension when fetching the patches from the actual image. A reduction by a factor of 20 was found to be large enough to greatly increase efficiency and yet preserve the details of the annotation. Larger strides such as those greater than 100 caused a large loss in precision, as illustrated in Figure 3.6.

3.3.3 Colour normalisation: three approaches

As discussed in detail in chapter 1, colour normalisation and its impact on deep learning algorithms is an ongoing research area in digital pathology. This project has attempted three of the solutions adopted by different researchers: the original image colours (Figure 3.7), Reinhard colour

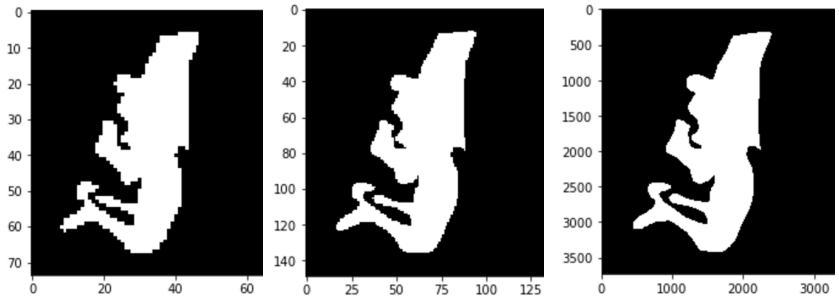


Figure 3.6: Illustration on a real mask showing the tradeoff between efficiency (larger stride of 300 on left) and accuracy (smaller stride of 10 on right).

normalisation (Figure 3.8), and conversion to grayscale. The openCV Python library was used to replicate this algorithm for normalising the whole slide images used in this research, with code snippets found online used as general guidelines (see Appendix A). By default, the experiments will be carried out using full, unaltered colour. By taking the maximum number of non-overlapping patch samples from the training images, this approach also maximises the CNN exposure to colour variation, as do methods which artificially generate more variation. It is expected that this approach will give the best results overall.

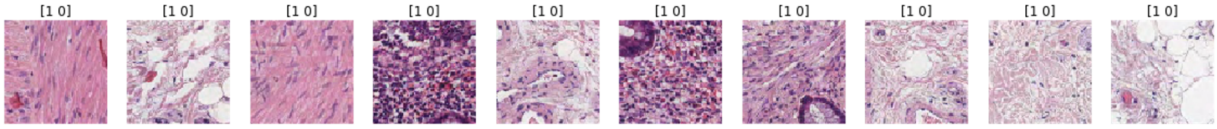


Figure 3.7: Ten sample patches before any colour normalisation.

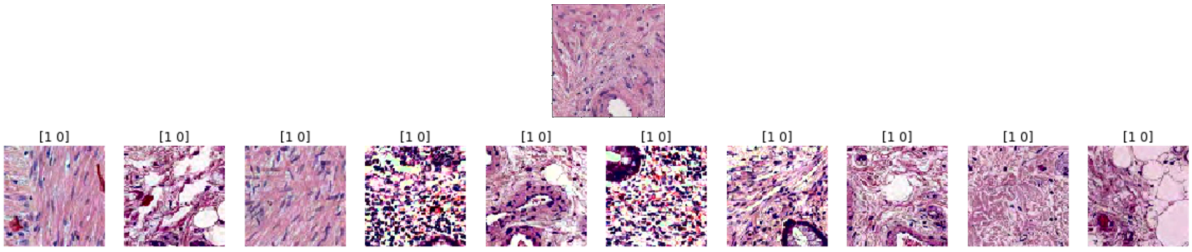


Figure 3.8: The ten sample patches after Reinhard colour transfer using pictured source image.

While this work will not involve a thorough comparison of all the colour normalisation methods and their impact on deep learning, the results of experiments involving these three approaches are discussed in chapter 6.

Chapter 4

Setting up the convolutional neural network

4.1 Overview

4.1.1 Choice of deep learning library

TensorFlow is an open-source machine learning library, currently very popular due to Google marketing and support. It provides a low-level API which gives developers full control. TensorFlow represents networks as graphs, and provides customisable visualisations. In addition, Google developers are actively contributing to and improving TensorFlow, which gives it excellent documentation and an advantage over Theano, a stable, more developed library also with a Python interface. Unlike Theano, TensorFlow also supports multi-GPU computing, a critical feature for efficiency with large-scale projects. TensorFlow also pre-compiles all its low-level operators and chains them together, providing a virtually instant compilation.

During the project planning phase, one consideration was to use Keras on top of TensorFlow due to its interoperability with TensorFlow, higher-level libraries and concise, minimalistic syntax. However, while the learning curve associated with TensorFlow was higher, various frameworks for basic convolutional neural networks already exist on the public domain, which helped speed up the learning process. The control and flexibility provided with TensorFlow eventually were more appealing than the friendly syntax offered by Keras.

4.1.2 Overall method description

Hvass-Labs (see Appendix A) provides a TensorFlow CNN which learns the MNIST digit dataset, classifying test images as digits between 0 and 9. Compared to the complex, dense patch samples from whole slide images, the MNIST digit dataset consists of low-resolution, grayscale, small images of relatively low complexity. The CNN was altered to produce a binary classifier which given a small image patch in 3-channel RGB colour, determines whether it is cancerous or normal. Once the CNN was trained with the training images from the training set, the learned filters were run convolutionally over each entire test image to perform the final segmentation. In the case of this project, this was equivalent to running every patch from the sampling grid of each test image through the trained network, and saving the results as a forced class and a probability.

4.2 The convolutional neural network

4.2.1 Components and basic structure

The basic structure of the CNN shown in Figure 4.1 was adopted from similar works using deep networks on whole slide images in [7], [39], and [36]. Each of the 10 sets of data, approximately 3GB of training and validation patches each, passes through a combination of the following network layers:

- Convolution layer: applies a set number of convolutional filters to the patch, returning an output feature map for every filter. Each filter is a small kernel window of specified size which is applied to every value in the patch. Each output value is the weighted sum of the kernel values and the corresponding input patch window.
- Max-pooling layer: downsamples the patch by taking the maximum value of every 2x2 window from the patch (with a stride of 2 so each window does not overlap) and discards all other values.
- ReLU function: replaces every negative pixel value with 0, introducing non-linearity to the transformation.
- Fully-connected or dense layer: performs classification on the features extracted from the previous layer by having every neuron connected to every neuron in the previous layer.
- Classification layer: a full-connected layer with only two neurons, each of which represents an output class (normal or cancerous). A softmax activation function returns a probability for each neuron output, the likelihood of the patch belonging to that class. The two softmax outputs sum to 1. For calculating accuracy, the maximum value is used to force a 0 or 1 prediction. For generating heatmaps and experimenting with threshold value, the softmax outputs are used.

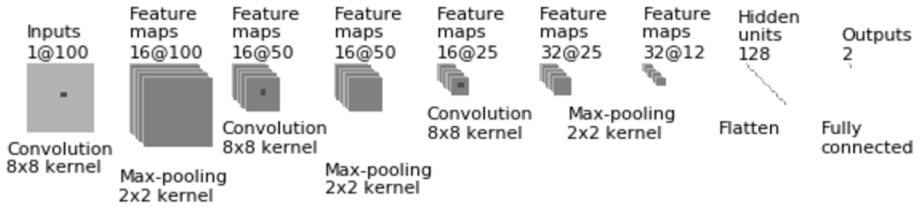


Figure 4.1: The basic convolutional neural network before experimentation with parameters.

The CNN structure developed as a base for experimentation has three convolutional layers, each with max-pooling and reLU, followed by a dense fully-connected layer and a final classification layer.

4.2.2 Saving the predictions and model

After the CNN has been trained on the training and validation sets, it must be run on the test images. As the test set is too large to be loaded all at once into memory, it is divided into two batches. Every non-background patch from each test image is passed through the CNN and a

simple function saves all the test patch predictions (softmax outputs) along with the corresponding meta data for each patch to a CSV file labeled after the CNN model. The learned CNN model is saved using the TensorFlow saver class. Since the structure of the network changes in each experiment, the complete model and graph must be saved. The following three files are generated for each saved model:

- .meta: stores the entire TensorFlow graph including all the network elements such as variables, layers, functions, etc.
- .data: contains all the variable values, namely all the learned weights and kernels.
- .index: stores metadata for all variables.

4.3 High-performance computing: ARC at Leeds

4.3.1 ARC3 and the container

Deep learning is known to be computationally expensive, and due to the size of the whole slide images, the number of training patches required, and the depth of the network, this project required high-performance computing (HPC) resources. The Advanced Research Computing centre (ARC) at the University of Leeds provides access to high-performance computing resources, including clusters with graphics processor units (GPUs). Containers instead of virtual machines are used in order to maximise space and efficiency, with Singularity on top of Docker taking care of container management. While Docker is a stable, widely used container management platform, it does not provide the security required for shared HPC clusters. Singularity, on the other hand, allows users to create containers from images and have root privileges within their container, while protecting the cluster.

The ARC3 cluster in ARC was chosen for this project due to its GPU availability. Built upon Linux CentOS 7 and possessing two Nvidia K80 nodes with 24 cores and 128GB of memory, ARC3 hard drives contain an additional 800GB per node and 350TB of Lustre storage. The University of Leeds is upgrading the ARC3 cluster to include 24 more GPUs in the fall of 2017, an additional motivation to become familiar with the system.

For training the CNN for this work, an image file for a Docker container with TensorFlow for GPUs was obtained from Docker Hub (see Appendix A). It was then altered to include OpenCV, the supporting graphics libraries for OpenCV, and updated for Python 3. The docker2singularity script provided by ARC staff then converted the Docker image file to a Singularity image file.

All code and important meta files were stored on a personal backed-up user directory with a 5GB quota on ARC3. The approximately 50GB of compressed image patches in .h5 format were stored in the no-backup high speed shared parallel file system. Once all HDF5 files were generated, they were copied to a personal University of Leeds OneDrive account for saving via Linux secure copy (scp).

4.3.2 Jupyter within the container and Git version control

One of the challenges with working within a container is visualisation; as there is no interface to the actual machine, users are limited to terminal-only access. To deal with this challenge, Jupyter notebooks were used extensively for each step of this project.

Jupyter is a powerful open-source web application for organised programming, interacting with live code, and in-line visualisation. Jupyter allows the creation of notebooks which can contain a

combination of code, displayed results, and stylised comments. While over 40 languages are supported by Jupyter, Python is its native language, and this project primarily used Python 3 kernels. Notebooks are hosted locally or on some other specified host. By running Jupyter notebooks inside the custom container created for this project with the –no-browser setting (Figure 4.2), and then using local ssh port forwarding (Figure 4.3), the notebook is able to be accessed via a local machine and all the results visualised from within the container.

```
[sc16rsmy@login1.arc3 ~]$ qrsh -l coproc_k80=1,h_rt=0:10:0 -pty y /bin/bash -i
[sc16rsmy@db12gpu1.arc3 ~]$ module load cuda
[sc16rsmy@db12gpu1.arc3 ~]$ module load singularity
[sc16rsmy@db12gpu1.arc3 ~]$ singularity shell --bind /nobackup --nv /nobackup/containers/deeplearn_1_gpu.img -c "export XDG_RUNTIME_DIR="
";/opt/conda/bin/jupyter notebook --notebook-dir=$HOME --ip='*' --port=8887 --no-browser"
Singularity: Invoking an interactive shell within container...

[W 13:13:01.744 NotebookApp] WARNING: The notebook server is listening on all IP addresses and not using encryption. This is not recommen
ded.
[I 13:13:01.752 NotebookApp] Serving notebooks from local directory: /home/ufaserv1_k/sc16rsmy
[I 13:13:01.752 NotebookApp] 0 active kernels
[I 13:13:01.752 NotebookApp] The Jupyter Notebook is running at: http://[all ip addresses on your system]:8887/?token=d512bc49b4e6efd58fc
b5c8bf95c85dd63ba084218af4e08
[I 13:13:01.752 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 13:13:01.790 NotebookApp]

    Copy/paste this URL into your browser when you connect for the first time,
    to login with a token:
        http://localhost:8887/?token=d512bc49b4e6efd58fcb5c8bf95c85dd63ba084218af4e08
```

Figure 4.2: Opening an interactive Jupyter session within the container.

```
ssh -L 11111:db12gpu1:8887 sc16rsmy@arc3.leeds.ac.uk
```

Figure 4.3: Local ssh port forwarding.

Typically, the development workflow for containers involves testing within a local instance of the container, then moving the container to the production environment. Container image files while not as large as virtual machines, are still substantial, so changing the container frequently uses valuable time. The local environment also has to match the production environment, which in this case was Linux CentOS. Rather than work within the container inside a virtual box on a local machine and export it to production, Git was used instead for simple version control. Development was done on a local machine, code changes committed to the master branch, then all changes pulled onto production from within the container. To avoid having to alter the container image during development, a list of Python modules was established early on in the project, and built into the container. As much as possible, code changes were made in the local environment and not in production. This allowed for a clean, efficient, controlled workflow (Figure 4.4).

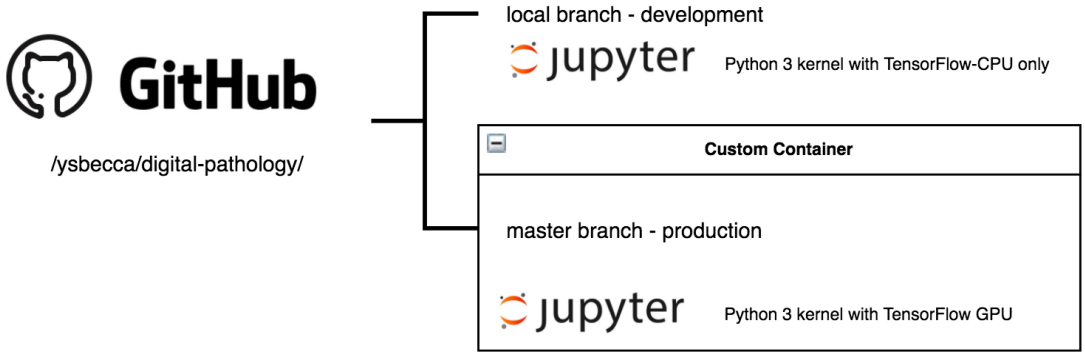
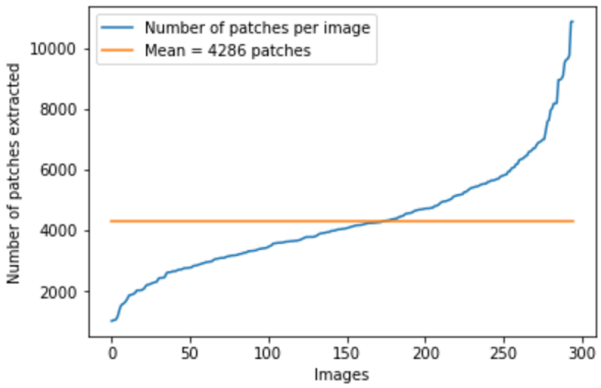


Figure 4.4: Development workflow.

4.3.3 Patch storage on ARC3 with HDF5

Turning every whole slide image into patches via the grid sampling algorithm in the previous chapter is a computationally expensive procedure, and could not be repeated for every session of experimentation. Instead the patches needed to be saved to disk on ARC3 and then loaded for training. Every patch is sampled at 100% quality in full 3-channel colour, with each 100x100 pixel patch taking up approximately 50MB of memory. With a total of over a million training patches, it was quickly evident that the patches would need to be generated once and then stored, but that in original form, they would consume too much memory. Using the grid sampling method, around 2,000 - 11,000 patches were retrieved from each image, depending on the size of the tissue sample featured. Complete patch statistics are shown in Figure 4.5.




```
Standard deviation of 1744 patches.
Minimum of 1018 patches per image.
Maximum of 10870 patches per image.
Total of 1264499 patches.
```

Figure 4.5: Statistics for patches sampled from all 295 whole slide images using grid method.

Various compression models were explored, and eventually HDF5 was chosen as a storage model. HDF5 is a data model which can accommodate a wide variety of data types, transparent file compression, and extremely large amounts of data. A Python module called h5py allows for easy usage. The patches from each whole slide image are stored in a single .h5 file and compressed by HDF5 using gzip. These files could be quickly read and loaded into NumPy arrays when needed. While gzip is not the best image compression method, it was the best compression method built into HDF5, which has the key advantage of offering transparent, efficient compression and decompression. Another main feature of .h5 files is the ability to read specific data items from them without reading in the entire file. Similar common storage formats such as comma-separated value (CSV), JavaScript Object Notation (Json) which can also save the images as numeric arrays do not have this capability. This feature is particularly useful in the training phase, as the patches need to be shuffled rather than read in sequential order.

The complete patch sampling process is shown in Figure 4.6.

In addition to the .h5 files, meta data for each patch was stored in CSV format. Information included image ID, classification label (0/1 normal/cancerous), and the coordinates of the patch location in the original image. Figure 4.7 shows eight example entries from a meta data file containing all the patches for training image 110042.

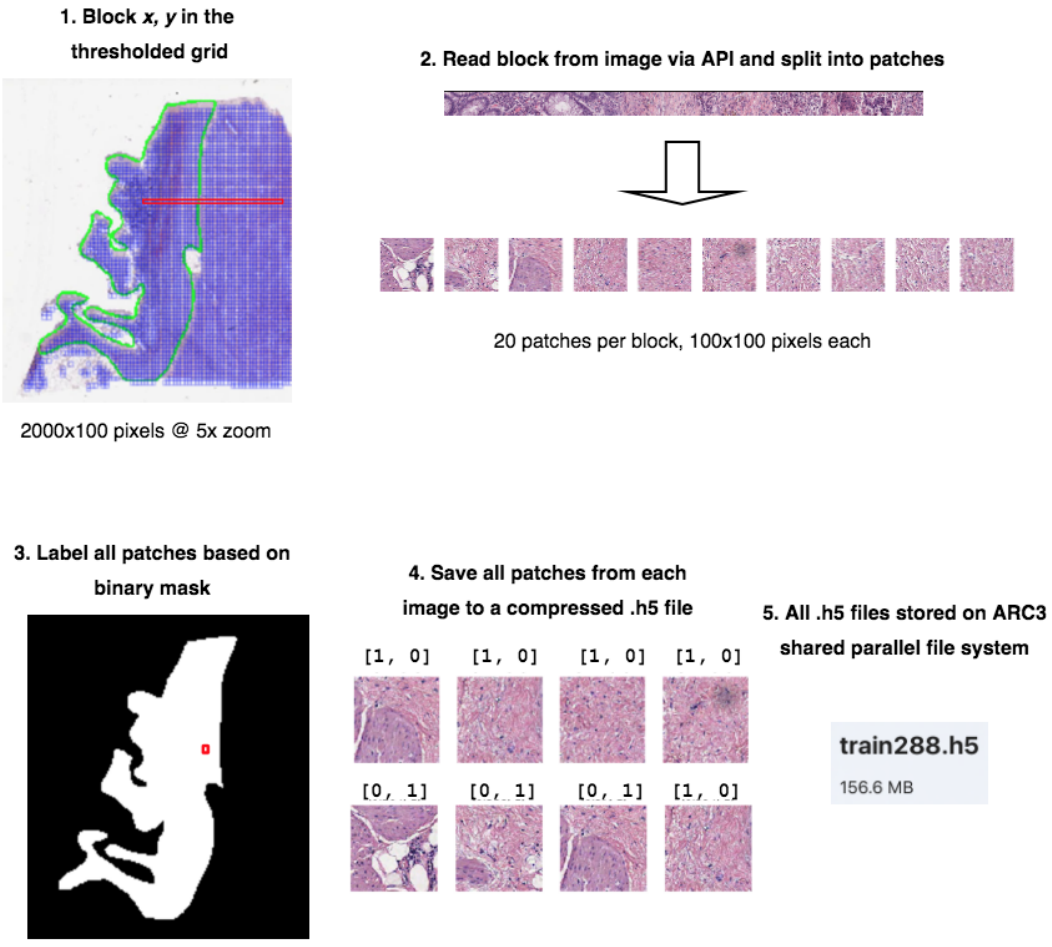


Figure 4.6: The patch sampling process.

1 110042.0 cancerous 6640 3920	1 110042.0 cancerous 6960 3920
1 110042.0 cancerous 6720 3920	1 110042.0 cancerous 7040 3920
1 110042.0 cancerous 6800 3920	1 110042.0 cancerous 7120 3920
1 110042.0 cancerous 6880 3920	0 110042.0 normal 8720 3920

Figure 4.7: The meta data stored for eight patches from image with ID 110042: label, image ID, class, x and y coordinates.

Chapter 5

The training phase

5.1 Setup

5.1.1 Division into training, validation, and test sets

The standard division of data is 80% training and validation, and 20% test. Typically, between 10% and 20% of the training data is set aside for validation, which can be used to prevent overfitting. In these experiments, 50 out of the 260 whole slide images (19.2%) were designated as test data so that none of the patches from one image used in training were included in testing. The division in shown in tabular form in Table 5.1.

Set	Number of whole slide images	Number of patches
Training and validation	210	844,378
Test	50	193,994

Table 5.1: Dataset division

5.1.2 Loading the patches

A naive initial attempt to train the CNN involved loading all the training patches at once, which allowed for very efficient in-memory processing and training. The MNIST digit classifier used as a template also loaded all the training, validation, and test images at once. While this method worked for smaller datasets, it was not feasible with the Eindhoven patches dataset even in the production environment within the container. Jupyter notebooks by default have a fixed memory limit, but even with that drastically increased, the Docker scheduler itself imposes memory limitations on the container, and closes any container attempting to use more than the allocated amount.

A second attempt used the TensorFlow feed dictionaries (placeholder variables) to load patches in batches during each training epoch, with a large batch size of 1,500 patches to reduce read frequency. While well within the container memory limits, training this way was extremely inefficient, taking over 18 hours on the two GPU nodes. In addition, there was no way to easily thoroughly shuffle the patches before training as the patches from each whole slide image are stored in a single .h5 file, and at each epoch, each batch was taken from a single .h5 file.

The final, most effective method divided the entire training, validation, and test data into sets. The memory limitations of the container permitted a maximum of approximately 180,000 patches

to be loaded at once, so all the patches were divided into 10 sets. Set i consisted of the i^{th} patch from each .h5 file, and was randomly shuffled before being fed to the feed dictionary for training. While this method does not minimise read frequency since every .h5 file must be read for each set, it returns a uniform sampling of patches across every whole slide image, and maximises training efficiency, thus taking advantage of the GPUs.

5.2 Learning

5.2.1 Basic learning requirements

Due to the nature of the dataset, a high test set accuracy is not expected nor desired, as all the ground truth labels are based on the original annotations. The CNN should demonstrate an ability to learn important features of carcinoma. Ideally, the CNN should not learn features of regression, such as fibrosis. A detailed discussion of what the CNN has learned as well as an evaluation on a small subset of re-annotated test images is provided in chapter 6.

Initial attempts to train the CNN demonstrated that two key elements were essential for learning. Firstly, with any fewer than 20,000 training patches, the CNN overfit the training set and was unable to get better than an average 50% accuracy on the test set, and tended to classify all the test images as a single class. Secondly, too low of a learning rate also encouraged overfitting the training set, and resulted in poor test performance. A high starting learning rate of 1.0e-4 discouraged the initial overfitting, and an exponential decay over time helped prevent divergence in the latter half of training.

5.2.2 Accuracy metrics

For quantifying CNN performance, several accuracy metrics were used. The segmentation problem over the whole slide images is essentially treated as a classification problem, where each patch of interest in the image (ignoring any background slide) is classified as normal or cancerous. Thus, instead of using accuracy metrics typically used in segmentation problems such as area ratios, shape similarity metrics, or mass centre location, the performance can be evaluated based on the test patches, their ground truth labels, and the CNN predictions. Every non-overlapping patch not filtered out through the thresholding function is sampled from each test image, and given a predicted value by the trained CNN. The softmax threshold then determines the final label for each patch of the test image. Each test image patch has two values: the predicted label output by the CNN+threshold, and the label provided by the original annotations. The segmentation accuracy can then be calculated very precisely by treating it as a classification problem by quantifying the true and false positives and negatives for all the patches in a test image.

The first and the second metric mentioned, F-measure and balanced accuracy, are the clearest indication of actual performance as they penalise both false negatives and false positives. For this dataset, there is not an equal division between normal and cancerous training and test patches. The traditional error calculation, the percent of correct classifications, does not reflect this imbalance, and may disproportionately reward classifying all the patches with the same label.

In the context of the pathology workflow, however, it is especially crucial to minimise false negatives. In the hypothetical scenario that a trained classification system were to be integrated into a software to support pathologists in identifying cancer, the presence of false positives would require a pathologist to examine the slide and observe that no cancer is present. At worst, the pathologist may trust the system and require the patient to undergo more testing, which would rule out carcinoma. False negatives, however, could encourage the pathologist to potentially ignore a

more careful verification of tissue which is in fact carcinoma. Misidentifying true carcinoma has the highest cost. For this reason, sensitivity (or recall) is a key accuracy metric. Sensitivity measures the true positives which are not misidentified; in other words, a high sensitivity minimises false negatives. In the medical imaging context, this is also an important metric.

- F-measure: is also known as F1 or F-score, and is the weighted harmonic average of both precision and recall.
- Balanced accuracy (BAC): is the arithmetic mean of sensitivity and specificity.
- Sensitivity or recall: the true positive rate, the proportion of positives which are correctly classified as positives.
- Error rate: the standard ratio of correct over total classifications.
- Specificity: the true negative rate, the proportion of negatives which are correctly classified as negatives.
- Precision: the consistency of classification results, or, the ratio of true positives over both true and false positives.

5.2.3 Experiments

The parameter values from the CNN outlined in chapter 4 are used as the starting base for the following experiments. The purpose of experimentation is to improve the quantitative accuracy of the CNN on the test set, but equally to qualitatively observe what the CNN is learning. Due to the poor quality of the annotations, it is also taken into consideration that too high of an accuracy would be unusual. The full results of each experiment discussed in the following sections are included in Appendix C, in chapter 8.

Softmax threshold

By default, the threshold for the softmax activation function is 0.5; classification layer outputs above this value are classified as 1, cancerous, while outputs below the threshold are classified as 0, normal. This experiment is run over multiple models (the trained CNN weights from the best performing set of parameters) to determine whether the threshold varies depending on the model, or if there is a fixed value or range of values for which all models perform the best. It was found that a small range of threshold values within the 0.65 - 0.75 range, does perform the best on all the tested models (Figure 5.1).

Figure 5.2 shows the result of several accuracy metrics applied to the complete test set as the softmax threshold value is varied.

Number of layers

Each layer of a CNN learns certain features from the training data, and deeper layers subsequently learn finer features, allowing a CNN to better learn difficult classification tasks. A network which is too shallow may result in the CNN being unable to learn the training data. However, if the training features can be learned in a shallower network and not overfitted, then extra layers only add additional parameters the CNN needs to learn, and increases the risk of overfitting the training data during the extra training epochs required to learn all the parameters. Thus, this experiment

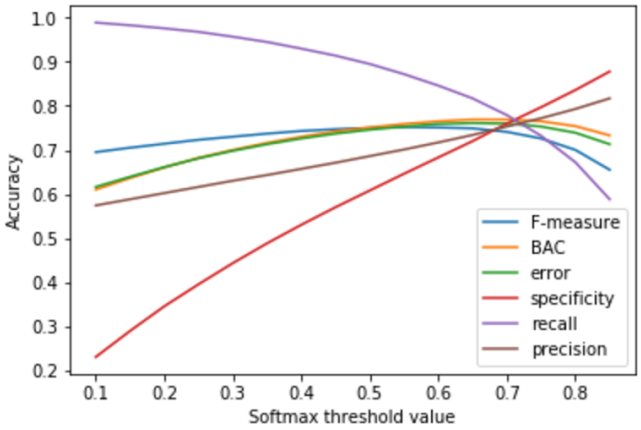


Figure 5.1: A softmax threshold within the 0.65 - 0.75 range returns the best accuracy for multiple learned models.

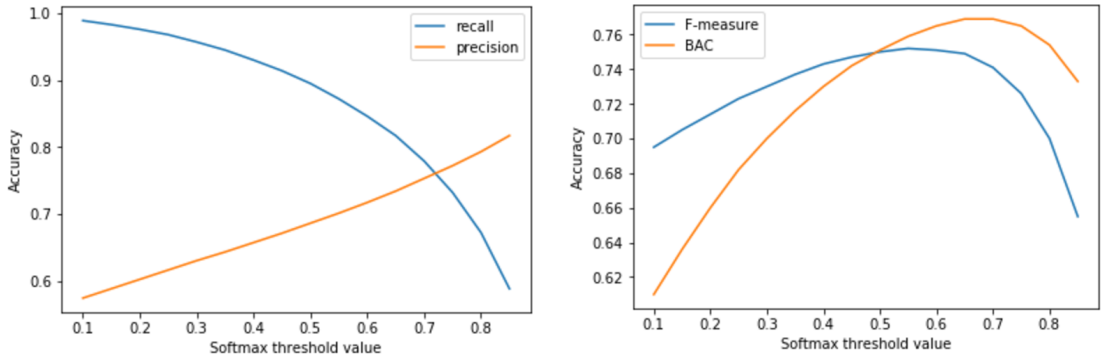


Figure 5.2: A clear illustration of the precision recall tradeoff (left). The key accuracy metrics observed in these experiments (right) are able to reflect the tradeoff and are used to find the optimal solution.

attempts to determine the fewest number of layers necessary to generalise well over unseen patches and not overfit the training data.

As shown in Figure 5.3, several experiments indicate that three layers is deep enough to learn the features of the dataset, and any shallower or any deeper negatively affects performance. It is important to note that in these experiments, the size of the fully-connected layer remains the same, meaning that regardless the number of parameters in the network, everything learned must be represented by a fixed number of features.

A careful qualitative evaluation provides some insight into which features are being learned, and in which layer. A patch passing through the best performing network with three layers is shown in Figure 5.4, demonstrating that general features are captured in the first and second layers, while the third layer appears to learn finer features. Even at the deepest layer, each filter has captured some information from the original patch.

The network deepened to five layers shows signs of overfitting the training data, and consequently does not generalise well. Figure 5.5 shows the training accuracy reaching unreasonably high values for the given dataset, decreasing validation accuracy, and high validation loss indicating divergence. With five layers and enough training epochs, the network has enough parameters

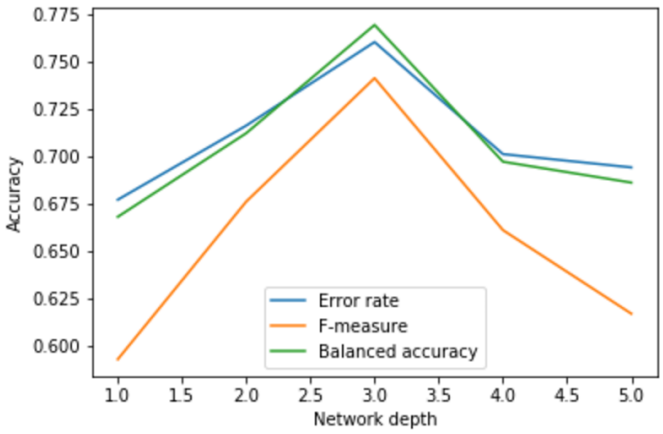


Figure 5.3: The performance of networks with up to 5 layers.

to essentially memorise the training set patches.

Number of filters

The results of the depth experiments already hint that too many filters will negatively affect learning, as increasing depth increases the number of features which are fed into the first fully-connected layer. Increasing the number of filters per layer also corresponds to more input features to the fully-connected layer. This can be qualitatively confirmed by Figure 5.6 showing a patch passed through a 3-layer CNN with 32, 32, and 64 filters per layer respectively. The first and second layers have several filters each which appear to not be learning any particular feature, while the majority of the third layer filters seem to learn nothing at all.

The declining accuracy as the number of filters increases further confirms these observations, as seen in Table 5.2. The best performing combination of filters per layer is in bold, with highest F-measure and balanced accuracy.

Number of filters	F-measure	Balanced accuracy	Error rate	Recall	Specificity	Precision
16, 32, 32	0.679	0.723	0.271	0.605	0.841	0.774
16, 16, 32	0.684	0.724	0.271	0.617	0.831	0.767
16, 16, 16	**0.741**	**0.769**	**0.240**	**0.779**	**0.76**	**0.753**
32, 32, 32	0.659	0.722	0.269	0.55	0.894	0.823
32, 32, 64	0.638	0.712	0.278	0.517	0.907	0.833

Table 5.2: Accuracy for 3-layer networks with varying number of filters per layer.

Filter window size

The filter window size should correspond to the size of the features of interest in the input patches. From a human perspective, the “features” which make carcinoma identifiable include a combination of abnormal morphologies. Figure 5.7 shows the results of a patch passing through the first layer of filters on two CNNs. The first CNN uses a 4x4 pixel filter window, and the second

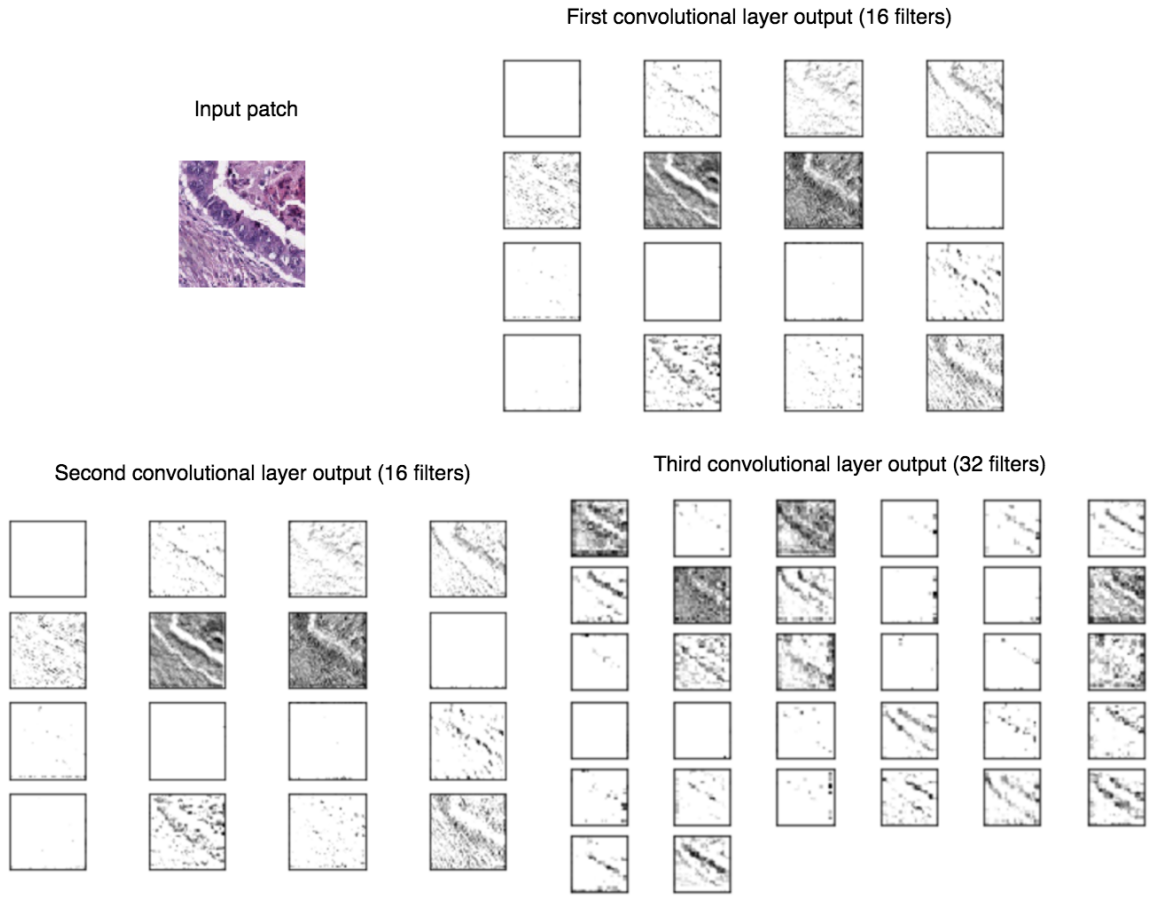


Figure 5.4: Visualisation of the features learned by the CNN with three layers of 16, 16, and 32 filters per layer respectively.

uses a 14x14 window. The small filter window has learned much finer features than the larger. The experiments with filter windows within this range indicate that the best sized window is 8x8 pixels.

Colour normalisation: grayscale and Reinhard

Converting the images to grayscale gives the CNN fewer features to learn, as the three colour channels are reduced to one. The default 3-layer CNN is too deep for the reduced number of features, as evident in Figure 5.8. A more shallow CNN avoids overfitting but does not generalise well on the test data. As discussed in chapter 1, pathologists generally believe that colour helps differentiate between tissue types and distinguish features. It appears that colour also helps the CNN to learn; the models perform better on coloured patches than grayscale. Because of the large number of sample patches taken for training, this approach most closely mirrors the approach where researchers artificially generate colour variation. The original unaltered images give the best results overall.

All the experimental results are listed in full in Appendix C.

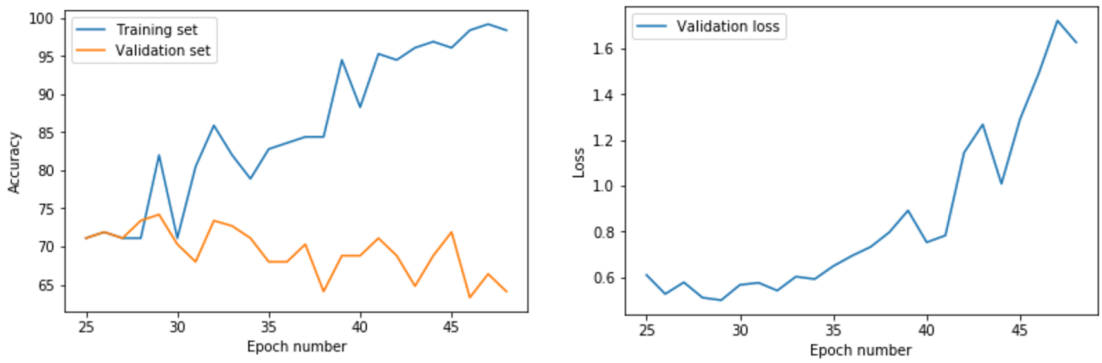


Figure 5.5: With too many layers, the training and validation accuracy diverge and validation loss increases, before reaching 50 epochs of training.

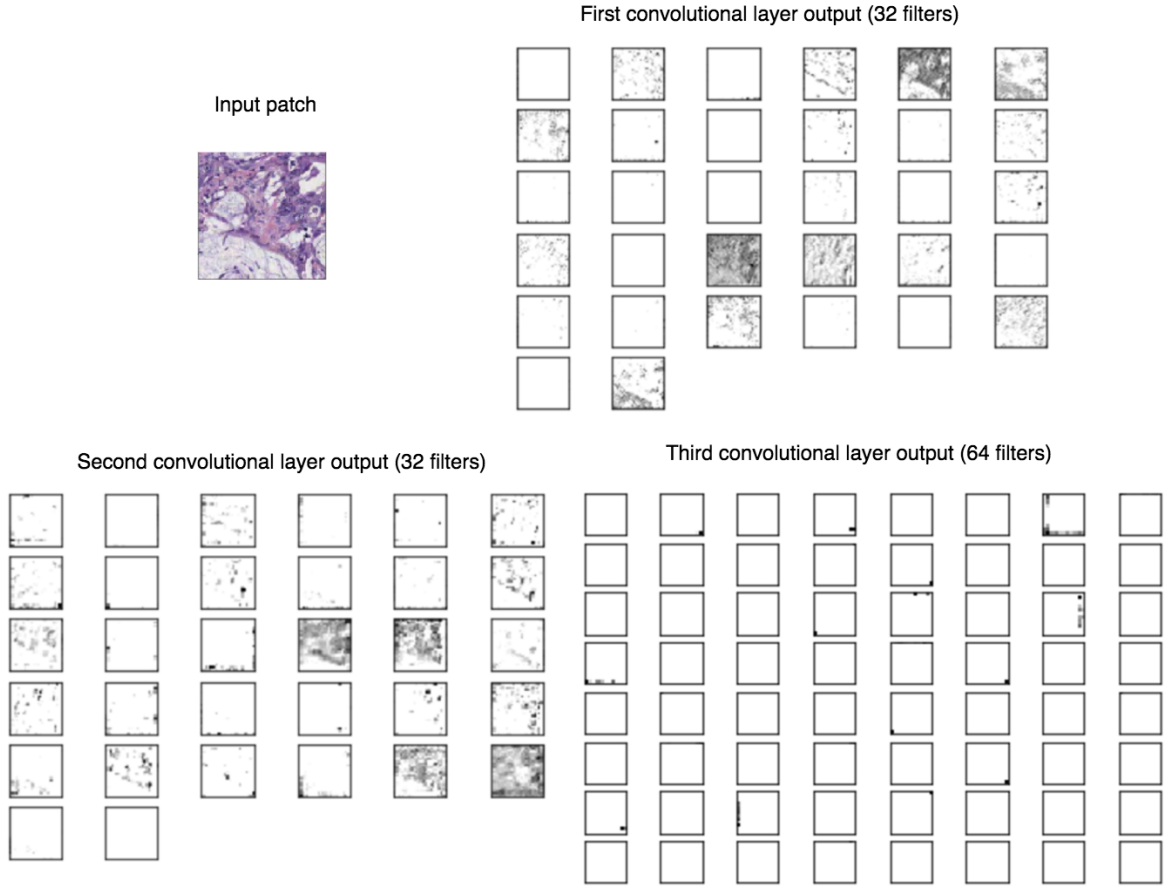


Figure 5.6: Visualisation of the features learned by the CNN with three layers of 32, 32, and 64 filters per layer respectively.

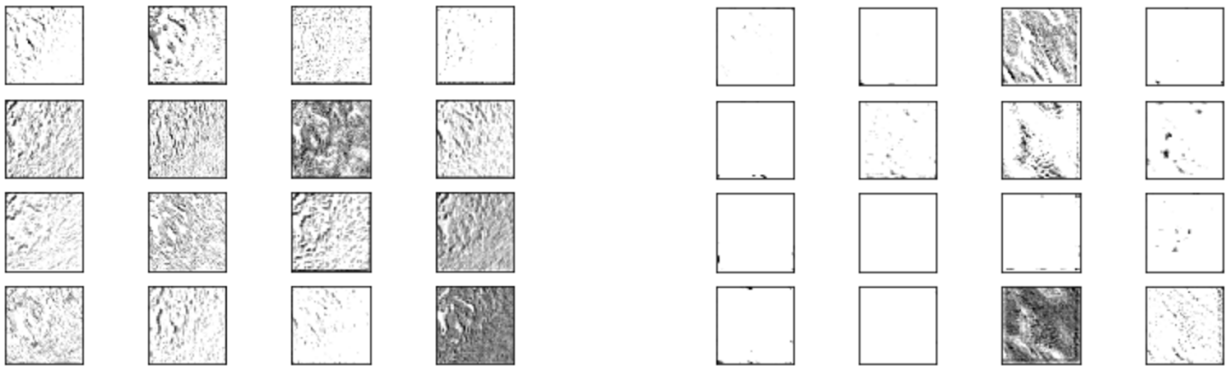


Figure 5.7: The results of a patch passing through the first layer of two networks, one with a filter window size of 4x4 pixels (left), and 14x14 pixels (right).

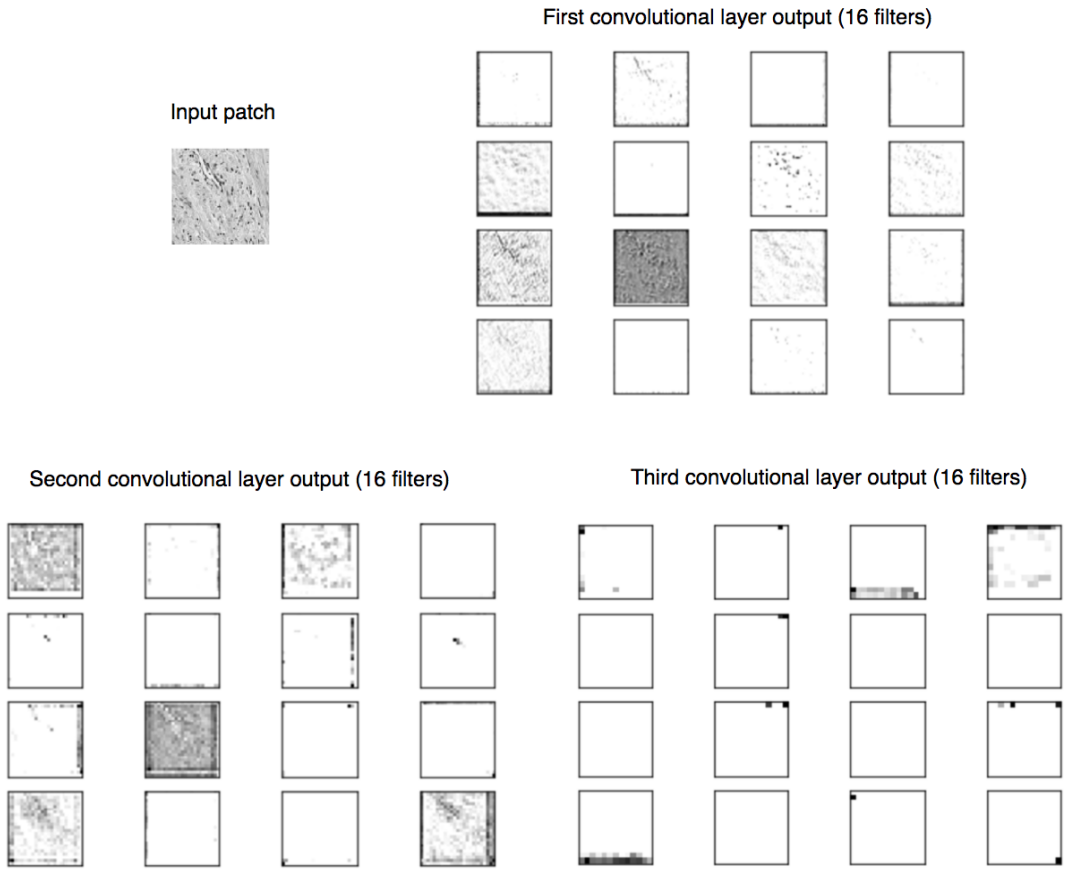


Figure 5.8: Patches converted to grayscale give the CNN far fewer features to learn; by the third layer of the network, the filters have learned very little from input patches.

Chapter 6

Evaluation

6.1 Results and visual analysis

Due to the nature of the annotations on the Eindhoven dataset, a result surpassing the most recent work by researchers in similar areas is not expected. However, the poor ground truth provides opportunities to demystify deep learning and examine what the CNN has learned. Several interesting observations can be made by looking at heatmaps generated using the CNN predictions. The predictions, in the range of 0.0 - 1.0 where 1.0 indicates a high likelihood of carcinoma and 0.0 indicates normal tissue, are used to colour the original patch locations on the test images. We can qualitatively see how well the CNN has performed by comparing the generated heatmaps with the original annotations. The heatmap colour scheme is shown in Figure 6.1.

Figure 6.1: The heatmap colour bar with yellow indicating high likelihood of carcinoma, and dark violet a low likelihood.

When examining the heatmaps, it appears that the CNN has learned to detect the general regions of interest. However, in the context of the poor annotations, a good heatmap is not always equivalent to a good carcinoma detection. Figure 6.2 displays a selection of test images for which the heatmaps indicate that the CNN has learned regression/fibrosis as well as carcinoma. Consequently, while it predicts the annotated region is cancerous, it also gives high probability to anything that looks similar to the regression outside of the region.

Other test images, however, such as those in Figure 6.3 indicate that the CNN is unsure about areas of regression within the original ground truth annotation, and gives them a lower probability.

Colouring the patches according to their confusion matrix category (true positive, true negative, false positive, or false negative), gives additional insight into what the CNN has learned. Figure 6.4 presents the colouring scheme, and Figure 6.5 displays several confusion matrix heatmaps.

There may also be potential for transferable learning across cancer types. One whole slide image in the test set has prostrate cancer as well as rectal. The network has surprisingly also been able to detect its features despite not having been trained on them. Figure 6.6 shows the CNN generated heatmap for the test image, as well as the ground truth annotation. In particular, the deformed prostrate glands in the top left of the tissue sample have been detected by the CNN.

6.2 Insight into actual accuracy

6.2.1 Re-annotated subset

After training and experimentation with the CNN was complete, an expert pathologist was able to analyse the test set, and re-annotate 10 images on which the CNN performed poorly so that only the viable carcinoma was enclosed. An additional five original annotations were judged acceptable by the same standards. This produced a small set of 15 whole slide images never seen by the CNN during training for which the ground truth is on par to those used in similar research in the domain, where the annotations segment tumour only. Some of the differences between the original annotations and the new annotations are highlighted in Figure 6.7.

On this subset of images, the CNN attained an F-measure score of 0.487 and balanced accuracy of 0.724. A visual examination reveals that on certain whole slide images the CNN has segmented significantly better than the original annotation, and much closer to the pathologist re-annotation. However, in other cases, it has confused regression for carcinoma. Overall, the network does very well at correctly classifying all true carcinoma, and does less well at determining what is not. It is understandable that the network is not good at distinguishing non-cancerous tissue, as it has been trained with incorrectly labeled ground truth, namely large amounts of normal tissue labeled as cancerous. In the context of digital pathology, it is more valuable to correctly classify true carcinoma than to correctly classify normal tissue, so a high sensitivity (recall) indicates that despite the rough training annotations, the CNN has still learned the crucial elements.

If the CNN has actually learned carcinoma features, raising the threshold value should increase accuracy, as it should filter out the patches which it is unsure about; namely, all tissue which is regression or normal tissue that looks like regression. It is found that a high threshold of 0.85 does in fact return the highest accuracy, improving the test performance to the values displayed below in Table 6.1.

Threshold	F-measure	Balanced accuracy	Error rate	Specificity	Recall	Precision
0.5	0.487	0.724	0.357	0.564	0.711	0.513
0.85	0.551	0.762	0.207	0.813	0.883	0.373

Table 6.1: Accuracy of best performing network on test subset, with default and adjusted threshold.

6.2.2 Comparison

The results of the CNN trained on sub-optimal data and evaluated on a small test subset of 15 correctly annotated images can be compared with the results obtained by researchers performing tumour segmentation on breast and rectal cancer with good quality annotations. It must be noted that due to the small subset size, a good average accuracy cannot be measured; however, Table 6.2 still illustrates that as is, the CNN with a high softmax threshold is not far away from comparable solutions on much better data sets. Note that the CNN methods listed below are not all purely CNN approaches; they may include extra classification tactics on top of the CNN.

Approach	Ref.	Cancer type	Balanced accuracy
Hand-crafted features	[5]	breast	0.772
CNN	[5]	breast	0.842
CNN	*	rectal	0.762
CNN	[40]	brain	0.840
CNN	[35]	breast	0.984

Table 6.2: Comparison of results from this work (from small test subset) and similar problems; * indicates this work.

6.3 The future of learning despite sub-optimal ground truth

Despite poor ground truth, the neural network has still managed to learn the key features of carcinoma. A neural network learns by error back-propagation starting from the output layer. For each training patch that is normal, but incorrectly labeled as cancerous, the non-linear separation boundary learned by the network is moved a small amount in the direction which would classify that patch as cancerous. If another training patch with similar morphological features passes through the network, and is correctly labeled as normal, the separation boundary is shifted back in the original direction to classify that patch as normal.

Theoretically speaking, it is possible to understand why the presence of a two similar patches labeled differently would cumulatively have little effect on the classifier. An equal number of inversely labeled patch pairs would allow the network boundary to effectively not be impacted by the incorrectly labeled patches. However, when given a test patch which is similar to the inversely labeled training patches, the CNN would be equally likely to correctly label it as to incorrectly label it. The network is unsure about the entire range of patches which exhibit regression-like features.

Future work with this particular dataset will explore using a very high softmax threshold (0.9 or greater) to allow the network to only classify the patches which it strongly believes are cancerous. There is also potential for better training the network using its own initial predictions; by running the training set through the trained network, re-labelling them with only the strongest predictions marked as cancerous and the rest marked as normal, the CNN may be able to learn a better separation boundary. This bootstrapping approach may result in a better final accuracy. Figure 6.8 shows four images from the re-annotated subset, demonstrating the potential of using a high softmax threshold on the training set to re-train the network. Higher threshold values produce predictions much closer to the actual ground truth. In addition, a probability or Markov model could be applied after the CNN to introduce some basic contextual information, such as discouraging patches from being classified differently than all their surrounding neighbours. This would produce a cleaner final segmentation and slightly improve accuracy.

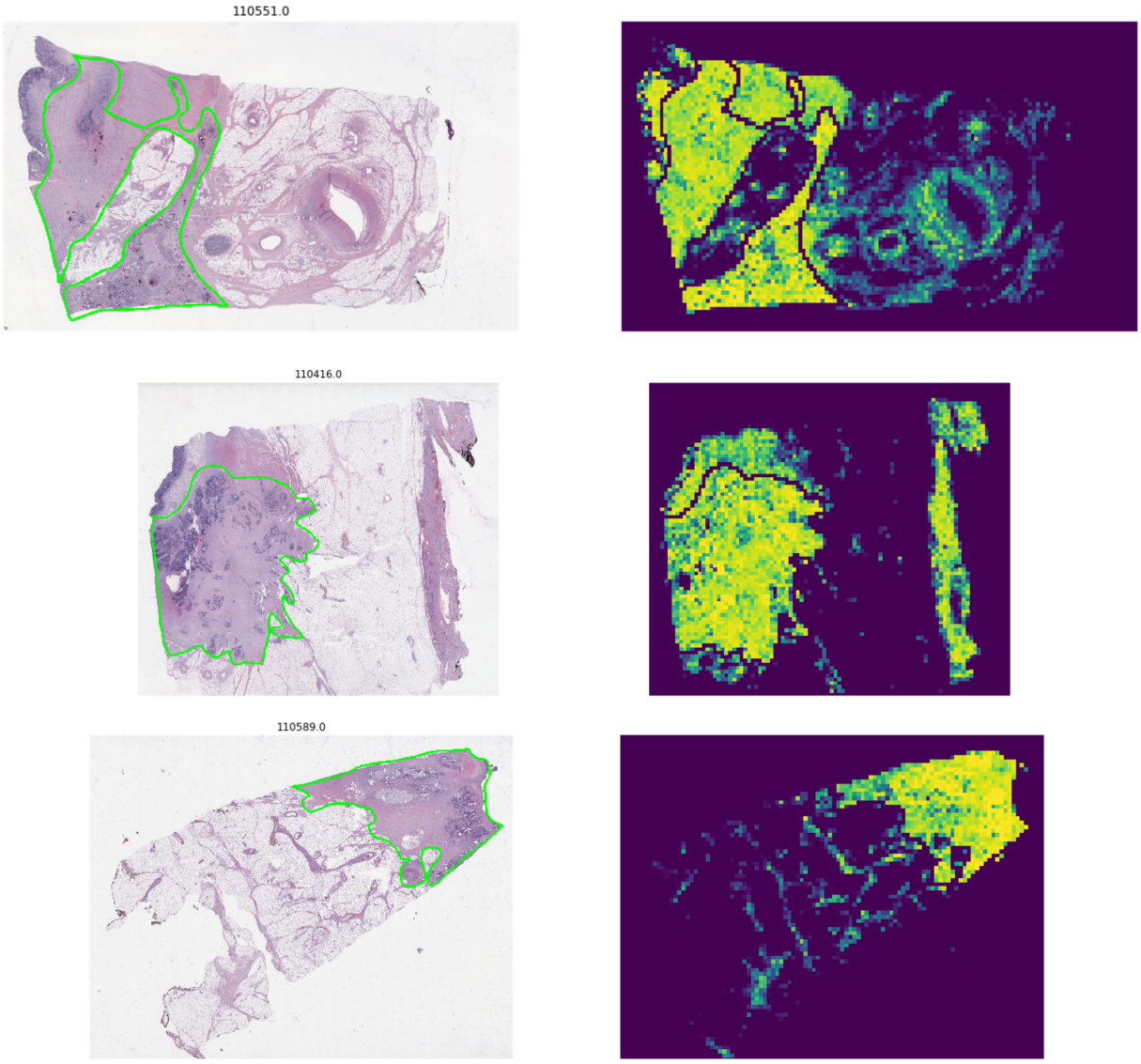


Figure 6.2: A selection of images and predictions from the test set, evidence that the CNN has unfortunately learned regression as well as carcinoma; however, it often appears less certain about the regression being cancerous than it is for the actual carcinoma.

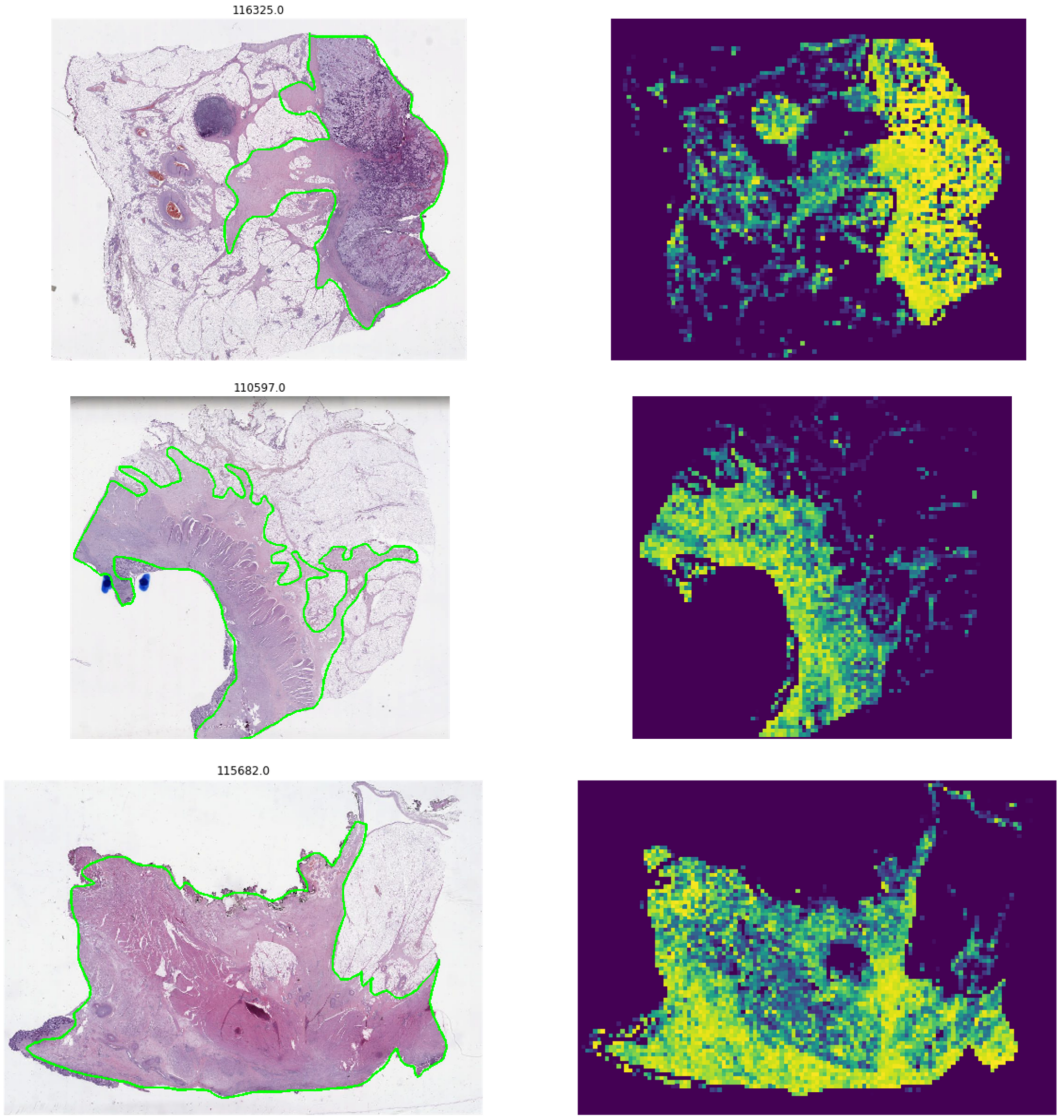


Figure 6.3: A selection of images and predictions from the test set where the CNN does better than the original ground truth annotation in detecting carcinoma; gives low probability to fibrosis in centre of slide (top), the entirety of the tissue sample (centre), and pink stroma in middle (bottom).

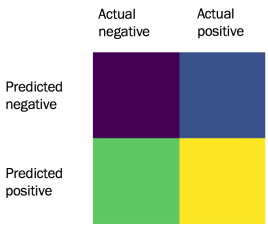


Figure 6.4: The colouring scheme according to confusion matrix category.

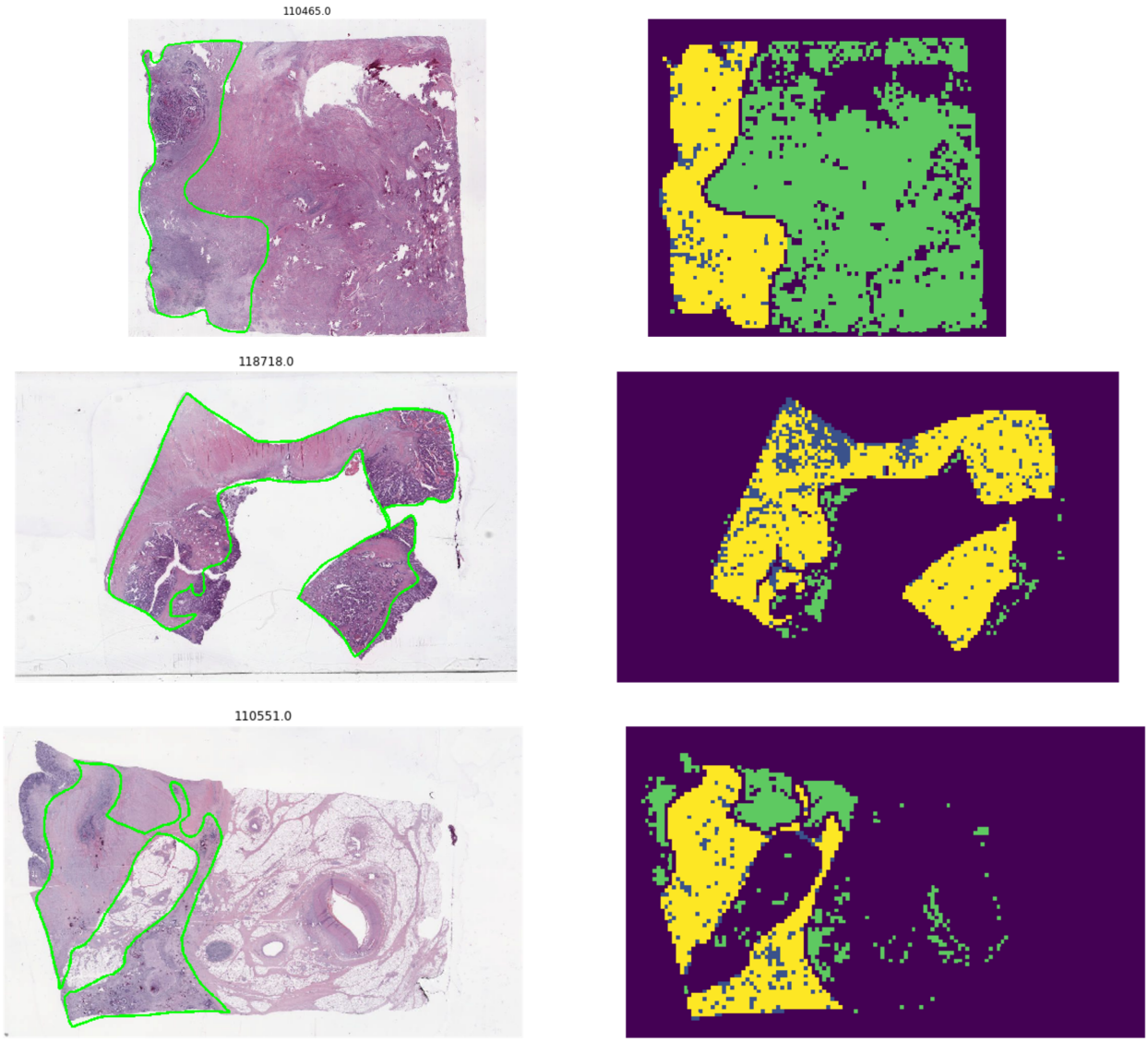


Figure 6.5: A selection of confusion matrix heatmaps demonstrating that the CNN has learned to classify regression as carcinoma.

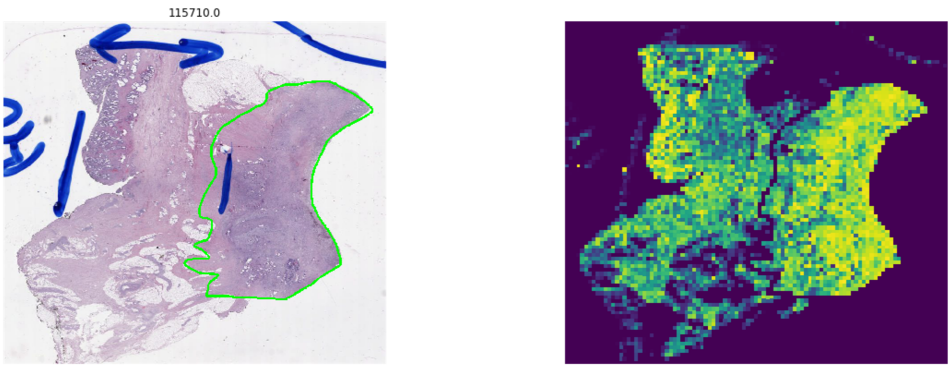


Figure 6.6: Prostrate cancer on the top left of the tissue sample, and rectal on the bottom left; both recognised by the trained network.

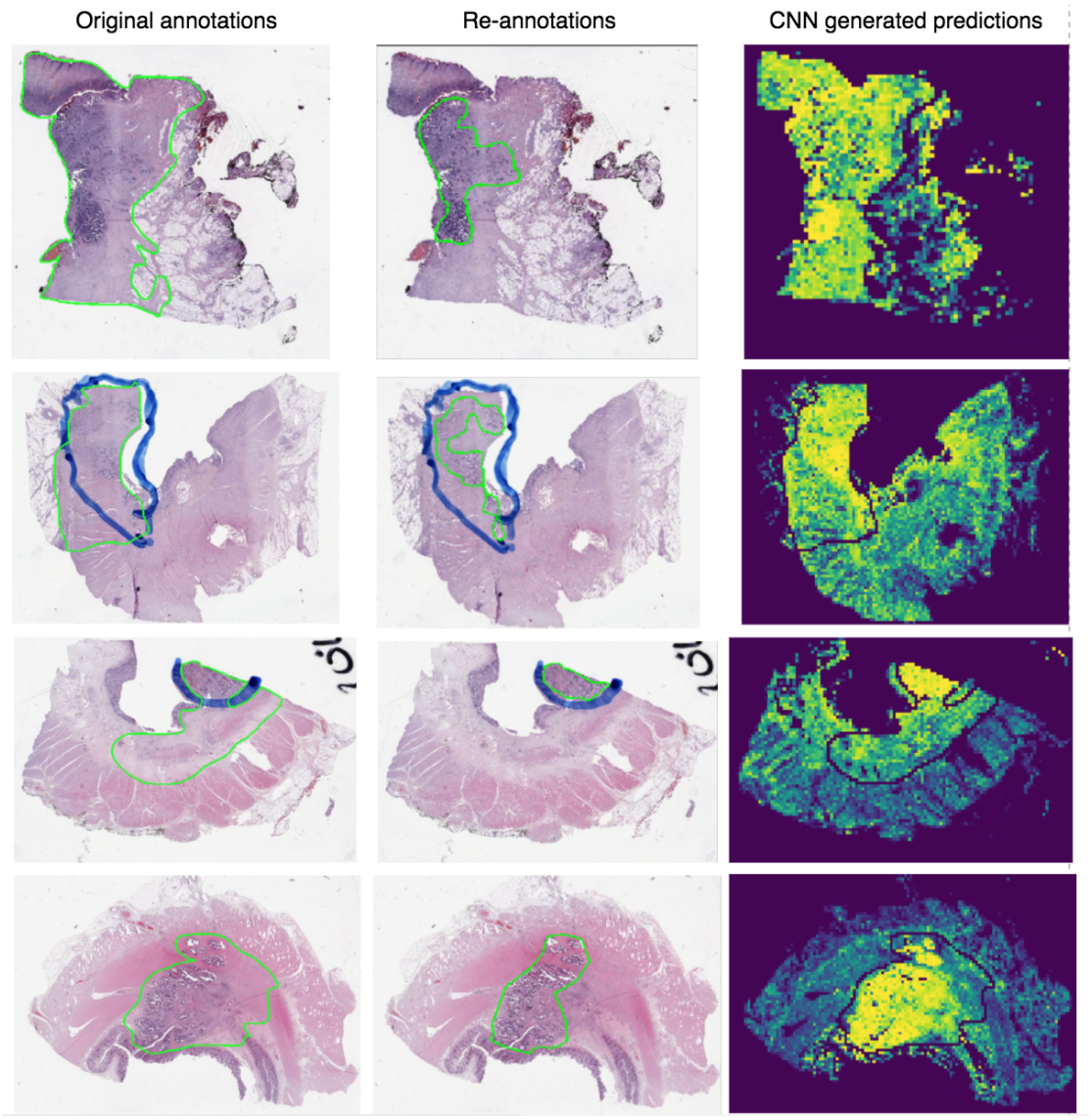


Figure 6.7: Four test images with original annotations, expert re-annotations, and CNN predictions.

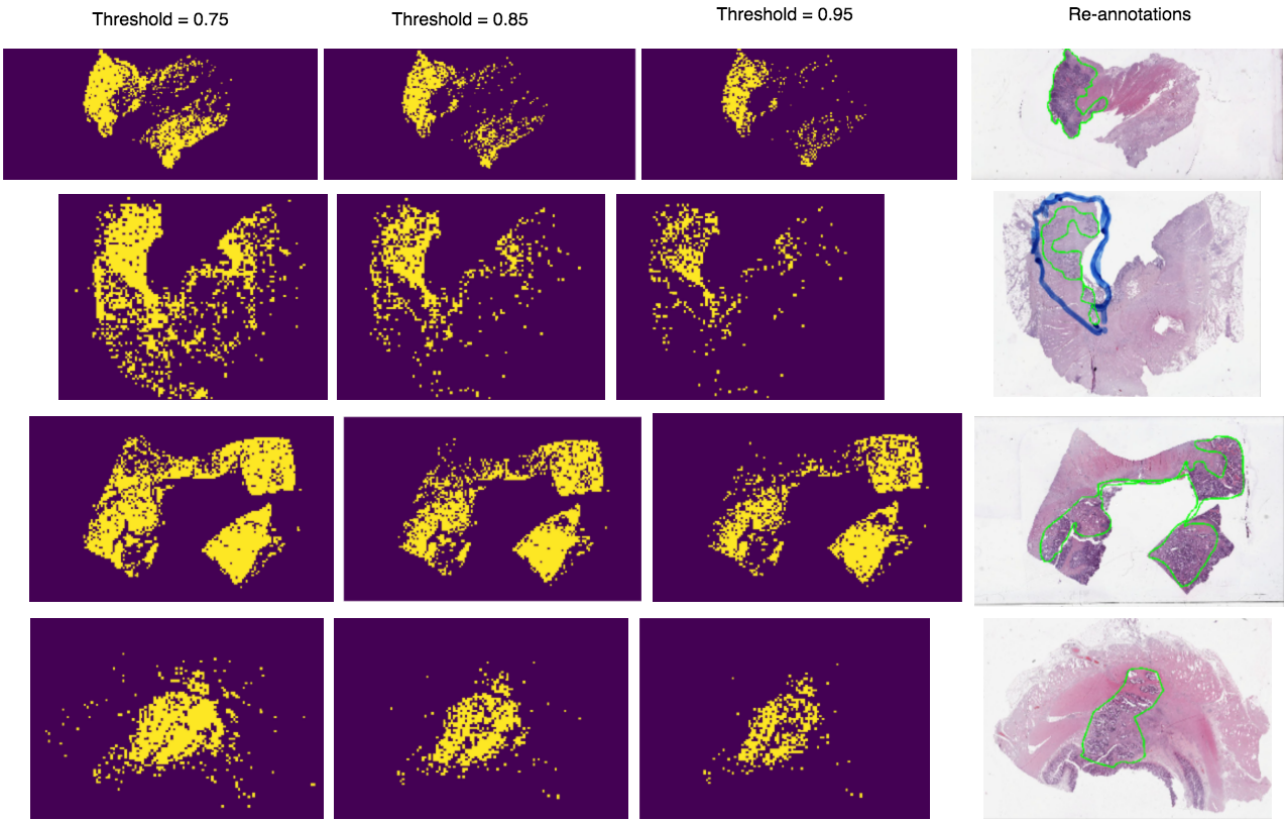


Figure 6.8: Four test images with CNN predictions using different softmax threshold values, alongside the expert re-annotations; the areas it is most certain about are more likely to be close to the ground truth.

Chapter 7

Conclusion

7.1 Project reflections

The two main obstacles in this project were the sub-optimal dataset and lack of direct access to the whole slide images. Neither obstacle is insurmountable, but due to the short timeline of the project (approximately 14 weeks from start until dissertation submission), the decision was made to progress despite them. In retrospect, a more thorough initial examination of the dataset and better understanding of the purpose of the annotations may have caused the use of the Eindhoven set to be reconsidered. However, using the Eindhoven dataset allowed for a more interesting evaluation and gave insight into the capabilities of a neural network.

The project timeline outlined in the initial scoping and planning phase, mentioned briefly in chapter 1, was closely followed throughout the project, although the initial work outlined in chapters 3 and 4 was more time-consuming than the later phases. Direct image access via a temporary mount to external file system from within ARC would have greatly sped up and simplified the patch sampling process, and the time gained could have been used for experimenting with different patch sizes and zooms in the training stage. As it required nearly a full week to convert each whole slide image into patches in HDF5 format with the conversion scripts running day and night, it was not feasible to repeat the process multiple times.

In addition, extra time could have been used to designate the training set and test set by separating all the images into five sets of equal size, and rotating which is used as the test set for each complete round of an experiment. This would require each experiment to be repeated five times. The final results could then be averaged over all the sets, reducing the possibility of a skewed result based on unusual test images.

During the parameter experimentation phase, the test set was used for evaluating the accuracy of each model. If there had been more data, it would have been better to use the validation set instead, thus eliminating the possibility of customising the CNN to the test set. Unfortunately, there was not enough time to see whether the validation set was large enough to obtain clear results for all the experiments.

7.2 A promising future

While the appeal of whole slide images over traditional slides continues to rise, opening up opportunities for machine learning, good quality annotations on whole slide images for the purposes of computer vision research are difficult to come by. Clinical pathologists generally roughly segment a tumour on a whole slide image at a magnification and quality far lower than those required by

a deep learning algorithm for training. Annotating at the ideal accuracy for a computer vision system would be a painstaking task. However, this work has proved that despite significant mislabelled ground truth, deep neural networks are still capable of learning the principal features which distinguish carcinoma from normal tissue. While the results understandably cannot compete with those of research performed on datasets with higher quality tumour annotations, they are still comparable and have clearly shown that a CNN does not need perfect ground truth in order to learn. This indicates that perhaps with bootstrap re-training and additional improvements to the model, rough annotations could be sufficient for obtaining state-of-the-art results. Future extensions to this work hope to explore these possibilities.

Intelligent deep learning systems could offer support in many areas of the pathologist workflow. Glass slides are placed in whole slide scanners and pass through the complete scanning process before any defective slides can be manually detected and either re-scanned or rejected. A deep learning system built into the scanners could be used for early detection of defects including large tissue tears or artefacts, major discolourations, and misaligned or marked slide covers. This could reduce the percentage of slides needing to be re-scanned or manually rejected, saving valuable time. In addition, highly accurate deep learning systems could be used for automatic analysis of removed polyps. Polyps are growths which appear in the colon surface and are commonly removed by physicians. Tissue samples from these polyps are then examined by pathologists for carcinoma, but only very few of polyps removed are actually cancerous. An automated system could be trained to support and speedup the screening of polyp tissue samples. These tasks are among many which could benefit from automated computer vision systems.

Deep learning algorithms have proven capable of performing difficult tasks in pathology and show enormous potential for having clinical applications. In work by Wang et al., it was observed that the errors of the deep learning system were not strongly correlated with those of the test pathologists annotating the same images [35]. Combining the conclusions of the trained CNN and those of a test pathologist thus gave a better overall accuracy. This strongly proves that deep learning algorithms have potential to support pathologists in a clinical environment. While, due to the nature of the domain, they may never be suited to replace the expertise of practicing pathologists, they promise to aid in tasks which can be tedious, painstaking, and subject to a degree of error and subjectivity.

The outlook is promising. If deep learning's heavy reliance on good-quality data and powerful computational resources can be reduced, a future where pathologists and intelligent computer systems work together is altogether imaginable.

Bibliography

[1] J Anitha, C Kezi Selva Vijila, D Jude Hemanth, and A Ahsina. Self organizing neural network based pathology classification in retinal images. In *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*, pages 1457–1462. IEEE, 2009.

[2] Francesco Bianconi, Alberto Álvarez-Larrán, and Antonio Fernández. Discrimination between tumour epithelium and stroma via perception-based features. *Neurocomputing*, 154:119–126, 2015.

[3] W Scott Campbell, Geoffrey A Talmon, Kirk W Foster, Subodh M Lele, Jessica A Kozel, and William W West. Sixty-five thousand shades of gray: importance of color in surgical pathology diagnoses. *Human pathology*, 46(12):1945–1950, 2015.

[4] Hao Chen, Qi Dou, Xi Wang, Jing Qin, and Pheng-Ann Heng. Mitosis detection in breast cancer histology images via deep cascaded networks. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 1160–1166. AAAI Press, 2016.

[5] Dan C Cireşan, Alessandro Giusti, Luca M Gambardella, and Jürgen Schmidhuber. Mitosis detection in breast cancer histology images with deep neural networks. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, pages 411–418. Springer, 2013.

[6] Emily L Clarke and Darren Treanor. Colour in digital pathology: A review. *Histopathology*, 2016.

[7] Angel Cruz-Roa, Ajay Basavanhally, Fabio González, Hannah Gilmore, Michael Feldman, Shridar Ganesan, Natalie Shih, John Tomaszewski, and Anant Madabhushi. Automatic detection of invasive ductal carcinoma in whole slide images with convolutional neural networks. In *SPIE medical imaging*, pages 904103–904103. International Society for Optics and Photonics, 2014.

[8] Arkadiusz Gertych, Nathan Ing, Zhaoxuan Ma, Thomas J Fuchs, Sadri Salman, Sambit Mohanty, Sanica Bhele, Adriana Velásquez-Vacca, Mahul B Amin, and Beatrice S Knudsen. Machine learning approaches to analyze histological images of tissues from radical prostatectomies. *Computerized Medical Imaging and Graphics*, 46:197–208, 2015.

[9] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, pages 6645–6649. IEEE, 2013.

[10] Allan Gray, Alex Wright, Pete Jackson, Mike Hale, and Darren Treanor. Quantification of histochemical stains using whole slide imaging: development of a method and demonstration of its usefulness in laboratory quality control. *Journal of clinical pathology*, pages jclinpath–2014, 2014.

[11] Jon Griffin and Darren Treanor. Digital pathology in clinical use: where are we now and what is holding us back? *Histopathology*, 70(1):134–145, 2017.

[12] Yue Huang, HAN ZHENG, CHI LIU, Xinghao Ding, and Gustavo Rohde. Epithelium-stroma classification via convolutional neural networks and unsupervised domain adaptation in histopathological images. *IEEE Journal of Biomedical and Health Informatics*, 2017.

[13] David H Hubel and Torsten N Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, 195(1):215–243, 1968.

[14] Humayun Irshad, Antoine Veillard, Ludovic Roux, and Daniel Racoceanu. Methods for nuclei detection, segmentation, and classification in digital histopathology: a reviewcurrent status and future potential. *IEEE reviews in biomedical engineering*, 7:97–114, 2014.

[15] Steve Lawrence, C Lee Giles, Ah Chung Tsoi, and Andrew D Back. Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 8(1):98–113, 1997.

[16] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[17] James S Lewis Jr, Sahirzeeshan Ali, Jingqin Luo, Wade L Thorstad, and Anant Madabhushi. A quantitative histomorphometric classifier (quhbic) identifies aggressive versus indolent p16-positive oropharyngeal squamous cell carcinoma. *The American journal of surgical pathology*, 38(1):128, 2014.

[18] Siqi Li, Huiyan Jiang, and Wenbo Pang. Joint multiple fully connected convolutional neural network with extreme learning machine for hepatocellular carcinoma nuclei grading. *Computers in Biology and Medicine*, 2017.

[19] Cheng-Yuan Liou, Wei-Chen Cheng, Jiun-Wei Liou, and Daw-Ran Liou. Autoencoder for words. *Neurocomputing*, 139:84–96, 2014.

[20] Anant Madabhushi, Scott Doyle, George Lee, Ajay Basavanhally, James Monaco, Steve Masters, John Tomaszewski, and Michael Feldman. Integrated diagnostics: a conceptual framework with examples. *Clinical chemistry and laboratory medicine*, 48(7):989–998, 2010.

[21] Anant Madabhushi and George Lee. Image analysis and machine learning in digital pathology: Challenges and opportunities, 2016.

[22] Derek Magee, Darren Treanor, Doreen Crellin, Mike Shires, Katherine Smith, Kevin Mohee, and Philip Quirke. Colour normalisation in digital histopathology images. In *Proc Optical Tissue Image analysis in Microscopy, Histopathology and Endoscopy (MICCAI Workshop)*, volume 100. Daniel Elson, 2009.

[23] Christopher D Malon and Eric Cosatto. Classification of mitotic figures with convolutional neural networks and seeded blob features. *Journal of pathology informatics*, 4, 2013.

[24] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[25] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1928–1937, 2016.

[26] Mateo Puerto, Tania Vargas, and Angel Cruz-Roa. A digital pathology application for whole-slide histopathology image analysis based on genetic algorithm and convolutional networks. In *Computational Intelligence (LA-CCI), 2016 IEEE Latin American Conference on*, pages 1–7. IEEE, 2016.

[27] Rajat Raina, Anand Madhavan, and Andrew Y Ng. Large-scale deep unsupervised learning using graphics processors. In *Proceedings of the 26th annual international conference on machine learning*, pages 873–880. ACM, 2009.

[28] Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. Color transfer between images. *IEEE Computer graphics and applications*, 21(5):34–41, 2001.

[29] Dennis W Ruck, Steven K Rogers, Matthew Kabrisky, Mark E Oxley, and Bruce W Suter. The multilayer perceptron as an approximation to a bayes optimal discriminant function. *IEEE Transactions on Neural Networks*, 1(4):296–298, 1990.

[30] Daniel L Ruderman, Thomas W Cronin, and Chuan-Chin Chiao. Statistics of cone responses to natural images: Implications for visual coding. *JOSA A*, 15(8):2036–2045, 1998.

[31] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[32] Korsuk Sirinukunwattana, Shan E Ahmed Raza, Yee-Wah Tsang, David RJ Snead, Ian A Cree, and Nasir M Rajpoot. Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images. *IEEE transactions on medical imaging*, 35(5):1196–1206, 2016.

[33] Dave Steinkraus, I Buck, and PY Simard. Using gpus for machine learning algorithms. In *Document Analysis and Recognition, 2005. Proceedings. Eighth International Conference on*, pages 1115–1120. IEEE, 2005.

[34] BWKP Stewart, Christopher P Wild, et al. World cancer report 2014. *Health*, 2017.

[35] Dayong Wang, Aditya Khosla, Rishab Gargeya, Humayun Irshad, and Andrew H Beck. Deep learning for identifying metastatic breast cancer. *arXiv preprint arXiv:1606.05718*, 2016.

[36] Haibo Wang, Angel Cruz-Roa, Ajay Basavanhally, Hannah Gilmore, Natalie Shih, Mike Feldman, John Tomaszewski, Fabio Gonzalez, and Anant Madabhushi. Mitosis detection in breast cancer pathology images by combining handcrafted and convolutional neural network features. *Journal of Medical Imaging*, 1(3):034003–034003, 2014.

[37] Yi-Ying Wang, Shao-Chien Chang, Li-Wha Wu, Sen-Tien Tsai, and Yung-Nien Sun. A color-based approach for automated segmentation in tumor tissue classification. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 6576–6579. IEEE, 2007.

[38] Martin Wiseman. The second world cancer research fund/american institute for cancer research expert report. food, nutrition, physical activity, and the prevention of cancer: a global perspective. *Proceedings of the Nutrition Society*, 67(03):253–256, 2008.

[39] Jun Xu, Xiaofei Luo, Guanhao Wang, Hannah Gilmore, and Anant Madabhushi. A deep convolutional neural network for segmenting and classifying epithelial and stromal regions in histopathological images. *Neurocomputing*, 191:214–223, 2016.

[40] Yan Xu, Zhipeng Jia, Yuqing Ai, Fang Zhang, Maode Lai, I Eric, and Chao Chang. Deep convolutional activation features for large scale brain tumor histopathology image classification and segmentation. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pages 947–951. IEEE, 2015.

[41] Wei Zhang, Maryellen L Giger, Yuzheng Wu, Robert M Nishikawa, Robert A Schmidt, et al. Computerized detection of clustered microcalcifications in digital mammograms using a shift-invariant artificial neural network. *Medical Physics*, 21(4):517–524, 1994.

[42] Wei Zhang, Akira Hasegawa, Kazuyoshi Itoh, and Yoshiki Ichioka. Image processing of human corneal endothelium based on a learning network. *Applied Optics*, 30(29):4211–4217, 1991.

Chapter 8

Appendices

8.1 Appendix A: materials provided

The following materials were provided to me either partially or entirely used, and are not the result of independent work:

- The basic implementation of the Reinhard colour transfer algorithm was taken from Adrian Rosebrock's open-source Github project, and adjusted to maximise efficiency (calculating statistics for the source image only once and turned into function form): `https://github.com/jrosebr1/color_transfer`
- The Hvass-Labs TensorFlow CNN for MNIST digit classification was used as a rough base for creating a 3-layer binary CNN classifier: `http://www.hvass-labs.org/`
- The public Cats vs Dogs dataset on Kaggle was used for practice training the CNN: `https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition/data`
- The following Tensorflow Docker image file was used as a base for creating a custom Docker container for use on ARC3: `https://hub.docker.com/r/tensorflow/tensorflow/`

8.2 Appendix B: ethical issues and risk addressed

The data used in this project has been fully anonymised and is legitimate. No attempt has been made during the course of this research to link images to patients. Image analysis work was pre-approved for this dataset, with ethics reference Leeds West LREC, 05-Q1205-220.

In the hypothetical situation that anyone were to misuse these experiment results in a clinical environment, and there were a misdiagnosis due to over-reliance on the algorithm results, patients could suffer psychological and physical damage. To this end, any use of this project in a clinical setting is not supported. The intention of this research is not to replace the expertise of a physician or pathologist. This project is not intended to apply to any kind of practical application.

8.3 Appendix C: complete experimental results

The following Table 8.1 displays the performance of all the CNN variations discussed in chapter 5. The best results in each category are in bold. The overall best performing model has no colour normalisation, three layers of 16 filters each, and filter window size of 8x8 pixels. The model names refer to the saved TensorFlow models available in the Github repository for re-running the CNN.

Number of filters	F-measure	Balanced accuracy	Error rate	Recall	Specificity	Precision	Model
16, 32, 32	0.679	0.723	0.271	0.605	0.841	0.774	model-2017-07-19-23:04
16, 16, 32	0.684	0.724	0.271	0.617	0.831	0.767	model-2017-07-20-14:18
16, 16, 16	0.741	0.769	0.240	0.779	0.76	0.753	model-2017-08-07-20:42
32, 32, 32	0.659	0.722	0.269	0.55	0.894	0.823	model-2017-07-20-20:31
32, 32, 64	0.638	0.712	0.278	0.517	0.907	0.833	model-2017-07-21-02:47

Number of layers	F-measure	Balanced accuracy	Error rate	Recall	Specificity	Precision	Model
1	0.593	0.668	0.323	0.497	0.838	0.735	model-2017-08-07-01:30
2	0.676	0.712	0.284	0.625	0.799	0.736	model-2017-07-23-18:51
3	0.741	0.769	0.240	0.779	0.76	0.753	model-2017-08-07-20:42
4	0.661	0.697	0.299	0.613	0.781	0.716	model-2017-07-24-02:36
5	0.617	0.686	0.306	0.519	0.853	0.761	model-2017-08-04-15:11

Filter size	F-measure	Balanced accuracy	Error rate	Recall	Specificity	Precision	Model
4, 4, 4	0.730	0.763	0.247	0.759	0.767	0.754	model-2017-08-05-01:39
6, 6, 6	0.694	0.735	0.259	0.619	0.850	0.788	model-2017-07-31-15:00
8, 8, 8	0.741	0.769	0.240	0.779	0.76	0.753	model-2017-08-07-20:42
10, 10, 8	0.674	0.715	0.28	0.611	0.818	0.752	model-2017-07-24-14:52
12, 12, 12	0.663	0.708	0.286	0.593	0.824	0.752	model-2017-07-29-18:27
14, 14, 14	0.601	0.677	0.314	0.499	0.854	0.755	model-2017-08-05-14:52

Colour	F-measure	Balanced accuracy	Error rate	Recall	Specificity	Precision	Model
normal	0.741	0.769	0.240	0.779	0.76	0.753	model-2017-08-07-20:42
Reinhard	0.669	0.72	0.273	0.584	0.855	0.784	model-2017-08-24-02:01
grayscale	0.617	0.686	0.306	0.519	0.853	0.761	model-2017-08-04-02:40

Table 8.1: Complete results for all experiments conducted with varying number of filters, layers, filter size, and colour normalisation; best parameters emphasised in bold.