Analysis of transfer learning approaches for patch-based tumor detection in Head and Neck pathology slides

Nishant Mishra, School of Computer Science

McGill University, Montreal

June 30, 2021

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of

Master of Computer Science

Abstract

Analysis of digitally scanned microscopic slides using machine learning is a rapidly advancing area of research that has the potential to enable faster identification, better localization, prediction of recurrence, and better decisions about the treatment of diseases, especially cancer. Several diverse techniques using computer vision and deep learning have been proposed and applied to histology images to perform a wide range of tasks such as classification, regression, and segmentation. Applying deep learning to digital histopathology comes with its own set of caveats. Most of these have to do with the high resolution of Whole Slide Images. Applying deep learning-based imaging on them is computationally intractable and resizing the slides to low resolution leads to loss of information. The primary objective of this thesis is to build a patch-based tumor detector or tissue classifier for head and neck pathology data and to analyze and benchmark state-of-the-art architectures in a fully supervised transfer learning paradigm for this task using a range of metrics. An end-to-end inference pipeline for tumor detection is also proposed that performs preprocessing, ROI segmentation, patch extraction, and prediction, returning whole slide coarse segmentation masks, and heatmaps for visualization. The slide-level tumor detection performance is also analyzed using an aggregation heuristic stacked on top of the patch tissue classifier. Various sampling strategies and coarse segmentation performance using metrics such as pixel-level IOU are also analyzed. The optimal performance obtained is an accuracy of 91% and an F1-score of 0.9 for unseen patch data and an accuracy of 89% for unseen data for Whole Slide classification into the tumor and non-tumor classes.

Acknowledgements

I am deeply grateful to my supervisors Peter Savadjiev and Reza Forghani for providing the opportunity to work under their guidance and for their invaluable supervision of my research and constant mentorship throughout my master's program. It would have been impossible to finish the thesis without their guidance and encouragement. I also extend my thanks to Peter for continuously monitoring, reviewing, and providing detailed feedback during the progress of this thesis. I thank Marc Philippe Pusztaszeri of Jewish General Hospital and McGill Department of Pathology for his help with the acquisition and expert annotation of the data used for this thesis without which the project would not have been possible. I extend my gratitude to lab members Farhad Maleki, and Nikesh Muthukrishnan for their constant help, discussions, and support with various aspects of the project. Farhad was a great help with his experience in the technical parts of the research through regular meetings. Finally, I am grateful to my parents, back in India, for their immense sacrifices and contributions to my life, for enabling me to pursue my masters and for standing like a rock behind me throughout and to my friends, especially Sweta, for their unconditional support whenever needed.

Table of Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

We are rapidly moving towards an increasingly automated society with more emphasis on human-machine and machine-machine interactions, that includes the medical science community. A large number of procedures from diagnosis to surgical procedures to drug discovery are now leveraging technologies like Artificial Intelligence, Machine Learning, Robotics, etc. with promising results. Medical experts, doctors, radiologists, and pathologists are highly trained professionals with vast in-domain knowledge involved in diagnosing diseases and their impacts on patients based on data obtained from patients. The AI-based techniques for diagnosis are being increasingly applied to medical imaging to assist radiologists and pathologists. Digital Pathology is a popular area in the field of medical imaging which has recently taken off with the advent of advanced digital microscopic scanners and novel computer vision algorithms to study and extract information from digitized scans.

Traditionally, histopathology slides are examined visually under the microscope by a trained pathologist who then makes a diagnosis. With the advent of high-resolution slide scanners, as well as with the advance in state-of-the-art computer vision and machine learning techniques, automated slide analysis known as digital pathology has become possible and more prevalent. Histopathological images have now become available in abundance to drive research related to automated analysis of these slides at scale.

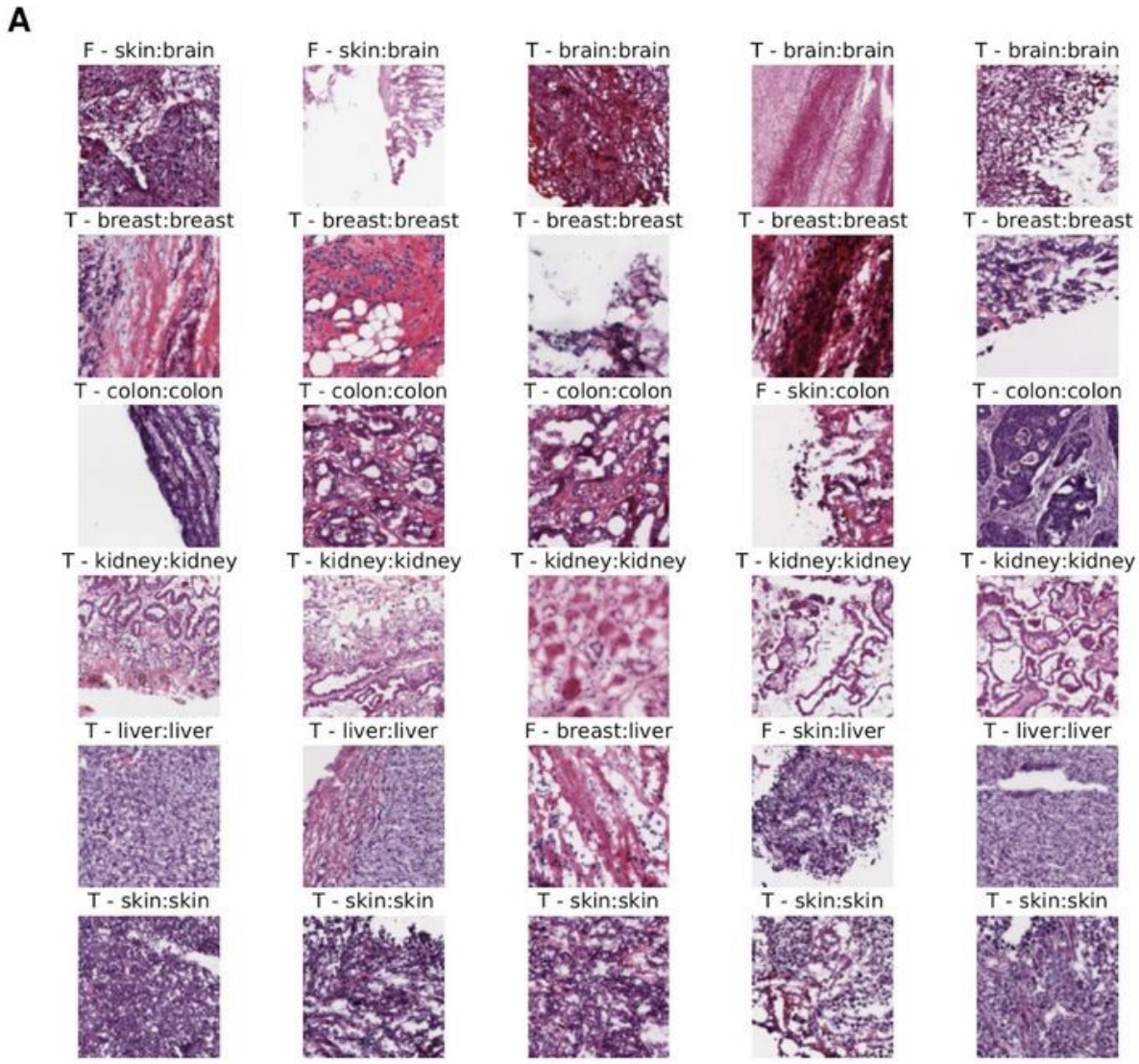

Figure 1.1: Whole Slide Image samples taken from publicly available The Cancer Genome Atlas (TCGA)[1], 5 from each of the following human tissues Brain, Breast, Colon, Kidney, Liver, and Skin

This automation helps speed up analysis for larger amounts of data. It is thus imperative to further explore the potential of Machine learning in providing reliable solutions for the automation of tasks such as diagnostics. Cancer prevalence and diagnosis have increased substantially in recent years. Accurate analysis, characterization, identification, and localization can go a long way in ensuring faster, more controlled treatment among patients. An automated biopsy analysis system can help pathologists by filtering and

pre-populating scans, improving efficiency and turnaround time. This not only reduces a lot of burden of redundant work from diagnosticians but also helps significantly improve the capacity and speed of hospitals and medical institutes in processing patients' data. Automation of diagnostics will also enhance the accuracy as it will help address large inter-observer variability arising from a subjective examination of the images. An automated, quantitative method can help overcome that and provide a more standardized reading.

The objective of this thesis is to not only theoretically explore the state of research in digital pathology, especially deep learning methods, but also to apply and analyze state-of-the-art deep learning algorithms for tumor detection in head and neck cancer histology data. It also aims to describe and build an efficient, high throughput pathology data preprocessing workflow that can be integrated with the training and inference pipelines.

1.1 Related Work

Medical Science, especially medical imaging has seen significant interest from the computational research community in recent times. It comes in the wake of a spike in data availability, computational resources, scanning devices, image analysis tools, and a growing necessity for interdisciplinary research using automated techniques for augmenting medical processes. This phenomenon has been especially marked in diagnostic systems for various diseases including cancer

Digital Pathology is no different, it is fast catching up in terms of the application of these novel methods for detection, classification, and prognostication of diseases in the fields of radiology[2, 3] and oncology[4, 5] with already well-established computational imaging practices. There has been extensive research going on for applying state-of-the-art imaging algorithms and learning techniques to scanned biopsy images for diagnosis, segmentation, and prognosis[6, 7]. This includes tissues from various regions of the body for tasks such as breast cancer detection(Mammography)[8, 9], cervical cancer anal-

ysis[10, 11], Squamous Cell Carcinoma[12, 13], Head and neck cancer detection[14], celiac disease analysis[15, 16], renal pathology[17], and many more.

Many recent works have been proposed that have proven to be highly successful in tasks relevant to digital pathology. They involved a variety of techniques including classical computer vision-based techniques as well as advanced deep representation learning-based ones. The primary tasks addressed are classification of diseases, segmentation of affected tissue regions, prediction of severity of diseases, prediction of treatment regime, etc. using different learning paradigms.

Several previous studies have implemented computer-assisted detection methods using histological images[18]. Colorectal epithelial and stromal tissues have been classified on histological images using support vector machines with hand-crafted features, such as color and texture[19, 20]. Barker et al.[21] in their 2014 work proposed a brain tumor type classification algorithm using local representative tiles. Their method utilizes a coarse-to-fine analysis of the localized characteristics in pathology images. There is an initial surveying stage for analyzing the diversity of coarse regions in the WSI that includes extraction of spatially localized features from tiled regions in the slide followed by dimensionality reduction of the features and clustering to create representative groups. A second stage provides a detailed analysis of a single representative tile from each group. An Elastic Net classifier is then used to produce a diagnostic decision value for each representative tile. A meta voting scheme then aggregates the decision values from these tiles by appropriately weighing them to obtain a diagnosis at the whole slide level. Their method achieved an accuracy of 93% for brain cancer classification into two possible types.

Additionally, convolutional neural networks (CNNs), which are a family of machine learning algorithms that learn to extract features from training images, have also been applied to classifying epithelium and stromal tissues from colorectal and breast cancers[22]. Several patch-based deep learning classification models have been used for various histology tasks. Patch-based CNNs have been used to classify Non-small cell lung cancers, including metastatic SCC to the lungs[23]. Another method for detecting lung cancers

in histological images of needle core biopsies used morphological and color features for classification with an ensemble of artificial neural networks[24]. Head and neck SCC has also been investigated but only in a couple of previous works, One such work was done for cell lines xenografted into mice, where a CNN was implemented with histological images to predict hypoxia of tumor-invaded microvessels[18, 25].

A more relevant and recent work for global classification by Halicek, Martin, et al.[14] performs patch-based localization and whole slide classification for Squamous Cell Carcinoma(SCC)(head) and Thyroid Cell Carcinoma(Neck) using CNN. It used the Inception-V4 model trained on coarsely annotated SCC and thyroid cancer datasets and tested for its performance on Thyroid Cell and Squamous Cell carcinoma detection as well as general cancer presence detection, it was also tested on the external CAMELYON 2016 Breast cancer lymph node metastases dataset. It used a patch-based approach training on 101 x 101 patches obtained using the coarse binary mask ground truth from downsampled WSIs.

Furthermore, computational imaging methods have also been developed for thyroid carcinomas to detect and classify malignant versus benign nuclei from thyroid nodules and carcinomas, including follicular and papillary thyroid carcinomas, on a cellular level with promising results[26, 27, 28]. However, most of the work related to thyroid carcinoma were done at a cellular or nuclear level leveraging hand-crafted features obtained using classical computer vision algorithms like texture or shape, and statistical learning algorithms such as support-vector-machines for nuclei classification, with a large number of algorithms using an ensemble-based approach such as boosting, bagging or stacking[27, 28, 29, 30, 31].

1.2 Challenges

Applying automated image analysis techniques for digital pathology comes with its own set of caveats and challenges[32]. Most of these have to do with the high resolution of

Whole Slide Images which are usually Gigapixel size images. They are computationally intractable and resizing them to low resolution leads to loss of information. Most of the research deals with this issue by breaking down the images into smaller tiles/patches for training and processing[7, 14, 21, 23]. Tiling/patch extraction breaks down the whole slide image into a set of sequentially extracted overlapping or non-overlapping patches of predefined sizes. This simplifies our problem to automated processing and designing algorithms for these individual patches instead of the original whole slide image. This reduces the processing cost and makes algorithms computationally tractable. Tiling also helps by essentially multiplying the amount of data we have for training deep learning models since it breaks down one slide into a large number of patches of different classes.

Most pathology slides are usually very high-resolution multi-level images. This poses a major challenge in terms of the computational tractability of algorithms designed for these slides. It is important to capture all the information present in a whole slide image without making the process too compute-intensive. Using lower-resolution versions of the slides for processing would lead to loss of sensitive information and in turn below par algorithms.

Patch-based approaches too have their issues. They do not capture visual context and require a lot of computation for whole slide-level tasks. Since in a patch-based approach we are concentrating only on a very specific patch of small size, we lose important contextual and spatial information present in the entire slide as well as the neighborhood of the patch in consideration. The context is very important in a variety of automated image analysis tasks such as classification, segmentation, generation, etc. Also, when we deal with specific patches in isolation we lose information pertaining to the relationship between different patches in a sub-region of the slide. This spatial relation between the different regions of a slide is usually very significant information for slide-level tasks. Under a specific patch size, patches drawn at a high magnification level lead to less contextual and spatial information whereas patches at lower magnification levels may not capture cell-level features

Since our whole slide images are massive images of gigapixel resolutions, breaking them down into small patches for faster processing results in a very large number of patches, which is a computational bottleneck when processing these patches sequentially. Aggregation of individual patch level predictions into slide level labels is also a major issue since it involves a lot of algorithmic design choices regarding how to best pool the patch level results to get a suitable slide level prediction. In the case of deep learning, major ways include soft and hard voting, average pooling, max pooling, and using linear models such as logistic regression, Support Vector Machines(SVMs), etc as meta learners, stacked on top of the patch-based deep learning model[14, 21, 23, 33, 34].

Another important challenge is the availability or lack thereof annotated data. Annotation of pathology slides requires medical professionals and expert pathologists. Deep learning-based computational pathology approaches either require manual annotation of gigapixel whole slide images (WSIs) in fully supervised settings or thousands of WSIs with slide-level labels in a weakly supervised setting and thus is a major bottleneck in training efficient algorithms. It is neither possible nor recommended for non-domain experts to annotate pathology slides to ensure our automated digital pathology algorithms don't learn spurious features that might make them unreliable. While the experts who can annotate are usually qualified doctors and pathologists who cannot dedicate too much time from their usual professions for data annotation. Also, the sensitivity of dealing with patients' medical data and the associated privacy-related norms limit the manpower that can be utilized for the task of data annotation or processing.

Another important caveat to take into account while dealing with pathology slides is that the regions of interest i.e tissues in these slides are sparse and far apart. Pixels representing tissue are often found grouped in small clusters embedded in large image regions with only uninformative, redundant background pixels. So a mechanism is needed that can isolate the regions of interest from the slide and strip off the background with high precision to not lose any important information. This will help speed up the analysis of slides, by bringing down the processing time for tiling, feature extraction, or prediction

since we only focus on relevant regions. This is usually done by a preliminary segmentation step that uses standard image processing algorithms leveraging thresholding based on slide properties such as color, intensity [35, 36, 7]. Deep Learning algorithms can also be used but they would require a large amount of training data and will have much higher latency and size. The tissue and background regions in pathology slides, especially H&E stained ones, are fairly easily separable based on color and intensity. Hence applying complex deep learning approaches for what is essentially a pre-processing step would be an algorithmic redundancy.

1.3 Method Overview

This project deals with the application of deep learning to digital pathology Slides for Head and Neck cancer detection. Essentially this can also be framed as a tissue classification problem, where we classify patches from the Whole Slide Images into the two classes: tumor and non-tumor. We leveraged state-of-the-art learning paradigms to experiment and benchmark their performance on head and neck cancer pathology data for both patch level and slide level aggregated tissue classification as well as for visual analysis of coarse segmentation using the same patch-based approach. This project involves a detailed study of the performance of the Transfer learning approach using several architectures trained on ImageNet[37] data for the tumor detection in head and neck tissue patches using fine ground truth annotations of tumor regions done at the pixel level. An entire experiment pipeline was developed that involved Data analysis, Image preprocessing such as denoising, normalization, tissue segmentation, and patch extraction, followed by data resampling, data augmentation, transfer learning, hyperparameter tuning, and finally evaluation on unseen data, whole slide level detection along with image reconstruction and output overlay.

We have used a patch extraction-based approach where we tiled our digitally scanned slides stained with hematoxylin and eosin. We trained state-of-the-art Convolutional

Neural Network(CNN) based architectures in a transfer learning setting[38] to classify tissue patches as containing Tumor or not. We benchmarked the performance with different architectures, sampling strategies, hyperparameters using various metrics such as accuracy, precision, recall, and F1 score for classification and Intersection over Union (IOU) for segmentation performance.

We trained on nearly 500 whole slide images obtained from 46 patients which produced $3*10^6$ non-tumor and 44,000 tumor patches. The Whole Slide Images were first preprocessed using standard image processing algorithms including Morphological Operations, thresholding, contour detection for tissue segmentation followed by patch extraction from these contours in a customized way for our purpose. These patches were then used for training ConvNets in a supervised setting to minimize classification error. We trained various state-of-the-art architectures for benchmarking and comparison. These were Residual Network(ResNet-34 and ResNet-50)[39] and EfficientNet (EfficientNet-B2)[40]. We tested our models on $5 * 10^5$ unseen patches and registered an accuracy of as high as 95% and an F1 score of 0.93 on validation data and accuracy of 91% and F1-score 0.9 on unseen patches. Once we had benchmarked different models for their performance on patch level classification, a metaheuristic was used to aggregate patch level predictions to generate slide level labels. The final slide level detection performance based on this heuristic stacked on top of the patch-classification model was measured. We also generated binary and overlaid prediction/localisation masks for each slide along with probability heatmaps for each slide using the patch level classifier which gave us a visual equivalent of coarse segmentation masks. Each patch was assigned as containing a tumor or not by thresholding the final activation values generated by the patch classifier. We then analyzed these coarse segmentations visually and by calculating Intersection over Union between the generated classifier mask and the initial tumor contour mask.

1.4 Outline of Thesis

This thesis is arranged as a series of Chapters. The introduction above was our first chapter which laid down a short but comprehensive overview of the project including the challenges, method overview, and problem introduction. Chapter 2 of the thesis is the Technical Background. The background chapter comprises sections explaining the necessary theoretical background for the project. They are arranged in terms of their application to the project as well as a general technical hierarchy of methods used for digital pathology. The theoretical descriptions also include relevant literature reviews as and when required. The background is followed in Chapter 3 by the methodology chapter where we discuss the overall methodology in much greater detail and the proper sequence of steps involved in the project. Apart from a detailed description of the steps involved in the project, we also lay out our entire experimental setup. Chapter 4 deals with a well-defined tabulation, visualization, summarization, and discussion of the results of our experiments. The last chapter, Chapter 5 concluded the thesis with general observation, conclusion, and discussion of future research directions relevant to this project.

1.5 Contribution

The main contributions of this thesis have been summarized below

- This thesis explores the efficacy of state-of-the-art Convolutional Neural Network models in a transfer learning paradigm for the detection of tumors in Head and Neck histology data.
- The work is among a rare few to study the performance of deep learning-based vision models for patch-based tumor detection in head and neck tissue samples in a fully supervised setting. Multiple CNN architectures such as ResNet-34, ResNet-50, and EfficientNet-B5 are trained, analyzed, and benchmarked for the classification of tissue patches into two classes, tumor and non-tumor.

- A high throughput, efficient, and reliable computer vision pipeline was implemented and used for preprocessing large giga-pixel sized Whole Slide Images for integrated noise removal, ROI segmentation, and patch extraction.
- Slide-level classification using a meta aggregation heuristic on top of patch classification was also implemented and benchmarked.
- Extensive analytic and comparative study of the different architectures, sampling strategies was done using an array of metrics such as F1 score, classification accuracy, IOU for coarse segmentation.
- An end-to-end workflow for inference was created that includes preprocessing such as tissue segmentation and patch extraction, classification using the trained model, and outputs not only the evaluation metrics but also visualizations of results using binary prediction masks, heatmaps, and overlaid WSIs.

Chapter 2

Background

In this section, we shall be studying the theoretical aspects of the various phases involved in Digital Pathology in general as well as concepts involved in the various steps of this project in particular. The discussion is divided across two implicit verticals: One based on the various tasks that can be performed on digital whole slide images, and the other based on the different techniques that can be applied to perform these tasks.

On a more abstract level, the different phases of a digital pathology workflow include a Pre-analytical phase that deals with the acquisition of tissue data(biopsy) from patients, slicing and preparation of these tissue samples to be put on glass slides for observation using glass slide microscopes, the analytical phase which deals with digitization of the whole slide images for storage and computation, as well as the post-analytical phase that encompasses computational analyses, data extraction, integration, and final output.

The finer workings of the computational image analysis of pathology can be divided into three main parts:

- ***Data Acquisition and Preparation*** This involves the process of acquiring data, storing them, visualizing, programmatically accessing, analyzing, removing outliers, and preparing them for the subsequent data pre-processing.
- ***Data Pre-processing*** In this step we prepare our data for being fed to the feature extraction or learning stages of the algorithm. This could involve standard prac-

tices such as Image normalization, denoising, artifact removal, tissue segmentation, patch extraction, graph construction, etc

- ***Learning*** This is the most important part of the image analysis. Here we use computer vision algorithms for modeling our data. Then we use further computer vision, machine learning, or deep learning algorithms for feature engineering and to learn useful representations from the model of the image content to predict some outcome of interest or for any other downstream task. So we can either use computer vision algorithms exclusively or use classical computer vision algorithms in combination with various learning algorithms.

We discuss the various pre-processing techniques, classical computer vision and feature engineering-based methods, the various deep learning-based algorithms, and their applications to classification and segmentation tasks in more general technical detail in the upcoming sections of this chapter. The various Learning Schemes or paradigms of Deep Learning that are popularly applied in the context of computational pathology which we shall discuss are outlined in the Figure 2.1. Based on these, various DL mod-

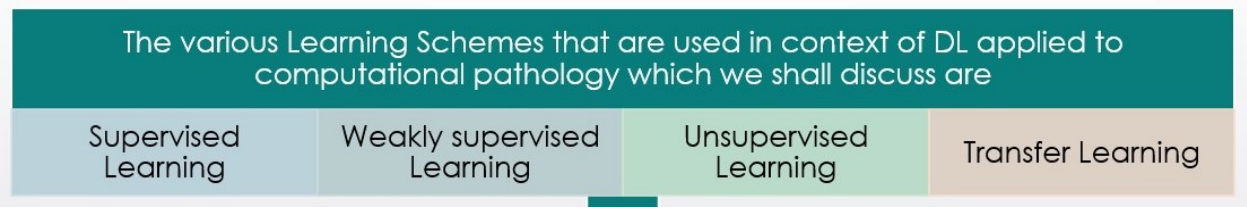

Figure 2.1: Four popular learning schemes applied to computational pathology

els have been applied, which are traditionally based on convolutional neural networks (CNNs), recurrent neural networks (RNNs), generative adversarial networks (GANs), auto-encoders(AEs), and various other variants.

Popular and relevant recent literature and state-of-the-art methods in the domain of digital pathology have also been discussed as and when required to consolidate the explanation of the various technical concepts or to visualize the possible applications to various steps in the digital pathology pipeline. But before that, the next section contains

a detailed definition of Digital pathology as well as a detailed description of its various phases.

2.1 Digital Pathology

Histology[41] is the branch of biology dealing with the microscopic structure of biological tissues. Histopathology[42, 43, 44] is a sub-domain of histology that deals with the diagnosis and study of diseases by examining tissues under a microscope to record changes caused by diseases such as cancer. Trained experts known as pathologists examine microscopic slides of patients' tissues in the affected region scanned under a microscope not only to identify and localize diseased regions but also to define the severity and layout of the prognosis or the treatment regime. It remains one of the most important clinical practices for diagnosing a variety of diseases such as cancer, Celiac disease in different parts of the body. The process by which pathologists take a small sample of tissues for further examination is known as a biopsy. Hematoxylin-Eosin (H&E) stained whole slides have been used by pathologists for over a hundred years now for cancer detection and prognosis. With such long history and proven applicability, histopathological imaging is a popular research paradigm for medical practitioners and is expected to stay among common clinical practices in the coming years.

The advent of ubiquitous whole slide digital imaging systems and developments in computational image analysis techniques have driven the rise of a new domain called Digital Pathology[45, 6]. Digital Pathology can be thought of as the convergence of digital clinical workflows such as virtual microscopy with imaging techniques with the goal of creating integrated computational systems in order to acquire, interpret and process pathology data. It's a departure from earlier where only the conventionally trained pathologists using light microscopy would analyze tissue slides. Now this entire digital environment where we use modern new techniques on digitally acquired slides enables

expedited clinical workflows and the development of new image analysis tools for pathology data.

Moreover, the application of machine learning algorithms has also led to new tissue investigation techniques and unique feature extractions which were erstwhile, not possible with physical interrogation as well as standard imaging techniques which have pushed the boundary of medical sciences in general. Hence digital pathology is a rapidly growing area of research and has fast reached applicability and accuracy levels to allow clinical deployment. Since 2017, the FDA has approved the use of commercial WSI platforms, facilitating the use of digital pathology as a tool for primary diagnosis. WSIs, which can now be scanned in less than a minute, can serve as an effective surrogate for traditional microscopy-based pathology The important analytical phases and their corresponding steps for Digital Pathology as shown in Figure 2.2 are:

- ***Pre-analytical Phase*** This phase encompasses the procurement step as well as parts of the histology step. It involves steps that are involved in tissue procurement and processing. It essentially relies on electron or glass slide microscopy and other advanced optical imaging technologies.

- ***Analytical Phase*** The analytical phase also involves histology steps such as selection of the stain to be used, optimization and validation of the staining procedure, etc. as well as digitization steps for scanning the whole slide images and proper curation of the dataset. This phase depends heavily on high-quality digital scanners capable of automatically producing very high-resolution images of the glass slides, and modern software systems capable of WSI data visualization, storage, and overall management.

- ***Post-analytical Phase*** The analytical phase involves extraction, analysis, and interpretation of results. Here we use the computer vision algorithms, both classical or modern deep learning-based, for modeling or simplifying our data by feature engineering or representation learning, followed usually by Machine Learning or

state-of-the-art Deep Learning algorithms or some combination of them to learn from these features and representations for specific tasks such as classification, prediction, segmentation, etc. It can also be accomplished using purely human visual analysis. Data extracted from the images using human and machine vision are integrated and reported.

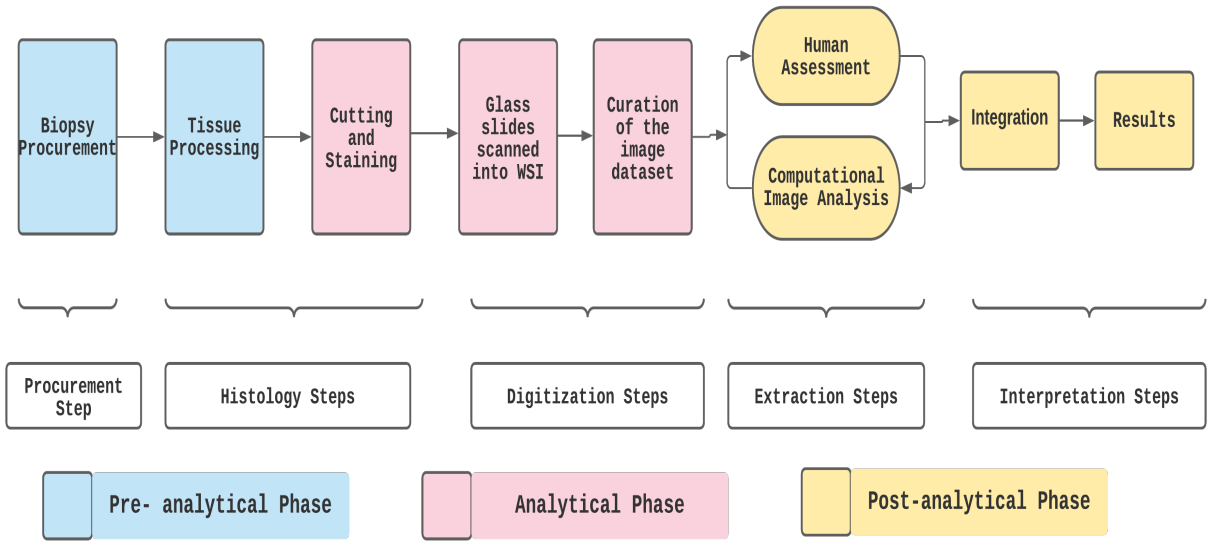

Figure 2.2: Analytical Phases of Digital Pathology, adapted from [46]

2.2 Procurement, Histology, and Digitization

Diving deeper to lay out the technical details of the Digital Pathology workflow requires that we discuss in appropriate detail all the primary steps leading up to the computational imaging are done. This includes acquisition/procurement, processing, visualization, staining, scanning of tissue samples.

In traditional pathology, the tissue sample to be examined is usually obtained through biopsy, autopsy, or operation. It is then processed by *fixation*, *wax embedding* before being cut into very thin slices of 2-5 μ width. These slices are then stained before being

transferred to a glass slide for being examined by pathologists using a light microscope. 12

The first step in tissue processing i.e fixating is done using a fixative solution. It is done to prevent tissue degradation by preserving its original state. This is followed by embedding it using paraffin wax[47]. This helps as the embedding agent infiltrates the specimen to provide a support matrix to toughen it up. This firming up allows very thin sectioning/slicing which is done using a microtome. The tissue is sliced into sections thin enough to allow light to pass through.

Histochemical staining[48] (also simply called staining) is done to provide contrast to tissue sections, making tissue structures better visible and easier to evaluate. Since tissues are normally colorless, applying a dye to the tissue section allows the cells and their components to be seen under a microscope. The most common technique used is the hematoxylin and eosin (H&E) stain. Other staining techniques such as Masson trichrome, alcian blue, reticulin staining, HER2, Ki-67, and others are sometimes used to demonstrate specific tissue components not captured by a H&E stain. H&E is the combination of two histological stains: hematoxylin and eosin. The hematoxylin stains cell nuclei a purplish blue, and eosin stains the extracellular matrix and cytoplasm pink, with other structures taking on different shades, hues, and combinations of these colors.

The common microscopy techniques involved in pathology are Optic aka Light microscopy and Electron microscopy. Optic microscopy uses light from the visible spectrum and combines it with multiple lenses to create a magnified image. The magnifying power of the objective (4x, 10x, 20x, 40x, or 100x) gets multiplied by the power of the ocular lenses (10x). Since tissues are relatively colorless, the magnifying properties of the optic microscope are not sufficient for proper visualization of a specimen, therefore necessitating the staining techniques described above.

Electron microscopy (EM) is a more modern form of microscopy. EM works by emitting parallel beams of electrons onto the tissue sample. It provides for a much higher magnification as well as higher resolution images. There are two types of EM: trans-

mission electron microscopy, which requires very thin sections of tissue, and scanning electron microscopy, which uses larger pieces of tissue and produces 3-dimensional images.

Digital pathology involves all of the same steps with the addition of methods for digitization, visualization, and storage of these tissue samples. The digitization is usually done using highly advanced optical scanners. It involves scanning glass slides to produce digital slides. Modern whole-slide scanners are capable of automatically producing very high-resolution images that replicate glass slides. Its also sometimes referred to as virtual microscopy. Currently, there is a plethora of high throughput, high precision whole slide scanners such as Aperio or InterScope. They use a multi-sensor array approach where a microscope objective lens projects an image onto multiple sensors, allowing concurrent image acquisition and data processing. These instruments typically contain a microscope with one or more objective lenses, digital cameras, robotics, and numerous other parts. They also come with state-of-the-art optic, robotic, and computing components such as bar code scanners, sensors, cartridges.

Not just scanning, Whole slide imaging requires highly capable software systems for visualization, processing, storage, and management of the scanned slides. These are referred to as virtual microscopy software. A large number of both active open-source WSI systems, as well vendor-owned proprietary whole-slide software systems are available currently driving research in digital pathology. Most of these Virtual Microscope software systems make use of image pyramid-based data management systems to support multi-resolution pan-and-zoom operations. The scanned Whole Slide Images are commonly encoded and stored in '.SVS' or (less frequently) '.tiff' formats.

These contemporary WSI systems include two integrated components (1) the scanner that handles image acquisition and (2) the workstation that includes a monitor as shown in Figure 2.3 depicting their workflow. Some examples of such systems are caMicroscope, the Digital Slide Archive, Aperio ImageScope, and QuPath. There also exists an increasing number of libraries such as OpenSlide[50], slideIO, HistomicsTK[51] in differ-

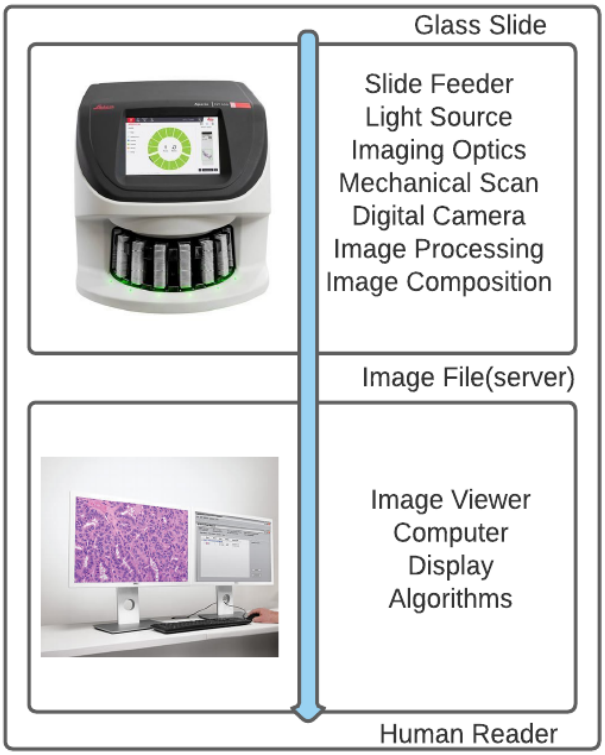

Figure 2.3: Overview of a Whole Slide Imaging system comprising of a scanner and integrated workstation along with typical sequence of steps, adapted from [49]

ent programming languages that are making it convenient to visualize, process, convert, perform image analysis and develop algorithms around them.

2.3 Computational Image Analysis

2.3.1 Whole Slide Image Preprocessing

While the most important part of the Computational imaging-based analysis of Pathology slides is the feature extraction and modeling techniques that allow us to perform prediction using our data, Image pre-processing is an essential step before that. Since the scanned Whole Slide Images are color images, it is imperative to perform certain color and texture-based preprocessing on them to enhance various color features for better feature extraction and to effectively exploit the color and spectral information present in the

images. These steps help improve the visibility, saturation, and overall representative ability of the images. Suitable usage of various combinations of preprocessing techniques supports and enables highly effective image analysis.

Pre-processing also helps make the imaging process less sensitive to extrinsic factors such as imaging conditions and scanner variations. Thus preprocessing is necessary to guarantee stability in image quality as well as in the performance of imaging algorithms. Pre-processing also helps segment and remove noise and other artifacts which might be present on the scanned image before processing. Typically, a large portion of a slide isn't useful, such as the background, shadows, water, smudges, and pen marks. Preprocessing can help rapidly reduce the quantity and increase the quality of the image data to be analyzed. When it comes to preprocessing there are no standardized workflows and this leads to significant variations and design decisions. There are a number of preprocessing methods[52, 53] that can be applied to WSIs, an important and non-exhaustive list of preprocessing steps includes color Calibration and enhancement, Colour deconvolution and color normalization, morphology, color thresholding, tiling, ROI segmentation, etc.

Colour calibration[54] aims to ensure accurate color information has been recorded and displayed, which can involve calibrating the intrinsics of a camera or the scanners, overcorrection of displayed colors Colour enhancement is useful in order to obtain optimal performance of image analysis algorithms. This is done to ensure robustness to imaging conditions and scanner variations. It involves techniques like Contrast enhancement, hue and saturation adjustment, interconversion among different color spaces such as RGB, YUV, YCbCr, HSV, etc that in turn can help improve e.g. segmentation or other subsequent tasks. Colour deconvolution allows us to effectively separate the contributions of stains localized in the same area and thus allows analysis of stain specific images,

Colour Normalization[55, 54] helps address the variations in color appearance of histopathology images due to scanner characteristics, chemical coloring concentrations, or different protocols. Color normalization is an image processing tool that has been developed to mitigate the effects of color variation by transforming the color properties of an image to

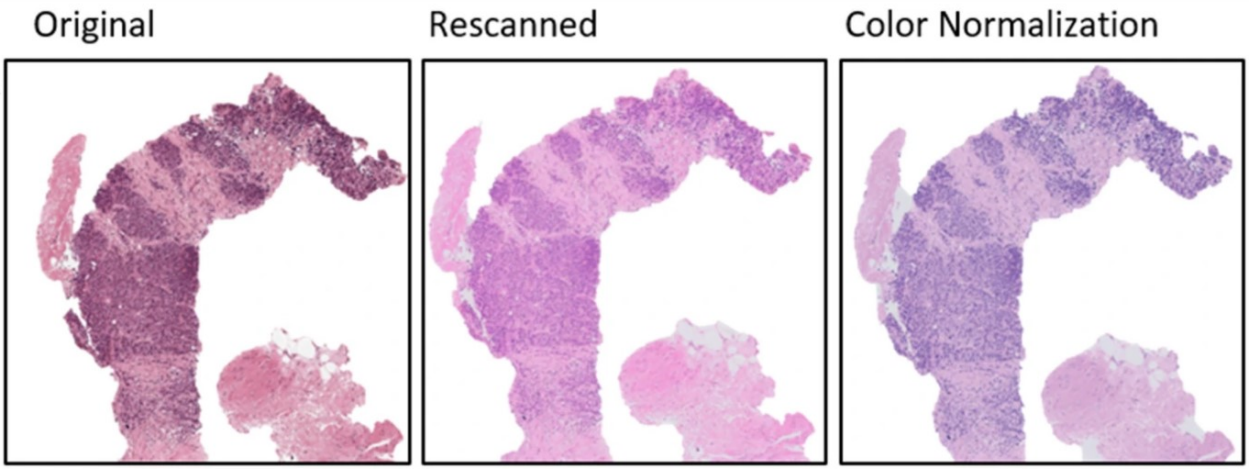

Figure 2.4: Figure depicting variations in Prostate tissue WSIs caused due to rescanning i.e scanner calibration and colour normalization [54]

align to a single standard. It helps reduce the staining variations and ensures all images adhere to a certain color standard thus helping in visualization. Several different color normalization approaches[56, 57, 58] have been developed that utilize standard image processing techniques like intensity thresholding, histogram normalization, stain separation, color deconvolution, and structure-based color classification.

In many cases, preprocessing also involves addressing the major disadvantages when dealing with Whole Slide Images as discussed in Section 1.2(Challenges). These are the huge gigapixel size of these images as well as the sparsity of these images. This means that large areas of pathology slides contain redundant background pixels which is a crippling computational overhead in downstream tasks. In order to help with high throughput, efficient imaging, we need to process WSIs to handle these issues. This involves tiling or patch extraction which is essentially breaking down the Whole slide image into smaller, more tractable sub-regions of a predefined size. The patch extraction can be performed on all the available resolutions of the slide and is usually done at their native resolutions.

It is also important to get rid of the redundant background information and segment only the tissue regions as our regions of interest. This is done through a series of image

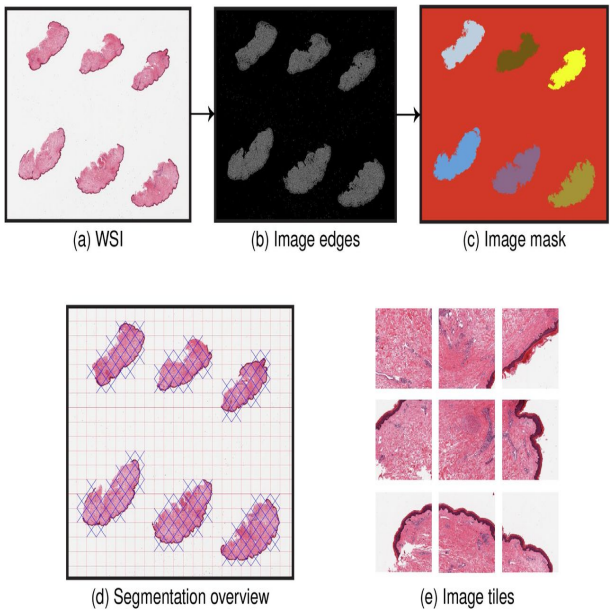

Figure 2.5: An example of a WSI preprocessing pipeline where a skin tissue sample goes through various steps such as edge detection, graph segmentation and tiling to remove background information and generate tiles from tissue regions.[61]

transformations to identify the foreground from the background, pass only foreground regions for further processing. The segmentation[59, 60] of the tissue region can be done by exploiting the large differences in colors of tissue and non-tissue region. This involves using image filters and other image processing techniques such as color space transformation, color thresholding, edge detection, contour detection, etc for segmentation of tissue regions. There is no standardized way and this pipeline involves a number of tunable parameters which affect the accuracy of segmentation. We utilized one such pipeline that was fairly accurate and generalizable for our experiments which is explained in Chapter 3.

2.3.2 Feature Extraction, Modeling and Learning

This is essentially the most important step in the computational imaging of digitized whole-slide images. In this step, we extract features and representations from patches or

Whole Slide Images depicting various properties of the different pathology slides which are utilized for downstream tasks such as prediction, segmentation, diagnosis, etc.

Quantitative feature modeling for tissue classification in the context of digital pathology is divided into two categories – ***handcrafted features*** that can be correlated with specific measurable attributes in the image and have some degree of interpretability, and unsupervised feature-based approaches such as ***deep learning-based methods*** which are less intuitive and rely on filter responses learned from large numbers of training data to characterize image appearance. There are a number of trade-offs to be made during the algorithmic decision-making process. While the former has the advantage of greater interpretability and control, the deep learning methods have proven to be highly successful in capturing feature combinations, spatial correlations that are difficult to be comprehended at a human level using primitive methods even by trained pathologists. Given substantial data, the deep learning models learn highly abstract representations and complex features which are also highly generalizable. Then again, handcrafted features are much less computationally expensive, easier to guide, and efficient while the deep learning-based methods require large amounts of carefully annotated data, as well as significant, compute resources. We discuss both these paradigms in greater detail in the following sections.

2.3.3 Handcrafted Features based Modeling

This section essentially deals with the classical computer vision-based techniques that have been developed over the years to perform digital pathology. Before the advances in computational capacity and learning-based algorithms which could learn abstract representations as an optimization problem, researchers working on Whole Slide Imaging would rely on handcrafted and engineered image features for diagnostic tasks such as segmentation, labeling, tissue classification, disease grading, and in precision medicine.

There have been a number of recent advances in handcrafted and unsupervised feature analysis approaches for use in digital pathology some of which also take inspiration

from algorithms developed for addressing radiomics problems. There are emerging research areas that involve the fusion of these computer extracted tissue biomarkers with genomic and proteomic measurements in a multimodal setting for improved prediction of disease aggressiveness and patient outcome.

The traditional approach has been to leverage algorithms for extracting imaging features. These features can be broadly classified as being pixel-level, object-level, or semantic-level based.

Pixel-level features are lowest in the information hierarchy, such as mathematical characterizations of color, texture, and spatial patterns. A small subset of specific pixel-level features includes Gray-level intensity profiles, Haralick Gray-level co-occurrence matrix features, wavelets, Gabor filter responses, and statistics and frequencies of color histograms[62, 63, 64]. Linder et al. (2012) [19, 65] employed a texture approach for segmentation of epithelium and stromal tissue partitions. They successfully used local binary patterns (LBPs) as features for the differentiation of stroma and epithelium with an area under the ROC curve of 0.995.

Object-level features help describe the characteristics of objects such as nuclei, nucleoli, and mitoses, as well as more complex aggregates of multicellular, histologic structures such as crypts, ducts, and blood vessels. They are, therefore, considered higher in the information hierarchy. We need segmentation of structures in our data in order to characterize these object-level features in a detailed manner. There is a significant body of literature dedicated to the documentation of methods developed for identification and segmentation of nuclei[66, 67] and more complex aggregate structures as well as for object detection, to identify the location or relative positions of different structures.

Madabhushi et al. [68] used cytoplasmic and stromal features to automatically segment glands in prostate histopathology. In [68] and [67], nuclear segmentation from breast and prostate cancer histopathology respectively was achieved by integrating a Bayesian classifier driven by image color and image texture and a shape-based template matching algorithm

Semantic-level features help to capture the biological classification of histological structures or regions. They help describe high-level concepts such as type of cell (e.g., epithelial or endothelial), presence of lymphocytes, and necrosis. Kothari et al. [69] describe semantic-level features as being histological objects with descriptive labels We can also use semantic information for classifying WSI regions. For example, we can classify regions within a WSI as belonging to a tumor region or non-tumor region, or even benign and malignant tumor regions.

Qaiser et al [70] in 2016 in their paper used Persistent Homology Profiles as distinguishing features in order to segment tumor regions in Hematoxylin & Eosin (H&E) stained colorectal cancer histology whole slide images (WSI) by classifying patches as tumor regions or normal ones. Persistent Homology Profiles are algebraic features that exploit the fact that nuclei in tumor regions have atypical characteristics such as nonuniform chromatin texture, irregularity in shape and size, and clustering of nuclei to distinguish tumor regions.

2.3.4 Deep Learning

Figure 2.6 gives a very detailed overview of the current state of research when it comes to the different Machine Learning Schemes applied for Digital Histopathology. They are broadly defined into four different paradigms i.e Supervised Learning, Weakly Supervised Learning, Unsupervised Learning, and Transfer Learning. The corresponding applications and specific deep learning architectures within the different learning schemes have also been clearly outlined in the figure.

A brief theory of deep learning applied to Digital Pathology along with relevant literature references has been discussed in the following sections divided based on the above mentioned four schemes. This section is largely based on the comprehensive survey paper titled *"Deep neural network models for computational histopathology: A survey"*[7] by Srinidhi et al.

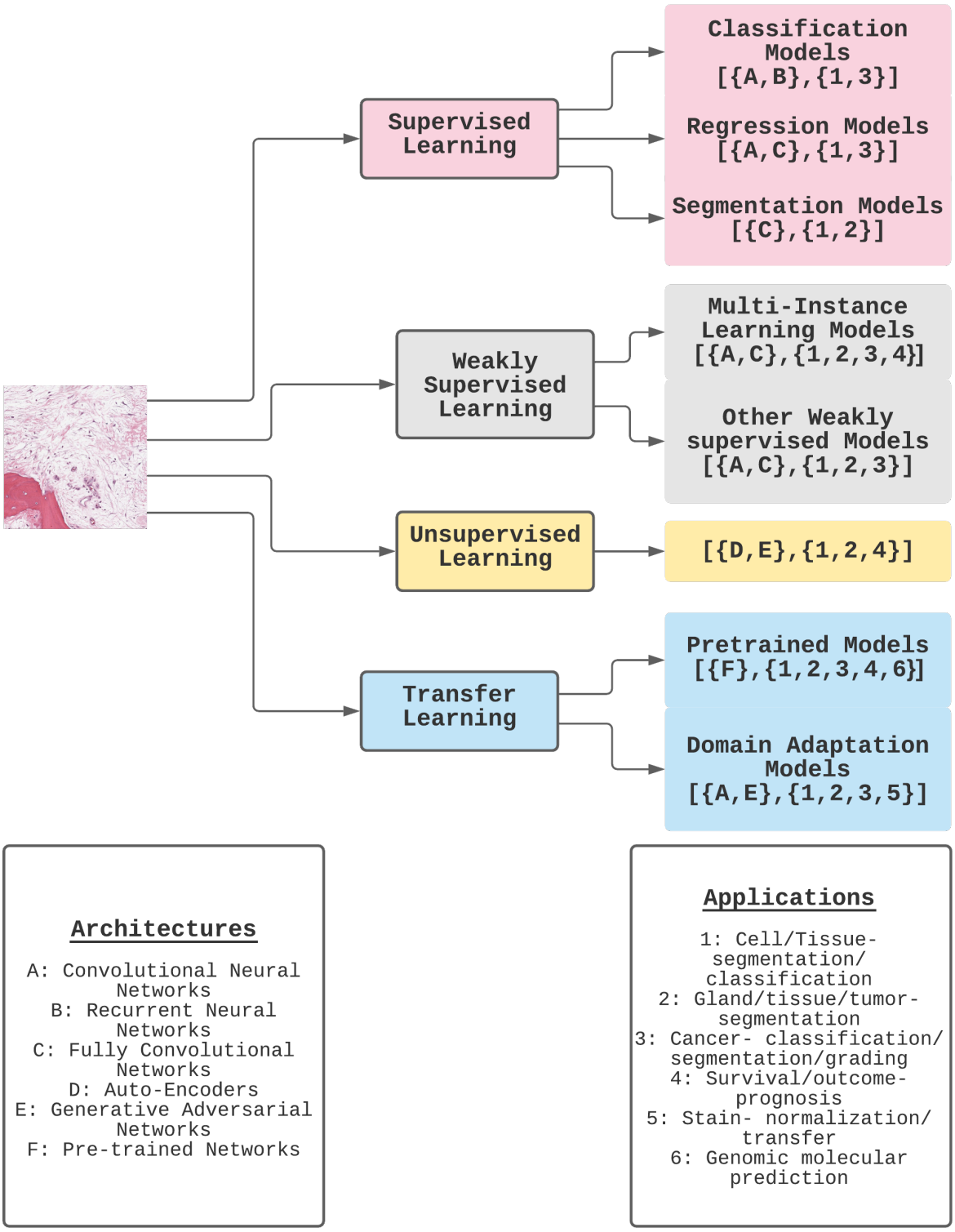

Figure 2.6: An overview of the different deep learning schemes used for digital pathology along with popular architectures and corresponding applications, adapted from [7]

Supervised Learning

Supervised Learning is a learning approach where the learning algorithm is provided with large amounts of labeled training data that contains both the data and its corresponding ground truth labels. So essentially the data is in the form of an ordered pair. Thus for N training samples we have of $\{x^i, y^i\} \ \forall \ i \in N$. The goal is to train a model that best learns to model the output given a new input based on a loss function.

The supervised learning-based models take three different canonical forms based on the problem they are solving, viz. classification which usually deals with tasks such as classification or object detection at a slide or patch level using a pixel-wise or sliding window-based prediction, regression which involves learning continuous-valued outputs and thus helps learning tasks such as prediction of severity of disease, or pixel-wise scoring for predicting the location, and segmentation which uses an encoder-decoder based fully convolutional network-based approaches for semantic and instance-level segmentation.

The supervised learning models for classification were identified as performing both local-level tasks and global-level tasks. Local-level classification models deal with the effective localization of objects in a region such as a cell or nuclei by using spatially pooled feature maps. CNN-based models have been really successful at pixel-wise prediction using a sliding window approach over image patches that are annotated by pathologists as regions containing objects of interest(cells/nuclei) or background. Most of these methods have a clear parallel to object detection algorithms meant for natural images [71]. Cirecsan et al. [8] was one of the most significant early successes in this domain in 2013, where they applied a CNN-based pixel prediction to detect mitosis in routinely stained H&E breast cancer histology images. Another popular and fairly successful set of work for these tasks involved hybrid approaches combining CNN extracted features with biologically interpretable handcrafted features, the likes of which we discussed in the previous section, which helped to reduce the requirement of a huge dataset for training CNNs.[72, 73, 74, 75]

Specialized research solutions like Multi-scale CNNs[76] and CNNs with data-specific stain augmentation[55] have also been proposed in several works in order to alleviate the issues caused by loss of information due to working with small localized windows and to deal with the loss of generalizability due to stain variations respectively. All in all, CNNs are the gold standard deep learning techniques when it comes to local histopathology tasks.

In global level classification, where we work to perform classification, grading at the slide level, most of the published work focuses on a patch-based classification approach for whole-slide level disease prediction tasks[77]. It can involve both patch level localization as well as whole slide level classification or grading of disease. Most works pertaining to these problems used simple CNN-based models. Cruz-Roa et al. [78] for example in 2014 proposed a simple 3-layer CNN for identifying invasive ductal carcinoma is breast cancer images which outperformed all the previous handcrafted methods by a margin of 5%. These methods and the global level tasks, in general, suffer from the issue of relatively long computational time required to carry out a dense patch-wise prediction over an entire WSI. Many new techniques are increasingly being utilized to address these problems. These include Monte-Carlo-based adaptive sampling strategy to focus only on highly relevant regions [78], task-driven visual attention models meant to selectively focus on the most diagnostically significant regions[79, 80, 81]. In fact, attention mechanism is rapidly becoming the method of choice. Xu et al.(2019) [80] proposed recurrent attention mechanisms to selectively attend and classify the most discriminate regions in WSI for breast cancer prediction. while Zhang et al. [77] (2019) proposed an attention-based multimodal framework to automatically generate clinical diagnostic descriptions and tissue localization attention maps, mimicking the pathologist which was inspired by solutions for visual question answering problems.

Regression models on the other hand solve the identification and localization of objects by directly regressing continuous-valued outputs for each pixel that gives a probability score of it being the center of an object. The reason it can potentially outperform simple

pixel-level classifiers is that it allows us to enforce topological constraints such as assigning higher scores to pixels near the center of an object than those farther from it thus taking a much more statistically nuanced approach. They use continuous-valued distance-based loss functions like L1, L2, and mean squared losses instead of cross-entropy loss.

These deep regression models are mostly based on CNNs and FCNs. Beyond the earlier iterations of simple FCN based models for regression, [82, 83], the more recent methods use modified loss functions as well as additional features for better performance. This includes a structured regression model proposed by Xie et al. [84] based on fully residual convolutional networks for detecting cells in four different tissue images that used a weighted MSE loss function with higher loss assigned to misclassified pixels nearer to the cell centers.

Another seminal work popularly known as HoVer-Net, short for Horizontal Vertical Network proposed by Graham et al. in 2019 [85] proposed a unified FCN model for simultaneous nuclear instance segmentation and classification. The proposed model effectively encodes both relative horizontal and vertical distance information of each nuclei pixel from their respective centers of masses as a number in -1,1. This information helps in accurate nuclei separation in multi-tissue histology images especially in case of occlusion or overlap.

Segmentation models aim to perform semantic segmentation of each pixel identifying them on the basis of the cells or nuclei to which they belong. Most segmentation models use an encoder-decoder architecture such as U-Net or FCNs. Patch-based sliding window approaches although effective is very computationally expensive. Common applications are the nucleus, gland or duct [86, 87, 88] segmentation. Swiderska-Chadaj et al.[89] did a comparative analysis of both FCN based segmentation model, de Bel et al.[90] and UNet architecture, Ronneberger et al.[91] and found U-Net to be more robust and generalizable owing to the upsampling path in its decoder as well as extra skip connections, thus yielding more accurate better quality segmentation maps.

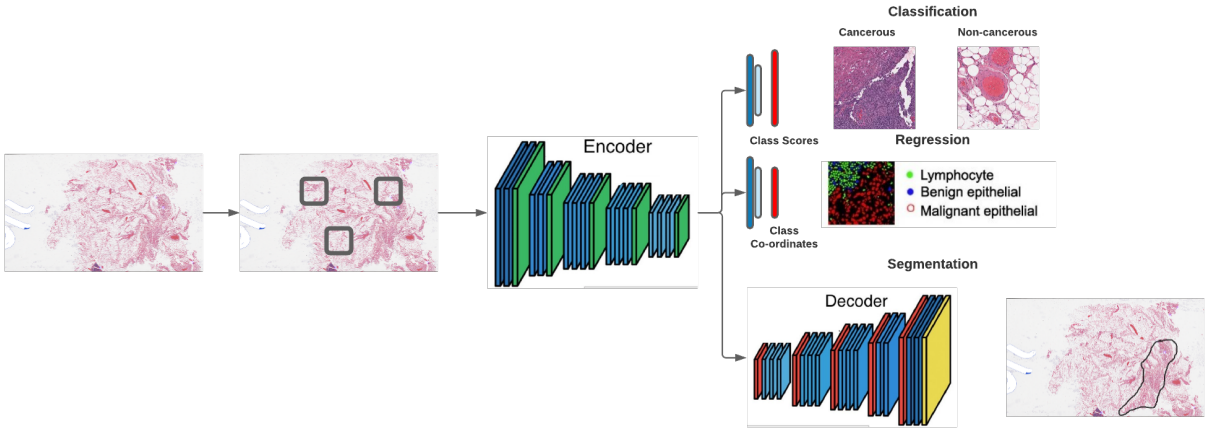

Figure 2.7: An overview of supervised learning methods

One very recent work that is worth highlighting is ***SlideGraph***[92] 2020 that devised a way to work around the computational intractability of using whole slide images. They used a method where the entire whole slide image is modeled into a graph which is a highly memory-efficient data structure before being fed to a Graph Neural Network model for training. Instead of extracting small patches from the WSI and doing analysis on a limited visual field for prediction, a pipeline is used which constructs a graph from the nuclei-level to the entire WSI-level.

It makes use of the seminal regression model HoVer-Net [85], that we discussed above for localizing and classifying the individual nuclei. Spatial clustering is then used to group together nuclei that share common properties. Each of the tissue clusters is then modeled as a node in a graph, with edges of the graph capturing the possible tissue signaling mechanisms. Once the graph construction is done, the task of WSI-level prediction can be modeled as a graph classification problem. A graph convolutional neural network(GCN) [93] with graph isomorphic network convolutional (GIN-Conv) layers can then be used for further training and downstream tasks. The mechanism has been outlined in the figure 2.8.

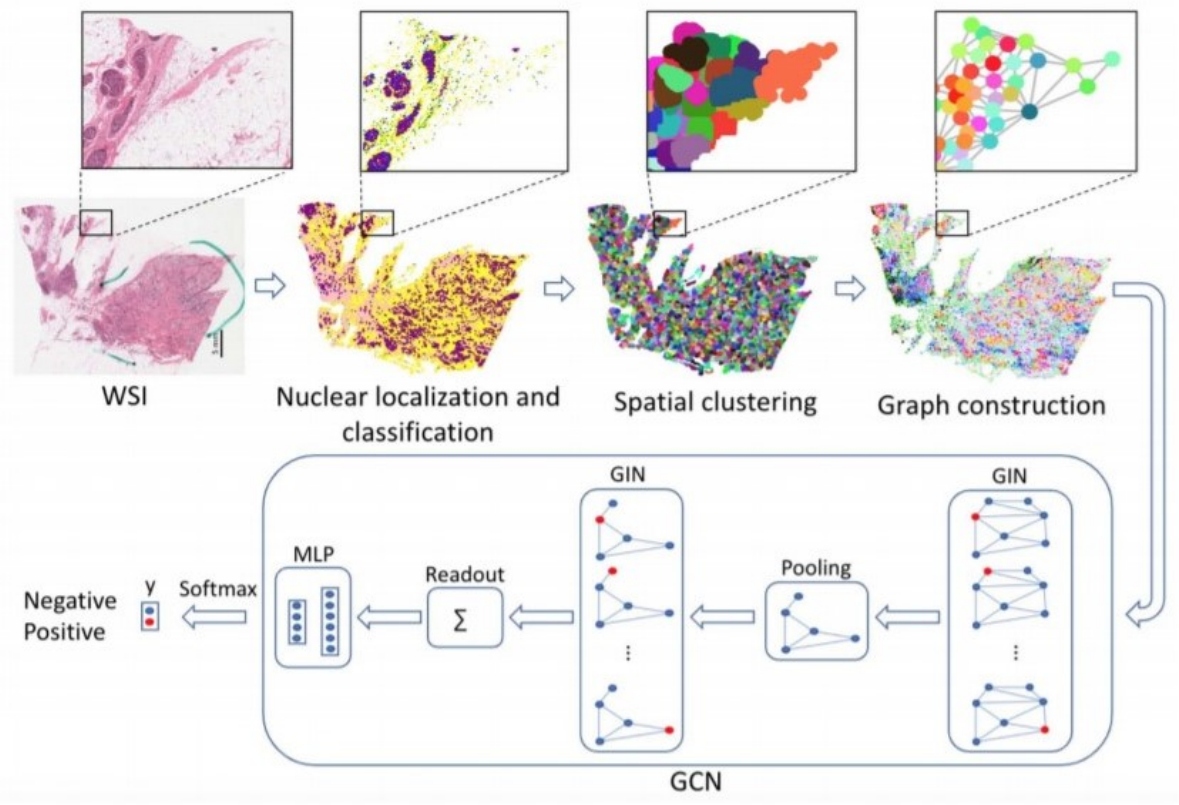

Figure 2.8: An overview of the SlideGraph approach (taken with permission from [92])

Slide Graph is computationally more efficient than patch-based models and opens the avenue of using WSI graph representations for solving other problems in computational pathology as well.

It is based on the assumption that cells in a tissue can organize in a certain way for specific functional states.

Weakly Supervised Learning

Weakly supervised Learning has also been extensively used recently in cases where we do not have access to complete or pixel-level data annotations or only have weakly annotated data, such as slide level annotation for classification of patches and/or localization of tumors. They make use of Multi-Instance Learning, or simply infer finer level information for pixels and patches from coarser slide level information which is more easily available.

This method is especially relevant in the domain of Digital Pathology given that getting expert annotations at pixel or patch levels is very hard and time-consuming and such an extrapolation dramatically reduces the burden from pathologists.

In Multiple Instance Learning[94, 95], we work with bags of training data, labeled as positive or negative, and each bag consists of many instances whose label is unknown. An example can be histology slides labeled cancerous/non-cancerous with each slide representing a bag and patches/pixels extracted from each slide being the instances. The goal usually is then to train models that can help with both bag level and instance level prediction using only weak annotation. Therefore in a broader sense even weakly supervised learning methods based on MIL can be categorized into three types

Global: Identifying target in the Whole Slide Images i.e at the bag level such as presence/absence of Cancer in the slide.

Local: Identification at a finer instance-level i.e in patches/pixel, such as localization of cancerous tissue patches or pixels.

Local and Global: A hybrid detection where both instance level and bag level targets are identifying, such as classifying a slide as cancerous along with localization of cancerous pixels/patches. This is arguably the most popular and clinically relevant MIL approach in histopathology.

A lot of weakly supervised algorithms make use of a combination of multiple weak annotations such as image-level tags,points, bounding boxes, polygons, and percentage of the cancerous region within an image [96, 97, 98, 99].

One popular weakly-supervised approach is *"LOOK, INVESTIGATE, AND CLASSIFY"* A three-stage deep hybrid attention method for Breast Cancer Classification by ***Xu et al*** [80]. In this work, the authors adaptively select a sequence of coarse regions from the raw image by a hard visual attention algorithm, and then for each such region, we are able to investigate the abnormal parts based on a soft-attention mechanism. A recurrent network is then built on top to classify the image region and also to predict the location

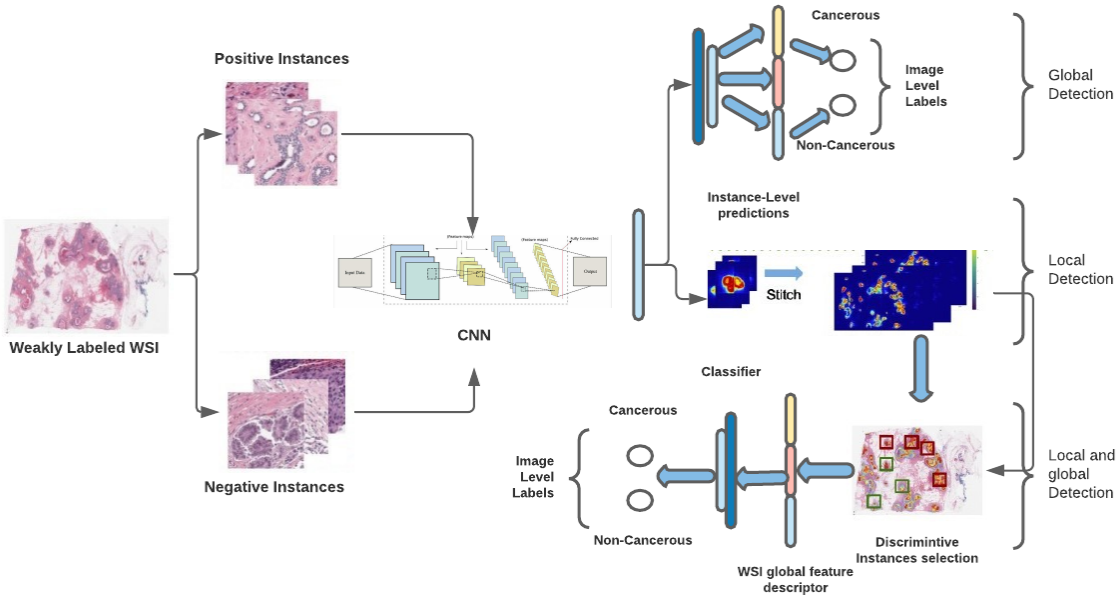

Figure 2.9: An overview of weakly supervised learning for digital pathology

of the image region to be investigated at the next time step. This way, only a fraction of pixels need to be investigated for the classification.

The classification problem is formulated as a Partially Observable Markov Decision Processes (POMDP). Slide-level annotations cast a weak label on all tiles within a particular whole slide image in the following way:

- If the slide is negative, negative for all of its tiles;
- If the slide is positive, positive for at least one of the tiles.

Following the patch-level training, slide level inference was done through aggregation, various Slide level aggregation techniques used were Max-pooling, Random Forest-based aggregation, and RNN based aggregation. Similarly, scale level aggregation was also done by taking tiles extracted at the same center pixel but at different magnifications(5x,10x, and 15x). The strategies used for the aggregation were naive multiscale aggregation using Average and Max Pooling as well as an RNN based multiscale aggrega-

tion using the S most interesting tiles. For both the slide level and scale level aggregation, the RNN based integration method proved to be the most effective.

Unsupervised Learning

Unsupervised Learning refers to the set of learning algorithms that are applied when we don't have any labeled data. Since there are no annotations to guide the training process, the emphasis of common unsupervised learning algorithms is to learn a distinctive pattern, clusters, or anything useful about the underlying data structures from the unlabelled data. The significance of studying and developing unsupervised learning algorithms should be abundantly clear given the scarcity of labeled data owing to regulatory concerns and the labor cost involved in data creation and annotation.

Usual ways to handle unlabeled data are to perform clustering, learn latent representations or perform generative modeling. Autoencoders and Generative Adversarial Networks are popular generative models for learning useful discriminative representations from unlabelled data[100, 101, 102]. The learning task in an unsupervised scenario is ambiguous since the given input can be mapped to infinitely many spaces, therefore most approaches are aimed at maximizing a constrained likelihood of the data thus achieving some meaningful clustering for the given task.

Fischer et al.[103] in their paper titled Sparse coding of pathology slides compared to transfer learning with deep neural networks applied an unsupervised dictionary learning to learn sparse representations of patches extracted using Local Competitive Algorithm that could minimize both reconstruction error as well as non zero neurons. The sparse representations generated using the learned dictionary were then used as a feature to train a tumor/non-tumor classifier. This method outperformed state-of-the-art CNN-based models, by reducing dependence on domain knowledge as well as patch-level annotation.

In early work, sparse autoencoders were utilized for unsupervised nuclei detection [100](Xu et al., 2015a). Detection performance improved by modifying the receptive fields of the convolutional filters to accommodate small nuclei [101]. Generative Adversarial

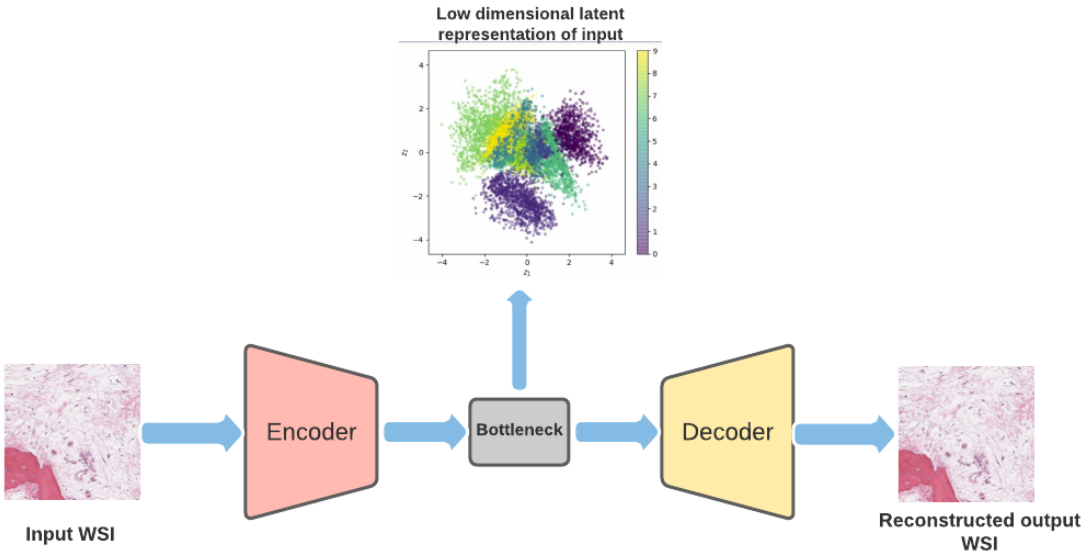

Figure 2.10: An overview of latent representation based unsupervised learning methods

Networks (GANs) have also recently been employed for more complex tasks, like tissue and cell classification. InfoGANs [102] were leveraged for extracting features, which in turn helped maximize the mutual information between the generated images and a pre-defined set of latent codes, to be applied to tasks such as cell-level classification, nuclei segmentation, and cell counting.

There is another category of approaches called unsupervised transfer learning, where instead of directly applying learned features on a target task, we learn mapping functions to be used as initialization for target tasks, possibly with very few labeled training images.

Using a loss term that is similar to the reconstruction objective of autoencoders,[104] trains a CNN using unlabeled images pertaining to a specific modality (e.g., brain MRI or kidney histology images), to learn filterbanks at different scales by sparsely encoding the input images of sizes 13×13 and 27×27 pixels. The resulting filters are shift-invariant, scale-specific, and can help uncover intricate patterns in various tasks, such as tumor classification of glioblastoma multiform or kidney renal clear cell carcinoma. In machine learning, this form of unsupervised learning is called "self-supervised" learning that can

help deal with larger images and thus, offers a promising alternative to clustering approaches in histopathology.

Transfer Learning

The Transfer Learning[105, 106] approach is the most popular and widely adopted technique in digital pathology is the use of. Transfer Learning is the paradigm where the goal is to apply knowledge or representations extracted from one domain to another domain. This way we transfer the learning from a source task/domain to a destination domain. This relies on relaxing the constraint on our data being independent and identically distributed. It also relies on the abstraction ability of deep learning models which tend to learn increasingly complicated features from the data. This means that the initial few layers learn very simple patterns like edges while the later layers learn increasingly complex features representing more specific structures. This ensures that the initial layers pre-trained on any task can easily be transferred to data from other domains since the simplistic features like edges and circles are task agnostic, while we can fine-tune later, more complex layers on our task-specific data.

Therefore, transfer learning techniques in medical imaging can be applied in two different contexts, a. Pretrained models as feature extractors, b. fine-tuning networks initialized with pre-trained weights for the given task, the latter being more ubiquitous in digital pathology. This ingenious approach ensures we don't have to train a new model from scratch every time on a new dataset, we can simply use weights learned on other more general, large datasets on a multitude of tasks and transfer the same weights on our data. This saves significant computational resources and time while not affecting the performance significantly.

Various pre-trained models have been successfully adopted in histopathology domain such as: VGGNet[107], InceptionNet[108], ResNet[39], MobileNet[109], DenseNet[110], etc. Their applications range widely from various cancer grading to prognosis tasks. A more relevant and recent work using transfer learning by Halicek, Martin, et al.[14] per-

forms patch-based localization and whole slide classification for Squamous Cell Carcinoma(SCC)(head) and Thyroid Cell Carcinoma(Neck) using CNN.

A ground-truth binary mask of the cancer area was produced from each outlined histology slide. The WSIs and corresponding ground-truths were down-sampled by a factor of four using nearest-neighbor interpolation. The downsampled slides were then broken into patches of 101 x 101 size. To ensure generalization the number of image patches was augmented by 8x by applying 90-degree rotations and reflections to develop a more robust diagnostic method.

Additionally, to establish a level of color-feature invariance and tolerance to differences in H&E staining between slides, the hue, saturation, brightness, and contrast of each patch were randomly manipulated to make a more rigorous training paradigm before being fed to the Inception-v4 model for detecting head and neck cancer.

Chapter 3

Methodology

In this chapter, the entire workflow has been described in detail. The first section is dedicated to the description, visualization, and analysis of the dataset used. The subsequent sections contain detailed overview of the steps involved in the project including the preprocessing, training, inference and evaluation explained in detail along with figures and pseudo-codes as necessary. They contain details about the segmentation and patch extraction steps, sampling strategies, models and architectures used, evaluation metrics used and the inference output, latency and computational requirements.

3.1 Dataset

The dataset, as is the case for any deep learning problem, was at the center stage of our project and a great deal of work involved in this project was centered around our data, be it acquisition, preprocessing, tiling, color adjustments, and training. Clean, well-annotated, high-quality data can help complex networks identify unique and distinctive information allowing novel tasks hitherto difficult to be automated, while lack of quality data can be the biggest bottleneck when trying to solving novel problems even with the most advanced algorithms.

Our data essentially consisted of digitally scanned head and neck biopsy Whole Slide Images. These were surgical tissue samples taken from patients that were Formalin-fixed and paraffin-embedded (FFPE). They were derived from head and neck regions such as the larynx, pharynx, neck, tongue, mouth, etc. Thin sections were then cut out of the fixed and embedded tissue samples. These sections were stained using Hematoxylin and Eosin staining which is the gold standard and the most widely used stain in medical diagnosis, especially for cancer. The H&E stained tissue sections were analyzed/reviewed by a pathologist.

Following H&E staining all slides were scanned using the *Aperio ScanScope AT Turbo* located at the Molecular Pathology Centre of the Segal Cancer Research Centre at Jewish General Hospital, Montreal. This helped us get the high-resolution digitized version of the slides for computational processing. We had a total of 1517 Digitized Whole Slide Images obtained from 41 patients.There was no inter-scanner variation among the slides since all slides were scanned using the same scanner. Each patient data was assigned a unique alphanumeric code for sorting.

The images were high-resolution files encoded in the '.SVS' format for accessing. Each of the files was very large in size, nearly gigapixel resolution. The whole slide images were created by laying out tiles of smaller sizes to create the larger resolution image. The whole slides each contained multiple resolutions allowing us to pan and zoom to visualize and process them at different resolutions. The four zoom levels expressed as downsample levels were 1x, 4x, 16x, and 32x respectively. We principally worked with the second-highest zoom level(4x downsample) for the project except for visualization, preprocessing steps. We used Aperio ImageScope as our virtual microscope to visualize the WSI at different scales easily. Figure 3.2 shows the distribution of the dimensions of the Whole Slide Images. The central scatter plot plots the width and height for each slide while the histograms on the side depict the distribution of heights and widths. The average dimensions of the slides were approximately 45185(height) x 69108(width)

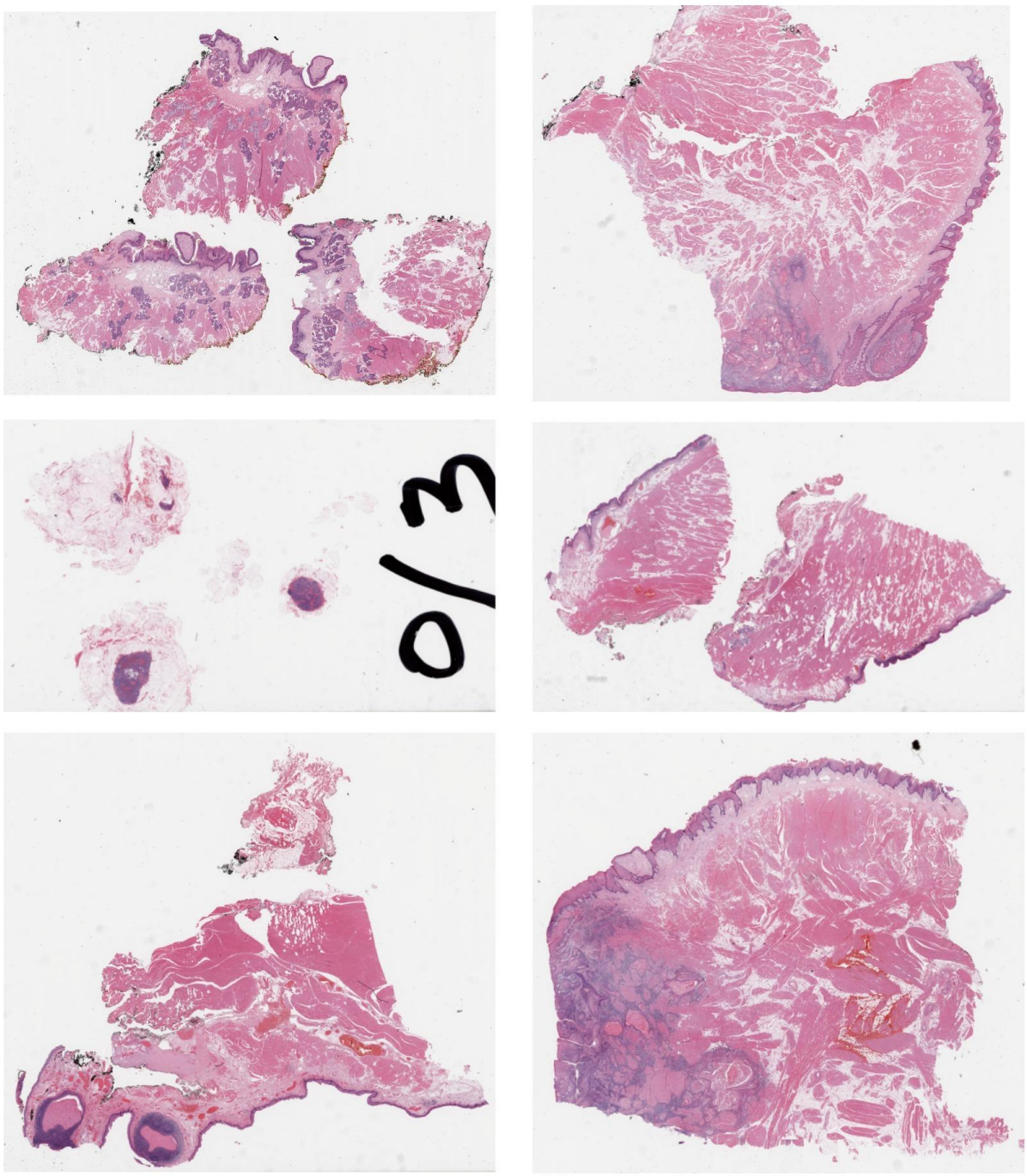

Figure 3.1: Selected samples from our head and neck histology data

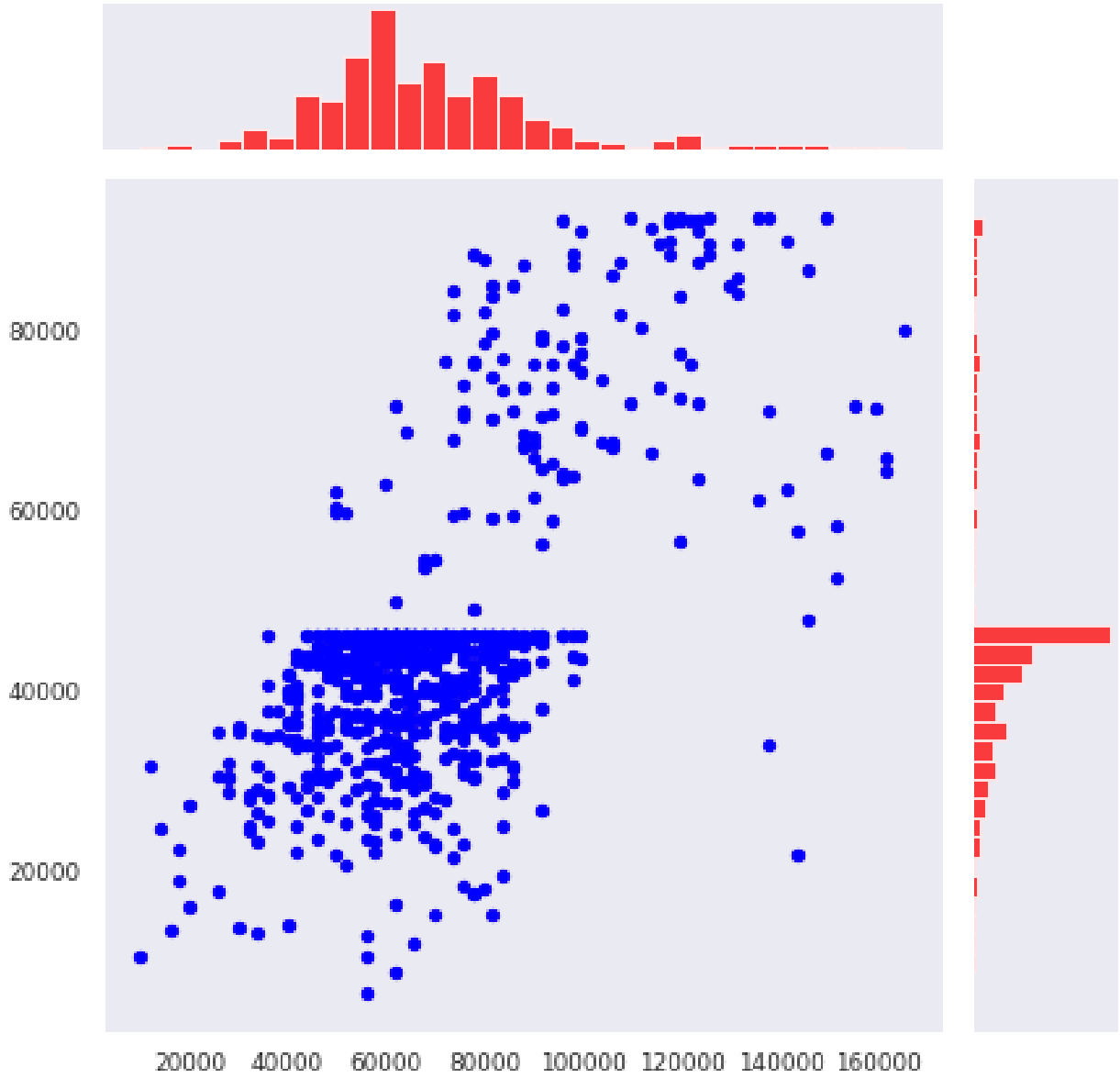

Figure 3.2: Distribution of the dimensions of the Whole Slide Image in pixels

The same ImageScope was used to generate manual annotations for our use case where an expert pathologist drew contours around the tumor regions in WSIs that were cancerous. The contour boundaries were stored as sets of coordinates in the highest resolution as XML files. WSIs with no tumor regions were left as is. These annotations served as our ground truth. A random subset of 600 WSIs out of 1500 odd were annotated since it was a time-consuming effort and they proved sufficiently successful for the scope of the task as we shall see later.

For accessing and processing these WSIs on our programming language of choice i.e Python, we primarily used the *OpenSlide* library, an open-source $C++$ based library that enables working on large Whole Slide Images using python. It works by loading WSIs as an object along with all its related information or metadata such as the number of levels, level downsamples, dimensions, etc. We also used several other tools and libraries such as *large_image*, *slideIO*, *OpenCV*, for processing WSIs as convenient.

3.2 Pre-processing

It was important to preprocess our data in order to make it viable for the application of deep learning best computational image analysis. This was also necessary because of the computationally inhibitive sizes of the images that ran into gigapixels. The images also consisted of vast redundant information that necessitated the use of a preprocessing pipeline that would segment out the regions of interest before further processing in order to streamline the whole training and inference process allowing high throughput. Therefore a major part of the project was to develop an efficient preprocessing pipeline replete with algorithms to analyze, process, sample, and prepare data for subsequent training. The preprocessing pipeline consisted of steps for artifact removal, ROI segmentation, Patch extraction, data normalization, etc which have been discussed in further detail in the following sections.

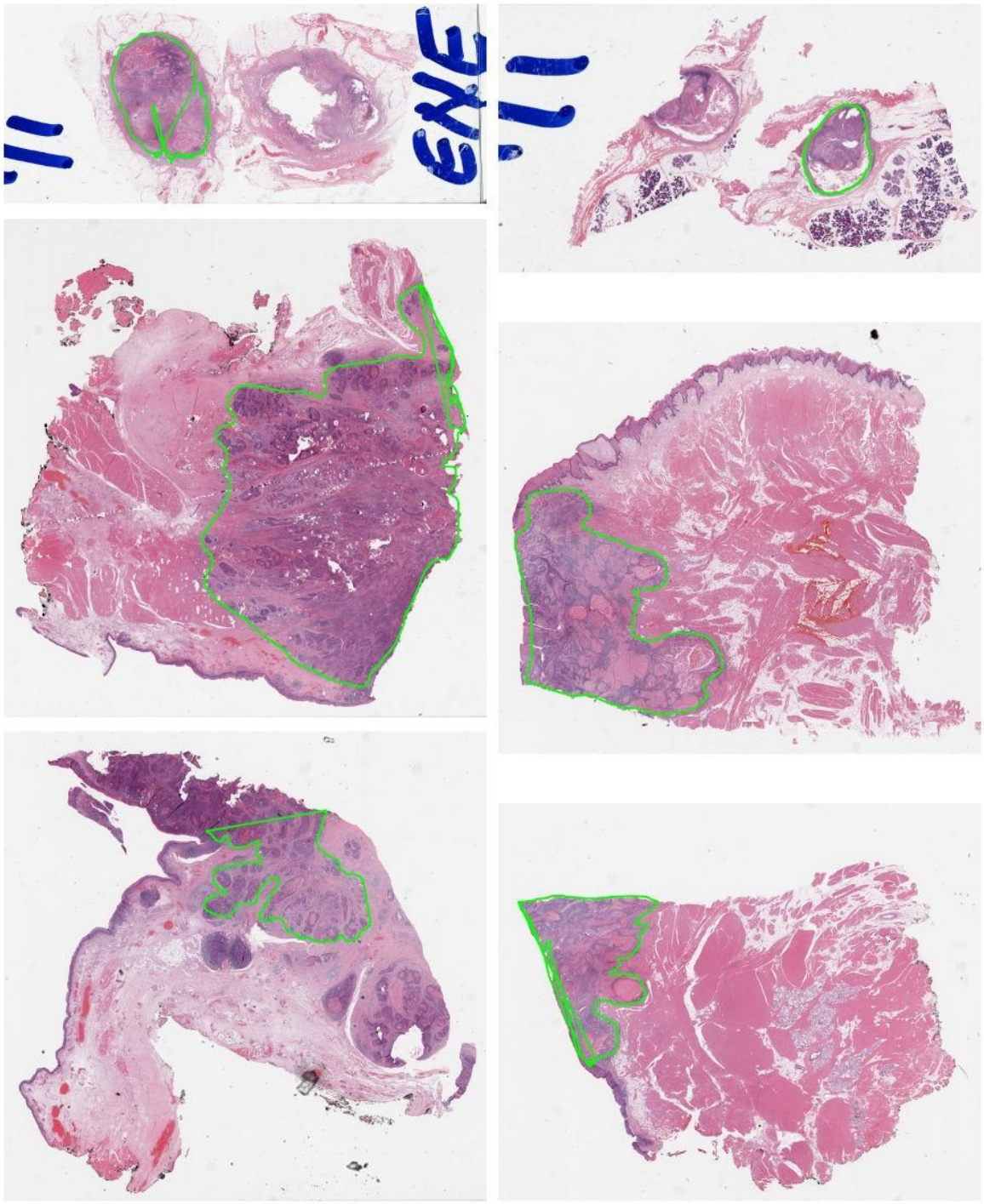

Figure 3.3: Selected WSIs with their annotation, the green boundary represents contours around tumor region

3.2.1 Data Cleaning

A large number of our Whole Slide Images contained artifacts such as Ink marks that had been used by the pathologists for identification of the tissue samples before digitization. The ink marks were usually text written in blue or black markers. They were redundant information for our digital pathology task since the IDs and other information were already available as metadata for the slides. Moreover, the artifacts are essentially noise for both the tissue segmentation as well as tumor detection models skewing their accuracy and leading to non-distinctive data.

A visual inspection of the dataset showed that these ink marks were only of two specific colors: blue and black, therefore we used an RGB thresholding-based image processing algorithm to remove them. We converted our WSI object into an array for ease of computation, we then created two binary masks by thresholding on two specific RGB value ranges, one each for blue and black. The binary masks had the highest value(255) assigned to the regions containing pixels in the said color ranges. We then performed morphological dilation of these masks to ensure they capture even the edges of the artifacts properly. Following this, we performed a bitwise OR operation of the masks with all the three channels of the image separately and then merged them to get the final version. The bitwise OR operation causes the regions in the image containing the ink marks to become white while letting the remaining pixels be as it is. A few samples of the working of the algorithm are shown below in Figure 3.4

```
def cleanData(image= <array>):
    mask1= threshold(image, range1) # binary threshold
    # range1=[[0,0,100], [50,50,255]] # blue
    mask1= morphologicalDilation(mask1)
    mask2= threshold(image, range2)
    # range2=[[0,0,0], [10,10,10]] # black
    mask2= morphologicalDilation(mask2)
```

```
        r,g,b= split_channels(image)
        r_int= bitwise_or(r,mask1)
        r_new= bitwise_or( r_int , mask2)
        ... # repeat for all three channels
        b_new= bitwise_or( b_int , mask2)
        image_new = merge_channels(r_new, g_new, b_new)

        return image_new
```

Listing 3.1: Python pseudocode for WSI noise removal

3.2.2 Tissue Segmentation

One of the biggest challenges when working with Whole Slide Images for digital pathology is that very frequently the WSIs contain a very limited region of interest that includes tissue and nuclei areas while a significant portion of the image is made up of redundant background pixels. With the WSIs being very high resolution in themselves, it's important to devise a way to get rid of the background and isolate the tissue regions in the WSIs for further processing thereby improving the computational efficiency.

Luckily the background and the tissue regions are very distinctive to the naked eye in their color, texture, and intensity as can be seen from the Figure 3.3 and can be easily separated. The two kinds of pixels are also easily separable by image processing techniques owing to their highly divergent properties. This difference is also a result of the H&E staining used on the samples. The task was to create an efficient and generalizable algorithm to accurately segment out the tissue regions. We used a pipeline of classical computer vision methods to achieve this.

The key step involved in the tissue segmentation pipeline was to convert the image from RGB color space to the HSV color space, this resulted in further intensifying the difference between the tissue and the background regions as seen in Figure 3.5. For the

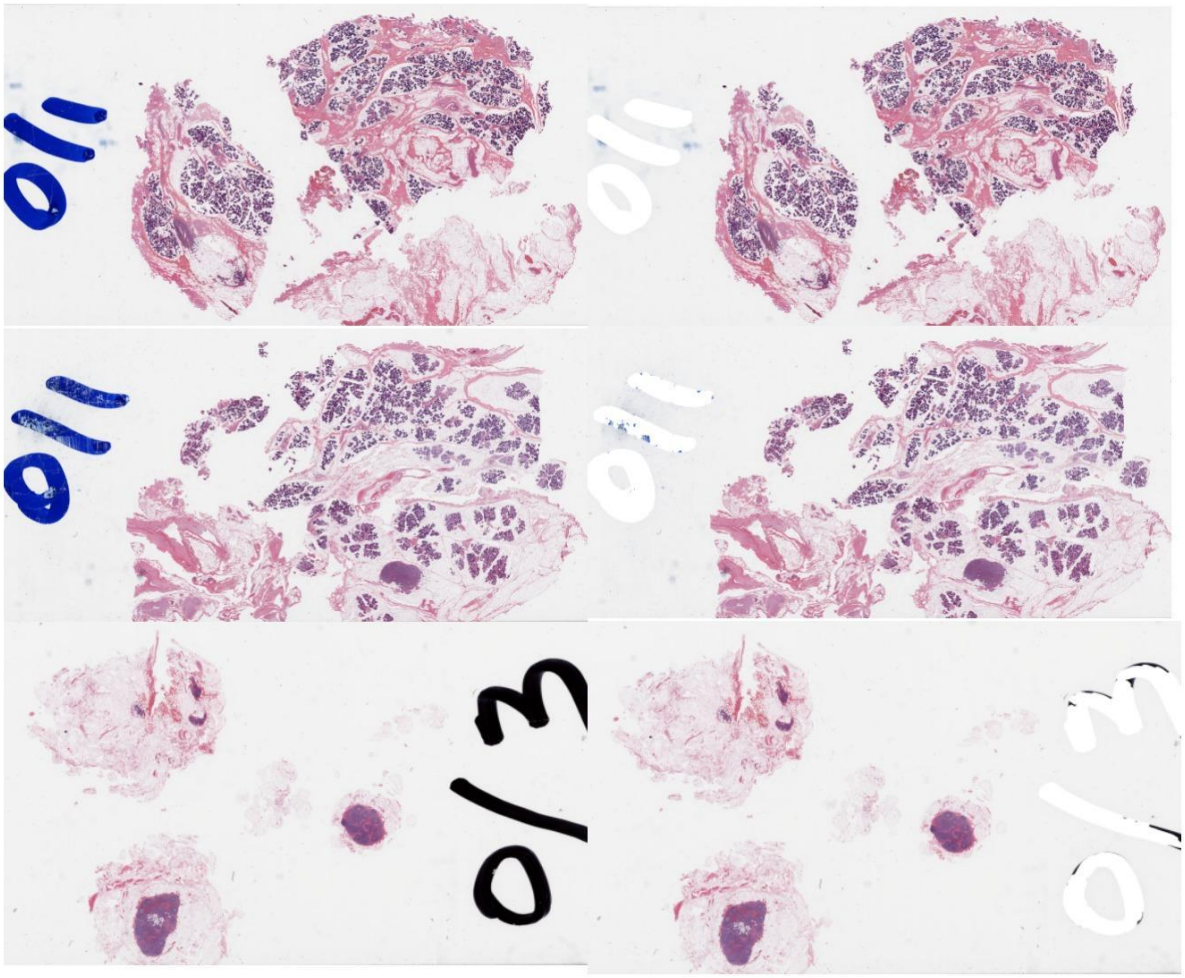

Figure 3.4: Samples from our dataset showing the artifact removal algorithm in action, original images on the left and final results on the right

thresholding, we used Otsu's thresholding algorithm. Otsu's algorithm is a binarization algorithm that iterates through all possible values of threshold and calculates a measure of spread for the pixel levels on each side of the threshold, i.e. the pixels that either fall in foreground or background. The goal is to find the optimum threshold value at which the sum of foreground and background spreads is at its minimum. All the steps involved in the segmentation step are outlined below

- First of all, the images were resized to a tractable size for the segmentation. The WSI is read into memory at a highly downsampled resolution (e.g. 32×32). This was done by simply taking the most downsampled level of our WSI for preprocessing.
- It was then converted from RGB to the HSV color space. Following this, a median blur is applied to the HSV image to smoothen the edges in the image.
- A binary mask for the tissue regions (foreground) is then computed based on thresholding the saturation channel of the image after median blurring. We used Otsu's thresholding algorithm.
- It is then followed up with additional morphological closing to fill small gaps or holes in the tissue regions of the mask.
- The approximate contours of the detected foreground objects are then filtered based on an area threshold and stored as arrays for downstream processing after scaling to the appropriate resolution.
- The contours for large holes within a tissue were also stored separately using contour hierarchy

```
def segmentTissue(image= <array>):
    image=cleanData(image) # remove artifacts
    image_hsv= convertRGB2HSV(image) # convert to HSV
       colorspace
    image_med= medianBlur(image_hsv[:,:,1], blur_threshold)
    # Apply median blur to saturation channel of HSV image to
       smoothen edges
    image_otsu= otsu_thresholding(image_med, threshold_value)
       # apply otsu thresholding
    image_otsu= morphologicalClosing(image_otsu)
```

```
        # perform morphological closing to remove small gaps
        contours, hierarchy= find_contours(image_otsu)
        # find admissible connected component based contours from
            the otsu-thresholded binary image
        tissue_contours, hole_contours = filter_contours(contours
           , hierarchy, area_threshold)
        # filter tissue and hole contours using contours and
           hierarchy based on their area
        return tissue_contours, hole_contours
```

Listing 3.2: Python pseudocode for WSI tissue segmentation

3.2.3 Patch Extraction

Once data cleaning and tissue segmentation were done, the contours were passed on to the last and most important part of the preprocessing pipeline, the image patch extraction system. Since we intend to train our model on patches for tumor detection, it is imperative to build a computationally efficient, high throughput patch extraction algorithm that can be easily integrated with both the preprocessing as well as machine learning training and inference pipelines.

We extract patches from only tissue regions that have been segmented. We pursue the patch extraction as a two-step process, one for non-tumor regions and another for tumor regions. Since we have the contours for the entire regions of interest, as well as contour coordinates for tumor regions for WSIs that contain tumors, we can iterate through each of these contours and extract patches to store them for further processing or for dynamic inference.

We first specify the patch size along with the patch level. Patch level decides the resolution of the WSI at which we extract the patches. The patch size we decided on was 256 x 256 pixels in size and the patch level was decided to be the second-highest zoom

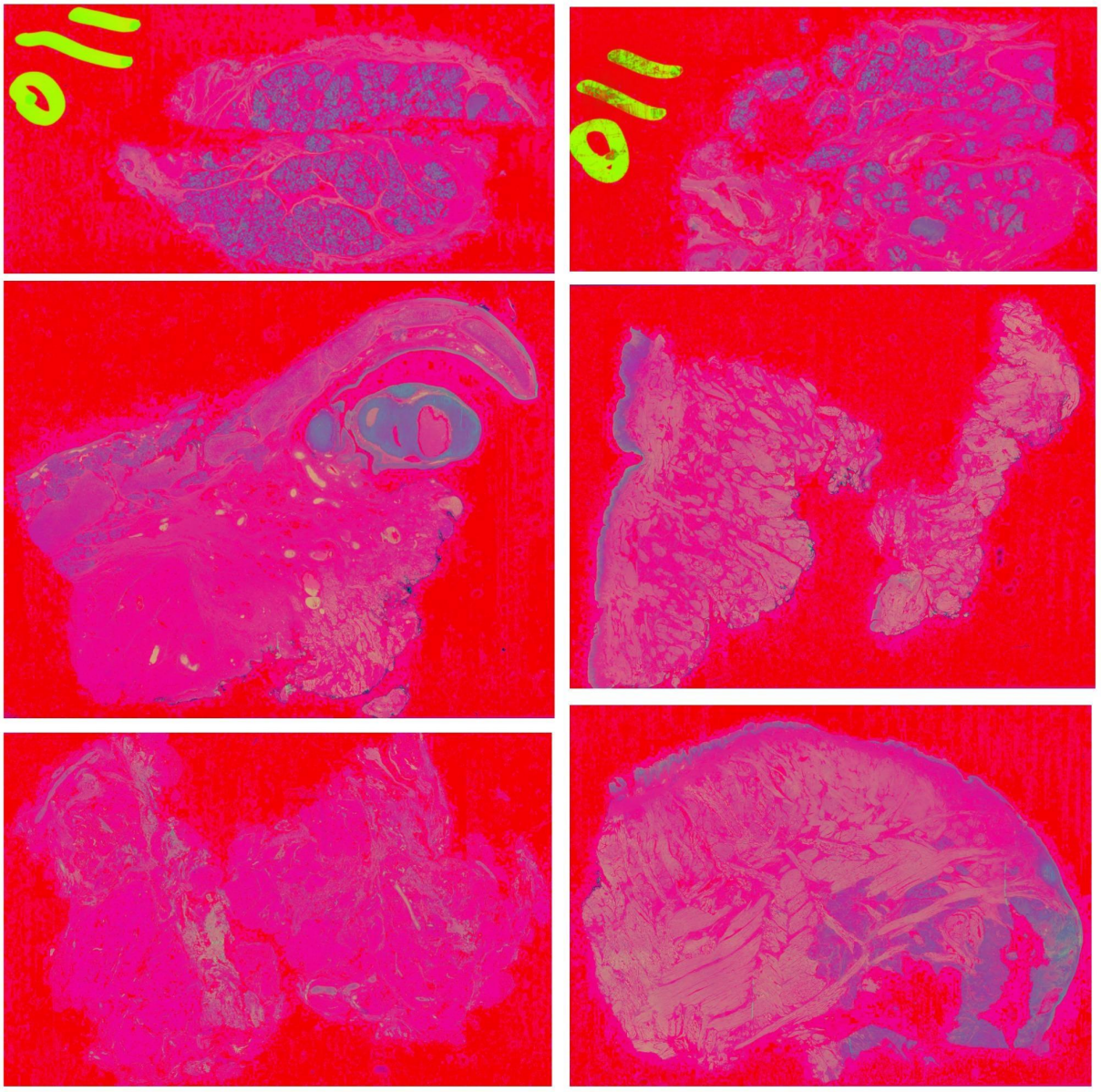

Figure 3.5: Sample WSIs converted to HSV space, it shows how this transformation helps to intensify the distinction between foreground and background pixels

level i.e 4x downsample. This helped with an optimum trade-off between the complexity of the patch extraction algorithm as well as the resolution of the patches.

Following this, we looped through each of the tissue contours returned by the segmentation function, for each contour we further went through them in a sliding window-based approach, with each window being of the same size as the patch size specified. The sliding windows had a stride of 128 pixels along with both the directions thus creating an overlap of 50% when looping through the contours.

For each of the sliding windows, the area was cropped out of the WSI at the patch resolution specified using *slideIO* library. The cropped image array, along with metadata such as patch coordinates, patch level, patch size, tumor label were saved in a dictionary which was converted to binary data format and encoded and stored in .h5py files for later access. Additionally, each patch array was also saved in '.tiff' format in a separate folder for each slide.

The patches were also checked for tumor content. A separate binary tumor mask was generated for slides that contained tumors using the contour coordinates stored as XML files. This was done by taking a zero array(black) of the same size as the WSI at 4x downsampling and drawing the contours and filling them in with ones. Now for each sliding window patch, a similar patch was extracted from the tumor mask at the same location. The proportion of tumor content in the patch was analyzed by counting the number of white pixels(tumor) and black pixels(non-tumor) and calculating their ratio. A ratio greater than 0.05 i.e 5% was considered tumorous and that particular patch was skipped. Thus we got only normal non-tumor patches using the tissue contours given by the segmentation algorithm, all arranged orderly with a 50% overlap.

The tumor patches were extracted separately from the tumor contours stored as XML files. The tumor regions in different slides were of highly varying sizes. Instead of sampling patches in the same sliding window approach as in the case of non-tumor patches, we randomly sampled a fixed number of patches from each of the tumor regions. This was done to ensure there is no bias towards tumor patches from slides containing large

tumors since we shall always get a fixed number of such patches. Also introducing randomness of patch selection was important to ensure coverage of all the different areas in a tumor region and improve generalisability of subsequent learning algorithms since we are not extracting all possible patches. We ended up extracting 200 patches for each tumor region after an inspection of the average tumor area.

In the Figure 4.2 in the result chapter, we can see the different tumor and non-tumor patches extracted from various slides. In figure 4.3 we can see the visualization of extraction patches, which is constructed by stitching together downsampled patches on a blank canvas of the size of a downscaled version of the original WSI. We can visualize the results of different patch extraction strategies for tumor and non-tumor regions.

```
def extractPatches(image= <array>, contours, tumor_contours,
   patch_level=1, patch_size=256, stride=128, num_tumors=200):
    '''Input params:
    Image: WSI
    contours: segmented contours
    tumor_contours: tumor contours
    patch_level: zoom level for patch extraction
    patch_size: size of each patch
    stride: stride of sliding window
    num_tumors: Number of tumor patches to extract from each
       tumor contour
    '''
    # extracting non tumor patches
    for cont in contours:
    # loop through all contours
        for x in range(0,width(contour),stride):
        # loop through all possible x coordinates of a contour
           with a given stride
```

```
            for y in range(0,height(contour),stride):
        # loop through all possible x coordinates of a contour
            with a given stride
                patch= crop_image(x,y, patch_size,patch_level)
                # crop out a patch of given patch size from
                    patch_level of WSI at point(x,y)
                gt_patch= crop(gt_mask, x,y, patch_size)
                # crop out a patch of given patch size from binary
                    ground truth tumor mask
                tumor_area=num_ones(gt_patch)/num_zeros(
                    gt_patch_mask)
                # calculate tumor ratio for the given patch
                if tumor_area<0.05:
                  save_patch(patch)
              # if tumor pixels are less than 5% save patch as non
                 tumor patch along with metadata
    # extracting tumor patches
    for cont in tumor_contours:
    # now looping through tumor regions
        x= random_sample(0, width(cont), num_tumors)
        y= random_sample(0, height(cont), num_tumors)
        # randomly sample as many pairs of (x,y) coordinates as
            number of tumor patches required, from each tumor
            contour
        patch= crop_image(x,y, patch_size,patch_level)
        # crop out a patch of given patch size from patch_level
            of WSI at point(x,y)
        save_patch(patch)
```

```
            # save patch as tumor patch along with metadata
    return patches
```

Listing 3.3: Python pseudocode for patch extraction

3.3 Experimental Setup

3.3.1 Data Split

Before proceeding with training our classifier with the patch data, we split the patches into three separate parts, the training, validation, and test data. This is standard practice in deep learning where we train our models on the training data and validate its performance on the validation data in order to tune various hyperparameters as well as to select the best model over multiple iterations. The test set is intended for testing our final model on completely unseen data to assess its final performance as well as to see how well the model generalizes.

We used the ***groupshufflesplit*** method from the sklearn library for the data split. This was done to ensure the patches belonging to a particular patient all were placed into the same split, i.e either train or validation or test. Since having patches belonging to the same slide or the same patient may skew the model performance and not provide correct insight into its performance. So the patches were grouped by the patient ID of the slide they were extracted from and then a 60:20:20 split was done along with random shuffling for the patches.

The final distribution of the data following the split is tabulated in Table 3.1 below

	Training Set	**Validation Set**	**Test Set**	**Total**
Tumor Patches	32,389	5,769	5,801	43,959
Non-Tumor Patches	2,259,014	173,334	570,828	3,003,176
Total	2,291,403	179,103	576,629	3,047,135

Table 3.1: Final distribution of tumor and non-tumor patches across the three data splits

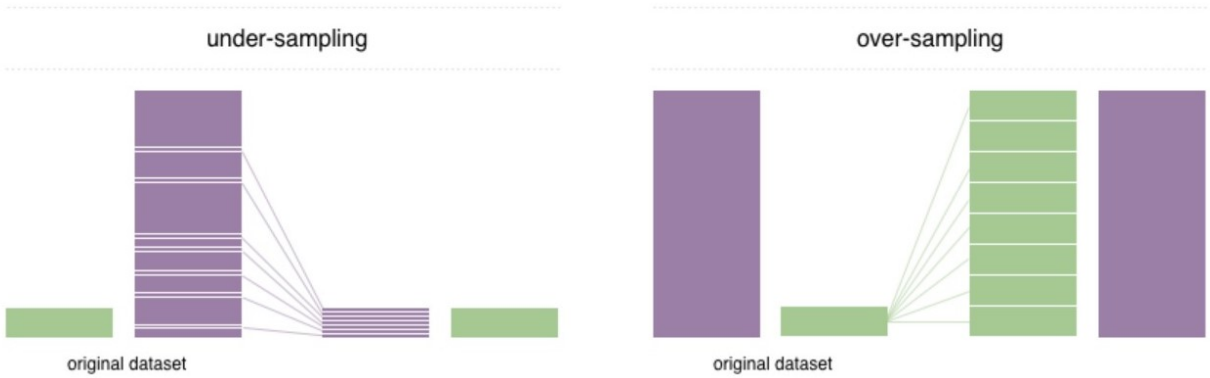

Figure 3.6: Pictorial representation of the two resampling strategies for an imbalanced dataset

3.3.2 Sampling

As we can see from Table 3.1 above the patches are heavily class imbalanced with tumor and non-tumor classes being present in a nearly 80:1 ratio. This needed to be fixed for our model to learn properly else it would be heavily biased towards the non-tumor class and hence non-reliable. While fixing this data imbalance it was also important to not simply make the two classes equal as that would not have been a realistic representation of actual data, since tissue samples/pixels/patches have a much higher probability of not containing tumor cells than otherwise. Hence the sampling had to be done keeping in mind this trade-off. The various sampling methods are:

Downsampling/Undersampling: In downsampling, we generally reduce the number of samples from the more frequent class or group by randomly sampling a given percentage of its samples and discarding the rest. In order to resample our data to be equal, we would have to downsample the non-tumor class by a factor of 30 i.e take one sample out of every 30 possible ones.

Upsampling/Oversampling: Upsampling on the other hand implies multiplying the number of samples present in the minority class. We can either do so by repetition, or by randomly sampling with replacement to increase the proportion of minority class.

Weighted Random Sampling: is a special sampling algorithm where instead of sampling from classes with uniform probability, we sample items based on probabilities determined by the relative weights assigned to each class. PyTorch framework provides an easy-to-use WeightedRandomSampler method to perform this sampling during the data-loading phase itself.

We used a combination of Downsampling and Weighted Random Sampling in our training phase, Uniform downsampling was first performed to bring the proportion of the two classes to a more representative level (e.g 10:1), and then weighted random sampling was performed during the training phase to ensure each batch of training data had a similar representation of both the classes. This along with the data augmentation techniques employed ensure our model neither overfit nor was biased towards one class despite the imbalance in the dataset.

3.3.3 Data Augmentation

Data augmentation is the technique employed to multiply the amount of data we have by introducing various affine, color, and intensity transformations to the images. The multiplied dataset helps to reduce the possibility of models overfitting and makes them more generalizable. These additions also make the model invariant to noise, blur orientation shifts that it might face when inferring on real-world unseen data.

The different augmentation techniques are affine transformations such as rotation, cropping, scaling, and shifting, image transformations such as blurring, noise addition, color feature transformations like hue, saturation, contrast variations, color jitter addition, normalization, etc.

The histology patches don't have any canonical orientation, which means the diagnosis isn't affected by their orientation. This provides us a great incentive to augment the data through various transformations. augmented the training data by using transformations such as horizontal and vertical flips, random rotations, random rescaling, gaussian blur, median blur, random Gaussian noise addition, normalization, random contrast and

brightness adjustment, color jitter, etc. The image augmentation was carried out with the help of *Albumentations* library in python. We used random rotation, gaussian blur, median blur, horizontal and vertical flips, and Contrast Limited Adaptive Histogram Equalization(CLAHE) with a range of parameter values and randomness for augmentation.

3.3.4 Training

The patches extracted were then used to train our tumor detection or the patch classification model. Three different state-of-the-art architectures were trained and analysed for this task. We used ResNet-34, ResNet-50 and EfficientNet-B2 for our analysis. All of these models were trained using a transfer learning approach. Models pre-trained on ImageNet dataset were fine tuned for our purpose. These models were then fine-tuned on our dataset by freezing the initial few layers' weights while removing the last fully connected layers and replacing them with a custom fully connected block for our use case. For the ResNet-34 and ResNet-50 models we froze the layers in the first six convolutional block and fine tuned latter layers along with our custom one layer deep fully connected classification block. For the EfficientNet-B2 we only fine-tuned the last MBConv block along with the final Fully Connected block with 4 linear layers. The theoretical aspects of the two primary CNN architectures we used have been discussed below

Models

Residual Network (ResNet) ResNet [39] is one of the most popular and powerful CNN architectures. It was the state-of-the-art architecture that outperformed all the previous architectures on the ImageNet data classification and was the first of its kind when it comes to the depth of the architectures. Before ResNet, the possible depth of the CNN architectures was severely limited by factors such as vanishing and exploding gradients due to the multiplicative effect of the gradients in backpropagation. ResNet managed to solve this problem by introducing the concept of residual blocks that contained skip connections which are essentially "identity shortcut connection" that skips one or more

layers. This helped workaround the exploding and vanishing gradient problems and enabled very deep networks to be built and hence a much more complex representation being learned. Figure 3.7 depicts a small residual block. There are many variations of the ResNet architecture based on the number of layers such as ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152. There have also been a number of new architectures inspired by the ResNet such as ResNeXt, Densely Connected Networks (DenseNets), etc. For our experiments we used ResNet-34 and ResNet-50 for training and analyzing their performance for tumor detection.

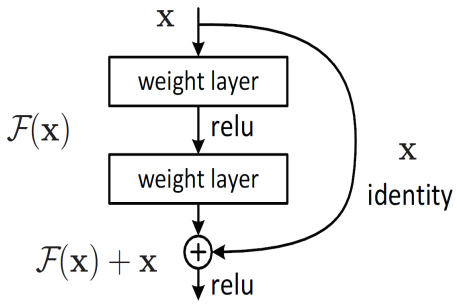

Figure 3.7: A residual block which is the building block of ResNet models

EfficientNet EfficientNet [40] is a new convolutional neural network architecture and scaling method that uniformly scales all dimensions of depth/width/resolution using a compound coefficient.

Its part of the AutoML paradigm meant for models to automatically adjust their features such as depth, scale based on data and computational resources. Unlike conventional practice that arbitrary scales these factors, the EfficientNet scaling method uniformly scales network width, depth, and resolution with a set of fixed scaling coefficients.

The main building block of EfficientNet architecture, is mobile inverted bottleneck MBConv, which was first introduced in MobileNetV2 architecture [109] meant for training very small models to be deployed on edge devices. It uses shortcuts directly between bottleneck layers which connects a much fewer number of channels compared to expansion layers, combined with depthwise separable convolution effectively reducing com-

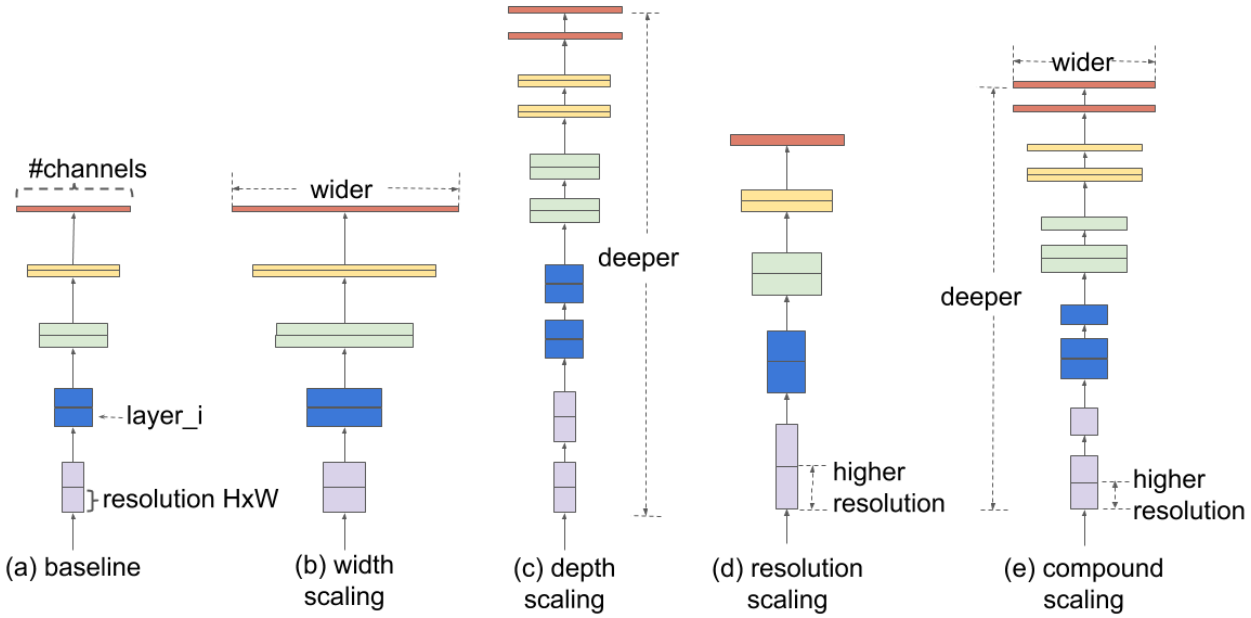

Figure 3.8: Comparison of different scaling methods (b)-(d) arbitrarily scale a single dimension of the network, while (e) is the compound scaling method used in EfficientNet [40]

putation by almost a factor of k^2, compared to traditional layers, where k denotes kernel size. This makes the EfficientNet much smaller in terms of the number of parameters compared to other popular architectures such as ResNet while achieving even better performance as can be seen in Figure 3.9

EfficientNet also has multiple variations such as EfficientNet-B0, EfficientNet-B1, EfficientNet-B3, EfficientNet-B5, EfficientNet-B7, etc based on the complexity and arrangement of the building blocks of the network. EfficientNet models also transfer well across tasks. We used the EfficientNet-B2 for our experiments.

Hyperparameters

Training a CNN model requires a number of hyperparameters to be decided and fine-tuned in order to get the optimum performance. These include activation functions, Optimizers, loss functions, number of epochs, batch size, etc.

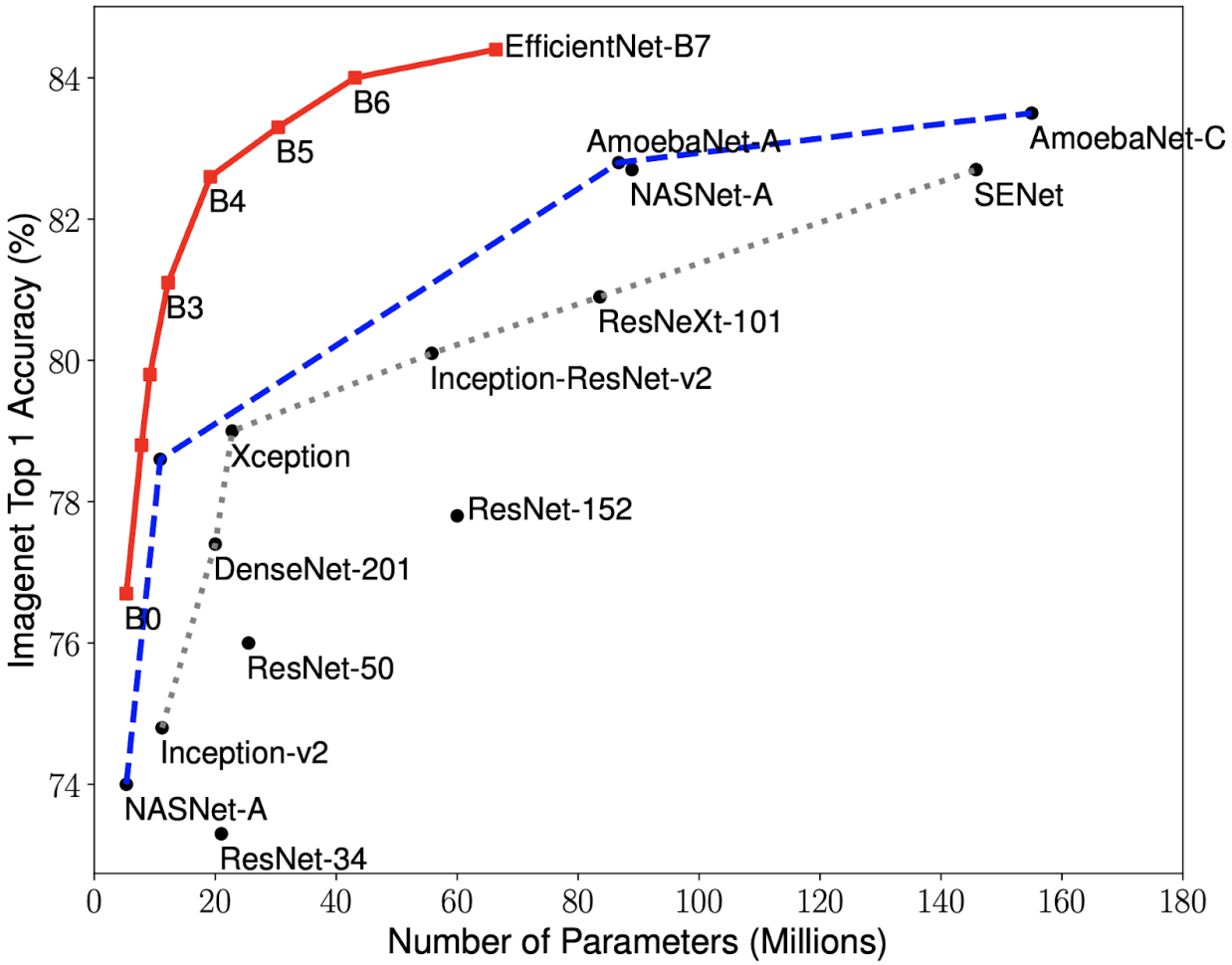

Figure 3.9: Graph depicting the ratio of performance to number of parameters of popular CNN architectures on ImageNet. All the versions of EfficientNet require much fewer parameters for achieving similar or better performance as other architectures. Graph taken from [40]

Activation Functions are transformations applied to the weighted sum of the inputs at a node in a neural network to get their output. Most often these activation functions are also used to introduce non-linearity into the values of the neurons. We used the Rectified Linear Unit (ReLU) activation that gets rid of negative values for our convolutional layers and is defined by the formula

$$R(x = max(0, x)$$

for ResNet models and a sigmoid activation for our final fully connected layer with one output unit to output a probability. The sigmoid function is given by

$$S(x) = \frac{1}{1 + e^{-z}}$$

For the EfficientNet model, we used the computationally efficient swish activation function for the convolutional building blocks of the model which is given by

$$f(x) = x \cdot \text{sigmoid}(\beta x)$$

where β is a learnable parameter and sigmoid again for the fully connected part.

Optimizer: Optimization algorithms are the algorithms used to find the optimum parameters or arguments of a function such that it outputs the maximum or minimum value. In the case of neural networks, optimization algorithms help find the optimum weights of the neural network such that an objective function also called the loss function is minimized. The most common optimization algorithms used in deep learning are gradient descent based first-order derivative optimization algorithms where we calculate the gradient of the function, then follow the gradient in the opposite direction (e.g. downhill to the minimum for minimization problems) using a step size (also called the learning rate). Common optimizers are stochastic gradient descent, RMSProp, Adagrad, Adam, etc We experimented with both RMSProp and Adam and ended up using Adam for training the model. Adam adaptively adjusts the learning rate on a per parameter basis using the moving average of the second moment of the gradients and also uses momentum based on the first moment of the gradients. The Adam optimization formula for a set of weights

ω is given as

$$\nu_t = \beta_1 * \nu_{t-1} - (1 - \beta_1) * g_t$$

$$s_t = \beta_2 * s_{t-1} - (1 - \beta_2) * g_t^2$$

$$\Delta\omega_t = -\eta \frac{\nu_t}{\sqrt{s_t + \epsilon}} * g_t$$

$$\omega_{t+1} = \omega_t + \Delta\omega_t$$

where, η : Initial Learning rate

g_t : Gradient at time t along ω^j

ν_t : Exponential Average of gradients along ω_j

s_t : Exponential Average of squares of gradients along ω_j

β_1, β_2 : Hyperparameters

Loss Functions: Loss functions are functions used to describe the cost or error between the model's output and the intended ground truth value. This loss value is then used by the optimization algorithm to fine-tune the weights and find the optimized parameters. The common loss functions used for classification are Cross-entropy loss, Binary Cross entropy loss, L2 loss, Hinge Loss, KL-Divergence loss. Since the tumor detection problem we are working on is a binary classification problem, we used the Binary Cross Entropy with Logits as our loss function. Binary cross entropy also called the log loss is given by the following function

$$-(y \log(p) + (1 - y) \log(1 - p))$$

where y is the ground truth label and p is the output of the neural network after sigmoid activation.

Apart from these principle hyperparameters other important values such as the number of epochs, batch size, learning rate were also experimented with and optimized. We also used a learning rate scheduler for dynamically changing our learning rate based on model performance. We used the *ReduceLROnPlateau* function that reduces the learning amount by a given factor every time a metric of interest plateaus i.e reduces from the highest achieved value or remains constant within a certain boundary. We used the F1

score as our metric of interest and set the patience value to 1 which essentially means the optimizer waits for at least two epochs of non-improvement before reducing the learning rate. This prevents the model from getting stuck oscillating near the optimum due to very high learning rates.

Metrics

Multiple metrics were used for the evaluation of the model in both training and testing phases. These were

Accuracy: Accuracy refers to the percentage of the correct prediction made by the model and is given by the formula

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

where TP, TN, FP, and FN refer to True Positives, True Negatives, False Positives, and False Negatives respectively In the case of imbalanced data such as the tumor and non-tumor patches in our case, accuracy is not a reliable metric since the model could be biased and predict the same label for entire data but still have high accuracy given the data ratio.

Recall: Recall is a metric that defines how many positive samples were identified out of all possible positive samples or fraction of the relevant instances retrieved, in our case, it defines how many tumor patches were detected by the model out of all possible tumor patches. It's also called sensitivity and is given by the formula.

$$\text{Recall } = \frac{TP}{TP + FN}$$

This metric is used to keep a tab on the number of False Negatives which can prove significant in a number of applications, especially medical applications such as ours, where a False negative diagnosis can prove fatal.

Precision: Precision is used to signify how many of the positive samples identified are actually positive samples and not False Positives, or the fraction of retrieved documents

that are relevant to the query. In our case, it defines how many of the tissue patches detected as containing tumors are actually tumorous. It's also called specificity and is given by the formula

$$\text{Precision } = \frac{TP}{TP + FP}$$

Precision can be used to keep a tab on the number of false positives which is a piece of important information for various applications.

F1 Score: Both recall and precision are better representative metrics than accuracy for imbalanced datasets and important values in general. Hence, to examine the effectiveness of a model we need both of these measures. Unfortunately, there is usually a trade-off to be made i.e precision suffers when we improve the recall and vice versa. Hence we need a metric we can focus on optimizing that will ensure the optimum value of both precision and recall is achieved by the model. This is done by F1-score, which is nothing but the harmonic mean of precision and recall and given by the formula

ROC curve and AUC: ROC curve or receiver operating characteristic curve is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters: True Positive Rate(TPR) and False Positive Rate(FPR) given by

$$\text{TPR} = \frac{TP}{TP + FN} \qquad \text{FPR} = \frac{FP}{FP + TN}$$

AUC stands for Area under the ROC Curve. That is, AUC measures the entire two-dimensional area underneath the ROC curve. AUC provides an aggregate measure of performance across all possible classification thresholds. These metrics were used to fine-tune the classification threshold

Apart from this we also used pixel-level **IOU**(Intersection over Union) to analyze objectively how well the patch tumor classifier could be used as a coarse segmentation model in the larger schemes of thinks. This was not done to measure segmentation performance as the model was not primarily trained for segmentation, but as another metric

to measure its performance as a classifier on a whole slide level. This was calculated using the whole slide binary prediction mask and ground-truth binary mask.

3.3.5 Inference

Once the model had been trained and validated to its optimal performance, we used the saved weights for inferring on new data at both patch and slide level. For inference, an integrated pipeline was built that would take a whole slide image, perform noise removal, tissue segmentation, patch extraction, and patch classification in a single end-to-end workflow. The outputs we got from the inference were binary prediction masks, heatmaps of prediction probability overlaid on top of a downscaled version of the WSI, IOU score and slide level prediction scores. So essentially we passed a set of WSIs and the model predicted on the patches present in the slide in batches to generate the desired outputs. All the patches were extracted using a sliding window of the patch size, since tumor and non-tumor patches were not provided separately and were to be inferred by the models.

The binary prediction mask was generated by taking a downsampled blank canvas of the same aspect ratio as the original slide and stitching together patches downsampled by the same scale as white if that region in the original slide was classified as positive by the tumor classifier. This mask was then used for creating an overlaid image with the predicted tumor mask on top of the actual Whole Slide Image. A heatmap of probability scores was also generated for better visualization of the classifier performance.

The IOUs were calculated on a per pixel basis using the binary prediction mask and the original ground truth tumor mask created from the annotations. This gave us high level information about the coarse tumor segmentation capacity of the patch-based tumor detection model. The IOUs suffered a due to minimal false positives since they were calculated at pixel level while the prediction was at the patch-level, so even a single misclassification lead to major negative effect on the IOU.

Apart from patch-level detection or tissue classification, slide level classification was also done by using a meta heuristic on top of the classifier, which classified the slides as containing tumor or not based on the average probability score of the top fifty highest probability patches according to the model output. This value was thresholded for assigning a tumor/non tumor label to the slide. We used a heap data structure to maintain a list of the top 50 highest scored patches values efficiently.

3.3.6 Computational Resources and Latency

We used a server with 12 GB RAM and two 12 GB NVIDIA GeForce GTX 1080Ti GPU memories. For training, we made use of a distributed parallel training regime to speed up training. For this, we made use of the Distributed Data parallel module in PyTorch to spawn multiple processes distributed over the two GPU machines for parallel training. The time required to train one ResNet-50 model was 45 minutes per epoch and that for EfficientNet-B2 was 30 minutes per epoch. The inference on the other hand took less than 16 seconds per slide and could be done both on GPU and CPU devices. A major part of latency during inference had to do with the preprocessing steps such as tissue segmentation and patch extraction.

Chapter 4

Results and Analysis

In this section, the results of the experiments discussed in Chapter 3 are presented in a sequential order corresponding to the steps mentioned in methodology along with a brief analysis of each of the obtained results.

4.1 Tissue Segmentation

In figure 4.1, we can visualize the results of the tissue segmentation pre-processing steps. Since there was no ground truth and since this was merely a preprocessing step in the tumor detection workflow, the segmentation of regions of interest was only analyzed visually and it was quite robust for all cases as can be seen in the figure. The disconnected tissue areas were closely segmented into contours. Holes within the tissue regions were also captured using the contour hierarchy and represented separately.

4.2 Patch Extraction

The patch extraction step was the most important part of the pre-processing workflow since we needed a low latency, high through-put accurate patch extraction method to convert WSIs into patches before training and inference with the patch-level tumor clas-

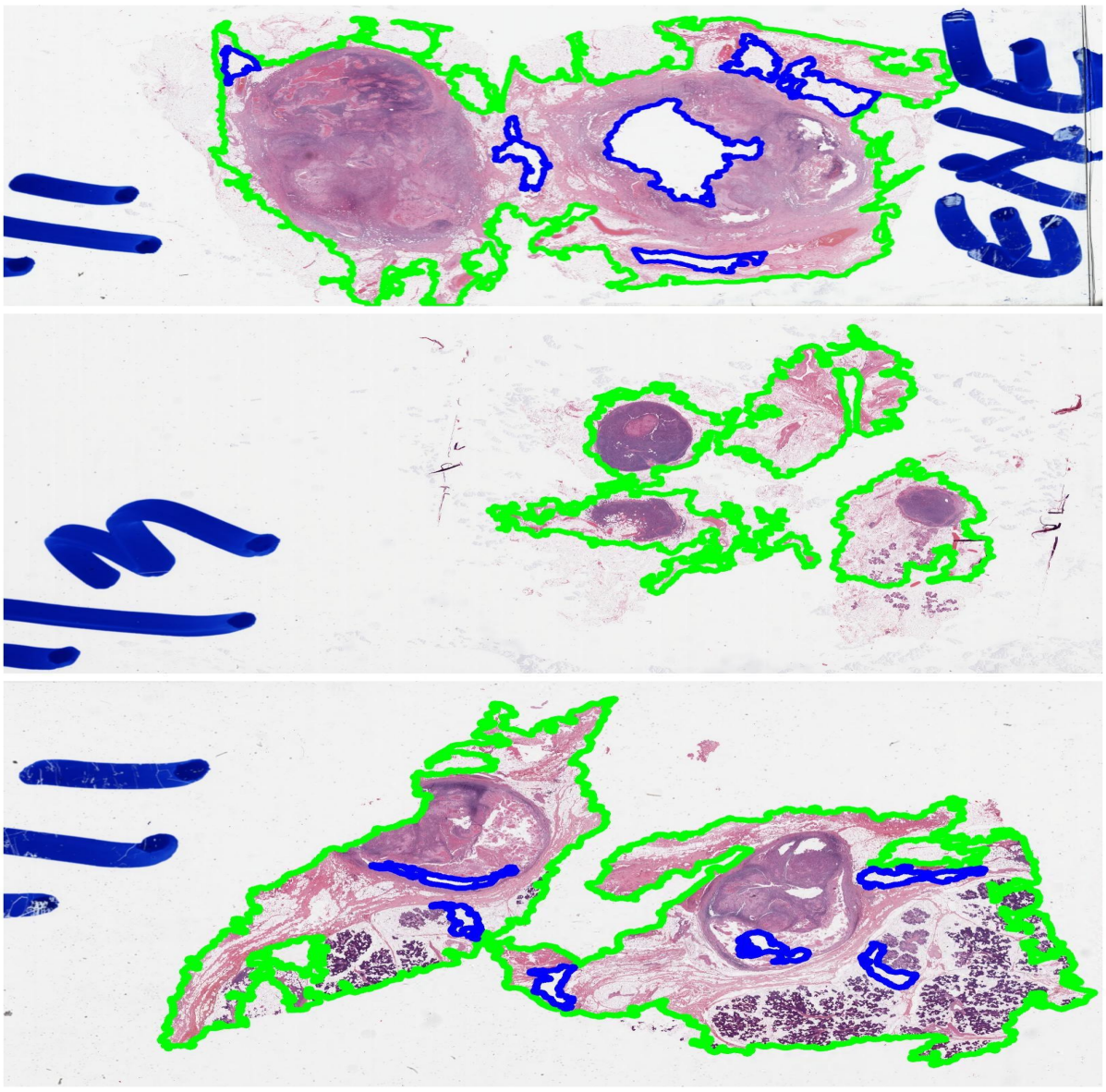

Figure 4.1: Some outputs of tissue segmentation algorithm, yellow contours represent tissues, blue represents holes

sifier. Figure 4.2 shows samples of tumor and non-tumor patches extracted from different slides.

The patches were extracted from the second-highest resolution of the WSI i.e 4x downsample and the size of each patch extracted was $256x256$. The patch extraction fits perfectly with the previous tissue segmentation step by taking only the tissue contours re-

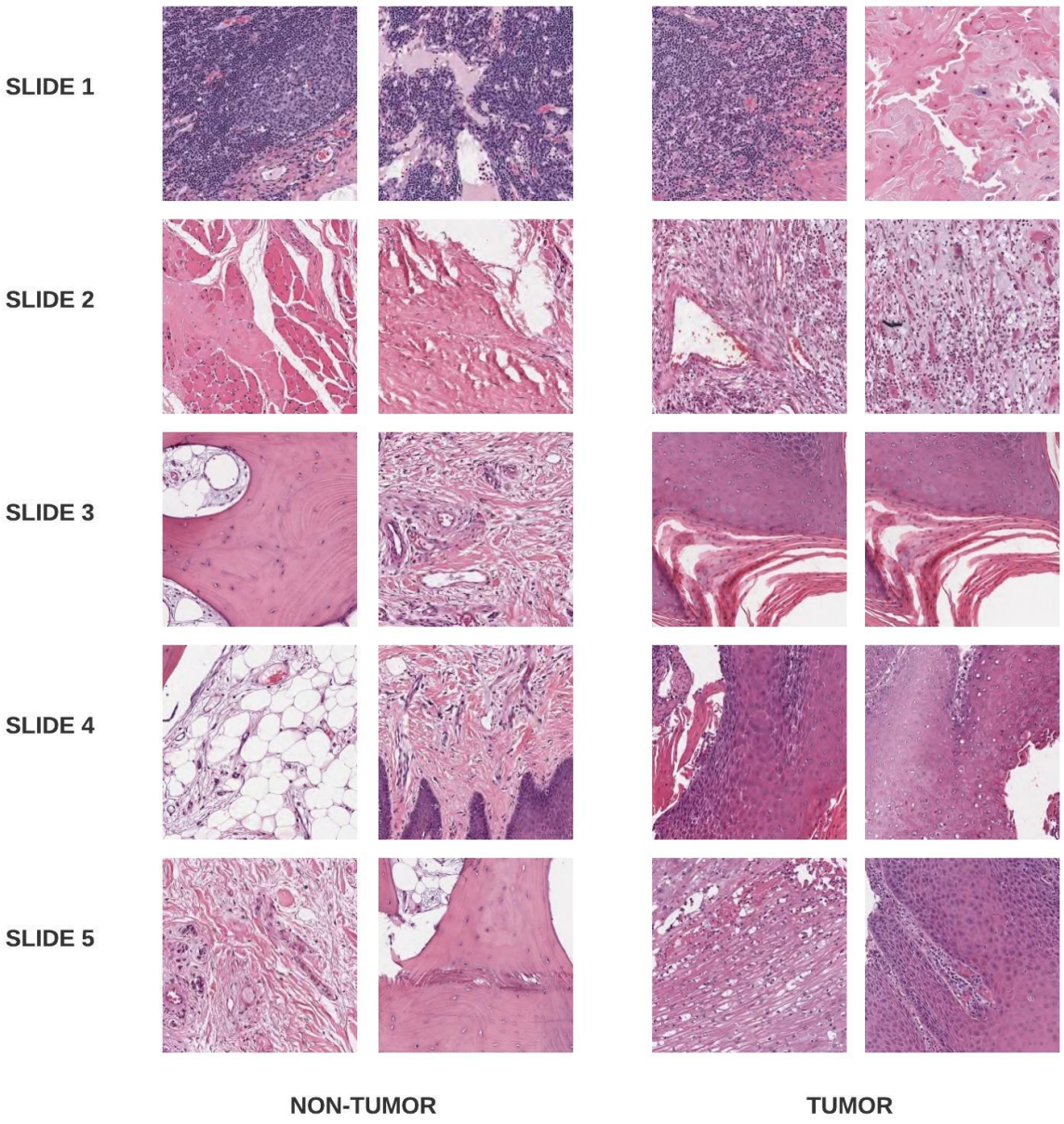

Figure 4.2: Samples of Tumor and Non-Tumor patches extracted from five different Whole Slide Images

turned by the segmentation step and performing patch extraction on them. For tumor regions, 200 patches were randomly extracted from the tumor contours provided as annotations while for non-tumor regions the patches were extracted sequentially. This difference in patch extraction strategy has been shown in Figure 4.3

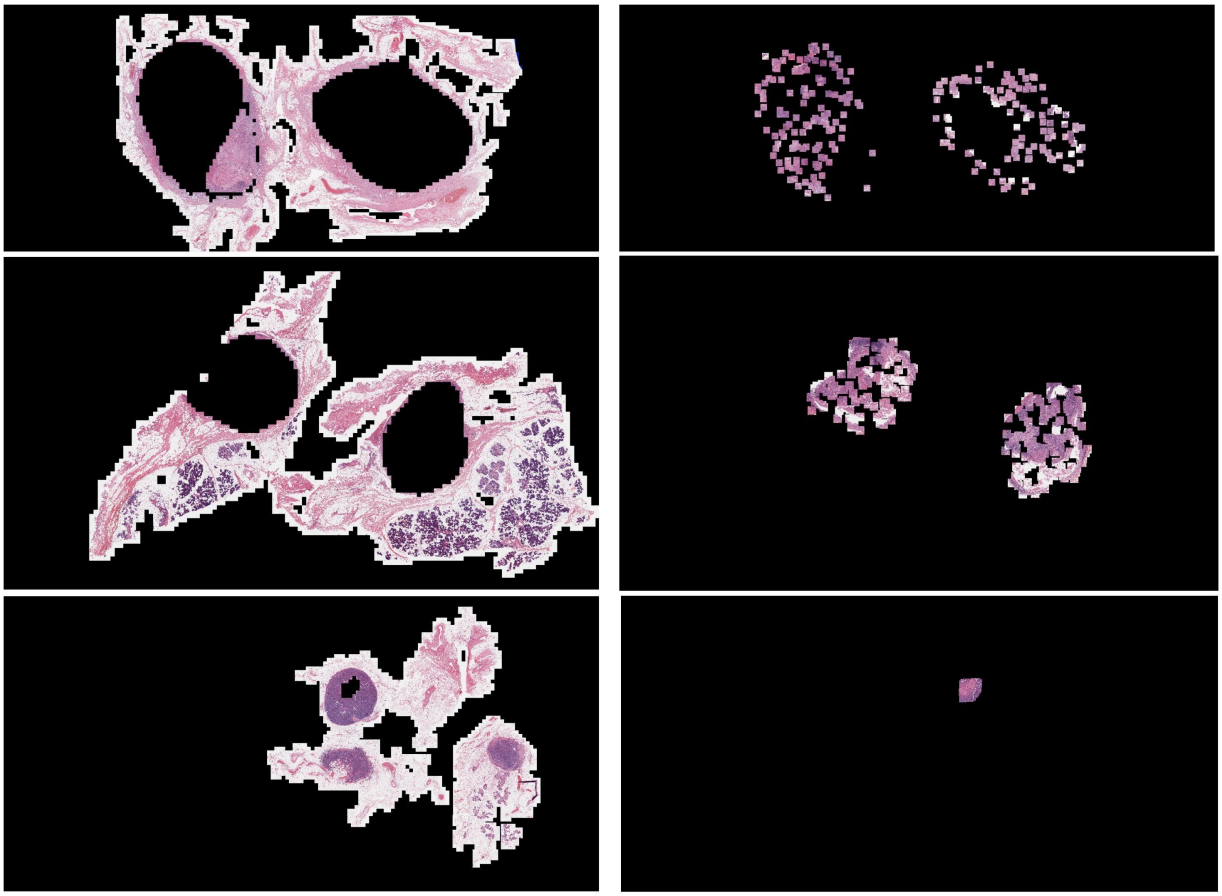

Figure 4.3: Patches extracted from WSIs stitched together on a dark background. (Left)non-tumor patches, (right) tumor patches. The difference in patch extraction strategies in non-tumor(sliding window-based) and tumor(randomly sampled) regions can be seen

4.3 Training

The patches extracted were then used to train our tumor detection model. Three different state-of-the-art architectures were trained and analyzed for this task. We used ResNet-34, ResNet-50 and EfficientNet-B2 for our analysis. The performance of these models on training, validation, and test data for patch-level tumor detection has been summarised in Table 4.1.

As we can analyze from the table, EfficientNet-B2 significantly outperforms both ResNet-34 and ResNet-50 on almost all metrics for validation and test data. EfficientNet-B2

Metrics		ResNet-34	ResNet-50	EfficientNet-B2
Training	Accuracy	0.95	0.96	0.95
	Recall	0.95	0.97	0.95
	Precision	0.94	0.955	0.92
	F1-Score	0.95	0.965	0.94
Validation	Accuracy	0.91	0.92	0.94
	Recall	0.85	0.89	0.93
	Precision	0.95	0.93	0.94
	F1-Score	0.9	0.91	0.935
Test	Accuracy	0.86	0.9	0.91
	Recall	0.82	0.84	0.88
	Precision	0.91	0.94	0.93
	F1-Score	0.86	0.88	0.90

Table 4.1: Table showing the training, test and validation performances of the three models for patch classification/tumor detection

achieves the highest F1-score among the three models by a significant margin. It generalizes better than the two models and has a lower false negative rate which is important for medical predictions. ResNet-50 was not significantly inferior to EfficientNet-B2, but it requires 3 times more parameters(27M) compared to EfficientNet-B2(9M). Hence the model size for ResNet-50 was considerably higher compared to EfficientNet-B2 for inferior performance. Figure 4.4 depicts the Receiver Operating Characteristic(ROC) curves for the ResNet-50 and EfficientNet-B2 model along with their AUC scores. Figure 4.5 contains convergence plots of important training and validation metrics obtained over the course of training ResNet-50. Similarly Figure 4.6 shows the convergence curves for EfficientNet-B2 for three different runs.

4.3.1 Resampling

We experimented with two resampling strategies to train efficiently on our imbalanced data. We used the ResNet-34 model to train using these strategies and benchmarking their performance. In the first iteration, we used the inbuilt weighted random sampling function in pytorch to sample data from classes using weights based on probabilities with

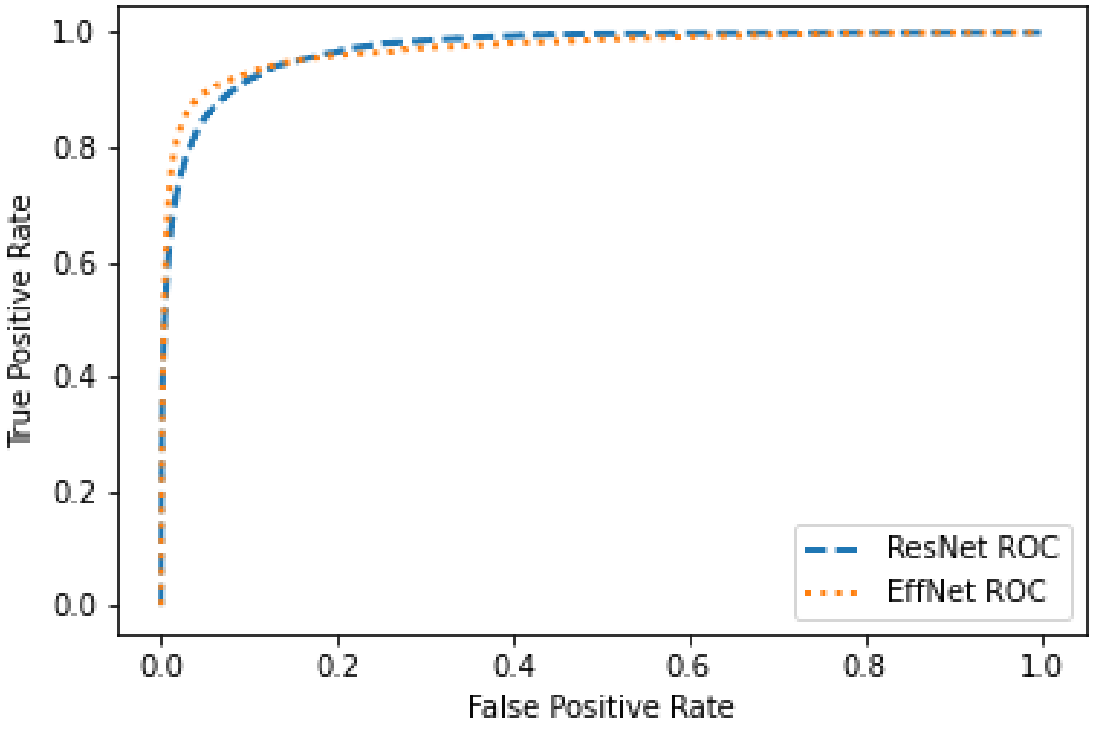

Figure 4.4: ROC Curves for ResNet-50 and EfficientNet-B2, AUC under the curves were 0.95 and 0.97 respectively

repetition. In the second iteration, we first undersampled the majority non-tumor data by a factor of 7 to bring the ratio of the two classes to 10:1, which made learning from it more tractable, before using the weighted random sampling. The second strategy performed better as the former overfit due to massive data imbalance and repetition which set a performance ceiling. In the latter case we reduced the ratio of the classes which caused lower repetition and hence the model did not overfit. Undersampling also significantly reduced the time required for training the model. The performance and observations have been tabulated in Table 4.2

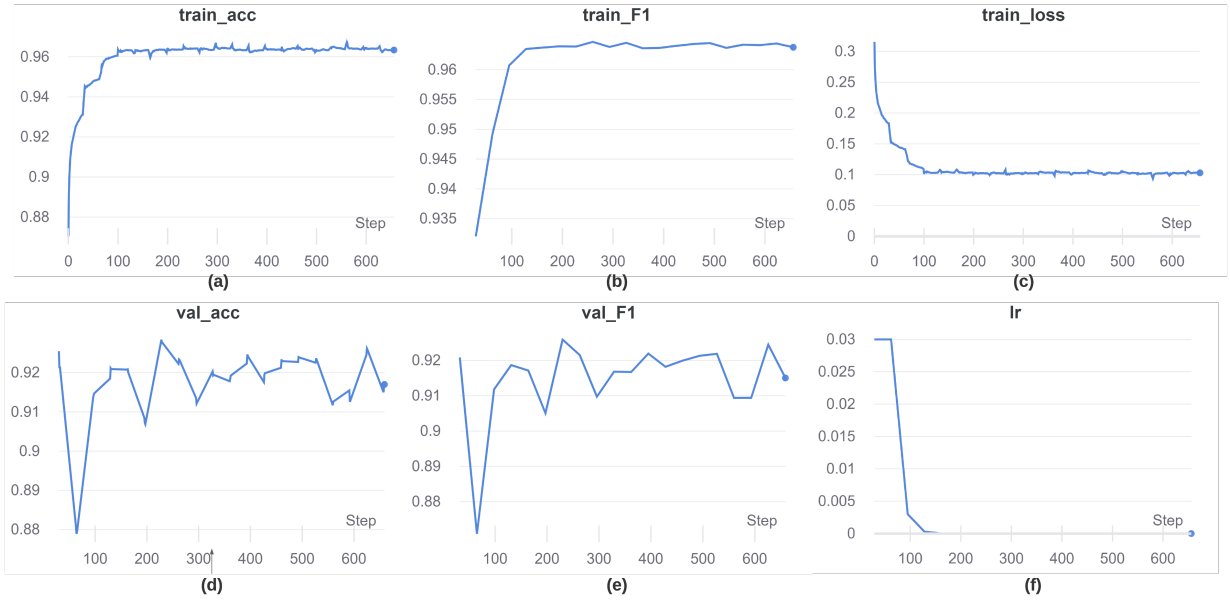

Figure 4.5: Curves showing changes in (a) Training Accuracy, (b) Training F1 Score, (c) Training Loss, (d) Validation Accuracy, (e) Validation F1 Score, (f)Learning Rate across the training steps for ResNet 50

	Weighted Random Sampling	**Undersampling + Weighted Random Sampling**
Training	Accuracy 0.97, F1 Score 0.95	Accuracy 0.96, F1 Score 0.95
Validation	Accuracy 0.85, F1 Score 0.84	Accuracy 0.9, F1 Score 0.9
Comments	Overfit in 4th epoch, Slower to train, Much lower recall(0.77)	Overfit in 8th epoch, much faster training, Much improved recall(0.85)

Table 4.2: ResNet-34 performance when trained using different resampling strategies

4.3.2 Hyperparameters

We experimented with a variety of hyperparameters over different runs of the training procedure for each architecture. These included learning rates, optimizers, batch sizes, the number of epochs, and the custom FC layers added on top of pre-trained convolutional layers. Table 4.3 summarizes the final optimum values of these hyperparameters used for training the ResNet-50 and EfficientNet-50 finalized after several runs. Since ResNet-50 has more parameters, we used a smaller batch size for training. ResNet-50 started overfitting early, hence 10 epochs were enough for optimally training it, whereas

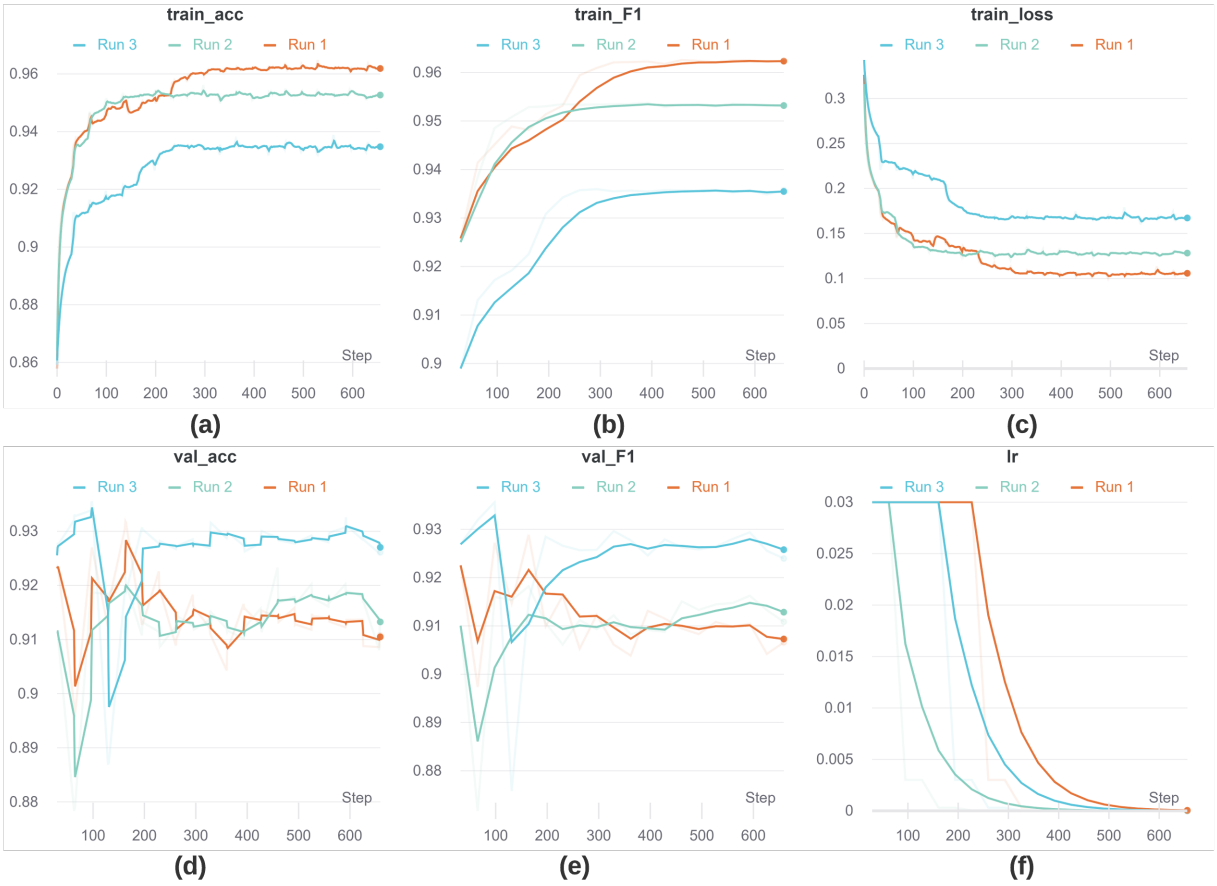

Figure 4.6: Curves showing changes in (a) Training Accuracy, (b) Training F1 Score, (c) Training Loss, (d) Validation Accuracy, (e) Validation F1 Score, (f)Learning Rate across the training steps for three different runs of EfficientNet-B2

EfficientNet-B2 was trained for 20 epochs. We used Adam optimizer for training both the models, EfficientNet-B2 with its smaller size allowed us to use a deeper fully connected layer on top of its convolutional layer without overfitting.

4.4 Inference and Slide level performance

For inference an end-to-end pipeline that took Whole Slide Images was taken, passed them through the denoising, segmentation and patch extraction, and detection/patch classification steps and returned binary prediction masks, heatmaps of prediction prob-

	ResNet-50	**EfficientNet-B2**
Learning Rate	1e-3(initially), then dynamically changed using a LR scheduler	3e-2(initially), then dynamically changed using a LR scheduler
Number of Epochs	10	20
Batch Size	128	256
FC Layer	Two linear layers with 512 and 128 units respectively	4 linear layers with 512,512, 128, 64 layers along with BatchNorm and dropout layers
Optimizer	Adam	Adam

Table 4.3: Values of selected hyperparameters used for training the ResNet-50 and EfficientNet-B2 models for tumor detection

	Slide Level Classification Accuracy	**Average IOU**
ResNet-50	0.86	0.62
EfficientNet-B2	0.89	0.7

Table 4.4: Performance of the patch-based classifier on the whole slide level, analysed on 91 test WSIs

abilities overlaid on top of a downscaled version of the WSI, IOU score, and slide level prediction performance. Table 4.4 shows the performance of ResNet-50 and EfficientNet-B2 in terms of slide level classification accuracy and average IOU score analyzed on 100 test Whole Slide Images.

Figure 4.7 and Figure 4.8 show the performance of our model on Whole Slide Images containing tumor while Figures 4.9 and 4.10 show the performance on non-cancerous WSIs. It is important to note that the IOU, heatmap, coarse segmentation mask are essentially meant to analyze visually the performance of the patch-level classifier and not the tasks for which the model is trained, hence these values are not to be taken as reflective of the model performance. The model performance for patch level classifier can be analyzed using Table 4.1

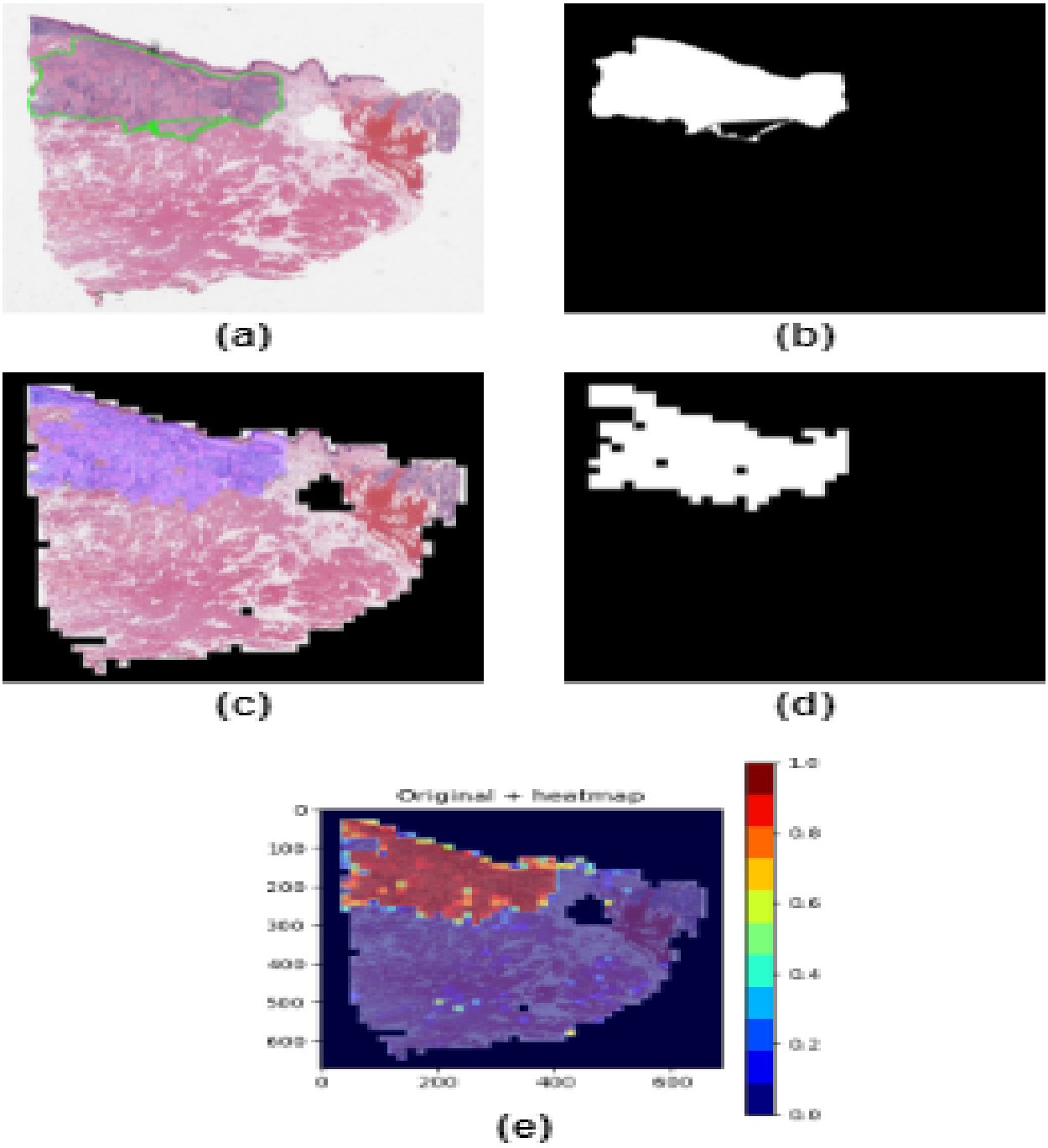

Figure 4.7: Sample results obtained for a WSI containing tumor(a) Annotated Slide, (b) Ground truth binary tumor mask, (c) WSI overlaid with prediction mask, (d) Binary Prediction Mask, (e) Prediction heatmap overlaid on tissue segmented WSI

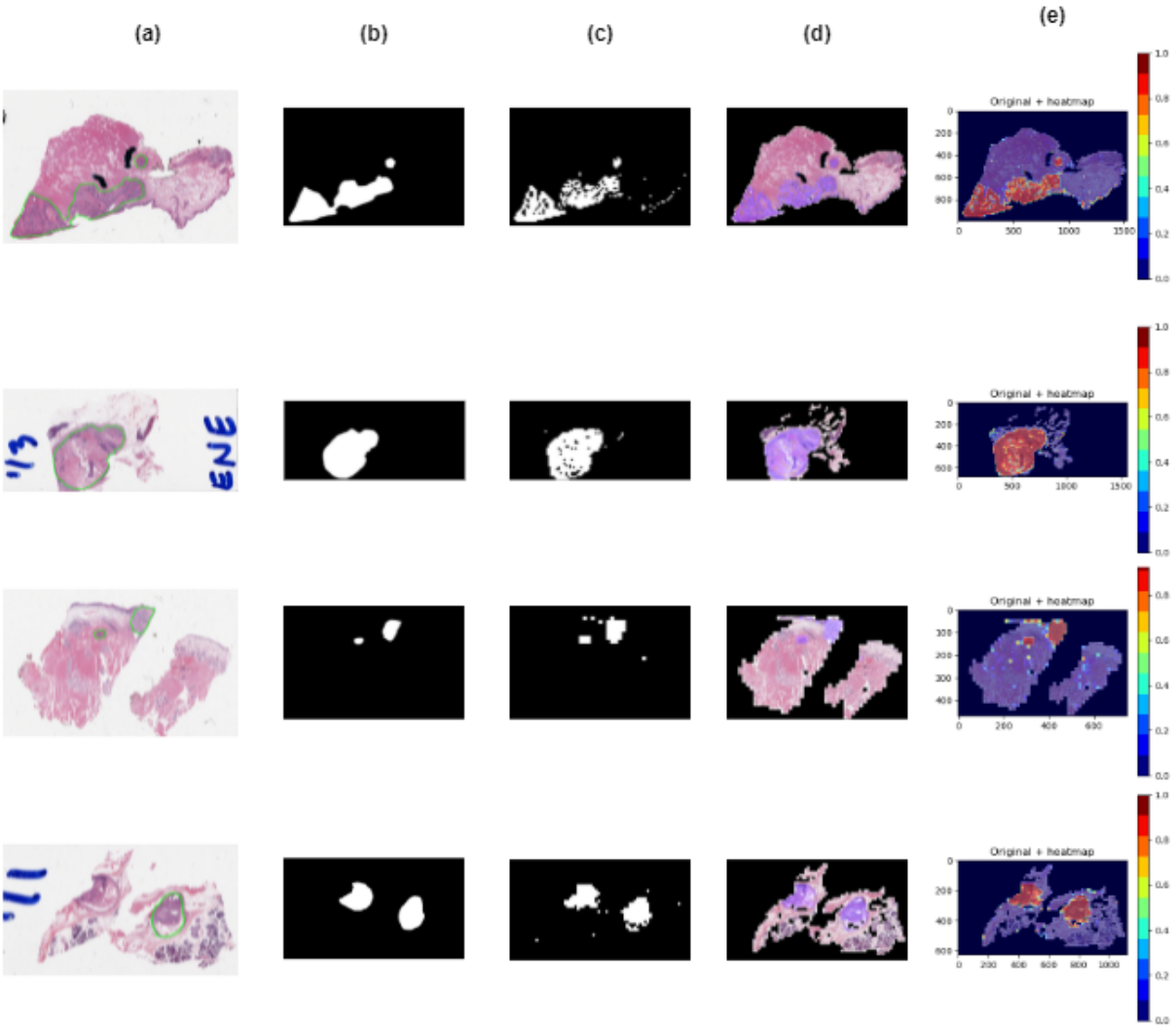

Figure 4.8: Few more results for WSIs containing tumor(a) Annotated Slide, (b) Ground truth binary tumor mask, (c) WSI overlaid with prediction mask, (d) Binary Prediction Mask, (e) Prediction heatmap overlaid on tissue segmented WSI

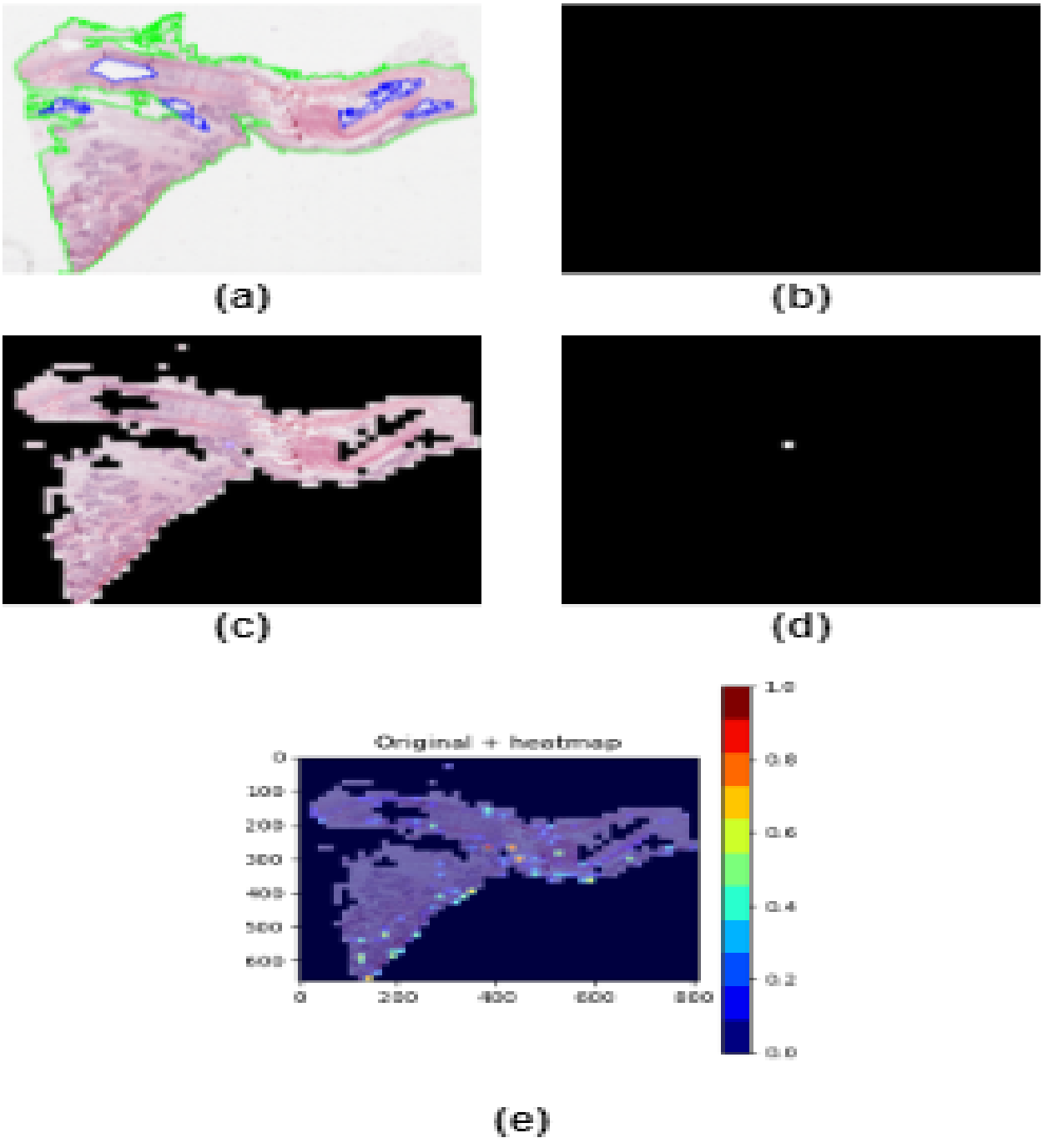

Figure 4.9: Sample results obtained for a WSI not containing tumor(a) Tissue segmentation mask, (b) Ground truth binary tumor mask, (c) WSI overlaid with prediction mask, (d) Binary Prediction Mask, (e) Prediction heatmap overlaid on tissue segmented WSI

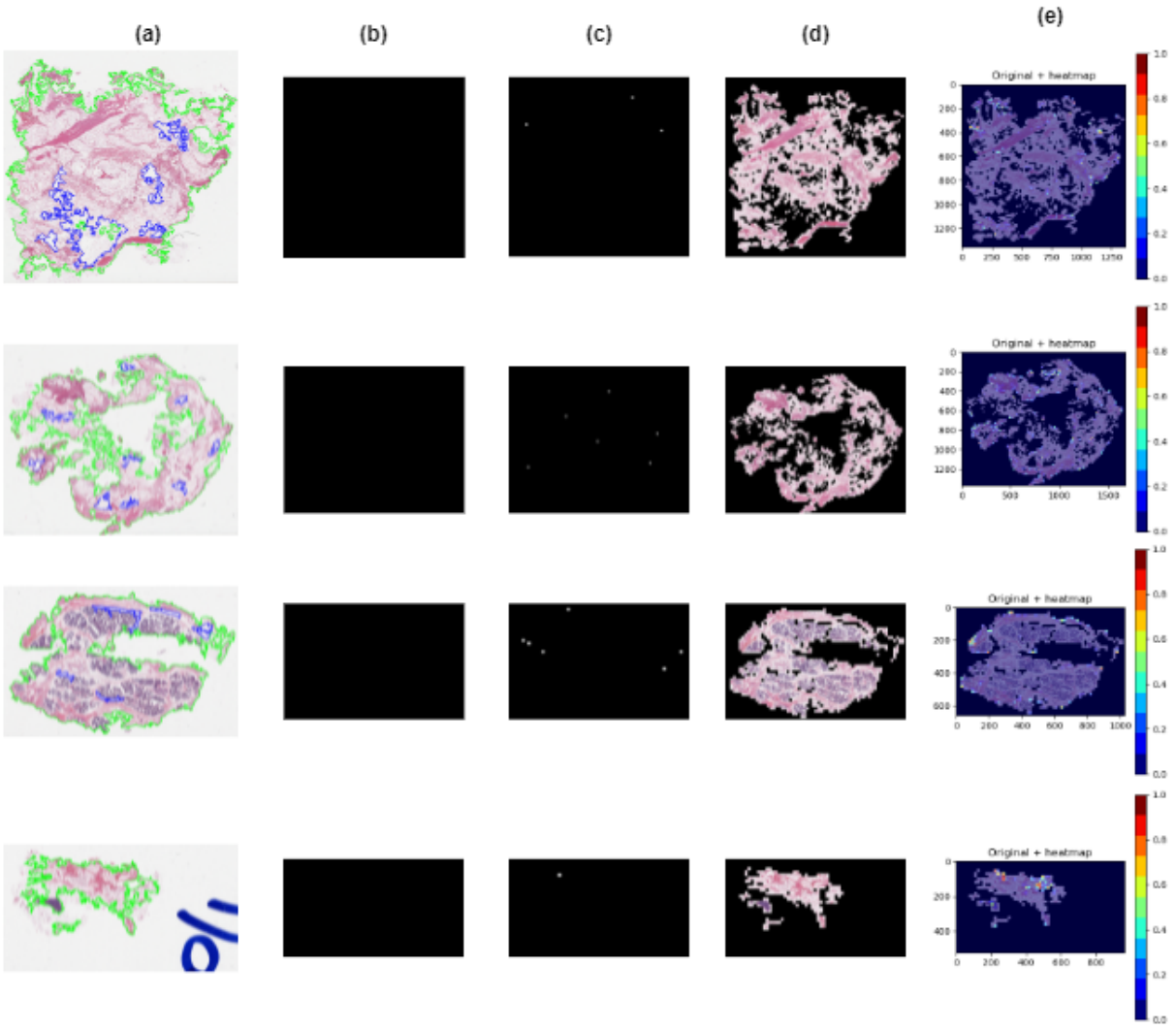

Figure 4.10: Few more results for WSIs not containing tumor(a) Tissue segmentation masks, (b) Ground truth binary tumor mask, (c) WSI overlaid with prediction mask, (d) Binary Prediction Mask, (e) Prediction heatmap overlaid on tissue segmented WSI

Chapter 5

Discussion

This thesis primarily involved exploring the application of CNN models for patch-based tumor detection in head and neck histology images. The goal was to build an accurate, efficient end-to-end algorithm to classify patches extracted from Whole Slide Images as cancerous or otherwise. The thesis intended to prove the capability of state-of-the-art CNN architectures trained on real-world data such as ImageNet object classification dataset[37] to be able to transfer to the digital pathology domain when fine-tuned on manually annotated patch level microscopy data. We also set out to build an efficient preprocessing modules which would help in denoising, region of interest segmentation, and patch extraction from the WSIs. Another important objective of the thesis was to build an end-to-end high throughput integrated system for preprocessing and inference on WSIs. Essentially this was both a proof-of-concept and a benchmarking based project with additional software byproducts such as the inference system.

Most of the relevant previous work in head and neck digital pathology involved standard computer vision algorithms for feature extraction and modeling. One of the popular recent works that leveraged deep learning for head and neck cancer detection [14] approached this problem in a weakly supervised setting taking random patches and assigning the same label as the whole slide, in the absence of finer annotation. They used much smaller patches of sizes $101x101$ randomly sampled from downsampled cancerous

and non-cancerous WSIs, They had an artificially balanced dataset, made so by sampling an equal number of tumor and non-tumor patches or instances which is not representative of actual data since tumor regions are rare and small. Working in a fully supervised setting, as done in this thesis, allows us to have a more realistic dataset since the patches have individual labels. The authors used an Inception-V4 CNN architecture while this thesis studies and analyzes state-of-the-art architectures such as ResNet and EfficientNet.

As we can see from 4.1, all the three models trained for tumor detection performed exceptionally well on all the metrics such as accuracy, F1-score, and AUC. The best-performing model used an EfficientNet-B2 architecture with a custom 4-layer fully connected network on top of the pre-trained convolution layers. It achieved an accuracy of 91% and an F1-score of 0.90 on unseen test patch data. It also achieved an accuracy of 89% at the whole slide level where the classification was done using an aggregation heuristic where the top 50 highest probability patches are averaged to get a slide level score. This is a significant yet expected improvement upon the work by Halicek et al [14] which achieved an accuracy of 85% in Head and neck squamous cell carcinoma in a weakly supervised setting using the Inception-V4 model. We also built a low latency end-to-end system for performing inference on WSIs using the tumor detection models. It is able to preprocess, extract patches, perform inference and generate prediction masks,a heatmap and other metrics taking about 16 seconds on average per Slide. The results are visualized in Figures 4.7, 4.8, 4.9, and 4.10. A detailed analysis of several architectures, sampling strategies, hyperparameters was also done along with a visual and pixel-wise IOU based analysis of coarse segmentation performance at the whole slide level. The results of these experiments analysed using a variety of metrics can be seen in Tables 4.1, 4.2, 4.3, and 4.4.

5.1 Limitation and Future Scope

This thesis lays a strong foundation for further research into the application of deep learning to head and neck histology while benchmarking the performance of state-of-the-art models on the tumor detection task. A large body of research is already being conducted towards leveraging deep learning for medical applications.

The major limitation of our tumor detection/patch classification system is the same as the limitation of any patch-based approach, which is the general loss of global contextual information due to limited region of focus when working with patches, as well as the computational overhead encountered due to fragmented processing. Other important limitations include the requirement of fine expert annotations for better performance, possible lack of generality when working with WSIs from other parts of the body, or WSIs stained differently, or in case of more varied, multi-class classification.

In order to mitigate this problem, Generative models that use an encoder-decoder architecture such as autoencoders or GANs can be trained to learn discriminative, low dimensional latent representation. Graph representation learning can also be leveraged to transform large WSIs along with all of their contextual and spatial information in computationally tractable graph data structures and training specialized graph neural networks on them. The models could also be used as the backbone while working with novel concepts such as visual attention, which would considerably lower the latency for slide level prediction by reducing the number of patches required for predicting by selecting only a few most representative patches using a weighting function.

Models trained in a fully supervised setting like the ones in this thesis can also be used a pre-trained models to be further fine-tuned in a semi-supervised and weakly-supervised scenario where we have only a fraction of the data labeled by experts. They can further be used as a backbone networks for the self-supervised or contrastive learning approaches which could prove very useful in digital pathology due to the general

unavailability of annotated data.This will help address the limited availability of annotated data

Further, the models can be quantized, compressed, frozen, and deployed as browser-based models or on edge devices like mobiles, embedded systems to make diagnosis more efficient, affordable, accessible, and advance the field of telemedicine. They can also be extended to predicting severity, treatment regime, as well as for finer segmentation.

5.2 Conclusion

Based on the results discussed above, it can be concluded that patch-based algorithms provide an efficient workaround to the problems of computational intractability caused due to the high resolution of Whole Slide Images. A bottom-up approach to slide level prediction is possible using a meta learner or some other aggregation heuristic on top of the patch classifiers. This thesis employs a simplistic heuristic to achieve this

Standard computer vision methods can be applied for efficient preprocessing and handling of digital pathology data as seen in this thesis where such classical vision algorithms were employed for denoising, tissue segmentation, etc. This is possible due to the distinctive nature of tissue regions, background, and artifacts present in the WSIs.

Image augmentation such as spatial transformations and color transformations helps improve the generalization ability of the models and prevent overfitting. Ideal transformations are rotation, flipping, contrast adjustment, blurring, noise addition, color jitter, brightness adjustment, etc.

Multiple trade-offs were made throughout the thesis including the decision of patch size and resolution to ensure optimal information is captured, trade-off during sampling to ensure less imbalance while also maintaining a realistic representation of the classes, trade-off in terms of the metrics such as precision and recall. In the specific case of diagnosis recall was the most significant metric based on which model decisions were made to ensure low false negatives. Accuracy turned out to not be an appropriate measure

of model efficacy in the case of an imbalanced dataset and F1-score was a much more important metric.

All in all the major contribution of this thesis was to show a deep learning patch-based end-to-end tumor detection system that can be built for digital pathology with low latency, high throughput, and reliable performance at both patch level and slide level that can be easily deployed, retrained and used to visualize results thus assisting pathologists in pre-screening.

Bibliography

[1] *The cancer genome atlas program - national cancer institute*. June 2018a. URL: https://www.cancer.gov/about-nci/organization/ccg/research/structural-genomics/tcga.

[2] Morgan P McBee et al. "Deep learning in radiology". In: *Academic radiology* 25.11 (2018), pp. 1472–1480.

[3] Maciej A Mazurowski et al. "Deep learning in radiology: An overview of the concepts and a survey of the state of the art with focus on MRI". In: *Journal of magnetic resonance imaging* 49.4 (2019), pp. 939–954.

[4] Mary Feng et al. "Machine learning in radiation oncology: opportunities, requirements, and needs". In: *Frontiers in oncology* 8 (2018), p. 110.

[5] EJ Limkin et al. "Promises and challenges for the implementation of computational medical imaging (radiomics) in oncology". In: *Annals of Oncology* 28.6 (2017), pp. 1191–1206.

[6] Anant Madabhushi and George Lee. "Image analysis and machine learning in digital pathology: Challenges and opportunities". In: *Medical image analysis* 33 (2016), pp. 170–175.

[7] Chetan L Srinidhi, Ozan Ciga, and Anne L Martel. "Deep neural network models for computational histopathology: A survey". In: *Medical Image Analysis* (2020), p. 101813.

[8] Dan C Cireşan et al. "Mitosis detection in breast cancer histology images with deep neural networks". In: *International conference on medical image computing and computer-assisted intervention*. Springer. 2013, pp. 411–418.

[9] Hai Su et al. "Region segmentation in histopathological breast cancer images using deep convolutional neural network". In: *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*. IEEE. 2015, pp. 55–58.

[10] Lisheng Wei, Quan Gan, and Tao Ji. "Cervical cancer histology image identification method based on texture and lesion area features". In: *Computer Assisted Surgery* 22.sup1 (2017), pp. 186–199.

[11] Bogdan Obrzut et al. "Prediction of 5–year overall survival in cervical cancer patients treated with radical hysterectomy using computational intelligence methods". In: *BMC cancer* 17.1 (2017), pp. 1–9.

[12] Dev Kumar Das et al. "Computational approach for mitotic cell detection and its application in oral squamous cell carcinoma". In: *Multidimensional Systems and Signal Processing* 28.3 (2017), pp. 1031–1050.

[13] Muhammad Shaban et al. "A novel digital score for abundance of tumour infiltrating lymphocytes predicts disease free survival in oral squamous cell carcinoma". In: *Scientific reports* 9.1 (2019), pp. 1–13.

[14] Martin Halicek et al. "Head and neck cancer detection in digitized whole-slide histology using convolutional neural networks". In: *Scientific reports* 9.1 (2019), pp. 1–11.

[15] Lubaina Ehsan et al. "Prediction of Celiac Disease Severity and Associated Endocrine Morbidities through Deep Learning-based Image Analytics." In: *medRxiv* (2021).

[16] Rasoul Sali et al. "Celiacnet: Celiac disease severity diagnosis on duodenal histopathological images using deep residual networks". In: *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE. 2019, pp. 962–967.

[17] Pinaki Sarder, Brandon Ginley, and John E Tomaszewski. "Automated renal histopathology: Digital extraction and quantification of renal pathology". In: *Medical Imaging 2016: Digital Pathology*. Vol. 9791. International Society for Optics and Photonics. 2016, 97910F.

[18] Fuyong Xing et al. "Deep learning in microscopy image analysis: A survey". In: *IEEE transactions on neural networks and learning systems* 29.10 (2017), pp. 4550–4568.

[19] Nina Linder et al. "Identification of tumor epithelium and stroma in tissue microarrays using texture analysis". In: *Diagnostic pathology* 7.1 (2012), pp. 1–11.

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

[21] Jocelyn Barker et al. "Automated classification of brain tumor type in whole-slide digital pathology images using local representative tiles". In: *Medical image analysis* 30 (2016), pp. 60–71.

[22] Jun Xu et al. "A deep convolutional neural network for segmenting and classifying epithelial and stromal regions in histopathological images". In: *Neurocomputing* 191 (2016), pp. 214–223.

[23] Le Hou et al. "Patch-based convolutional neural network for whole slide tissue image classification". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2424–2433.

[24] Zhi-Hua Zhou et al. "Lung cancer cell identification based on artificial neural network ensembles". In: *Artificial intelligence in medicine* 24.1 (2002), pp. 25–36.

[25] Gustavo Carneiro et al. "Weakly-supervised structured output learning with flexible and latent graphs using high-order loss functions". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 648–656.

[26] Edward Kim, Zubair Baloch, and Caroline Kim. "Computer assisted detection and analysis of tall cell variant papillary thyroid carcinoma in histological images". In: *Medical Imaging 2015: Digital Pathology*. Vol. 9420. International Society for Optics and Photonics. 2015, 94200A.

[27] J Angel Arul Jothi and V Mary Anita Rajam. "Automatic classification of thyroid histopathology images using multi-classifier system". In: *Multimedia Tools and Applications* 76.18 (2017), pp. 18711–18730.

[28] Wei Wang, John A Ozolek, and Gustavo K Rohde. "Detection and classification of thyroid follicular lesions based on nuclear structure from histopathology images". In: *Cytometry Part A: The Journal of the International Society for Advancement of Cytometry* 77.5 (2010), pp. 485–494.

[29] Balasubramanian Gopinath and Natesan Shanthi. "Computer-aided diagnosis system for classifying benign and malignant thyroid nodules in multi-stained FNAB cytological images". In: *Australasian physical & engineering sciences in medicine* 36.2 (2013), pp. 219–230.

[30] Antonis Daskalakis et al. "Design of a multi-classifier system for discriminating benign from malignant thyroid nodules using routinely H&E-stained cytological images". In: *Computers in biology and medicine* 38.2 (2008), pp. 196–203.

[31] John A Ozolek et al. "Accurate diagnosis of thyroid follicular lesions from nuclear morphology using supervised learning". In: *Medical image analysis* 18.5 (2014), pp. 772–780.

[32] Thomas J Fuchs and Joachim M Buhmann. "Computational pathology: challenges and promises for tissue analysis". In: *Computerized Medical Imaging and Graphics* 35.7-8 (2011), pp. 515–530.

[33] Chensu Xie et al. "Beyond classification: Whole slide tissue histopathology analysis by end-to-end part learning". In: *Medical Imaging with Deep Learning*. PMLR. 2020, pp. 843–856.

[34] Davood Karimi et al. “Deep Learning-Based Gleason grading of prostate cancer from histopathology Images—Role of multiscale decision aggregation and data augmentation”. In: *IEEE journal of biomedical and health informatics* 24.5 (2019), pp. 1413–1426.

[35] Hawraa Haj-Hassan et al. “Classifications of multispectral colorectal cancer tissues using convolution neural network”. In: *Journal of pathology informatics* 8 (2017).

[36] Ming Y Lu et al. “Data-efficient and weakly supervised computational pathology on whole-slide images”. In: *Nature Biomedical Engineering* 5.6 (2021), pp. 555–570.

[37] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[38] Lisa Torrey and Jude Shavlik. “Transfer learning”. In: *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. IGI global, 2010, pp. 242–264.

[39] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[40] Mingxing Tan and Quoc Le. “Efficientnet: Rethinking model scaling for convolutional neural networks”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 6105–6114.

[41] Michael H Ross and Wojciech Pawlina. *Histology*. Lippincott Williams & Wilkins, 2006.

[42] Karim M Khan et al. “Histopathology of common tendinopathies”. In: *Sports medicine* 27.6 (1999), pp. 393–408.

[43] Walter Frederick Lever et al. “Histopathology of the Skin.” In: *Histopathology of the skin.* (1949).

[44] Michael M Paparella. “A Review of Histopathology”. In: *Annals of Otology, Rhinology & Laryngology* 89.2_suppl3 (1980), pp. 1–10.

[45] Shaimaa Al-Janabi, André Huisman, and Paul J Van Diest. "Digital pathology: current status and future perspectives". In: *Histopathology* 61.1 (2012), pp. 1–9.

[46] Laura Barisoni et al. "Digital pathology and computational image analysis in nephropathology". In: *Nature Reviews Nephrology* 16.11 (Nov. 2020), pp. 669–685. ISSN: 1759-5061, 1759-507X.

[47] Andrew H Fischer et al. "Paraffin embedding tissue samples for sectioning." In: *CSH protocols* 2008 (2008), pdb–prot4989.

[48] Andrew H Fischer et al. "Hematoxylin and eosin staining of tissue and cell sections". In: *Cold spring harbor protocols* 2008.5 (2008), pdb–prot4986.

[49] Liron Pantanowitz et al. "Twenty years of digital pathology: an overview of the road travelled, what is on the horizon, and the emergence of vendor-neutral archives". In: *Journal of pathology informatics* 9 (2018).

[50] Adam Goode et al. "OpenSlide: A vendor-neutral software foundation for digital pathology". In: *Journal of pathology informatics* 4 (2013).

[51] David A Gutman et al. "The digital slide archive: A software platform for management, integration, and analysis of histology for cancer research". In: *Cancer research* 77.21 (2017), e75–e78.

[52] Joshua J Levy et al. "PathFlowAI: a high-throughput workflow for preprocessing, deep learning and interpretation in digital pathology". In: *PACIFIC SYMPOSIUM ON BIOCOMPUTING 2020*. World Scientific. 2019, pp. 403–414.

[53] Yves-Rémi Van Eycke et al. "Image processing in digital pathology: an opportunity to solve inter-batch variability of immunohistochemical staining". In: *Scientific reports* 7.1 (2017), pp. 1–15.

[54] Zaneta Swiderska-Chadaj et al. "Impact of rescanning and normalization on convolutional neural network performance in multi-center, whole-slide classification of prostate cancer". In: *Scientific Reports* 10.1 (2020), pp. 1–14.

[55] David Tellez et al. “Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology”. In: *Medical image analysis* 58 (2019), p. 101544.

[56] Abhishek Vahadane et al. “Structure-preserved color normalization for histological images”. In: *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*. IEEE. 2015, pp. 1012–1015.

[57] Andrew Janowczyk, Ajay Basavanhally, and Anant Madabhushi. “Stain normalization using sparse autoencoders (StaNoSA): application to digital pathology”. In: *Computerized Medical Imaging and Graphics* 57 (2017), pp. 50–61.

[58] Amit Sethi et al. “Empirical comparison of color normalization methods for epithelial-stromal classification in H and E images”. In: *Journal of pathology informatics* 7 (2016).

[59] Péter Bándi et al. “Comparison of different methods for tissue segmentation in histopathological whole-slide images”. In: *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*. IEEE. 2017, pp. 591–595.

[60] VB Surya Prasath et al. “Segmentation of breast cancer tissue microarrays for computer-aided diagnosis in pathology”. In: *First IEEE Healthcare Technology Conference: Translational Engineering in Health & Medicine, Houston, TX, USA*. 2012, pp. 40–43.

[61] *Apply filters for tissue segmentation*. URL: https://developer.ibm.com/technologies/data-science/articles/an-automatic-method-to-identify-tissues-from-big-whole-slide-images-pt2/.

[62] Thaına A Azevedo Tosta et al. “Evaluation of statistical and Haralick texture features for lymphoma histological images classification”. In: *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization* (2021), pp. 1–12.

[63] S Simonthomas. "Pattern Analysis for detecting Pathology Using Haralick Texture Features". In: *International Journal of Computer Science and Network Security (IJCSNS)* 15.11 (2015), p. 53.

[64] Omar S Al-Kadi. "A gabor filter texture analysis approach for histopathological brain tumor subtype discrimination". In: *arXiv preprint arXiv:1704.05122* (2017).

[65] Riku Turkki et al. "Assessment of tumour viability in human lung cancer xenografts with texture-based image analysis". In: *Journal of clinical pathology* 68.8 (2015), pp. 614–621.

[66] Mitko Veta et al. "Automatic nuclei segmentation in H&E stained breast cancer histopathology images". In: *PloS one* 8.7 (2013), e70221.

[67] Shivang Naik et al. "Automated gland and nuclei segmentation for grading of prostate and breast cancer histopathology". In: *2008 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. IEEE. 2008, pp. 284–287.

[68] Shivang Naik et al. "Gland segmentation and computerized gleason grading of prostate histology by integrating low-, high-level and domain specific information". In: *MIAAB workshop*. Citeseer. 2007, pp. 1–8.

[69] Sonal Kothari et al. "Pathology imaging informatics for quantitative analysis of whole-slide images". In: *Journal of the American Medical Informatics Association* 20.6 (2013), pp. 1099–1108.

[70] Talha Qaiser et al. "Persistent homology for fast tumor segmentation in whole slide histology images". In: *Procedia Computer Science* 90 (2016), pp. 119–124.

[71] Olga Russakovsky et al. "Imagenet large scale visual recognition challenge". In: *International journal of computer vision* 115.3 (2015), pp. 211–252.

[72] Haibo Wang et al. "Mitosis detection in breast cancer pathology images by combining handcrafted and convolutional neural network features". In: *Journal of Medical Imaging* 1.3 (2014), p. 034003.

[73] Muhammad Nasim Kashif et al. “Handcrafted features with convolutional neural networks for detection of tumor cells in histology images”. In: *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*. IEEE. 2016, pp. 1029–1032.

[74] David Romo-Bucheli et al. “Automated tubule nuclei quantification and correlation with oncotype DX risk categories in ER+ breast cancer whole slide images”. In: *Scientific reports* 6.1 (2016), pp. 1–9.

[75] Fuyong Xing, Yuanpu Xie, and Lin Yang. “An automatic learning-based framework for robust nucleus segmentation”. In: *IEEE transactions on medical imaging* 35.2 (2015), pp. 550–566.

[76] Youyi Song et al. “Accurate segmentation of cervical cytoplasm and nuclei based on multiscale convolutional network and graph partitioning”. In: *IEEE Transactions on Biomedical Engineering* 62.10 (2015), pp. 2421–2433.

[77] Zizhao Zhang et al. “Pathologist-level interpretable whole-slide cancer diagnosis with deep learning”. In: *Nature Machine Intelligence* 1.5 (2019), pp. 236–245.

[78] Angel Cruz-Roa et al. “High-throughput adaptive sampling for whole-slide histopathology image analysis (HASHI) via convolutional neural networks: Application to invasive breast cancer detection”. In: *PloS one* 13.5 (2018), e0196828.

[79] Talha Qaiser and Nasir M Rajpoot. “Learning where to see: A novel attention model for automated immunohistochemical scoring”. In: *IEEE transactions on medical imaging* 38.11 (2019), pp. 2620–2631.

[80] Bolei Xu et al. “Look, investigate, and classify: a deep hybrid attention method for breast cancer classification”. In: *2019 IEEE 16th international symposium on biomedical imaging (ISBI 2019)*. IEEE. 2019, pp. 914–918.

[81] Aıcha BenTaieb and Ghassan Hamarneh. “Predicting cancer with a recurrent visual attention model for histopathology images”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2018, pp. 129–137.

[82] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.

[83] Hao Chen, Xi Wang, and Pheng Ann Heng. "Automated mitosis detection with deep regression networks". In: *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*. IEEE. 2016, pp. 1204–1207.

[84] Yuanpu Xie et al. "Efficient and robust cell detection: A structured regression approach". In: *Medical image analysis* 44 (2018), pp. 245–254.

[85] Simon Graham et al. "Hover-net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images". In: *Medical Image Analysis* 58 (2019), p. 101563.

[86] Neeraj Kumar et al. "A multi-organ nucleus segmentation challenge". In: *IEEE transactions on medical imaging* 39.5 (2019), pp. 1380–1391.

[87] Nikhil Seth et al. "Automated segmentation of DCIS in whole slide images". In: *European Congress on Digital Pathology*. Springer. 2019, pp. 67–74.

[88] Jingxin Liu et al. "An end-to-end deep learning histochemical scoring system for breast cancer TMA". In: *IEEE transactions on medical imaging* 38.2 (2018), pp. 617–628.

[89] Zaneta Swiderska-Chadaj et al. "Learning to detect lymphocytes in immunohistochemistry with deep learning". In: *Medical image analysis* 58 (2019), p. 101547.

[90] Thomas de Bel et al. "Automatic segmentation of histopathological slides of renal tissue using deep learning". In: *Medical Imaging 2018: Digital Pathology*. Vol. 10581. International Society for Optics and Photonics. 2018, p. 1058112.

[91] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

[92] Wenqi Lu et al. “Capturing Cellular Topology in Multi-Gigapixel Pathology Images”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2020, pp. 260–261.

[93] Si Zhang et al. “Graph convolutional networks: a comprehensive review”. In: *Computational Social Networks* 6.1 (2019), pp. 1–23.

[94] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. “Solving the multiple instance problem with axis-parallel rectangles”. In: *Artificial intelligence* 89.1-2 (1997), pp. 31–71.

[95] Gwenolé Quellec et al. “Multiple-instance learning for medical image and video analysis”. In: *IEEE reviews in biomedical engineering* 10 (2017), pp. 213–234.

[96] Gabriele Campanella et al. “Clinical-grade computational pathology using weakly supervised deep learning on whole slide images”. In: *Nature medicine* 25.8 (2019), pp. 1301–1309.

[97] Hui Qu et al. “Weakly supervised deep nuclei segmentation using points annotation in histopathology images”. In: *International Conference on Medical Imaging with Deep Learning*. PMLR. 2019, pp. 390–400.

[98] Lin Yang et al. “Boxnet: Deep learning based biomedical image segmentation using boxes only annotation”. In: *arXiv preprint arXiv:1806.00593* (2018).

[99] Zhipeng Jia et al. “Constrained deep weak supervision for histopathology image segmentation”. In: *IEEE transactions on medical imaging* 36.11 (2017), pp. 2376–2388.

[100] Jun Xu et al. “Stacked sparse autoencoder (SSAE) for nuclei detection on breast cancer histopathology images”. In: *IEEE transactions on medical imaging* 35.1 (2015), pp. 119–130.

[101] Le Hou et al. “Sparse autoencoder for unsupervised nucleus detection and representation in histopathology images”. In: *Pattern recognition* 86 (2019), pp. 188–200.

[102] Xi Chen et al. “Infogan: Interpretable representation learning by information maximizing generative adversarial nets”. In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. 2016, pp. 2180–2188.

[103] Will Fischer et al. “Sparse coding of pathology slides compared to transfer learning with deep neural networks”. In: *BMC bioinformatics* 19.18 (2018), pp. 9–17.

[104] Hang Chang et al. “Unsupervised transfer learning via multi-scale convolutional sparse coding for biomedical applications”. In: *IEEE transactions on pattern analysis and machine intelligence* 40.5 (2017), pp. 1182–1194.

[105] Zhimin Gao et al. “HEp-2 cell image classification with deep convolutional neural networks”. In: *IEEE journal of biomedical and health informatics* 21.2 (2016), pp. 416–428.

[106] Mira Valkonen et al. “Cytokeratin-supervised deep learning for automatic recognition of epithelial cells in breast cancers stained for ER, PR, and Ki-67”. In: *IEEE transactions on medical imaging* 39.2 (2019), pp. 534–542.

[107] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[108] Christian Szegedy et al. “Going deeper with convolutions”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9.

[109] Andrew G Howard et al. “Mobilenets: Efficient convolutional neural networks for mobile vision applications”. In: *arXiv preprint arXiv:1704.04861* (2017).

[110] Gao Huang et al. “Densely connected convolutional networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708.