Deep Weakly Supervised Learning for Whole Slide Image Representation: A Multimodal Approach

by

Amir Safarpoor Kordbacheh

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2021

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Dr. Anant Madabhushi Professor, Dept. of Biomedical Engineering, Case Western Reserve University
Supervisor:	Dr. Hamid Tizhoosh Professor, Dept. of Systems Design, University of Waterloo
Internal Member:	Dr. Jonathan Kofman Professor, Dept. of Systems Design, University of Waterloo
Internal Member:	Dr. Apurva Narayan Adjunct Assistant Professor, Dept. of Systems Design, University of Waterloo
Internal-External Member:	Dr. Mark Crowley Assistant Professor, Dept. of Electrical and Computer Engineering, University of Waterloo

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Pathologists examine stained specimens under a microscope to diagnose a multitude of diseases. With the advancement of scanner technology in recent years, the entire tissue glass slide can now be scanned and saved as a digital image known as a *whole slide image* (WSI). The digitization of glass slides enables the use of digital image analysis tools to evaluate WSIs. Machine Learning (ML) and, more specifically, Deep Learning (DL) has piqued practitioners' interest because they deliver cutting-edge results without the need for feature engineering. However, the DL performance comes at a cost; training deep models necessitates massive amounts of data that must be manually labeled by domain experts. Hand-labeled datasets take time and resources to create, especially when specialized knowledge is required, as in medicine. As a result, practitioners are increasingly focusing on techniques that require less supervision. Furthermore, due to existing hardware limitations, high-dimensional digital slides impede the application of cutting-edge deep learning models. As a result, most learning methods frequently require pixel-level annotation and are best suited for simplified scenarios like working with small and manageable images.

In this thesis, two methods for representing WSIs with weakly labeled data will be proposed to address the challenges of WSI representation learning in digital pathology. First, a pipeline for learning WSI representation at low magnification is proposed. This algorithm allows for the low-cost use of deep learning methods at the slide level. A WSI is embedded into a fixed-length compact feature vector using a deep model, making it suitable for computer vision tasks such as classification and search. A multitask multi-instance learning paradigm based on Vision Transformers (ViTs) is also proposed to learn visual descriptors by learning to predict gene expressions from H&E WSIs. Not only does this approach connect the tissue morphology and transcriptomics domains, but it also generates a WSI representation that taps into the wealth of information contained in the molecular signature of the input. As a result, it is now possible to learn visual representations using rich gene-level data as the ultimate source of biological information while also providing a mechanism for translating visual knowledge to transcriptomics.

Finally, the proposed models have been trained and evaluated using renal cell cancer subtyping as a case study. TCGA, a publicly available cancer repository, is used to train the proposed models. The performance and superiority of the models are demonstrated by comparison to state-of-the-art models developed for WSI classification, search, and gene expression prediction. Lastly, the generalizability of the trained models was demonstrated by testing them on an independent (external) test cohort.

Acknowledgements

Throughout the writing of this dissertation, I have received a great deal of support and assistance. I would first like to thank my supervisor, Professor H.R. Tizhoosh, whose expertise was invaluable in formulating the research questions and methodology. Your insightful feedback pushed me to sharpen my thinking and brought my work to a higher level. You provided me with encouragement and patience throughout the duration of this project.

I would like to acknowledge my colleagues at KIMIA Lab for their wonderful collaboration. I would particularly like to single out Sobhan, I want to thank you for your patient support and for all of the opportunities I was given to further my research. I would like to extend my sincere thanks to Mitra for her unparalleled support. I cannot forget the friends who went through hard times together, cheered me on, and celebrated each accomplishment, Danial, Milad, Maryam H., and Soheila.

I would also like to thank my committee members, Professor Kofman, Dr. Crowley, and Dr. Narayan, for their valuable guidance throughout my studies. You provided me with the tools that I needed to choose the right direction and successfully complete my dissertation.

In addition, I would like to thank my parents for their wise counsel and sympathetic ear. You are always there for me. Finally, I could not have completed this dissertation without the support of my friends, Amirali, Hadi, Mahta, Mandana, Sahand, Maryam, Mohammad Ali, Hossein, Mostafa, Amir, Mahsa, Nargess, Hossein S., Hamid, Ali, Daniel, Benyamin, Shahim, Rui, Rinat, Jose, George, who provided stimulating discussions as well as happy distractions to rest my mind outside of my research.

Dedication

This thesis is dedicated to my mother. I doubt I would have finished my graduate studies if it weren't for her unending support and encouragement. I am grateful for all that you have done for me.

Table of Contents

List of Figures

List of Tables

Abbreviations

AP@K Average Precision@K 33, 64

AUC Area Under the Curve xiii, xiv, 37, 39, 40, 42, 62, 64

BH Benjamini-Hochberg xvi, 56

CAD Computer-Aided Diagnostic 7, 15

CAM Convolution Activation Map xiii, 41, 43, 44

ccRCC Clear Cell Carcinoma xiii, 31–33, 44, 54

CI Confidence Interval xiii, 37, 40, 42

CNN Convolutional Neural Network 1, 2, 15–18, 20, 22, 25, 27, 28, 30, 41, 65

crRCC Chromophobe Renal Cell Carcinoma 32, 33, 54

DeiT Data efficient image Transformers 25

DL Deep Learning 6–8, 13, 15, 16, 20, 27, 30, 45

FFPE Formalin-Fixed, Paraffin-Embedded 32, 53

FPKM Fragments Per Kilobase of transcript per Million xvi, 13, 53, 54

FPKM-UQ Fragments Per Kilobase of transcript per Million mapped reads Upper Quartile 46, 53

GBT Gradient Boosting Tree 16

GELU Gaussian Error Linear Units 48

GO Gene Ontology 69

GP Gaussian Processes xii, xiii, 37–39, 42, 64

GRU Gated Recurrent Unit 22

H&E Hematoxylin and Eosin 2, 5, 8, 9, 16, 32, 65–67

HS Holm-Šidák xvi, 56

IHC Immunohistochemistry 9

KL divergence Kullback–Leibler Divergence 35

LR Logistic Regression 36, 38

LSTM Long Short-Term Memory xi, 15, 22, 24

MAE Mean Absolute Error xvi, 56, 57

mAP@K Mean Average Precision@K 64

MI Mutual Information 42, 43

MIL Multiple Instance Learning 2, 4, 46, 65

ML Machine learning 1, 3, 4, 6, 7, 13, 18, 27

MLP Multi-layer Perceptron 17, 52

P@K Precision@K xii, xvi, 33–36, 38, 64

PCA Principal Component Analysis xiv, xv, 58, 60, 63

PLA Patch Level Aggregation 36, 38

PPA Patch Probabilities Aggregation 36, 38

pRCC Papillary Renal Cell Carcinoma 31–33, 54

RCC Renal Cell Carcinoma xii, xiii, 27, 31, 32, 35, 37, 41, 43, 58, 68

ReLU Rectified Linear Unit 19, 52

RMSE Root Mean Square Error xvi, 56, 57

RNN Recurrent Neural Network xi, 22, 24

ROC Receiver Operating Characteristic xiii, xiv, 37, 39, 40, 42, 62

RRMSE Relative Root Mean Squared Error xvi, 56, 57

SD Standard Deviation xii, xvi, 29, 35, 36, 38

t-SNE t-Distributed Stochastic Neighbor Embedding xii, xiii, 34, 35, 39, 42

TCGA The Cancer Genome Atlas xii–xiv, xvi, xvii, 3, 4, 15, 32–39, 44, 45, 47, 49, 52–60, 62, 64, 65, 68

ViT Vision Transformer 25

WSI Whole Slide Image xii–xvii, 1, 2, 4, 5, 8, 10, 15–17, 27–31, 33–39, 41–49, 51–54, 58–61, 63–68

Chapter 1

Introduction

1.1 Motivation

The evaluation of biopsy samples by pathology is the gold standard for cancer diagnosis. Pathologists traditionally have been using light microscopy to examine glass slides. As a result of the increasing adoption of whole slide imaging, pathology is currently undergoing a change. Remote primary diagnostic work, teleconsultation, workload efficiency, central clinical review, image analysis, virtual training, and research are some of the prospects of this transformation [104]. The massive digitized slide datasets were made possible thanks to the whole slide imaging. The combination of big data produced by whole slide imaging technology, Machine learning (ML), and computer vision algorithms opens up a world of computational pathology possibilities. For instance, a Convolutional Neural Network (CNN), and its various topologies surpassed all other traditional computer vision algorithms in a wide range of clinical applications, including cancer subtyping [41], WSI search [47], mitosis detection [113], and grading [15].

Data representation (or feature extraction) has a significant impact on the performance of machine learning algorithms. Pattern recognition algorithms require extensive preparation pipelines and data transformations in order to enable successful object and scene identification. Therefore, most of the research efforts in implementing machine learning algorithms are devoted to designing robust and expressive representation [13]. In particular, image representation is the footstone of high-level computer vision frameworks, regardless of the task. As a result, establishing a crisp visual representation is crucial. However, due to a lack of labeled data, hyper-dimensionality of images, polymorphism in pathology images, and the non-binary nature of the diagnosis, learning representations from pathology

images is a daunting problem [50, 104]. Various learning schemes can be used depending on the information available for the training set. An example of supervised learning is a training set that has a collection of instances. Each example is labeled with the name of the class to which it belongs. In addition, the objective of weakly supervised learning is to train a model that can use coarse-grained (i.e., slide-level) annotation. Then infer fine-grained (i.e., pixel/tile-level) labels using the trained model. Furthermore, a low-dimensional space is created by transforming unlabeled raw data into a low-dimensional space that captures the most key qualities of the input in unsupervised learning. On the other hand, transfer learning methods are concerned with the transfer of information from one domain to another [92].

Supervised learning. The classification, segmentation, and regression problems fall under traditional supervised learning algorithms. A pixel-by-pixel prediction is made in local-level classification tasks utilizing a small sliding window and annotation map. For instance, Cireşan *et al.* [20] applied CNN based pixel prediction to detect mitosis in Hematoxylin and Eosin (H&E) breast cancer histology images. An alternative approach is a global-level task in which the model focuses on a tile-based classification approach for WSI-level disease prediction. Cruz-Rao *et al.* proposed a simple CNN for detecting invasive ductal carcinoma in breast cancer images [22, 24]. Later, they [23] suggested a combination of CNN and adaptive sampling, as well as a gradient-based adaptive method, to solve the computational complexity caused by a dense tile-wise prediction. Regression models often focus on object detection and localization by regressing the likelihood of an object's central pixel. One of the applications for which regression models are suitable is the detection of cells and their nuclei in histopathology images [70, 118].

Weakly supervised learning. The concept behind weakly supervised learning is to take coarse-level annotations and infer fine-grained annotations automatically from them. Because instance-level labeling is less expensive than pixel-level annotation, the weakly supervised learning paradigm is well suited to many problems in the histopathology domain. Multiple Instance Learning (MIL) is one type of weakly supervised learning. Each sample in MIL is made up of a bag of instances and a label. In pathology, each WSI may be seen as a "bag of tiles". The WSI label might be primary diagnosis and/or primary site. The MIL method can be used to learn to detect global (bag level), local (instance level), or global-local (fusion of both) patterns in the bag, depending on the objective. For example, Campanella *et al.* [16] developed a MIL paradigm in which only slide level diagnosis was utilized to classify WSIs.

Unsupervised learning. The objective of unsupervised learning is to learn critical information about the data's inherent structure. These approaches are uncommon, as they are among ML strategies that are mostly in their infancy (perhaps with exception of clustering). Due to the possibility of mapping inputs into as many subsets as possible, the task in unsupervised learning is ambiguous. Most of these methods maximize the data's probability distribution within certain constraints. Given the constraints, they are able to narrow the solution space and form the required groupings. For example, Xu *et al.* [119] employed sparse autoencoders to perform unsupervised nuclei detection in histopathology images of breast cancer.

Transfer learning. The objective of transfer learning is to extract information from one domain and apply it to another domain by loosening the requirement of the train and test sets being independently identically distributed. For transfer learning in digital pathology image analysis, models trained suing ImageNet images such as VGG [89], Inception [97], ResNet [37], DenseNet [42], and a variety of other models are usually utilized. Riasatian *et al.* [80] have shown that by training deep models utilizing ImageNet pre-trained weights and large histopathology datasets such as TCGA, one can construct an exclusive feature extractor more suitable for the pathology image domain.

Taking into consideration the properties of learning schemes, each has benefits and limitations. Supervised learning shines when a big, high-quality expert-annotated dataset is available. Transfer learning uses an existing well-trained model to transfer knowledge to a domain of interest at a lower cost. Unsupervised learning, on the other hand, investigates the underlying structure of data in order to extract and possibly distil knowledge from it. However, specialist knowledge is still required for validation. Finally, weakly supervised learning uses vast amounts of weakly labeled data to learn and uncover meaningful information in the data. As a result, weakly supervised methods lay in between fully supervised and unsupervised techniques, reaping the benefits of both. When looking at the broad picture, a weakly supervised setup may be ideal for the histopathology image analysis problem since there are large digitized datasets that have been weakly labeled that are publicly available.

Pixel-level annotated datasets are usually small and disease-specific. To give an example, the Camelyon dataset [62] is comprised of hundreds of slides from metastatic breast cancer patients, some of which include pixel-level annotation defining the metastatic areas. The TCGA dataset, on the other hand, has over 33,000 slides from more than 11,000 patients encompassing 25 anatomic sites and 32 cancer types. Each patient may contain numerous slides, RNA-seq data, clinical data, and many more variables, making the TCGA,

despite some deficiencies, a goldmine for cancer research. As a result, each learning method is well-suited to a certain topic and data, whereas techniques with looser assumptions may be used to a broader range of datasets (i.e., weakly supervised learning in comparison with supervised learning).

Motivated by the advantages of weakly supervised learning and by having access to a large dataset like TCGA, two methods for WSI representation learning are suggested in this dissertation. In chapter 3 an efficient pipeline for WSI representation learning is proposed. The suggested method is capable of learning relevant patterns at the tile level first. Then it filters out irrelevant information such as background and ink marker. Eventually, an entire gigapixel slide is represented in a fixed-length feature vector. All of this is accomplished only through the use of slide-level diagnosis.

In Chapter 4, the gene expression is introduced as an extra modality to enforce the model to learn diagnostically relevant representation for a slide constrained to recovering bulk RNA-seq of the sample from the image. A MIL method is used to develop the WSI representation learning and pixel-to-gene expression translation. In a multitask MIL framework, a Transformer-based model is utilized to analyze a bag of samples in order to predict both bulk RNA-seq and the primary diagnosis. Following that, the internal representation is retrieved and utilized as the code for a gigapixel WSI. This model's performance is then compared to state-of-the-art techniques in the pixel-to-gene expression and WSI representation learning methodologies for WSI classification and search.

Generalization is arguable the most critical attribute of many ML algorithms. An external test cohort is required to evaluate the performance of any algorithm, especially in medicine. As a result, all of the approaches proposed in this dissertation are compared to their counterparts using a fairly big external dataset that includes the same kidney cancer subtypes as TCGA subset used in this study.

1.2 Objectives

This work is part of a larger pilot project at Kimia Lab, University of Waterloo, dedicated to developing a search engine for gigapixel histopathology images. A WSI search engine can find comparable cases in the database, which can assist pathologists in reducing inter-observer variability [104]. Moreover, unlike classification, image search does not make a direct diagnostic judgment on pathologist's behalf; rather, it searches for comparable images and retrieves them, together with relevant metadata (i.e., pathology reports), and presents them to the pathologist as decision support [47]. As a result, an AI-assisted approach improves pathologists' capacity to make more confident decisions efficiently.

The objective of this thesis is to overcome the challenges of representing WSIs for computational pathology. It also intends to develop an algorithm for learning WSI representation by injecting the corresponding molecular signature, as well as linking the tissue morphology and transcriptome domains.

1.3 Contributions

Two major deep frameworks are introduced in this thesis. The primary contribution of the first model is the low-cost representation learning and embedding of WSIs. The main contributions of the second method are three-fold. For representation learning from WSIs, first, a weakly supervised method is suggested. Second, a multitask paradigm is used to inject the molecular signature into the proposed model architecture to constrain visual representation learning. Third, bulk RNA-seq can be predicted from H&E WSIs using the final trained model.

1.4 Thesis Outline

The layout of this thesis is as follows. Chapter 2 discusses the background and relevant work in the literature. It starts with some basic background knowledge on digital pathology and transcriptomics. Following that, a summary of deep learning techniques in digital pathology is presented. Finally, a brief background section discusses deep learning architectures used in this study. The third Chapter is devoted to a strategy for learning WSI representation at low power. This Chapter contains parts on technique, data descriptions, experiments, and findings. The technique for transcriptomic learning from H&E WSIs is introduced in Chapter 4. This chapter contains the approach, data description, experiments, and results. The Chapter 5 summarizes the thesis and identifies potential directions.

Chapter 2

Deep Learning in Digital Pathology

2.1 Introduction

Pathology is the study of diseases' causes and consequences via the examination of tissue, cells, and body fluid. The procedure begins with the collection of a biopsy sample from a suspicious tissue. The material is then cut into thin sections. The sample's thin slices are and stained as needed [67]. Finally, pathologists analyze stained specimens using a light microscope, which has been widely the sole instrument available. Many attempts to capture and store a digital version of glass slides have been made in recent years. These experiments resulted in the development of whole-slide imaging and the first digital scanners [74]. Researchers utilize digital image analysis algorithms to assist in activities like diagnosis, prognosis, and knowledge discovery in the digital pathology era, thanks to enormous databases of digitized slides made available by whole slide imaging technology. Although digital pathology, in conjunction with computer vision and ML, has opened up new avenues for histopathology image analysis, it also comes with its own set of problems. The principal barriers include a lack of adequate labelled data, poly- and pleomorphism, the non-boolean nature of the diagnosis, and the lack of transparency of current approaches [104, 50, 12]. This chapter provides a brief outline of opportunities and challenges for Deep Learning (DL) applications in digital pathology.

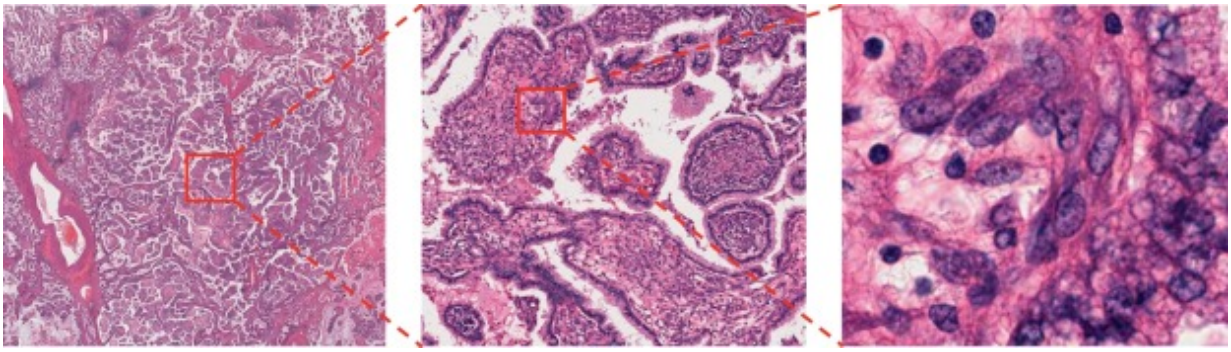


Figure 2.1: This image demonstrates a diversity of details in multiple magnifications. From left to right, the magnification increases. Image from [50].

2.2 Digital Pathology: Challenges and Opportunities

The introduction of whole-slide imaging sparked a surge in clinical, and research interest in digital pathology [50, 92]. Acquisition, management, exchange, and interpretation of pathology information such as slides, data, and a digital environment are all part of digital pathology. To comprehend this perspective, one must first grasp the value and significance of the histopathology slide. Histopathology glass slides have traditionally been utilized for diagnosis because they are quick to obtain, less expensive than alternative molecular profiling tests, and can be acquired in real-time during surgery [12].

Pathologists are trained to follow straightforward algorithmic decision criteria to stratify patients into duplicable subgroups based on tumour type and degree of malignancy in order to preserve consistency, and inter-observer agreement [26, 12]. Despite the fact that these algorithms focus on binary and adequately distinct characteristics, inter-observer variability endures even among well-practiced pathologists in an identical sub-specialty [116, 57, 109]. This ambiguity encourages the use of expensive molecular tests to assist pathologists in distinguishing between indistinct lesions [68, 12]. Despite the availability of a variety of molecular tests, pathologists continue to rely on morphological inspection of histopathology slides, and the application of conventional decision algorithms [12, 84]. As a result, Computer-Aided Diagnostic (CAD) systems provide the possibility of reducing ambiguity while increasing inter-observer concordance [99, 103]. DL-based techniques have established the gold standard among all computer vision and ML algorithms available for medical image analysis, outperforming nearly all conventional approaches in the field [12, 92, 50]. However, DL algorithms encounter a number of problems, including high-dimensional data, the multi-magnification nature of histopathology slides, and a lack of

annotated data [104, 50]. Fig. 2.1 shows a section of a H&E WSI at different magnifications. The visual information differs at each level of magnification due to the massive size of the images.

The remainder of this chapter is organized as follows; Initially, the specific procedures involved in the collection of digital pathology slides are described. Following that, the collection of RNA-seq data is discussed and the value and relevance of molecular profiling. Then, some applications of DL models for histopathology image processing that is already available are discussed. Next, a general review of current state-of-the-art deep learning approaches is presented.

2.2.1 Image Acquisition in Digital Pathology

The procedure for creating the tissue slide begins when a physician requests confirmation of a diagnosis by histology after a physical examination, radiography, and/or laboratory results [12]. There are several biopsy techniques available to collect a tissue sample, such as fine-needle aspiration, needle biopsy, excision biopsy, or extraction of a lesion in its entirety [12, 7]. The biopsy technique has a direct effect on the outcome of the diagnosis since the size of the sample has a strong correlation with a more accurate diagnosis. The larger sample size ensures that the cellular context is preserved, which is essential for a more reliable and accurate diagnosis [12]. Following tissue collection, a pathologist examines, measures, and trims the specimen, preparing it for future procedures.

Next, the soft, thick, and transparent tissue undergo several steps to prepare for visual assessment. The specimen is submerged in a fixation solution. That prevents the cells from degrading and the tissue from being changed by the microorganisms [12]. The fixed tissue is then embedded in a hardening substance, such as paraffin wax to assist subsequent sectioning. Fixation and sectioning are essential processes because they inhibit self-destruction, preserve the tissue close to its living condition, and limit shape and volume changes in the following steps [12]. Finally, a small glass sheet is placed over the tissue fixed on a glass slide. This results in a uniform thickness and prevents the microscope lens from coming into contact with the sample. Tissue preparation takes an average of 9 to 12 hours. After preparation, the prepared glass slide can be digitized [12]. A tissue preparation workflow is shown in 2.2.

Because the tissue is colourless, the tissue samples are stained to highlight cellular components and structures. H&E staining is one of the most popular, affordable, quick, and simple staining techniques [102, 117]. In H&E staining, the nuclei are stained purple with hematoxylin, while the extracellular matrix and cytoplasm are stained pink with eosin [30].

Fig. 2.1 presents a glass slide stained in H&E protocol. Alternative staining techniques exist, such as Immunohistochemistry (IHC), which uses the principle of antibodies binding particularly to antigens in biological tissues to selectively detect antigens (proteins) in cells of a tissue sample [79]. Although IHC provides detailed answers, it necessitates a more complex process and is more expensive. As a result, the H&E is regarded as a gold standard staining method for both diagnostic and research, and more H&E datasets are publicly available [92, 12].

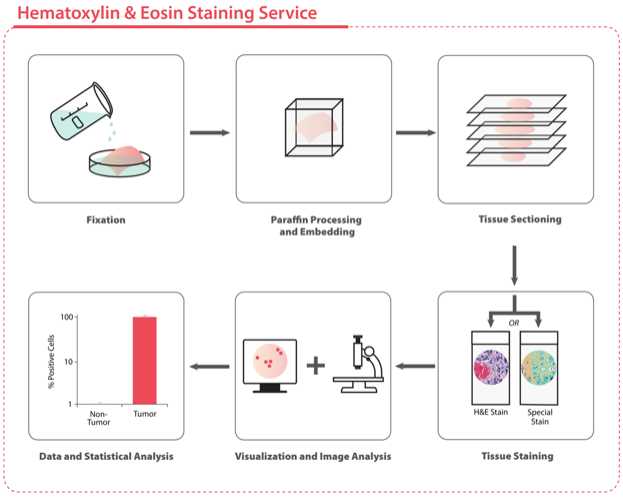


Figure 2.2: Tissue slide preparation workflow. Image from [4].

2.2.2 Tissue Slide Digitization

The whole slide scanner is a computer-controlled optical microscope. These microscopes have highly specialized cameras paired with sophisticated optical sensors capable of capturing images as small as 0.25 micron per pixel [9]. A complete slide scanner typically consists of a microscope with lens objectives, a light source, robotic parts for loading and moving slides, digital cameras, a computer, and software for manipulating, managing, and viewing pictures. An example of a digital tissue scanner is illustrated in Fig. 2.3. Most whole slide scanners use tiling or a line-scanning procedure to produce the final image. The final image is created by stitching and smoothing smaller scans using one of these approaches [12].

The magnification and resolution notion is one of the most significant distinctions between light microscopy and digital pathology. The magnification is determined by multi-

Figure 2.3: An example of a whole slide imaging scanner manufactured by Philips [5]. The whole slide scanner is on the left, the image management software shown on the monitor in the middle, and the operator holding glass slides on the right.

plying the power of the objective by the power of the eyepiece [12]. Due to the fact that the image captured at the original size may be magnified or shrunk depending on the size of the screen on which it is displayed, the notion of magnification does not apply to digital pathology [12]. As a result, the resolution is determined by the objective used to scan the slide. The resolution is expressed in micrometres per pixel. The resolution is referred to as magnification in this dissertation.

The size of the digitized slides ranges from hundreds of megabytes to tens of gigabytes, depending on the tissue size and scanning objective [115]. When compared to other medical imaging modalities, pathology WSI's are much bigger in size. WSIs is stored using both lossy (e.g., JPEG2000) and lossless (e.g., TIFF) compression techniques [12, 39]. Additionally, some manufacturers employ the SVS format, which saves a WSI as a multi-layered pyramid. Presently, digital pathology lacks a standardized and unified format comparable to DICOM for digital radiography [115].

2.2.3 Artifacts

In digital pathology, artifacts are irregularities in the generated tissue image. The artifacts are introduced in the preparation and digitization steps and can compromise the quality of the image analysis, and diagnosis [75]. Fixation, sectioning, and staining all can produce artifacts. When the process is manual, artifacts are dependent on human skills; when it is automated, they are dependent on human monitoring, machine maintenance, and solution preparation [75]. Moreover, artifacts might be introduced throughout the digitization process due to the slide scanner's overall quality and resolution. Pathologists are trained to recognize artifacts and either remove them or ignore them (sometimes they may order new samples when artifacts are preventing reliable tissue inspection). While manual techniques might be impervious to artifacts, automated approaches can be quite susceptible to them. Thus, many preprocessing approaches are proposed to discover, improve, or eliminate these components before the final algorithm is applied [82, 66, 18].

2.3 Transcriptomics

A gene is a DNA or RNA sequence of nucleotides that contain the code for the synthesis of a gene product, which can be RNA or protein [3]. A visual example is depicted in Fig. 2.4. During gene expression, DNA is translated into RNA. The RNA might be functional on its own or serve as a template for a protein that performs a specific function. Genes are the building blocks of heredity, and the transfer of genes from one organism to its offspring is the foundation of phenotypic trait inheritance. Multiple mutations can occur in the sequence of a gene. These gene variations are known as alleles. Subsequently, the alleles code for slightly different forms of the same protein, resulting in phenotypical differences.

To figure out what genetic process is at work when comparing normal and mutant cells, one must look at changes in gene expression [84]. Each cell has chromosomes, which carry genes. High-throughput sequencing reveals which genes are transcribed and in what amounts. To put it another way, RNA-seq technology can be utilized to evaluate gene expression in various cells and to figure out why the mutant cells behave the way they do. Transcriptomics investigates at RNA levels across the genome in both qualitative and quantitative perspectives [35]. Many cancers have been demonstrated to have significant changes in gene expression as a result of mutations, and characterization of gene expression aids in clarifying disease processes and prioritizing appropriate treatment options [85, 56].

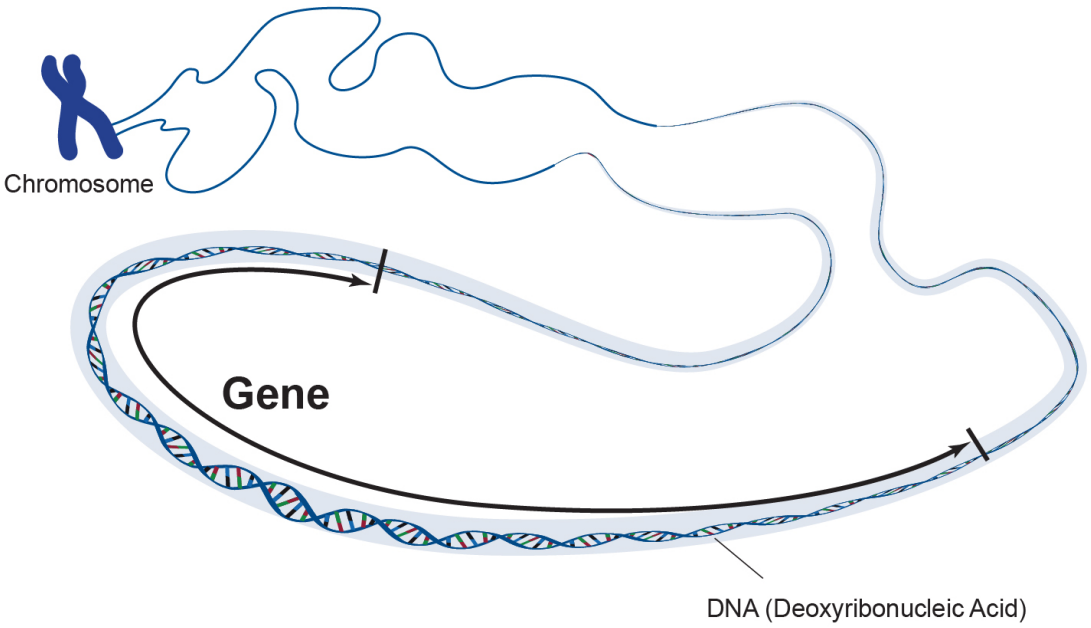


Figure 2.4: In both human, animal, and plant cells, chromosomes are thread-like structures found inside the nucleus. Proteins and a single molecule of deoxyribonucleic acid (DNA) make up each chromosome. DNA, which is passed down from one generation to the next, includes the particular instructions that give rise to the diversity of life on Earth. A gene is represented by a sequence of nucleotides in DNA that is included in the chromosome. Image from [3].

2.3.1 RNA Sequencing

The process to generate RNA sequences, short RNA-seq, has three steps, namely prepare a sequencing library, sequencing, and data analysis [6, 73, 55]. Isolating the RNA, splitting it into small fragments, turning the RNA fragments into double-stranded DNA, adding sequencing adapters, PCR amplification, and performing quality control is the first step [6]. The second step is sequencing. The DNA fragments are arranged vertically in a flow cell grid. The RNA-seq machine is equipped with fluorescent probes that are colour-coded according to the nucleotide type they can bind to [6]. These probes are then attached to the first nucleotide in each sequence [6]. The machine takes a picture of the flow cell from above after the probes are connected [6]. The first nucleotide in each sequence can

be identified using the image. The machine then removes the colour from the probes [6]. The new colour-coded probes bind to the next nucleotide in the sequence, and the process repeats until all nucleotides are recognized [6]. In the end, filtering away garbage readings[1], matching high-quality reads to a genome and calculating the number of genes per gene is the third, and the last step in the RNA-seq process [6].

Both the genome and the reads are split into small parts in order to align them to the genome [6]. By slicing the sequences into small fragments, reads that are not precise matches to the reference genome can be aligned. The read segments are then compared to genomic fragments. The chromosome and location of the read may be identified using the reference genome segment [6]. A matrix of genes versus the number of reads per sample is created following the count of reads per gene. In the "bulk" RNA-seq, the term "sample" refers to a pool of cells (usually six million cells) that can either contain normal, and disease samples [6]. "Single-cell" RNA-seq, on the other hand, treats each cell as a unique sample.

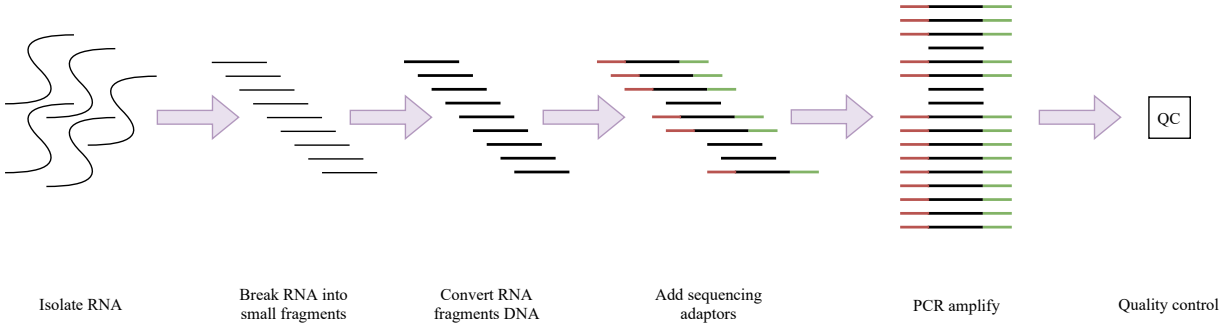
RNA-seq or gene expression data is commonly provided in a normalized format, such as FPKM. The normalizing with regard to sequencing depth and gene lengths are represented by the "Million" and "Kilobase" portions, respectively. A significant number of low-quality reads or varying levels of concentration on the flow cell may be the cause for normalizing the different amounts of reads in each sample. A schematic of RNA sequencing steps is shown in Fig. 2.5.

2.4 Deep Learning Applications in Digital Pathology

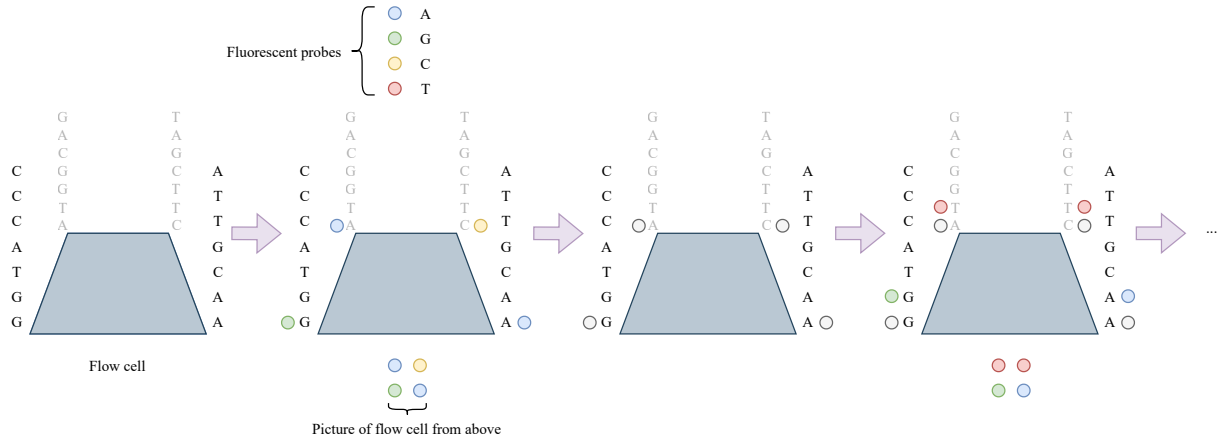
Advances in DL have re-defined benchmarks in several fields by surpassing other ML predecessors. There is no exception to this rule in the field of digital pathology; DL methods can attain clinical expert accuracy [10] in digital pathology. Historically, hand-crafted methods needed subject-matter expertise and advanced engineering in order to generate a strong data representation. On the other hand, deep learning algorithms can directly analyze raw data. Due to the fact that hand-crafted techniques were often designed specifically for a task, they did not generalize well to other tasks or datasets. Deep features have demonstrated superior performance compared to handcrafted features in histopathological image analysis [44, 90].

DL applications are classified into two broad categories, clinical and non-clinical problems. Clinical tasks include the identification of histologic primitives such as cells and nuclei and predictive modelling of indicators such as cancer diagnosis, categorization, and

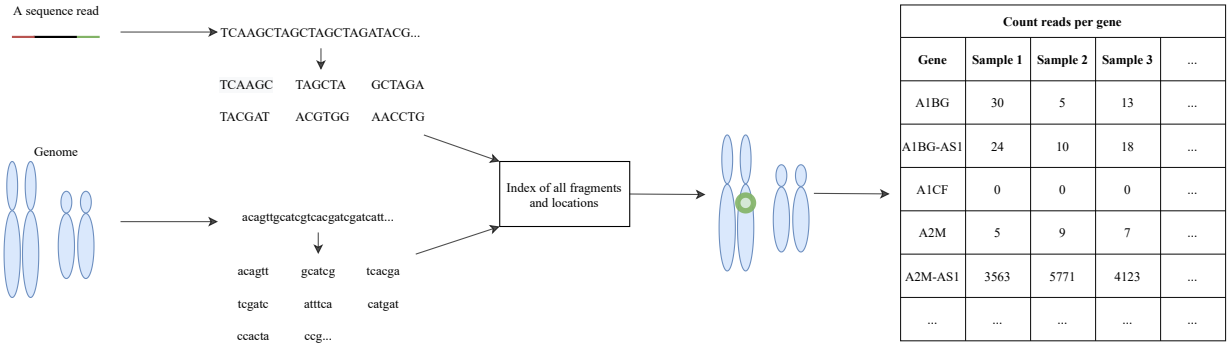
[1]Garbage reads includes (1) reads with poor nucleotide calls and (2) reads that are chemical artifacts.



(a) preparing an RNA-seq library.



(b) Sequencing.



Count reads per gene				
Gene	**Sample 1**	**Sample 2**	**Sample 3**	...
A1BG	30	5	13	...
A1BG-AS1	24	10	18	...
A1CF	0	0	0	...
A2M	5	9	7	...
A2M-AS1	3563	5771	4123	...
...	...	...	...	...

(c) Aligning the reads to a genome.

Figure 2.5: A diagram of RNA sequencing steps (Illumina protocol): (a) preparing an RNA-seq library, (b) sequencing, and (c) aligning the reads to a genome [6].

survival analysis. On the other hand, there are inevitable byproducts of CAD applications that are not clinical in nature, such as stain normalization and synthetic picture creation. The current dissertation is primarily dedicated to developing deep learning algorithms for the learning representations from gigapixel histopathology images. First, an overview of studies for the WSI representation is provided. Following that, a body of work for the use of transcriptome data in histopathology image analysis is discussed.

2.4.1 Whole Slide Image Representation

Given the challenges and complexities of gigapixel image analysis, researchers typically make assumptions in order to solve simpler versions of the problem. The most common method for processing WSIs is to combine the results of smaller tiles[2] taken from the whole image. Faust *et al.* [29] fine-tuned a deep CNN to classify 13 different tissue and lesion classes common in surgical specimens of the central nervous system. Coudray *et al.* [21] trained a deep CNN to differentiate lung cancer subtypes at 5× and 20× magnification. Riasatian *et al.* [80] trained a CNN using 1000 × 1000 pixel high-cellularity tiles at 20× from 32 cancer subtypes available on TCGA to construct a visual extractor, particularly for general histopathology image representation. Hou *et al.* [41] suggested a tile selection scheme to reduce sample redundancy while working at high magnifications. Bejnordi *et al.* [11] proposed a framework for training a stacked CNNs on both high and low magnification tiles. Kalra *et al.* [46] devised an approach, called "Yottixel", for representing WSIs using a handful of high magnification tiles sampled from distinctive colour regions. Later they applied the Yottixel search engine [46] on the TCGA WSIs and presented their findings regarding WSI search in [47].

On the other hand, there are few works on WSI analysis at the gigapixel level, namely representing the WSI in its entirety. Graham *et al.* [33] classified WSIs using features that are extracted from the probability maps of a trained CNN. Kong *et al.* [51] suggested a method based on 2D LSTM networks to embed the spatial context into the representation. Lin *et al.* [61] proposed, "ScanNet", a new neural network architecture that can be trained using larger regions. Wang *et al.* [114] presented a weakly-supervised approach to WSI representation based on ScanNet [61]. Tellez *et al.* [101] embedded all tiles of a WSI and then learn a representation based on the latter tensor of the WSI tiles. Shaban *et al.* [86] proposed a framework, that applied multitask learning, namely classification, and segmentation, on a tensor of spatially ordered tile embeddings, while utilizing attention. As an alternative to these methods, Lu *et al.* [64] proposed "CLAM", a DL weakly-supervised

[2]Also known as patches.

approach that uses attention-based learning to automatically detect high diagnostic value sub-regions across the entire slide, while also employing instance-level clustering over the identified representative regions to refine the embedding space.

Overall, the vast majority of WSI representation schemes either omitted spatial information or were computationally costly. Furthermore, the majority of the offered algorithms do not give a technique for representing WSI as they are making a decision based on patch aggregation.

2.4.2 Transcriptomic Data: An Extra Source of Information

DL models are commonly applied in digital pathology to answer clinical questions such as prognosis and diagnosis. However, there have been a few attempts to connect morphological characteristics to molecular signatures [84, 60, 36, 100]. Models that can relate histological characteristics with mutations in organs such as the lung and prostate have been shown in recent studies [21, 83]. Mutations and epigenomic modifications are known to cause large variations in gene expression. Therefore, characterization of the gene expression can be vital for diagnosis and treatment [85]. Even though more affordable whole transcriptome sequencing tools for studying gene information have been established, they are still a long way from being widely used in medical centers [48]. On the other hand, the recovery of molecular features from H&E stained WSIs is one of the faster and less expensive options. The capability to predict gene expression using WSIs, either as an intermediate modality or as an outcome, has been demonstrated to aid diagnosis and prognosis [84, 100, 36, 45]. Previous studies have drawn attention to gene expression prediction using WSI; however, the nature of WSI and available labels still impose challenges. In particular, sample selection and WSI representation is an open topic that is often handled arbitrarily.

He *et al.* suggested ST-Net based on DenseNet-121 architecture for predicting 250 genes from 224×244 tiles [36]. They used a dataset including spatial transcriptomics associated with 30,612 spots on 68 tissue sections from 23 patients [36]. Tavolara *et al.* presented a method for predicting gene expression using WSIs [100]. To detect a condition in mice, they employed predicted gene expression values as an intermediary modality [100]. They trained Gradient Boosting Tree (GBT)s to identify five predictive gene transcripts. Then, they created several bags of randomly selected thousands of 32×32 tiles to train their model. To discover an embedding for a bag of features linked with a WSI, they first used a CNN to embed all tiles. The attention weights for each instance (tile embedding) were then computed, and the final embedding was generated by combining the weighted average of all embeddings of the tiles. After that, the model's last fully connected layer projected

the embedding to five gene expression values. Finally, Schmauch *et al.* [84] proposed a Multi-layer Perceptron (MLP) and a training paradigm for learning bulk RNA-seq from WSIs. First, they embedded 8,000 randomly selected tiles of each WSI using ResNet-50 [37]. Next, they clustered 8,000 tiles into 100 clusters. By taking the average of the tiles in each cluster, they created the second type of bag of instances from the WSIs. They trained their model using pairs of WSI bags and the bulk RNA-seq.

Because this is a new area of study, there has not been as much research done on it as elsewhere. Currently available solutions either call for tile-level information or involve a significant computing expense. However, no one has yet imposed molecular information while it is learning a visual representation for a WSI.

2.5 Deep Learning Architectures

This section is comprised of a brief introduction of CNN architectures and popular models that are used for image recognition tasks. Followed by CNNs, Transformers are introduced. In the end, some notable and state-of-the-art Transformer architectures for the image recognition task are presented. Overall, this section provides background information on crucial deep learning models widely utilized in the field and is discussed multiple times throughout this work.

2.5.1 Convolutional Neural Networks

CNNs, which were created by LeCun *et al.* in 1989 and are named after a linear process called convolution, is one of the most successful neural networks [58]. The adjective convolution refers to all networks with at least one convolutional layer. For a real-valued function, the convolution operation is defined as

$$s(t) = \int x(a)w(t-a)da, \tag{2.1}$$

and can also be written as

$$s(t) = (x * w)(t). \tag{2.2}$$

Here, x and w refer to the input and kernel, respectively, in both equations 2.1 and 2.2, whereas s refers to the feature map. Due to the fact that the equations 2.1 and 2.2 can only take integer values, discrete convolution may be expressed as

$$s(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a). \tag{2.3}$$

Furthermore, assuming that the input and kernel are both zero except for a finite number of points, the concept of convolution may be expanded to include multidimensional arrays, commonly known as tensors. For example, for a two-dimensional image, I and kernel K, the convolution is defined as

$$S(i,j) = (I * K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n), \tag{2.4}$$

or

$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i-m, j-n)K(m,n).. \tag{2.5}$$

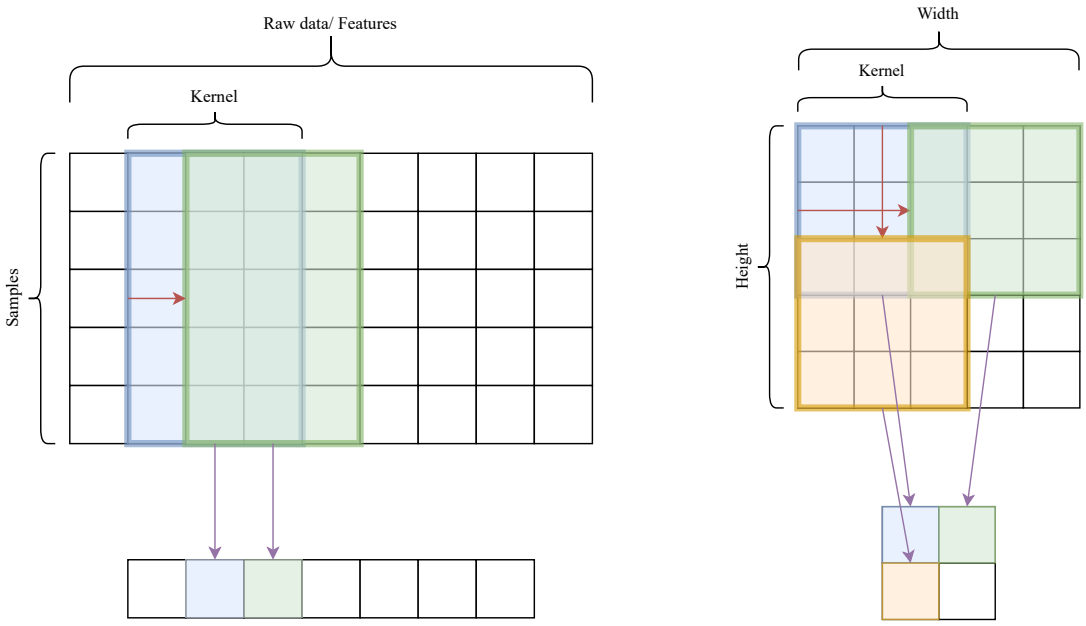
The primary purpose for flipping in the convolution formula is to keep the commutative property. Although in many ML library implementations, an alternate "cross-correlation" is employed instead:

$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i+m, j+n)K(m,n). \tag{2.6}$$

An example of 1D and 2D convolution is depicted in Fig. 2.6.

A large number of weights connect neurons in traditional fully-connected neural networks, resulting in hyper-parameters for learning. The same parameters are shared by more than one function in the CNN, which means the network has tied weights together, resulting in the same set of parameters at all locations. The term "weight sharing" refers to this architectural attribute [32].

The latter situation, equivariant representations, results in an equivariance to translation attribute in the convolution as a result of the particular kind of parameter sharing. Equivariance is a characteristic of a function in which a change in one input causes the function's output to change in the same way. Convolution is equivariant to translation, but



(a) 1D convolution (b) 2D convolution

Figure 2.6: Diagram of 1D and 2D convolution operation. The stride is how much the filter should be moved at a time. The stride is shown with red arrows in the figures. (a) 1D convolution operation for kernel size 3, stride one and without padding. (b) 2D convolution operation for kernel size 3, stride (2,2) and without padding.

not to transformations like scaling, or rotation [32]. When the objective is to determine if a feature exists or not, rather than its precise position, translation invariance is desirable.

Each convolution block has three stages: convolution or affine transform, non-linearity function[3], and pooling [32]. The pooling stage replaces each value with summary statistics, like maximum or average, in a rectangular neighbourhood around that value. This maintains the invariance to the small translations and adds robustness to the noise. It will be statistically efficient when the assumption is correct. Also, pooling over the outcome of separately parametrized convolutions can introduce invariance to specific transformations [32]. Moreover, pooling also decreases memory requirements by reducing the size of the output of each convolution layer.

[3]e.g., the rectifier or Rectified Linear Unit (ReLU) activation function [69].

2.5.2 Notable Convolutional Neural Network Architectures

Various CNN designs have been introduced in recent years, contributing to the progress of computer vision and DL. For handwritten digit recognition, LeNet-5, commonly known as the CNN architecture, was presented by LeCunn *et al.* in 1998 [59]. The top-5 error rate of image classification in the "ImageNet" competition was one of the key benchmarks that offered a good measure during these years.

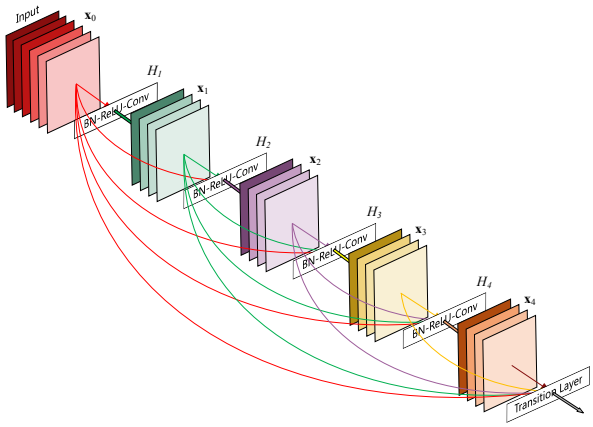
AlexNet. In 2012, Krizhevsky *et al.* presented "AlexNet", the winner of "ImageNet" that year, which outperformed the second-best accuracy by %9 [54]. They used two regularisation approaches to prevent overfitting, in addition to using %50 dropout on fully connected layers and conducting data augmentation. After the non-linearity stage of the first two convolution layers, they included "local response normalization" [54].

GoogLeNet. Szegedy *et al.* presented "GoogLeNet", a novel design that uses subnetworks and inception modules to reduce the top-5 error rate in ImageNet to under %7, resulting in 10 times fewer parameters than its predecessors [97]. "GoogLeNet" can capture patterns at different scales because of the inception modules, which contain many convolutional layers with varying kernel sizes [97].

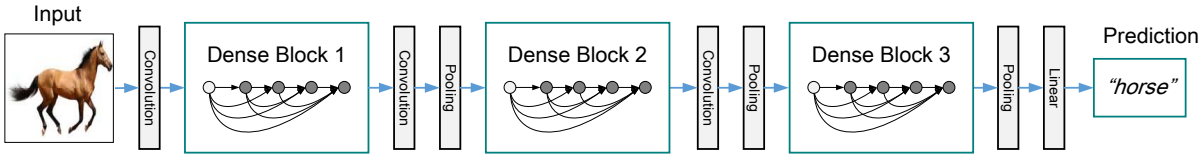
VGG. The VGG design, presented by Simonyan *et al.*, has a number of layers ranging from 16 to 19, making it deeper than other known structures at the time [89]. They also used 3×3 kernels with a stride of one to reduce the number of parameters in the model as it was improved [89].

ResNet. He *et al.* devised another notable structure, known as "ResNet", which had 152 layers and was the winner of "ImageNet" in 2015 [37]. They were able to train such a deep network because of the introduction of skip connections. The skip connections allow the network to proceed even when several layers have not yet begun to learn while also speeding up the learning process [37].

DenseNet. Huang *et al.* introduced the "DenseNet", which demonstrated that deep networks might achieve greater accuracy while being efficient when the connections between layers are shorter [42]. Unlike conventional neural network setups, in which an L-layer network has L connections, DenseNet feeds all previous layers' feature maps into each layer



(a) A dense block schematic.



(b) A deep DenseNet with three dense blocks.

Figure 2.7: DenseNet architecture. (a) a dense block diagram, in which each convolution layer has a skip connection to all successor convolution layers. (b) a DenseNet model that consist of several dense blocks. (a) and (b) are from [42].

as an input. This approach overcomes the vanishing gradient problem while simultaneously improving feature propagation and lowering the number of parameters [42]. Figure 2.7 depicts a dense block and a three-layer DenseNet model.

2.5.3 Transformers

A *Transformer* is an attention-based architecture that was originally designed for natural language processing. It was initially introduced in "Attention is all you need" [112]. Since the release of the Transformers, several projects, such as Google BERT and OpenAI's GPT, have been built on its basis and have single-handedly surpassed prior benchmarks [25, 77]. Due to its architecture, A Transformer excels in processing sequential data, such as text. Transformers are built on "Encoder" and "Decoder" layers at their heart. Certain Transformer variants do not have a Decoder at all. An Encoder is made of a "Self-

Attention" and a "Feed-forward" layers. In addition to Self-Attention and Feed-forward layers, a Decoder has an additional "Encoder-Decoder Attention" layer. There are also residual skip connections and normlayers in the "Encoder". The "Attention mechanism"[4] is responsible for the Transformers' ground-breaking performance. The original Encoder-Decoder design for Transformer, presented in [112], is depicted in Fig. 2.8.

RNNs, as well as their close relatives, LSTMs and Gated Recurrent Unit (GRU)s, were once the usual choice for processing sequential data (see Fig. 2.9). However, they have two main flaws: (1) they are limited in their ability to handle lengthy dependencies, and (2) they are slower since they process data sequentially. In the same way, CNNs have limits in terms of long-range dependencies. On the other hand, transformers overcome these restrictions by processing sequential data in parallel and capturing all dependencies, regardless of sequence length.

Attention

While processing each instance, *attention* allows the model to focus on closely connected components of the sequence. The Attention layer accepts three parameters as input: Query, Key, and Value. The Attention score is then calculated by combining these variables using the "Scaled Dot-Product Attention" formula

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{2.7}$$

where d_k is the embedding size. Fig. 2.10a shows a diagram of scaled dot-product attention. In the Encoder's Self-attention unit, the input sequence pays attention to itself. Therefore, the Encoder's input is passed to all three parameters, Query, Key, and Value. The Encoder's Self-attention generates an encoded representation for each instance in the input sequence, including the Attention scores.

Transformers leverage various "Attention scores" for each instance to manage numerous nuances and semantics in a sequence. Each Attention processor is referred to as an Attention Head, which is repeated multiple times in parallel. This is known as the "multi-head attention". It increases the discriminating strength of its Attention by merging many similar Attention calculations. Fig. 2.10b shows a multi-head attention unit. A multi-head attention unit can be summarized as

[4]The Deep Learning Attention mechanism is based on the notion of directing the model's focus, and it pays more attention to particular variables when processing data.

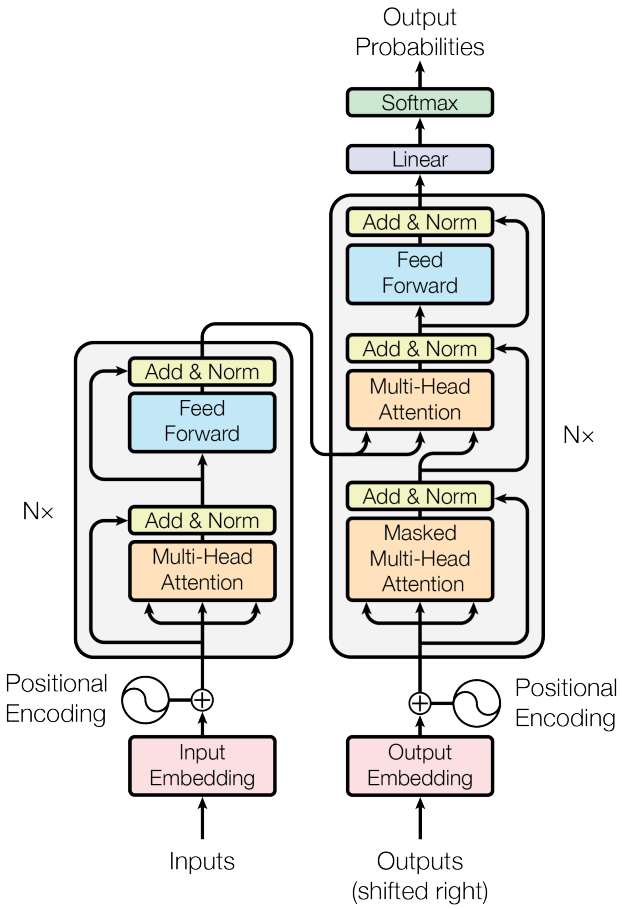
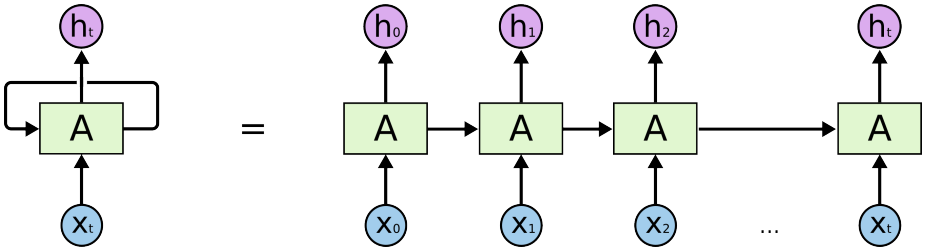
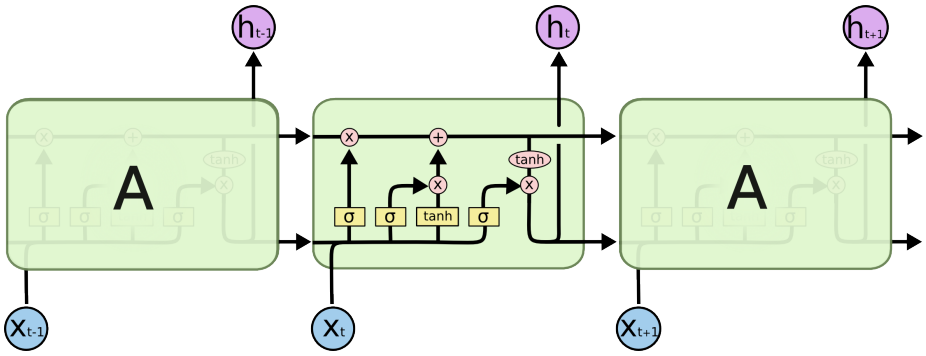


Figure 2.8: The original Transformer [112] including both Encoder and Decoder blocks for a sequence-to-sequence task (i.e., language translation). First, the input sequence (i.e., a sentence) is embedded and supplied into the Encoder. The Encoder takes the input and converts it into an encoded version of the input sequence. The start-of-sequence token is given to the Decoder. The Decoder stack processes its input along with the encoded representation from the Encoder and creates the encoded version of the target sequence. The embedding is converted to a vector of probabilities corresponding to the next instance (i.e., a word) in the target sequence in the output layer. The new target sequence is added to the start-of-sequence token and processed until it reaches the end-of-sequence token. Image from [112].

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_\text{h})W^O, \tag{2.8}$$



(a) An unrolled RNN.



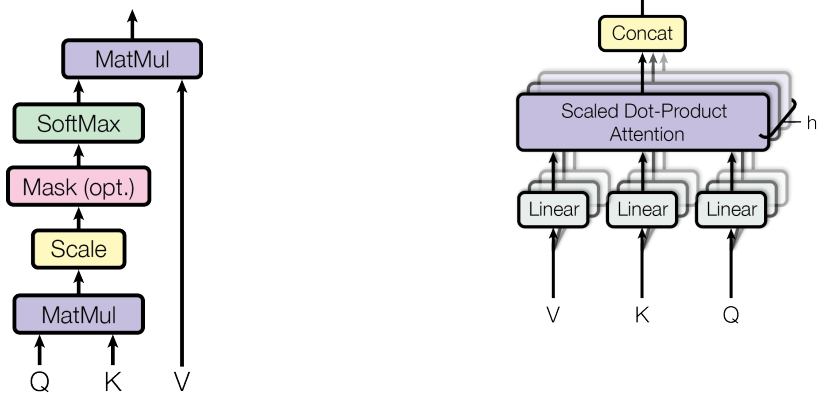
(b) The repeating module in an LSTM.

Figure 2.9: A diagram of RNNs. (a) An RNN is made up of many copies of the same network, each sending a message to the next. As a result, the output must be calculated in a sequential manner. (b) underlying LSTM operations and how output is handled at each iteration. (a) and (b) are from [72].

where

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V). \tag{2.9}$$

Consider a sequence-to-sequence language translation model. Given the input sentence "The cat drank the milk because it was hungry," self-attention should give semantic information so that the words are connected with the proper terms in the sentence. To appropriately translate the word "it" in the sentence, attention assists the model in focusing on the closely related terms in the text. For example, in the given sentence, "it" refers to the "cat," not the "milk." There might also be nuances in the sentences. When translating the word "it," for example, both "cat" and "hungry" should be considered. The model's multi-head attention allows it to collect many connections if necessary. Fig. 2.11 shows a graphic representation of how a self-attention model might assign scores to dis-



(a) Scaled Dot-Product Attention. (b) Multi-Head Attention.

Figure 2.10: (a). (b) Multi-head attention is made up of many attention layers that work in parallel. (a) and (b) from [112].

tinct instances in a sequence. The same self-attention notion is applicable to any sequential format, such as a collection of image tiles or a sequence of features.

2.5.4 Transformers in Vision

First introduced in [27], Vision Transformer (ViT)s have taken the computer vision by storm. The paper's major objective was to show that, if altered to deal with data from the visual domain, Transformer could compete with some of the most high-performing CNNs. The vanilla ViT splits the image into tiles, flattens, and projects them into an embedding space to produce tile embeddings. Then a learnable class token is concatenated with the embeddings. Next, the positional embedding[5] is added to these embeddings. Ultimately, Encoder processes the outcome tensor to deliver the desired task. ViT's performance was only acceptable when trained on smaller datasets (i.e., ImageNet with 1M images) while being state-of-the-art when trained on large datasets (i.e., JFT-330M with 300M images). Later in Data efficient image Transformers (DeiT), the Transformers were shown to be competitive with CNNs utilizing major data augmentation, stochastic depth, and hard-label distillation [107]. Multi-head self-attention in bottleneck blocks of the last stage of a ResNet [93], Conditional Positional Encodings [19], Transformer in Transformer design

[5] A collection of learnable vectors that allows the model to retain its position.

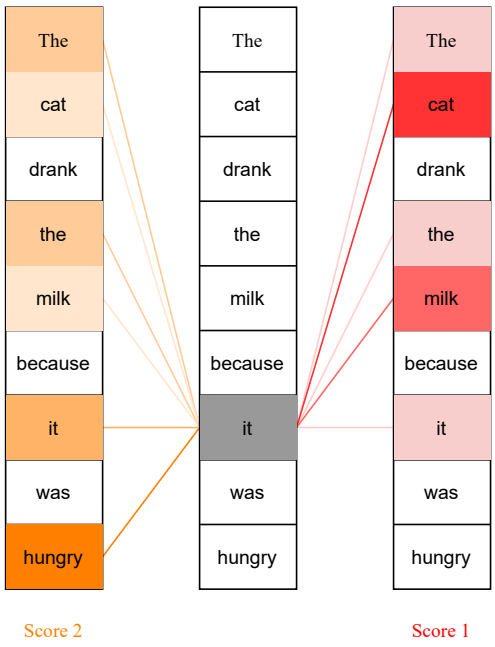


Figure 2.11: A dummy example of how multi-head attention can capture semantics in sequential data. For the given input sentence, "The cat drank the milk because it was hungry," attention helps the model focuses on related instances to seize the semantics in the sequence.

[34], and deeper ViTs utilizing class-attention layers [108] are only a few examples of the advances accomplished in recent of years.

Chapter 3

Learning Whole Slide Image Representation at Low Power

3.1 Motivation

Pathology WSIs are often comprised of a variety of distinct areas that may or may not include useful information for any given task. Due to their massive scale, these images may contain a broad range of patterns, ranging from local to global textures. State-of-the-art methods like as CNNs cannot be easily applied to WSIs because of current hardware and memory limitations. Extracting information from these images involves several steps and is, understandably, difficult [71, 104]. An overview of existing solutions is presented in Chapter 2, Section 2.4.1.

In this chapter, a novel approach is demonstrated based on DL that can be used for classification and search of RCC subtypes WSIs. The proposed framework is capable of learning WSI patterns using CNNs at a low cost at 2.5× magnification. The application of low magnification WSIs has been reported by other researchers previously. For instance, WSIs at 2.5× [76], 4× [114], and 5× [21, 46] are used by other researchers for wide range of ML and computer vision applications in pathology. The proposed model can learn task-specific features efficiently. It can also represent a large image with a fixed-length compact representation useful for a fast similarity search in image databases or other applications.

3.2 Method

This section contains all of the algorithm's stages. First, a description of preprocessing will be provided. An algorithm for localizing tissues is then discussed. Representation learning comes next. Finally, the WSI embedding is presented.

3.2.1 Stain Normalization

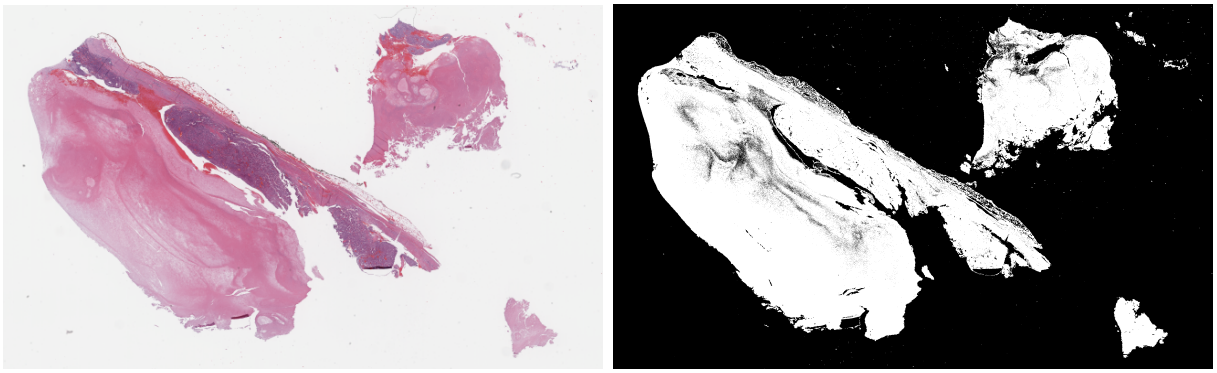
Colour differences in sample preparation, raw material use, staining methods, and slide scanners can all affect visual examination and computer-aided image analysis. As a result, several approaches for normalizing image stain have been presented in recent years [106]. In the realm of digital pathology, the influence of stain normalization on image processing methods and CNNs has been investigated [96]. In this study, a stain normalization technique proposed by Macenko *et al.* [65] is deployed. The colour of WSIs is normalized at 2.5× magnification to reduce stain variation and to enhance the colour appearance. An example of stain normalization is illustrated in Fig. 3.1.

(a) Original thumbnail (b) Color normalized thumbnail

Figure 3.1: An example of stain normalized thumbnail using [65].

3.2.2 Tissue Localization

The next step in the algorithm is to localize the tissue region. The benefits of localizing tissue are twofold. First, tiles can be extracted from the informative part of the WSI and avoid background pixels. Second, the same masks can be used to confine the deep feature maps extracted from the CNN model to important information. The tissue masks computed from WSIs by applying the following steps:



(a) Thumbnail (b) Mask

Figure 3.2: An example of the tissue localization algorithm's outcome. (a) a tissue 2.5× thumbnail and (b) the outcome binary mask delineating the tissue.

1. First, red, green, and blue pen markers[1] were filtered out using a set of fuzzy rules[1].
2. The SD was calculated within the three colour channels. All pixels with SD < 5 were considered as background regions.
3. The resulting primary mask is then closed using morphological operation with a 3×3 structuring element. All pixels associated with the background region are set to white (i.e., $[255, 255, 255]$ for a pixel in a 24-bit RGB image).
4. By using RGB to HSD (hue, saturation, and density), colour space transformation [110], the density channel of each image is evaluated. By applying a global threshold, 0.05, the rest of the background pixels are transformed to white pixels on the mean filtered density channel.
5. All white pixels are assigned to the background pixels, while the rest of the pixels are considered tissue pixels.

An example of the tissue localization algorithm is included in Fig. 3.2.

When all WSIs and their tissue masks had been preprocessed automatically, they are then visually examined for quality control. Visual inspection was put in place in order to ensure that the automatic preprocessing stage did not include any low-quality occurrences (i.e., WSIs extensively contaminated with artifacts). Automated and manual processes

[1] manual markings on glass slides

were combined to guarantee reliable and high-quality data for the next stage of the algorithm.

3.2.3 Representation Learning

A visual feature extractor is a cornerstone of any image recognition task. It extracts key patterns from an image and quantifies them as a feature vector. With the introduction of DL, CNNs outperformed handcrafted features in image recognition tasks by a considerable margin [54]. Therefore, the DenseNet-121 model [42] is used as the visual feature extractor in this work. The DenseNet-121 is frequently utilized in image recognition applications [78, 46, 36]. DenseNet-121 is made up of many convolutional, pooling, and fully connected layers, similar to most CNNs.

The following steps are taken to train a DenseNet-121 [42] on the histopathology data. First, the tissue localization algorithm takes preprocessed 2.5× thumbnails to compute the tissue mask. The tissue tiles are then extracted from the thumbnail on a grid using the tissue mask from the previous step. The deep model is initialized with ImageNet weights. Finally, the model is trained on a dataset of small tiles to learn relevant patterns associated with different cancer subtypes.

3.2.4 Representing An Entire Whole Slide Image

After training a DenseNet-121 [42] model using tiles associated with the cancer subtypes, the model has learned how to represent salient morphology patterns at 2.5× magnification. Consequently, the model can be applied to extract feature maps from an image of any size. Although the proposed method can facilitate feature extraction from the entire WSI, the feature maps may still contain redundant information (i.e., background pixels in filter responses). As a result, the feature maps are masked using tissue masks that were extracted earlier from each WSI.

First, each tissue mask is downsampled to match the height and width of the associated feature map. Then, positive values are maintained within the tissue region, and the remainder of the values are set to zero for each feature map. Each feature map now depicts how each area of the WSI has a particular pattern. Finally, to produce a fixed-length feature vector related to the WSI, each feature map is replaced with its average value. In this study, each WSI has a feature-length of 1024 since DenseNet-121 has a last convolutional layer with 1024 feature values. The outline of the proposed algorithm is shown in Fig. 3.3.

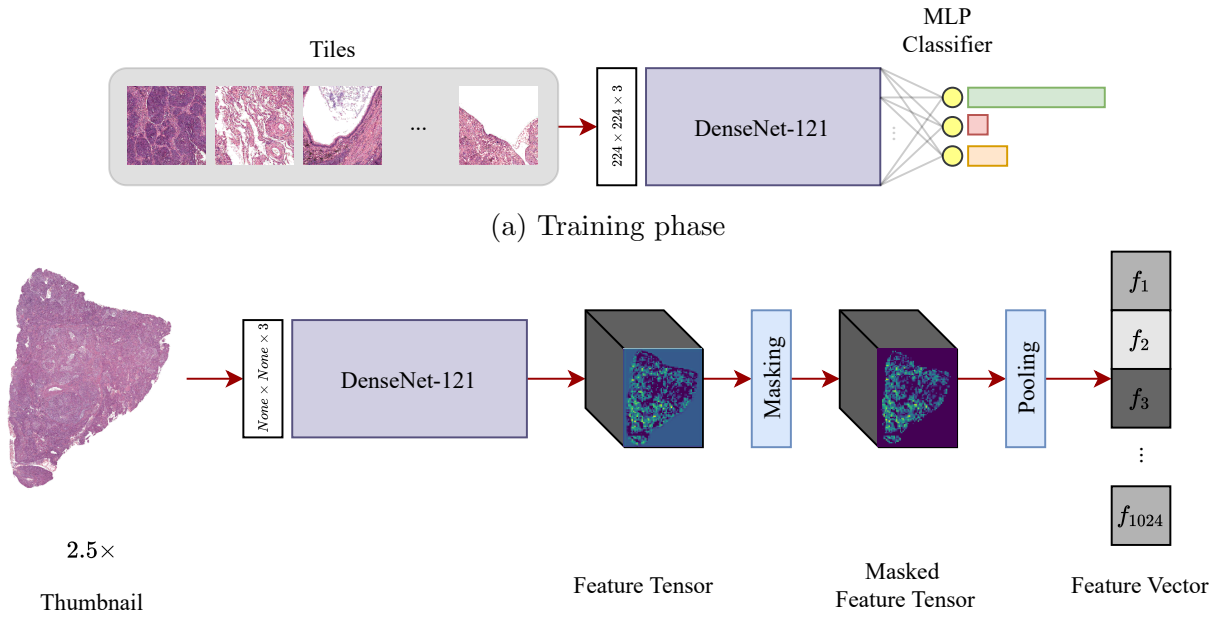


Figure 3.3: The outline of the proposed algorithm for encoding WSIs. (a) shows the training step in the algorithm. In this step, a DenseNet-121 [42] is trained using 224×224 tiles extracted from 2.5× WSIs in the training subsets to classify different RCC subtypes. The DenseNet-121 [42] is initialized with ImageNet weights. (b) depicts how the model encodes an entire WSI at 2.5× magnification. First, a 2.5× WSI is passed into the model, and the feature maps are calculated. Then, the feature maps are masked using the associated tissue mask. As the WSI goes through a series of convolution operations with a stride larger than one, the ultimate size of the feature map is smaller than the actual WSI. As a result, the tissue mask is resized first. Finally, an average of the positive values that are inside the tissue region within each feature map is taken to calculate the ultimate feature vector that represents the WSI.

3.3 Data

According to the most recent global cancer statistics report, in 2020, there were an estimated $431,288$ new cases of kidney cancer and $179,368$ deaths globally [95]. The RCC is the most common kidney cancer that is responsible for 85% malignant cases [87]. From a single malignant phenotype to a heterogeneous group of tumors, our knowledge about RCC has evolved over time [87]. Among all RCC histologic subtypes, ccRCC, Papillary Renal

Cell Carcinoma (pRCC), and Chromophobe Renal Cell Carcinoma (crRCC) make almost 75%, 16%, and 7% of the whole RCC cases, respectively [87]. RCC subtypes differ in their histology, molecular characteristics, clinical outcomes, and therapeutic responsiveness as a result of this heterogeneity. For instance, because the 5-year survival rate differs across different subtypes, proper subtype diagnosis is critical [98]. All methods in this dissertation are applied on RCC slides to identify the subtypes using search and classification.

This section includes detailed information about the public and private datasets used in this chapter. First, the TCGA dataset is introduced, and its associated statistics are provided. Next, the external test cohort from the Ohio State University is presented in detail.

3.3.1 TCGA Kidney Dataset

The data used in this study came from the TCGA [2]. The H&E-stained Formalin-Fixed, Paraffin-Embedded (FFPE) diagnostic slides were selected. The retrieved cases included three subtypes, ccRCC, ICD-O 8310/3, crRCC, ICD-O 8317/3, and pRCC, ICD-O 8260/3. The data was split case-wise into train (80%), validation (10%), test (10%) sets, respectively. In other words, each patient only belonged to one of the sets. A summary of TCGA dataset used in this chapter is presented in Table 3.1.

Table 3.1: TCGA kidney dataset splits for training, validation and testing.

	Train		Validation		Test	
Subtype	Slides	Cases	Slides	Cases	Slides	Cases
ccRCC	393	390	47	47	48	47
crRCC	95	86	10	10	13	10
pRCC	222	208	28	26	27	25

3.3.2 The Ohio State University Kidney Dataset

This is an internal (private) dataset used to evaluate the internal representation of the model. The pathology department's surgical pathology files were examined for consecutive cases of renal cell carcinoma classified as ccRCC, crRCC, or pRCC. The dataset was created at the end of the search, and it contained 141 instances of renal cell carcinoma. The WSIs from ccRCC, crRCC, or pRCC were 48, 44, and 49, respectively. Each patient

had one representative cancer slide that was examined by a board-certified pathologist[2] before being scanned at 20× utilizing an Aperio XT scanscope (Leica biosystems, CA). The board-certified pathologist reviewed the WSI images and validated the classifications a second time to guarantee the image quality and correctness of the diagnosis.

Ethics statement. This study was approved by the Ohio State University institutional research board. Informed consent was obtained from all individual patients included in the study. All the data was de-identified using an honest broker system.

3.4 Experiments and Results

A 25%, 50%, 75%, and 100% of the cases are randomly selected from the training set to create four different subsets. The hybrid approach was applied to preprocess the data. After preprocessing the dataset, no manual intervention was needed. The model is trained using tiles of size $224 \times 224 \times 3$ extracted at 2.5× magnification level for each subset. The validation set was used for error estimation throughout the training. The network parameters are set to the best parameter set attained in training on the ImageNet dataset. The kidney cancer tiles from ccRCC, crRCC, and pRCC subtypes in the training set are used to optimize all weights. The backpropagation algorithm guided by the cross-entropy loss function and the Adam optimizer [49] is used to update all parameters of the network.

The performance of the proposed technique is assessed in two scenarios, namely classification and search. In WSI classification and search, the method is compared against state-of-the-art methods [64, 46]. All methods were evaluated on the same test set, and their performance was measured using established criteria. The "weighted F1 score" is used to compare the proposed technique with [64] in the classification experiment. The F1 scores are determined for each label, and the number of true cases then weighs the average. The P@K is used as a metric for the WSI search experiment. P@K is the number of relevant results among K retrieved cases [8]. Also, Average Precision@K (AP@K) is defined as the mean of P@i for $i = 1, \ldots, K$.

3.4.1 Experiment 1: Tissue Localization and Dataset Size Effect

First, the models trained with various training sets were used to compute P@K on the TCGA test set for the WSI search. First, following Section 3.2.4, all test WSIs are em-

[2] Anil V. Parwani, MD, Ph.D., MBA

bedded using the model. Next, the distance matrix is computed for each query WSI and returned the K nearest WSIs as the search results to identify the top-K instances for search. The distances between two feature vectors were calculated using the Pearson correlation. To avoid obtaining samples from the same patient in the results, a "leave-one-patient-out" strategy is used for the search. The outcomes for masking feature maps are also included. Table 3.2 summarizes these findings. As it is depicted in Table 3.2 overall P@K value increased proportionally to the size of the training set. Also, masking feature maps enhanced and stabilized the P@K outcomes when more cases were retrieved (from P@3 to P@10). According to Table 3.2, masking increases AP@10 from 0.873 to 0.906.

Table 3.2: The impact of different training dataset sizes and masking feature maps on P@K using TCGA test set.

		Precision									
Training set	Methods	@1	@2	@3	@4	@5	@6	@7	@8	@9	@10
25%	NM	0.89	0.84	0.80	0.78	0.77	0.75	0.73	0.73	0.72	0.70
	M	0.86	0.85	0.84	0.83	0.83	0.82	0.81	0.80	0.80	0.80
50%	NM	0.85	0.85	0.83	0.82	0.80	0.78	0.77	0.76	0.76	0.74
	M	0.88	0.88	0.88	0.87	0.87	0.85	0.83	0.82	0.81	0.80
75%	NM	0.86	0.89	0.87	0.85	0.83	0.83	0.81	0.80	0.79	0.79
	M	0.85	0.87	0.89	0.90	0.89	0.89	0.88	0.88	0.88	0.87
100%	NM	0.94	0.94	0.90	0.89	0.87	0.86	0.85	0.84	0.83	0.81
	M	0.90	0.93	0.93	0.91	0.91	0.91	0.90	0.90	0.89	0.88

NM and M are corresponding to "not masked" and "masked" approaches, respectively.

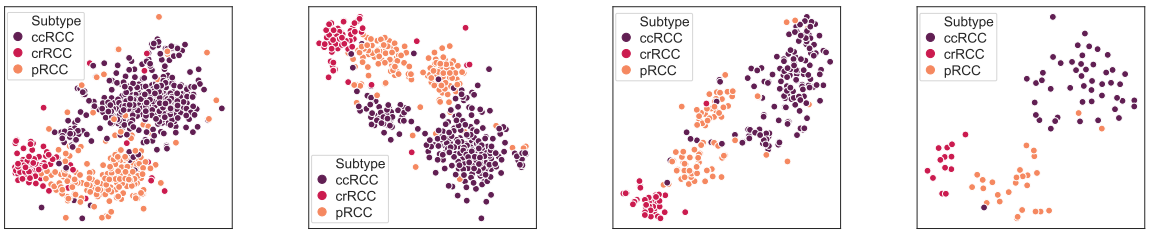
3.4.2 Experiment 2: Whole Slide Image Search - TCGA Dataset

For the second experiment, the performance of the method was tested against Yottixel described in [46]. To construct a more extensive search database for the WSI search, first, the samples that were left out of the training set were added to the test set. It was checked that the left-out set and the training set do not share patients. As a result, while training, the model has not seen any of the slides in the search database. Then, the proposed method was used to embed all samples in the search database.

For visualization, the t-SNE method is used to reduce the dimensionality of the WSI embeddings [111]. t-SNE is a dimensionality reduction approach that is especially well

suited for the display of high-dimensional datasets [111]. The t-SNE method is divided into two phases. To start, t-SNE creates a probability distribution across pairs of high-dimensional objects, assigning greater probabilities to comparable items and lower probabilities to dissimilar points. Next, t-SNE forms a similar probability distribution over the low-dimensional map's points and minimizes the Kullback–Leibler Divergence (KL divergence) between the two distributions with regard to the map's point locations. While the initial algorithm's similarity measure is based on the Euclidean distance between objects, this may be modified as needed [111]. Fig. 3.4 shows the two-dimensional t-SNE embeddings of the samples in the search databases for various models. The method could represent various RCC subtypes in distinct clusters, as illustrated in t-SNE diagrams in Fig. 3.4. The suggested method was compared with the Yottixel search engine [46] in terms of WSI retrieval over the search datasets.

The P@K diagrams for three RCC subtypes for both methods are shown in Fig. 3.5. The SDs are shown in the diagrams by horizontal lines, indicating that Yottixel [46] used a random tile selection strategy in the WSI representation process. The Yottixel results are the outcome of ten independent runs, with the mean and SDs used to create the graphs. In the RCC subtype WSI search, it is shown that proposed technique performs superior. However, it must be acknowledged that Yottixel is designed as a universal search engine and has been tested for all TCGA anatomical sites and their subtypes. Finally, top-3 searches for the TCGA test dataset was included in the qualitative evaluation of the WSI search framework in Fig. 3.6.



(a) (b) (c) (d)

Figure 3.4: The two-dimensional t-SNE embedding of WSI features. (a)-(d) are the two-dimensional t-SNE embedding for excluded samples from the 25%, 50%, 75%, and 100% subsets of the training set alongside the test set, respectively.

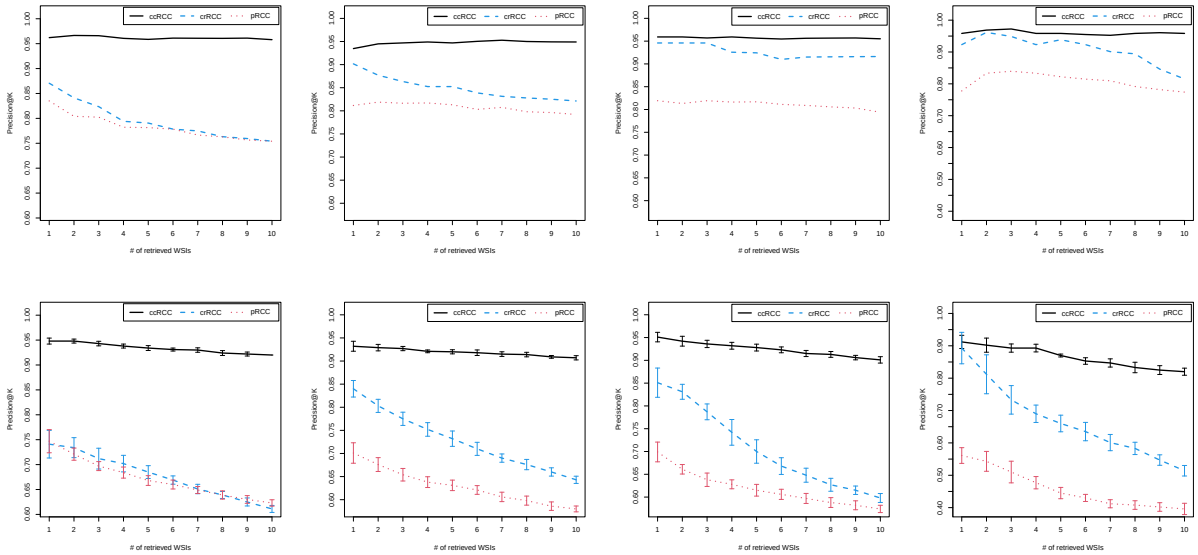


Figure 3.5: P@K for the proposed method and Yottixel [46]. The first row is P@K diagrams for the proposed method. The second row is the P@K diagram for Yottixel. The SD of ten runs of Yottixel [46] is shown by vertical lines on the diagram. From left to right, each column includes diagrams for excluded samples from the 25%, 50%, 75%, and 100% subsets of the training set alongside the test set, respectively. Each column has a fixed range for P@K to provide an easier visual comparison.

3.4.3 Experiment 3: WSI Classification - TCGA Dataset

For the third experiment, the framework's performance was investigated against CLAM [64] in a classification task. The weighted F1 scores of different approaches on the TCGA test set are reported in Table 3.3. The first two approaches, namely Patch Level Aggregation (PLA) and Patch Probabilities Aggregation (PPA), are tile-based methods for predicting WSI labels based on tile-level labels. The model was used to estimate probabilities for tiles of size $224 \times 224 \times 3$ in the test dataset for tile-based models. The bulk of the tile labels in a WSI are used to compute the WSI label in PLA. In other words, the predicted label for WSI was chosen from the class with the most tiles. However, in PPA, the average of all tile probabilities in a WSI was considered, and the WSI label based on the class with the highest average. More significantly, because the method converts a WSI into a fixed-length feature vector, various classifiers could be applied directly on the WSI features. Thus, different strategies were deployed to do this, namely Logistic Regression (LR) and

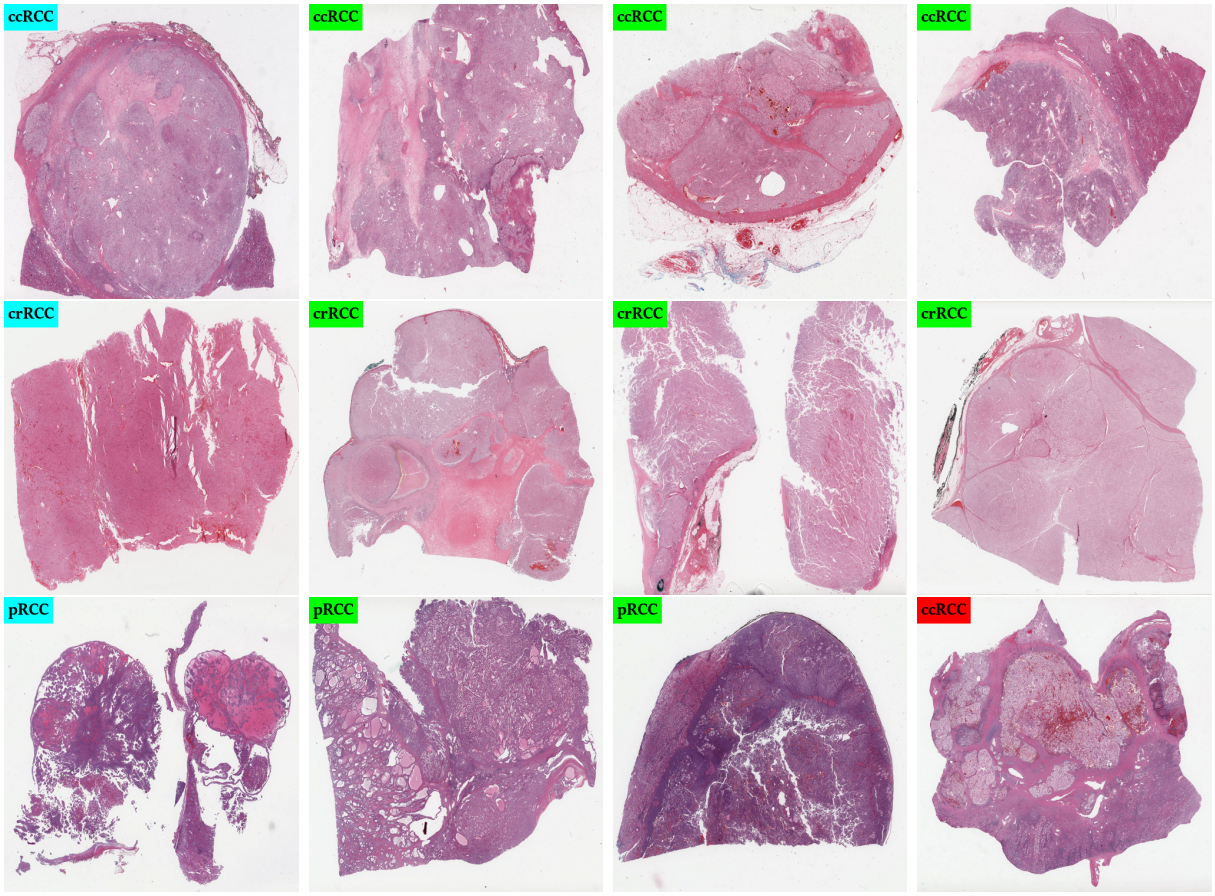


Figure 3.6: The top three search results for queries related to different RCC subtypes from the TCGA search dataset. The images with a light blue tag are the query WSIs, while the correct and the wrong retrievals are shown in green and red tags, respectively. A variety of similar cases show the method's robustness in terms of different colours, shapes, sizes, and the number of tissue segments. For visualization purposes, all images are resized to a fixed-sized square. So, the size of the WSIs may vary.

GP. The embedded WSIs in the training set are used to train all classifiers. Also, the F1 score on the test set using the approach described in [64] is reported. In terms of weighted F1 score, all approaches based on the trained model outperform CLAM [64], as shown in Table 3.3.

The micro ROC curves are drawn and AUC are provided alongside their associated CIs for different classification strategies, as shown in Fig. 3.8. It can be concluded that the GP

Table 3.3: The weighted F1 score of WSI classification using different approaches considering different training subsets.

	% of training set used (number of slides)			
Methods	25 (176)	50 (354)	75 (531)	100 (710)
PLA	0.84	0.91	0.93	0.95
PPA	0.85	0.90	0.92	0.95
LR	0.84	0.90	0.93	0.94
GP	0.86	0.90	0.94	0.94
CLAM [64]	0.63	0.50	0.69	0.66

classifier trained with WSI-level embeddings had the best overall performance, as reported in Table 3.3 and depicted in Fig. 3.8. The GP classifier's confusion matrices are shown in Fig. 3.7. It can be seen that, according to the confusion matrices, that increasing the number of training samples enhanced the performance of the algorithm.

3.4.4 Experiment 4: WSI Search - External Validation

A private dataset from the Ohio State University is used to assess better the generalizability of the method beyond the TCGA dataset. The WSI search performance is assessed on 141 WSIs from the Ohio State University dataset. In Table 3.4, the P@K for the framework is provided alongside Yottixel [46]. To demonstrate the benefits of the steps incorporated in the suggested approach, the search results for the DenseNet-121 [42] with ImageNet weights are presented as well. The proposed approach exceeds alternative methods by a substantial margin, as demonstrated in Table 3.4. The top-3 searches are included for the external dataset from the Ohio State University in the qualitative evaluation of the WSI search pipeline in Figure 3.9.

Table 3.4: The P@K for WSI search in Ohio State University dataset. Due to the selection process in Yottixel [46], the values reported here have SD under 2% for ten independent runs.

	Precision									
Methods	@1	@2	@3	@4	@5	@6	@7	@8	@9	@10
Proposed method	0.85	0.80	0.78	0.78	0.76	0.75	0.75	0.74	0.74	0.73
DenseNet-121	0.76	0.71	0.68	0.65	0.64	0.64	0.62	0.61	0.59	0.59
Yottixel [46]	0.78	0.76	0.74	0.72	0.71	0.70	0.69	0.68	0.67	0.66

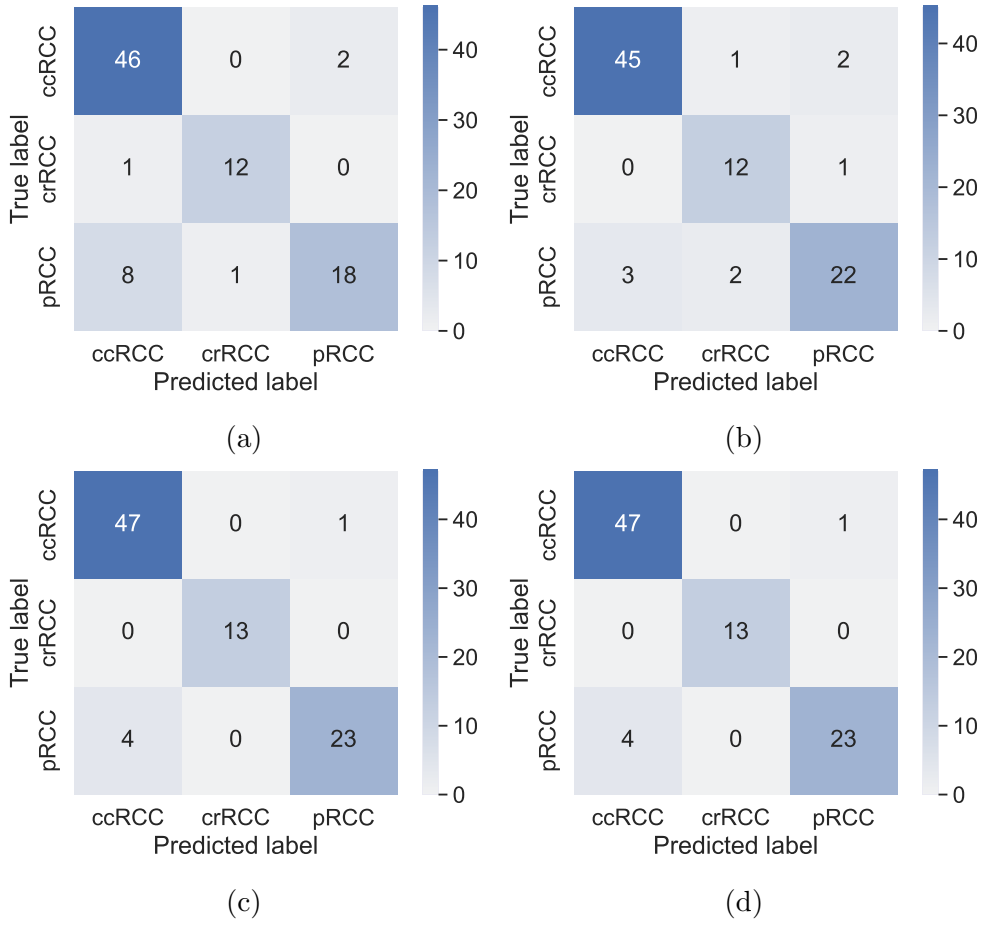


Figure 3.7: Confusion matrices for GP classifier trained on TCGA training set WSI embeddings and tested on TCGA test set WSI embeddings. (a)-(d) are the confusion matrices for 25%, 50%, 75%, and 100% subsets of the training set, respectively.

3.4.5 Experiment 5: WSI Classification - External Validation

For the last experiment, the external dataset is used to test the performance of the classification algorithms mentioned in the preceding sections. The GP classifier (the suggested approach) generated the best AUC among all other strategies, according to the ROC curves in Fig. 3.11. The confusion matrix for the GP classifier, as well as the two-dimensional t-SNE representations of the WSIs, are displayed in Fig. 3.11.

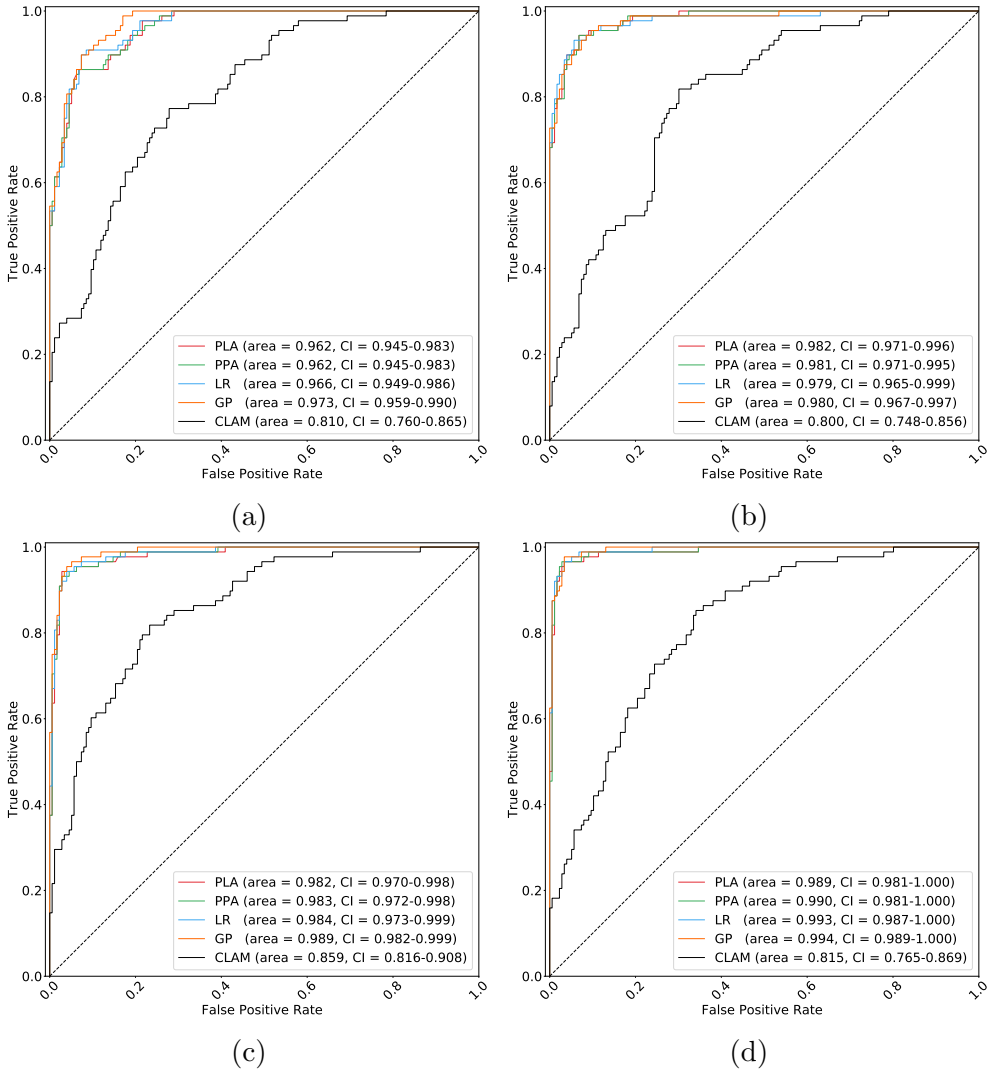


Figure 3.8: ROC curves and AUC values for different classification approaches. (a)-(d) are the confusion matrices for 25%, 50%, 75%, and 100% subsets of the training set, respectively. The AUC's CI are reported for all classifiers.

3.4.6 Activation Map Visualization and Interpretability

Along with the phenomenal success of deep neural networks, there is rising skepticism over their black-box nature. Interpretability is a desirable quality for deep networks to become strong tools in many sensitive disciplines such as medical image analysis [120]. Visualization

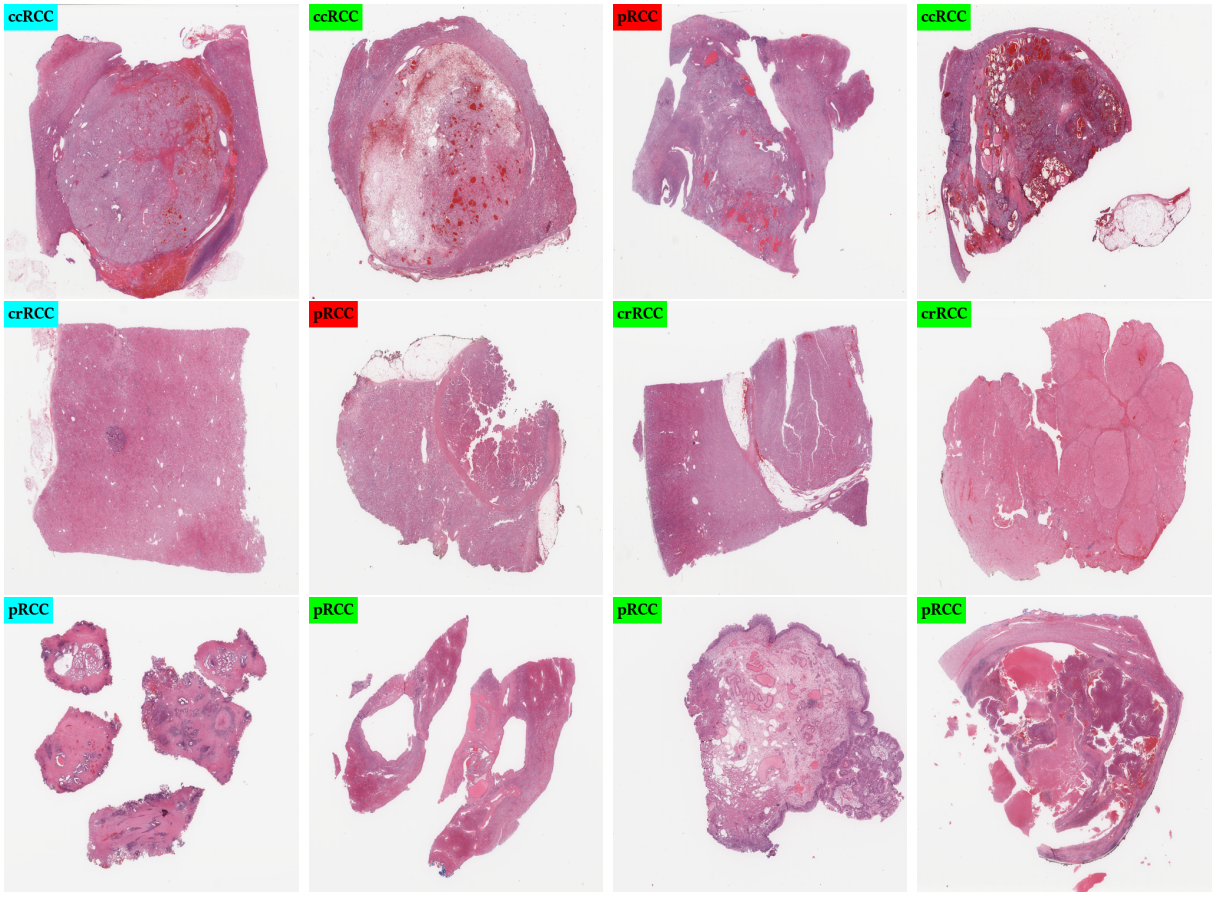


Figure 3.9: The image depicts the top three search results for queries related to different RCC subtypes from the Ohio State University dataset. The images with a light blue tag are the query WSIs, while the correct and the wrong retrievals are shown in green and red tags, respectively. A variety of similar cases show the method's robustness in terms of different colours, shapes, sizes, and the number of tissue segments. For visualization purposes, all images are resized to a fixed-sized square. So, the size of the WSIs may vary.

of convolutional activation maps is one method for compensating the lack of interpretability in deep learning. Visualization of the CAM is previously utilized for pathology images for visual validation of the features learned by a deep CNN [28]. By comparing pathologist[3] annotations with patterns that are relevant to the model, the interpretability of the model is demonstrated. To do so, first, the most important features were selected by computing the

[3]Ricardo Gonzalez, MD

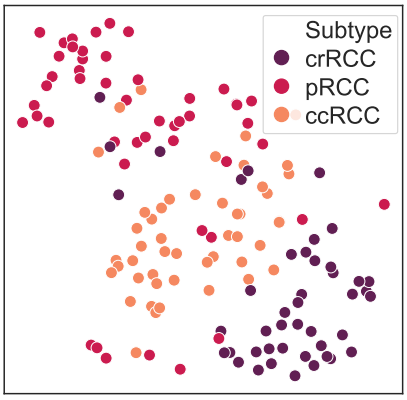
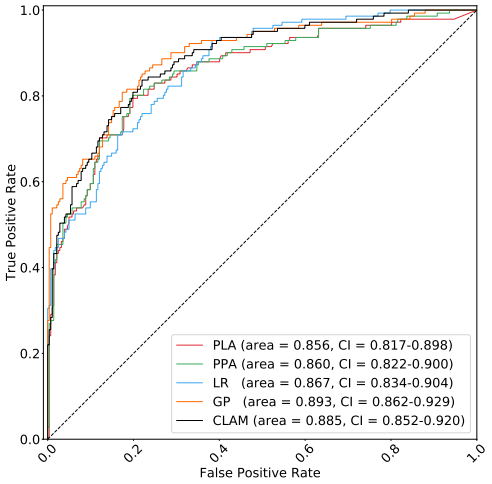
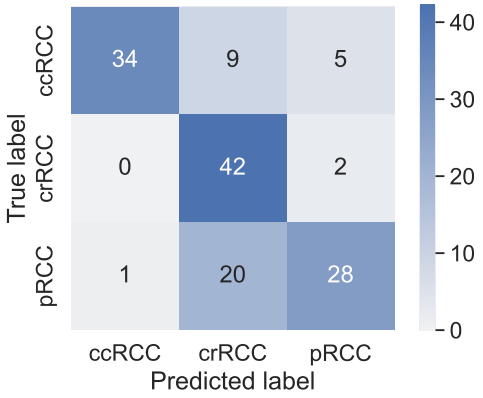


Figure 3.10: Two-dimentional t-SNE embeddings of the external dataset.



(a) ROC curves and AUC values.



(b) Confusion matrix.

Figure 3.11: Classification performance on the external dataset. (a) ROC curves and AUC values for all classifiers. The CI for all classifiers are reported in front of the AUC in the figure. (b) the confusion matrix for the GP classifier.

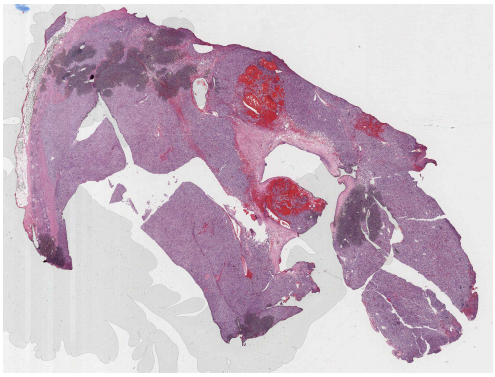
Mutual Information (MI) between the training set WSI features (continuous variables) and class labels (discrete variable). MI between two random variables is a non-negative number that indicates how dependent the variables are on each other. It is zero if and only if two

random variables are independent, while larger values indicate more dependence between them. As stated in [53] and [81], the employed function uses nonparametric approaches based on entropy estimates using k-nearest neighbors distances. Both techniques are based on a concept that was first stated in [52].

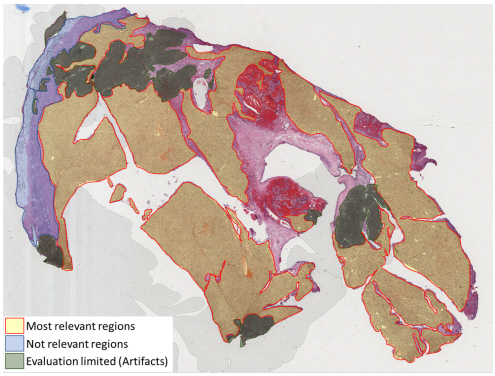
In this experiment, the MI values ranged from 0.68 to 0. Then, those deep features were selected that had MI greater than 0.5. Out of 1024 features, a subset of 82 features was selected. Because the number of features was reduced, it was easy to explore each one visually, one at a time. Fig. 3.12 depicts the CAM corresponding to a deep feature in the model that complies with the pathologist's annotation. There is a high degree of consistency between the annotated area and the heatmap, demonstrating the suggested algorithm's interpretability. According to Fig. 3.12, the model seeks a particular pattern associated with the diagnosis while excluding irrelevant data and artifacts.

3.5 Conclusions

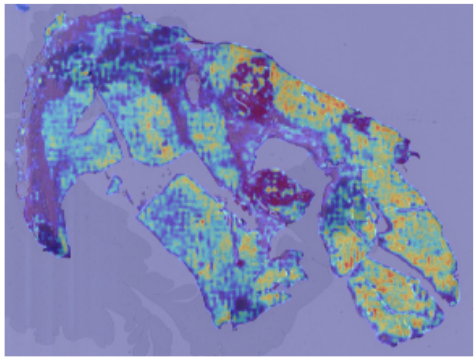
A pipeline is presented for representation learning for kidney WSIs. It was shown that by utilizing only slide-level labels and no extra annotations, the method achieves better performance in WSI multi-class classification and search in comparison with state-of-the-art algorithms. It was demonstrated that the method enhances data efficiency and generalizability over an independent test cohort. The method learns relevant textural information for RCC subtyping at 2.5× magnification. Choosing a magnification of 2.5× helps balance the computational cost while still obtaining essential information at both the cell and structural levels. At test time, the technique processes the WSI in a single step, whereas competing approaches require tens of thousands of tiles to cover the WSI at 20× [64]. In addition, unlike methods that rely on tile selection for WSI representation, a fixed-length vector is created to characterize each instance. The latter benefit allows us to minimize the number of comparisons at retrieval time significantly. It also assists in the removal of variance in the outcomes by dealing with the unpredictability caused by sample selection. Overall, the pipeline provides a more efficient and accurate approach for encoding gigapixel kidney WSIs, which can be utilized to design and train classifiers and search engines.



(a) A WSI thumbnail.



(b) The annotated WSI.



(c) The CAM visualization.

Figure 3.12: Interpretability and visualization of the CAM. (a) a ccRCC WSI from the TCGA test set, (b) annotated WSI by the pathologist, and (c) the CAM for a deep feature. The activated regions in (c) are in good agreement with the most relevant regions marked by the pathologist in (b). The model concentrated on the most important portions of the image and disregarded irrelevant regions and artifacts.

Chapter 4

Transcriptomic Learning for Representing Whole Slide Images

4.1 Motivation

In this chapter, **tRNAsformer**[1], a DL model for end-to-end gene prediction and learning WSI representation at the same time, is introduced. **tRNAsformer** employs transformer modules built on the attention mechanism to gather information required for learning to represent WSIs. To train the model, kidney WSIs, as the primary site, and their related RNA-seq data have been gathered from The Cancer Genome Atlas (TCGA) public dataset. For WSIs, the findings related to gene prediction and internal representation are presented in this chapter. Finally, the proposed model has been tested using an external kidney cancer dataset from the Ohio State University to verify its generalization.

4.2 Methods

In this section, first, gene expression and WSI preprocessing steps are discussed in detail. Next, the suggested architecture for **tRNAsformer** is explained in Section 4.2.3.

[1] **tRNAsformer** sounds like t·r·n·a·sfor·mr.

4.2.1 Gene Expression Preprocessing

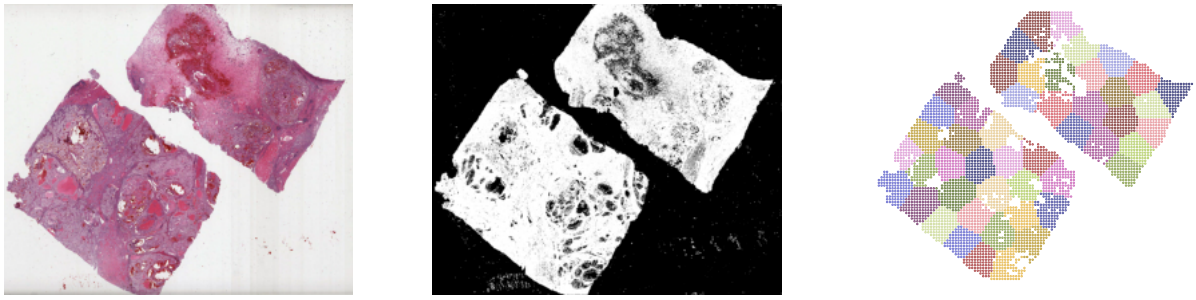
The Fragments Per Kilobase of transcript per Million mapped reads Upper Quartile (FPKM-UQ) files contained 60,483 Ensembl gene IDs [43]. The genes with a median of zero across all kidney cases were excluded. As a result, the final gene expression vector was of size 31,793. Other studies have adopted the same strategy to improve the interpretability of the results [84]. The $a \rightarrow \log_{10}(1+a)$ transform was used to convert the gene expressions since the order of gene expression values changes considerably and can impact mean squared error only in the case of highly expressed genes [84, 100].

4.2.2 WSI Preprocessing

The size of the digitized glass slides may be $100,000 \times 100,000$ in pixels or even larger. As a result, processing an entire slide at once is not possible with present technology. These images are commonly divided into smaller, more manageable pieces known as *tiles*. Furthermore, large WSI datasets are generally weakly labelled since pixel-level expert annotation is costly and labour-intensive. As a result, some of the tiles may not carry information that is relevant to the diagnostic label associated with the WSI. Consequently, MIL may be suitable for this scenario. Instead of receiving a collection of individually labelled examples, the learner receives a set of labelled bags, each comprising several instances in MIL. For making *bags of instances*, the first step is to figure out where the tissue boundaries are. Using the algorithm described in Chapter 3, Section 3.2.2, the tissue region was located at the thumbnail ($1.25\times$ magnification) while the background and the marker pixels were removed. Tiles of size 14 by 14 pixels were processed using the $1.25\times$ tissue mask to discard those with less than 50% tissue. Note that 14 by 14 pixel tiles at $1.25\times$ is equivalent to area of 224×224 pixels at $20\times$ magnification.

The k-means algorithm is deployed on the location of the tiles selected previously to sample a fixed number of tiles from each WSI. The value of k was set to 49 for all experiments in this chapter. After that, the clusters are spatially sorted based on the magnitude of the cluster centers. The benefit of spatially clustered tiles is twofold; (1) the concept of similarity is more likely to be true within a narrow radius [88, 31], and (2) clustering coordinates with two variables is computationally less expensive than high-dimensional feature vectors. The steps of the clustering algorithm are shown in Fig. 4.1.

After sampling from the clusters, a bag of samples including 49 tiles was generated. Each slide is sampled 100 times and to generate 100 bags. This is done to reduce the randomness in the bagging procedure. Random sampling works as an augmentation technique



(a) WSI thumbnail (b) tissue mask (c) k-means clusters

Figure 4.1: An example of clustering for creating a bag of tiles from a WSI.

for training. It also helps in the incorporation of uncertainty during testing. Finally, every tile was embedded using a *DenseNet-121* model with ImageNet weights [42]. Each bag reduces the dimension of each WSI to $\mathbb{R}^{k \times d}$, where k and d are the numbers of clusters and number of the deep features[2], respectively. Given the number of slides in the TCGA dataset, about four million $224 \times 224 \times 3$ tiles were selected and embedded to form bags of instances needed to train and evaluate the **tRNAsformer** model.

4.2.3 The tRNAsformer Architecture

The **tRNAsformer** is made of L standard transformer encoder layers [27] followed by two heads, namely the classification and the gene prediction head. Fig. 4.2 depicts the architecture of the proposed method. The Transformer Encoder learns an embedding (also known as the *class token*) for the input by treating it as a sequence of feature instances associated with each WSI. It learns internal embeddings for each instance while learning the class token that represents the bag or WSI.

The classification head, which is a linear layer, receives the WSI representation $\mathbf{c}$. Next, the WSI representation is projected using a linear layer to the WSI's score $\hat{y}$. **tRNAsformer** then uses cross-entropy loss between the predicted score $\hat{y}$ and the WSI's true label $\mathbf{y}$ to learn the primary diagnosis. The use of the Transformer Encoder and the classification head enables the learning of the WSI's representation while training the model.

Considering a bag $\mathrm{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k]$, where $\mathbf{x}_i \in \mathbb{R}^d$, $i = 1, \ldots, k$ are the embedded tiles by DenseNet-121, an L-layer standard Transformer can be defined as

[2]For instance, DenseNet-121 has 1024 deep convolutional features right after the last pooling layer.

$$\mathbf{z}_0 = [\mathbf{x}_{class};\ \mathbf{x}_1\mathbf{E};\ \mathbf{x}_2\mathbf{E};\ \ldots;\ \mathbf{x}_k\mathbf{E})] + \mathbf{E}_{pos}, \qquad \mathbf{E} \in \mathbb{R}^{d\times D},\ \mathbf{E}_{pos} \in \mathbb{R}^{(k+1)\times D} \tag{4.1}$$

$$\mathbf{z}'_\ell = \mathrm{MSA}(\mathrm{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1}, \qquad \ell = 1, \ldots, L \tag{4.2}$$

$$\mathbf{z}_\ell = \mathrm{MLP}(\mathrm{LN}(\mathbf{z}'_\ell)) + \mathbf{z}'_\ell, \qquad \ell = 1, \ldots, L \tag{4.3}$$

$$\mathbf{c} = \mathrm{LN}(\mathbf{z}^0_L), \tag{4.4}$$

$$\hat{y} = \mathrm{L}(\mathbf{c}), \tag{4.5}$$

where MSA, LN, MLP, L, $\mathbf{E}$, and $\mathbf{E}_{pos}$ are multi-head self-attention, layernorm, multi-layer perceptron block (*MLP*), linear layer, tile embedding projection, and position embedding (for more information see [27]). The variables $\mathbf{E}$ and $\mathbf{E}_{pos}$ are learnable. The layernorm applies normalization over a minibatch of inputs. In layernorm, the statistics are calculated independently across feature dimensions for each instance (i.e., tile) in a sequence (i.e., a bag of tiles). The multi-layer perceptron block is made of two linear layers followed by a dropout layer. The first linear layer has Gaussian Error Linear Units (GELU) activation function [38]. The embedding is projected to a higher dimension in the first layer and then mapped to its original size in the second layer. Fig. 4.2b shows the structure of a *MLP* block in a Transformer Encoder.

The remaining internal embeddings are passed to a dropout layer followed by a 1D convolution layer for the gene prediction head. The gene prediction head uses a dropout layer and 1D convolution layer as the output layer similar to the HE2RNA model introduced in [84]. However, the first two layers, which were two 1D convolution layers responsible for feature extraction in HE2RNA, were replaced with a Transformer Encoder to capture the relationship between all instances. As the model produces one prediction per gene per instance, the same aggregation strategy described in [84] was adapted for computing the gene prediction for each WSI. In particular, Schmauch et al. sampled a random number n at each iteration and calculated each gene's prediction by averaging the top-n predictions by tiles in a WSI (bag) [84]. They suggested this approach acts as a regularization technique and decreases the chance of overfitting [84]. As there were 49 tile embeddings in each bag, n was randomly selected from $\{1,\ 2,\ 5,\ 10,\ 20,\ 49\}$. For a randomly selected n during training, gene prediction outcome can be written as

$$\mathbf{s} = \mathrm{CONV1D}(\mathbf{z}_L^{1:\mathrm{end}}), \tag{4.6}$$

$$\mathbf{S}(\mathrm{n}) = \sum_{i=1}^{n} \frac{\mathbf{s}^i}{n}, \tag{4.7}$$

where $\mathbf{z}_L^{1:\text{end}} \in \mathbb{R}^{D\times k}$, $\mathbf{s} \in \mathbb{R}^{D\times k}$, and $\mathbf{S}(\text{n}) \in \mathbb{R}^{d_g}$ are the internal embeddings excluding the class token, the tile-wise gene prediction, and slide-level gene expression prediction, respectively. During the test the final prediction $\mathbf{S}$ is calculated as an average of all possible values for n as

$$\mathbf{S} = \sum_{i=1}^{k} \frac{\mathbf{S}(\text{i})}{i}. \tag{4.8}$$

The mean squared error loss function is employed to learn gene predictions.

Finally, the total loss for **tRNAsformer** is computed as

$$\mathcal{L}_{\text{Total}}(\theta) = \mathcal{L}_{\text{classification}}(\theta) + \gamma\mathcal{L}_{\text{prediction}}(\theta) + \lambda\mathcal{L}_{\text{regularization}}(\theta), \tag{4.9}$$

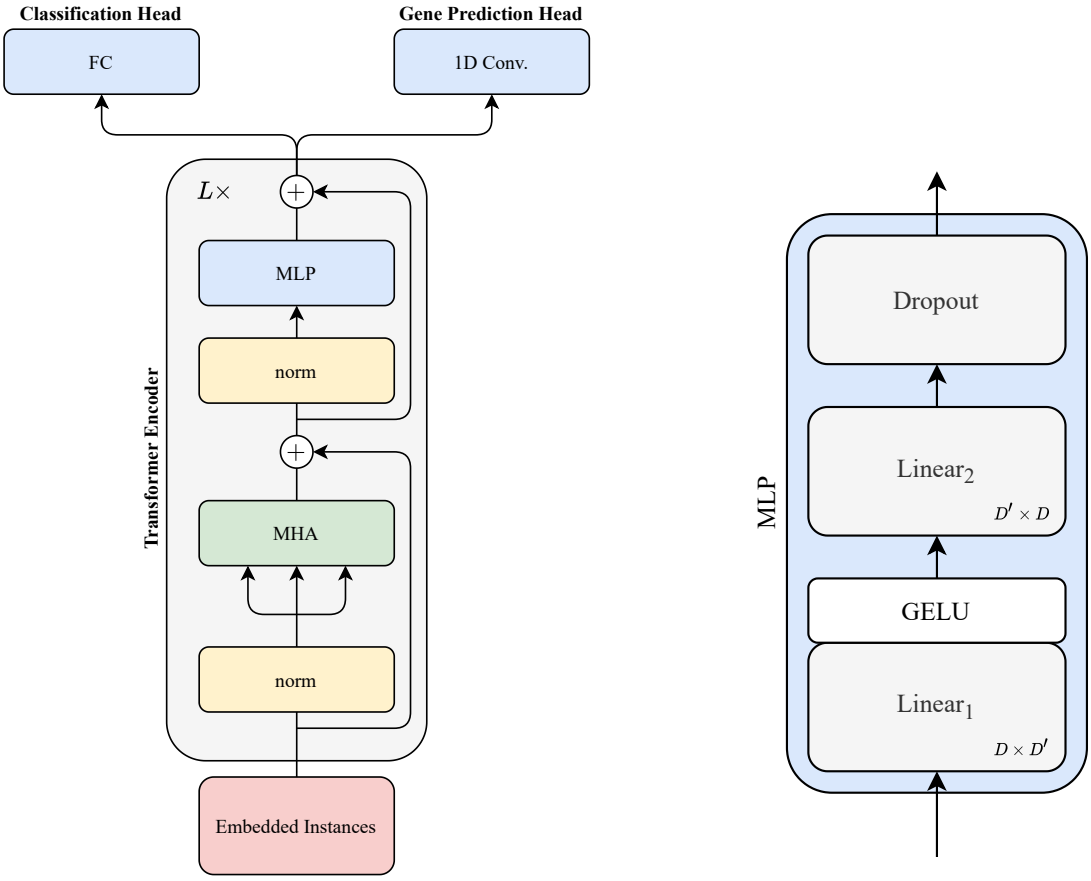
$$= \frac{1}{B}\sum_{i=1}^{B}\left(-\mathbf{y}_i\mathbf{log}(\hat{y}_i) + \gamma|\mathbf{y}_i^g - S_i|\right) + \lambda\left\|\theta\right\|_2^2, \tag{4.10}$$

where θ, λ, γ, B, $\mathbf{y}^g$ are the model parameters, weight regularization coefficient, hyper-parameter for scaling the losses, number of samples in a batch, and true bulk RNA-seq associated with the slides.

4.3 Training and Evaluation

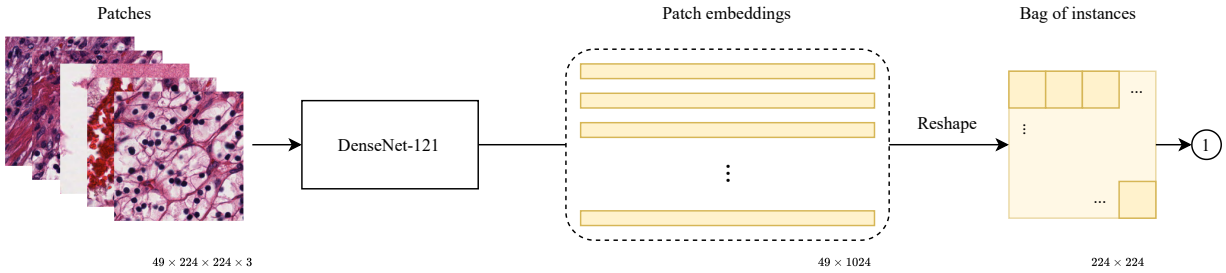
Training settings for training tRNAsformer models. To begin, TCGA cases are split into 80%, 10%, and 10% subsets for the training, validation, and test sets. Each case was associated with a patient and could have contained multiple diagnostic WSIs or RNA-seq files. The 100 bags were sampled from each WSI. As a result, the training set comprised of 63,400 bags (see Table 4.2).

The **tRNAsformer**'s internal representation size was set to 384. The *MLP* ratio and the number of self-attention heads were both four. The **tRNAsformer** was trained for 20 epochs with a minibatch of size 64. The AdamW was chosen as the optimizer with a starting learning rate of 3×10^{-4} [63]. The weight regularization coefficient was set to 0.01 to avoid overfitting. The reduce-on-plateau method was chosen for scheduling the learning rate. Therefore, the learning rate was reduced by ten every two epochs without an improvement in the validation loss. The scaling coefficient γ was set to 0.5. The last dropout layer's

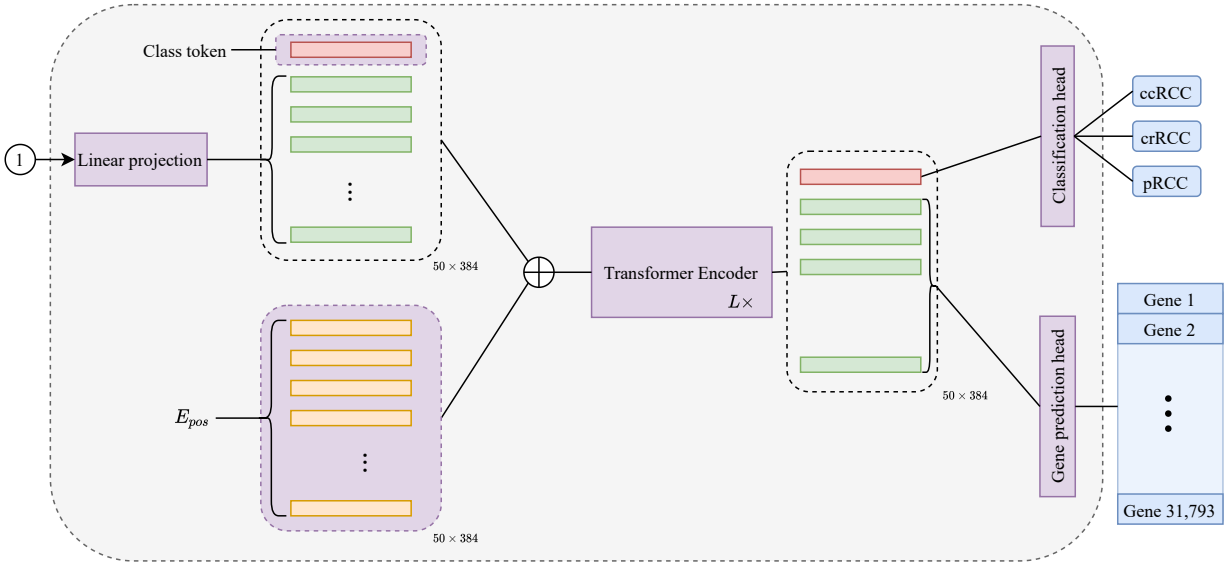


(a) **tRNAsformer** model (b) structure of the *MLP* block

Figure 4.2: The **tRNAsformer** model architecture – (a) a standard Transformer Encoder comprises layernorm, multi-head attention, multi-layer perceptron block, and residual skip connections. Because it is a multi-head self-attention module, the first layernorm's output embedding is provided to the multi-head attention as the query, key, and value. Each model can have L blocks of Transformer Encoder. The classification head transforms the internal representation to the number of classes, whereas the gene prediction head maps it to the number of genes. (b) a detailed diagram of multi-layer perceptron block (MLP). The letter D refers to the size of internal representation in the Transformer Encoder, and $\frac{D'}{D}$ is referred to as MLP ratio.



(a) Creating a bag of instances from a WSI.



(b) Internal schematic of how data flows in **tRNAsformer**.

Figure 4.3: A diagram showing how **tRNAsformer** works. (a) 49 tiles of size $224 \times 224 \times 3$ selected from 49 spatial clusters in a WSI are embedded with a DenseNet-121. The outcome is a matrix of size 49×1024 as DenseNet-121 has 1024 deep features after the last pooling. Then the matrix is reshaped and rearranged to 224×224 matrix in which each 32×32 block corresponds to a tile embedding 1×1024. (b) applying a 2D convolution with kernel 32, stride 32, and 384 kernels, each 32×32 block has linearly mapped a vector of 384 dimensional. Next, a class token is concatenated with the rest of the tile embeddings, and $\mathbf{E}_{pos}$ is added to the matrix before entering L Encoder layers. The first row of the outcome, which is associated with the class token, is fed to the classification head. The rest of the internal embeddings that are associated with all tile embeddings are passed to the gene prediction head. All parts with learnable variables are shown in purple.

Table 4.1: The number of parameters and one epoch's wall clock processing time for **tRNAsformer** and $\text{HE2RNA}_{\text{bb}}$ models. When the minibatch is set to 64, the processing time is the wall clock time for one epoch of training or validation.

		Processing time (s)	
Model	Number of parameters	Training	Validation
$\text{tRNAsformer}_{L=1}$	14,429,876	128	61
$\text{tRNAsformer}_{L=2}$	16,204,340	133	60
$\text{tRNAsformer}_{L=4}$	19,753,268	146	61
$\text{tRNAsformer}_{L=8}$	26,851,124	173	64
$\text{tRNAsformer}_{L=12}$	33,948,980	205	65
$\text{HE2RNA}_{\text{bb1024}}$	34,687,025	335	81

probability was set to 0.25. The values for the model with the lowest validation loss are reported in Section 4.5. All experiments are conducted using a single NVIDIA GeForce RTX 2080 SUPER graphic card. The desktop's CPU was Intel(R) Core(TM) i9-10900X.

Training settings for training MLP model. Another model was trained based on the MLP architecture described in [84] for fair comparison. The fully connected layers were replaced with successive 1D convolutions with kernel size one and stride one to slide data due to practicality in the MLP design [84]. A dropout layer is applied between successive layers, and the activation function was ReLU. The model based on MLP design suggested in [84] is referred to as $\text{HE2RNA}_{\text{bb}}$[3] as it was trained on TCGA training set used in this dissertation. The $\text{HE2RNA}_{\text{bb}}$ model is made of three 1D convolutional layers. The first two layers each contained h input and output channels, whereas the last layer had the same number of output channels as the number of genes. In other words, h is the size of the model's internal representation. The h was set to 1024 for $\text{HE2RNA}_{\text{bb1024}}$. The model was trained for 20 epochs using AdamW optimizer and a starting learning rate of 3×10^{-4} [63]. If no improvement is observed for the validation loss for two epochs, the learning rate was reduced by 10. The minibatch size was set to 64. The values for the model with the lowest validation loss are provided. The number of parameters of each model in Table 4.1 for comparison. The wall clock time for a single epoch for training and validation is also provided in the same table as the number of parameters.

For the evaluation of the model, 100 bags from each test WSI in TCGA and external dataset were created, similar to the training set.

[3] bb stands for **b**ack**b**one

4.4 Data

This study uses the TCGA (public) and the Ohio State kidney (private) datasets. The TCGA dataset is divided into train, validation, and test subsets, as described in Chapter 3. In addition to slides, bulk RNA-seq data from TCGA is obtained and utilized to train and assess the models. The data description is included below in section 4.4.1[4]. Finally, the Ohio State University slides are used as an external test cohort. The details for the external validation data can be found in Chapter 3, Section 3.3.2.

4.4.1 TCGA Kidney Image-Gene Dataset

First, the TCGA dataset was searched for the kidney diagnostic FFPE glass slides and kidney cases RNA-seq files, separately. Then the results of both searches were compared against each other to find instances with at least one glass slide and one bulk RNA-seq file. The detailed information regarding the cases is included in Table 4.2. The FPKM-UQ were selected as the transcriptomic data. The FPKM-UQ normalization technique is based on a modified version of the FPKM approach. The following is the formula for calculating FPKM-UQ values:

$$\text{FPKM-UQ} = \frac{\text{RM}_\text{g} \times 10^9}{\text{RM}_{75} \times L_g}, \tag{4.11}$$

where RM_g is the number of reads mapped to the gene, RM_{75} is the number of reads mapped to the 75th percentile gene in the alignment, and L_g is the length of the gene in base pairs. Each FPKM-UQ file is a text file with two columns. The first column contains the Ensembl ID, which serves as a unique identifier for each gene. The second column includes the above-mentioned normalized value for the number of reads per gene.

4.5 Experiments and Results

Several experiment series were carried out to quantify the performance of the proposed model, which can predict gene expression as well as construct WSI level embedding. In each task, **tRNAsformer** is compared to the best model for that task. The benchmark methods were chosen based on model performance, comprehensive assessment, and code and data

[4]The dataset is created under the supervision of Anil V. Parwani, MD, Ph.D., MBA.

Table 4.2: TCGA kidney dataset split for transcriptomic learning (the number of cases, slides, and FPKM files per subtype per subset).

	Train			Validation			Test		
Subtype	Cases	Slides	FPKMs	Cases	Slides	FPKMs	Cases	Slides	FPKMs
ccRCC	369	372	373	43	43	45	46	47	48
crRCC	47	47	47	8	8	8	7	7	7
pRCC	195	215	195	26	28	26	24	26	24

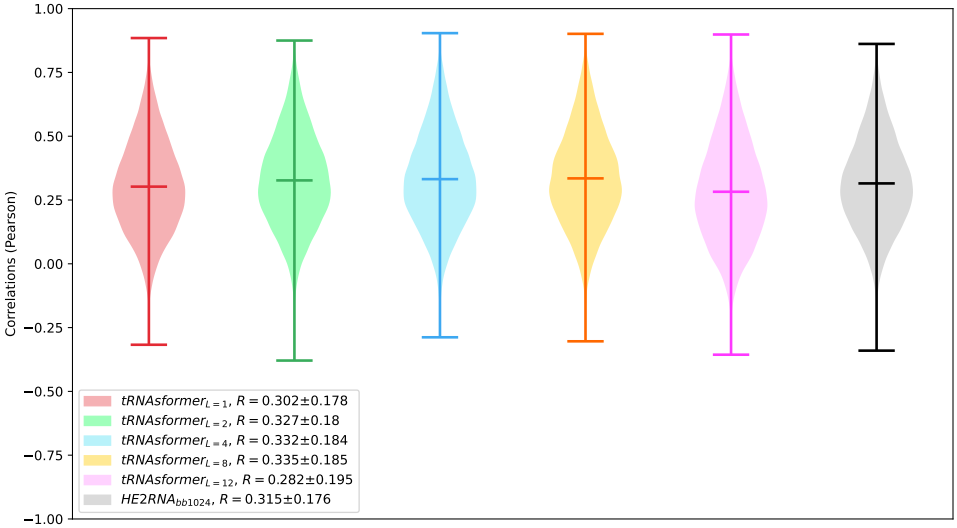
availability. The **tRNAsformer** model is compared to the HE2RNA architecture [84] in terms of gene prediction performance in Section 4.5.1. Next, **tRNAsformer** is compared to state-of-the-art WSI representation models in WSI classification and search to measure the quality of the internal representations. As a result, the CLAM [64] and Yottixel [46] are utilized for benchmarking the WSI categorization and search. A description of all three methods is presented in Chapter 2.

4.5.1 Experiment Series 1: Predicting Gene Expression

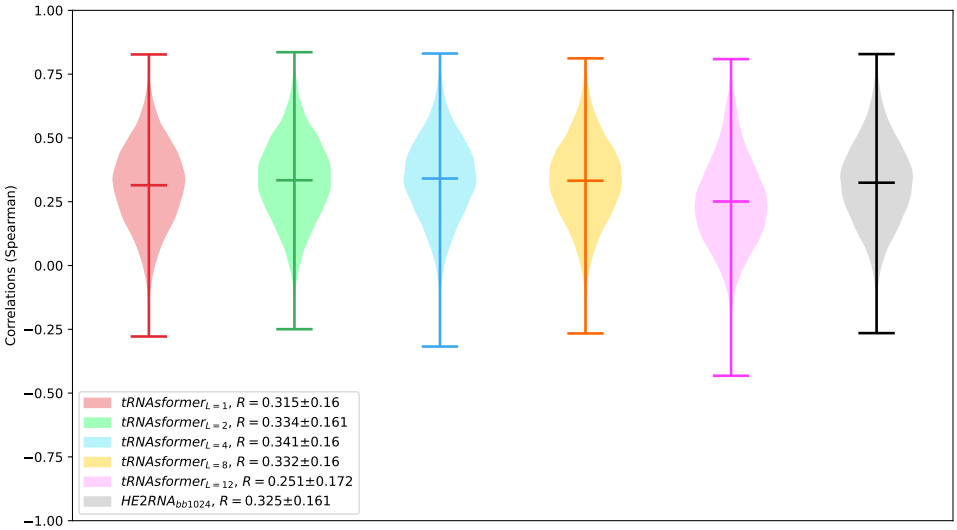
Both models, **tRNAsformer** and HE2RNA, were compared for three different subtasks, namely mean correlation coefficient of predictions, the number of genes predicted significantly better than a random baseline, and the prediction error.

In the first experiment, the correlation is assessed for each gene separately using Pearson and Spearman's correlation coefficient. If the datasets are normally distributed, the Pearson correlation coefficient measures the linear connection between them. The Pearson correlation coefficient varies between -1 and $+1$. A correlation of -1 or $+1$ denotes a perfect linear negative or positive relationship, respectively, whereas a correlation of 0 denotes no correlation. The p-value roughly represents the probability that an uncorrelated system can produce datasets with a Pearson correlation at least as high as the one calculated from these datasets. The Spearman correlation, unlike the Pearson correlation, does not require that both datasets be normally distributed. Fig. 4.4 displays the distribution of correlation coefficient for 31,793 genes predicted by different models.

The mean correlation coefficient R grew with depth from $L = 1$ to $L = 8$, as seen in Fig. 4.4. The mean R value declines after eight blocks of Transformer encoders, suggesting that increasing the number of layers does not enhance gene expression predictions. Another



(a) Pearson correlation.



(b) Spearman correlation.

Figure 4.4: The distribution of the correlation coefficients between 31,793 genes predicted and their true value for TCGA test set. The violin diagrams depict the distribution, min, max, and mean values of the correlation coefficients. (a) violin diagrams for Pearson correlation coefficients and (b) violin diagrams for Spearman's correlation coefficients. The violin diagrams are plotted for tRNAsformer$_L$ for $L = (1, 2, 4, 8, 12)$ and HE2RNA$_{\text{bb1024}}$. The mean and standard deviation of the correlation coefficients are included in the legend.

Table 4.3: The number of genes were predicted with a statistically significant correlation (p-value < 0.01) under HS and BH correction. The total number of predicted genes is 31,793. These values are computed using the TCGA test dataset.

	Pearson		Spearman	
Model	HS	BH	HS	BH
tRNAsformer$_{L=1}$	29,990	30,797	30,427	31,042
tRNAsformer$_{L=2}$	30,338	31,014	30,695	31,141
tRNAsformer$_{L=4}$	30,433	30,996	30,858	31,266
tRNAsformer$_{L=8}$	30,344	31,002	30,741	31,181
tRNAsformer$_{L=12}$	28,933	30,187	28,938	30,210
HE2RNA$_{\text{bb1024}}$	30,249	30,937	30,663	31,163

important observation is that **tRNAsformer** has higher mean correlation coefficients than its counterpart for $L = 2$ to $L = 8$.

The Pearson and Spearman's correlation coefficients and p-values were computed between the predicted and the true value of the gene expression for each gene. Two multiple-hypothesis testing methods, namely HS and BH, were utilized to adjust the p-values. If the p-value of the R coefficient was less than 0.01 after correction for multiple-hypothesis testing, the prediction was significantly different from the random baseline [40, 14]. Similar to [84], multiple-hypothesis testing was done using both HS and BH correction. The results are shown in Table 4.3 for all architectures.

As it is demonstrated in Table 4.3, increasing the depth of the **tRNAsformer** from one to eight increases the number of genes that are significantly different from a random baseline. Similar to the results in Fig. 4.4, there is a decrease in the number of genes when the depth reaches 12 blocks of Transformer Encoder. On the other hand, the model based on the design of HE2RNA scored inferior to nearly all other **tRNAsformer** models (except for $L = 1$).

MAE, RMSE, and RRMSE [91] were selected to calculate the error between the prediction and real gene expression values. MAE, RMSE, and RRMSE are defined as

Table 4.4: Prediction error for **tRNAsformer** and $\text{HE2RNA}_{\text{bb1024}}$ models quantified by MAE, RMSE, and RRMSE. All errors are calculated using TCGA test set.

Model	MAE	RMSE	RRMSE
$\text{tRNAsformer}_{L=1}$	1.31 ± 1.04	1.67 ± 1.20	1.02 ± 0.16
$\text{tRNAsformer}_{L=2}$	1.30 ± 1.03	1.65 ± 1.17	1.02 ± 0.16
$\text{tRNAsformer}_{L=4}$	1.30 ± 1.08	1.63 ± 1.19	0.98 ± 0.11
$\text{tRNAsformer}_{L=8}$	1.37 ± 1.02	1.69 ± 1.13	1.11 ± 0.27
$\text{tRNAsformer}_{L=12}$	1.50 ± 1.10	1.79 ± 1.26	1.10 ± 0.43
$\text{HE2RNA}_{\text{bb1024}}$	1.29 ± 1.08	1.63 ± 1.20	0.96 ± 0.08

$$\text{MAE} = \frac{\sum_{(x_i,y_i)\in D_{\text{test}}} |\hat{y}_i - y_i|}{|D_{\text{test}}|}, \tag{4.12}$$

$$\text{RMSE} = \sqrt{\frac{\sum_{(x_i,y_i)\in D_{\text{test}}} (\hat{y}_i - y_i)^2}{|D_{\text{test}}|}}, \tag{4.13}$$

$$\text{RRMSE} = \sqrt{\frac{\sum_{(x_i,y_i)\in D_{\text{test}}} (\hat{y}_i - y_i)^2}{\sum_{(x_i,y_i)\in D_{\text{test}}} (\bar{y} - y_i)^2}}, \tag{4.14}$$

where D_{test} denotes the test set, (x_i, y_i) is the i-th sample x_i with ground truth y_i, $\hat{y}_i$ is the predicted value of y_i, $\bar{y}$ is the mean value over the targets in the test set, and $|D_{\text{test}}|$ is the number of samples in the test set. The results are given in Table 4.4.

Similar to the results in Fig. 4.4 and Table 4.3, increasing the number of Transformer Encoder blocks from eight to 12 significantly degrades the performance of the model. In addition, the **tRNAsformer** has a comparable for $L = 1$ to $L = 8$, considering the fact that **tRNAsformer** handles multiple tasks rather than a single gene prediction task. Overall, according to the correlations coefficients, p-value tests, and prediction errors, $L = 8$ appears to be a critical threshold after which the model becomes overparametrized for the gene prediction task, according to the results.

4.5.2 Experiment Series 2: WSI Classification

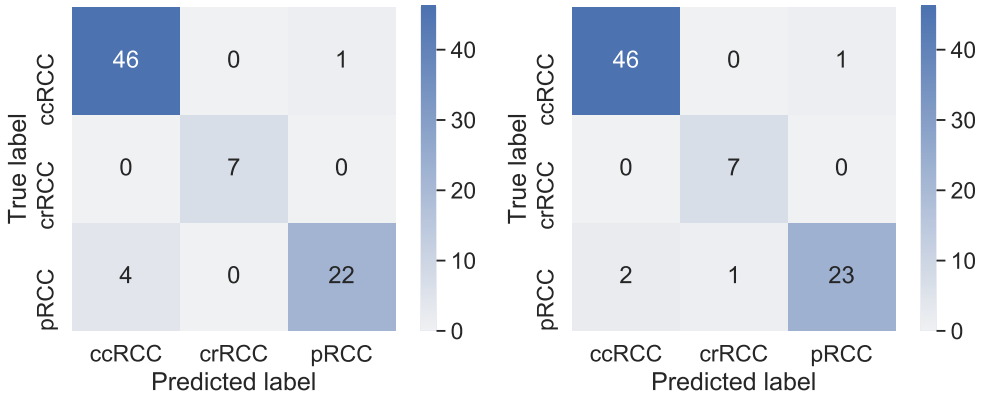
WSI classification - TCGA dataset. The classification experiments were conducted to assess the quality of internal representation learned by the proposed model. To begin,

100 bags have been created from each TCGA test WSIs. According to Table 4.2, a total of 8,000 bags were created from TCGA test set, as there were 80 WSIs. The same models that were trained in the previous section to predict RCC subtypes were assessed for the classification task as well. The accuracy, macro, and weighted F1 scores is presented for all models in Table 4.5. The confusion matrices of different models are displayed in Fig. 4.5. All values reported here are based on slide-level classification results. The prediction is made for all bags in order to calculate slide-level values. Each test slide's label predication is chosen as the most common prediction among all bags created from that slide. The WSI representations learned by the models are projected onto a plane created by the first two principal components found using PCA to depict the internal representation of our models in two-dimensional space. The two-dimensional PCA projections are shown in Fig. 4.6.

WSI classification - External dataset. Because of variations in hospital standards and methods for tissue processing, slide preparation, and digitization protocols, the appearance of WSIs might vary significantly. As a result, it is important to ensure that models built using data sources are resistant to data-source-specific biases and generalize to real-world clinical data from sources not used during training [94]. For testing the generalization of our trained models, 142 RCC WSIs are used from the Ohio State University as an independent test cohort (see Section 3.3.2 in Chapter 3 for detailed information about the external kidney dataset).

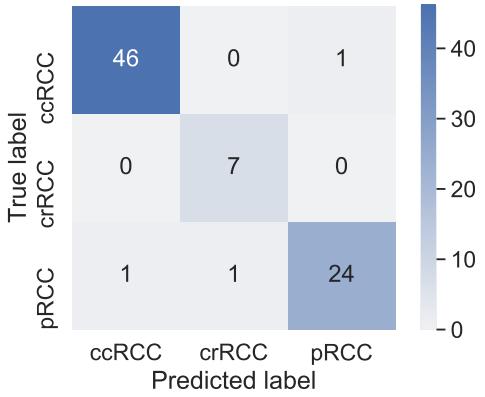
First, 100 bags were created from each external test WSIs. According to Table 4.2, a total of 14,200 bags were created from TCGA test set, as there were 142 WSIs. Same models that were trained in the previous section to predict RCC subtypes are used to report classification results for the external dataset. The accuracy, macro, and weighted F1 scores are reported for all models in Table 4.5. The confusion matrices of different models are displayed in Fig. 4.7. The WSI representations learned by the models are projected onto a plane created by the first two principal components found using PCA to depict the internal representation of the models in two-dimensional space. The two-dimensional PCA projections are shown in Fig. 4.9.

Referring to Figures 4.5, 4.7, 4.8, and Table 4.5, the accuracy and F1 scores on TCGA and the external dataset drop when the model has 12 blocks. This is consistent with the results for the gene prediction in Section 4.5.1. For tRNAsformer$_{L=4}$ seems to slightly overfit to TCGA data as the model's performance slides towards TCGA. In other words, the tRNAsformer$_{L=4}$ improves slightly for the TCGA dataset while decreasing marginally for the external dataset.

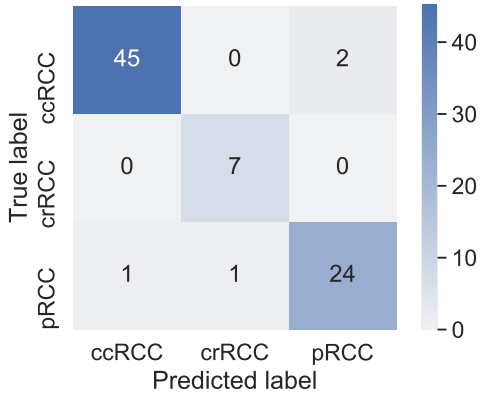


(a) tRNAsformer$_{L=1}$

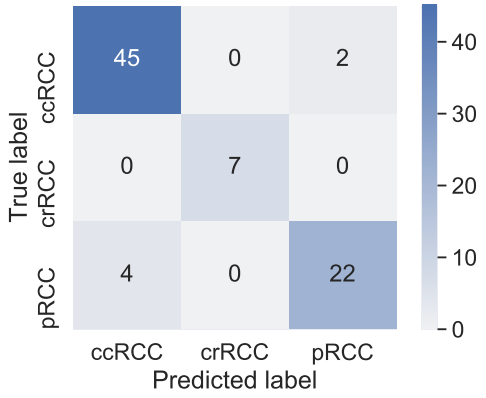
(b) tRNAsformer$_{L=2}$



(c) tRNAsformer$_{L=4}$

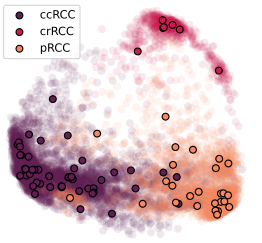


(d) tRNAsformer$_{L=8}$

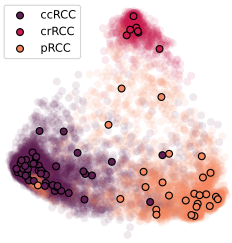


(e) tRNAsformer$_{L=12}$

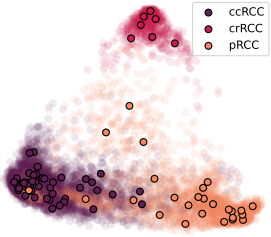
Figure 4.5: The confusion matrices for different models applied on 8,000 bags created from 80 TCGA test WSIs. (a)-(f) are for tRNAsformer$_L$, $L = (1, 2, 4, 8, 12)$, respectively.



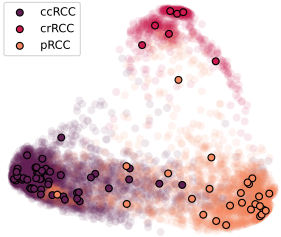
(a) tRNAsformer$_{L=1}$



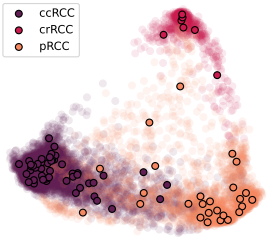
(b) tRNAsformer$_{L=2}$



(c) tRNAsformer$_{L=4}$

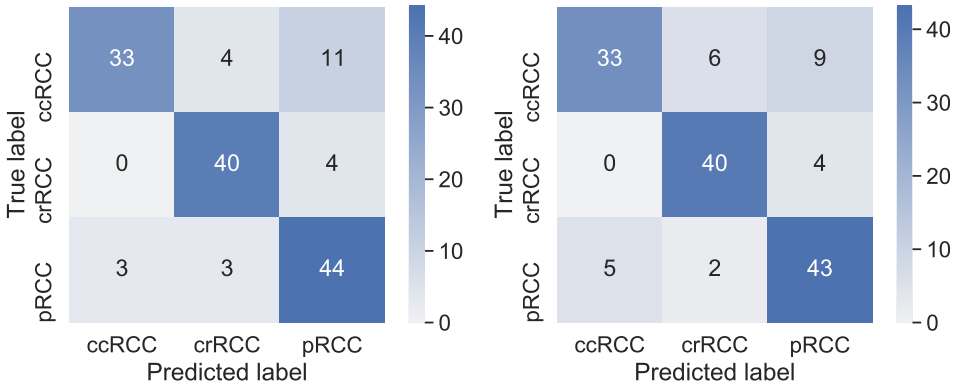


(d) tRNAsformer$_{L=8}$



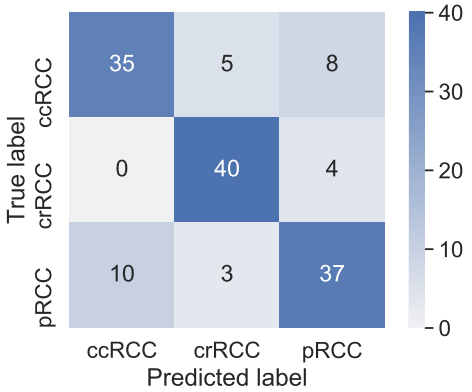
(e) tRNAsformer$_{L=12}$

Figure 4.6: The two-dimensional PCA projection of TCGA test WSI features. (a)-(f) are for tRNAsformer$_L$, $L = (1, 2, 4, 8, 12)$, respectively. Each TCGA test WSI is represented by 100 bags of features. All bags of features associated with the test set are shown with transparent circles. The average of PCA projection of each WSI (average of 100 bags associated with each WSI) is shown in bold circles with black edges.

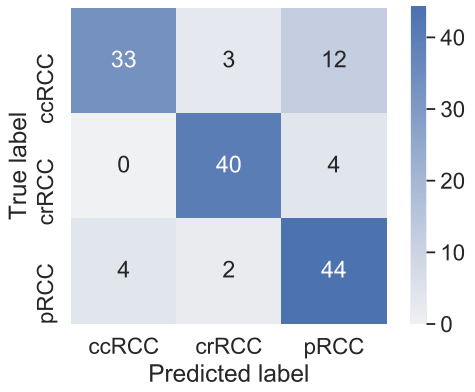


(a) tRNAsformer$_{L=1}$

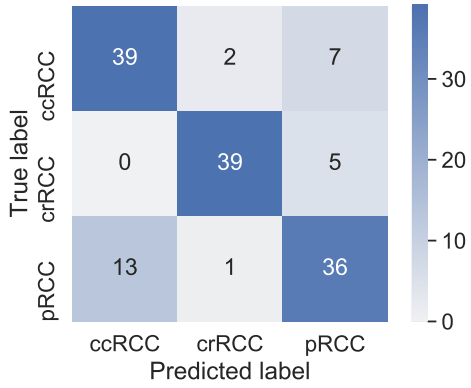
(b) tRNAsformer$_{L=2}$



(c) tRNAsformer$_{L=4}$

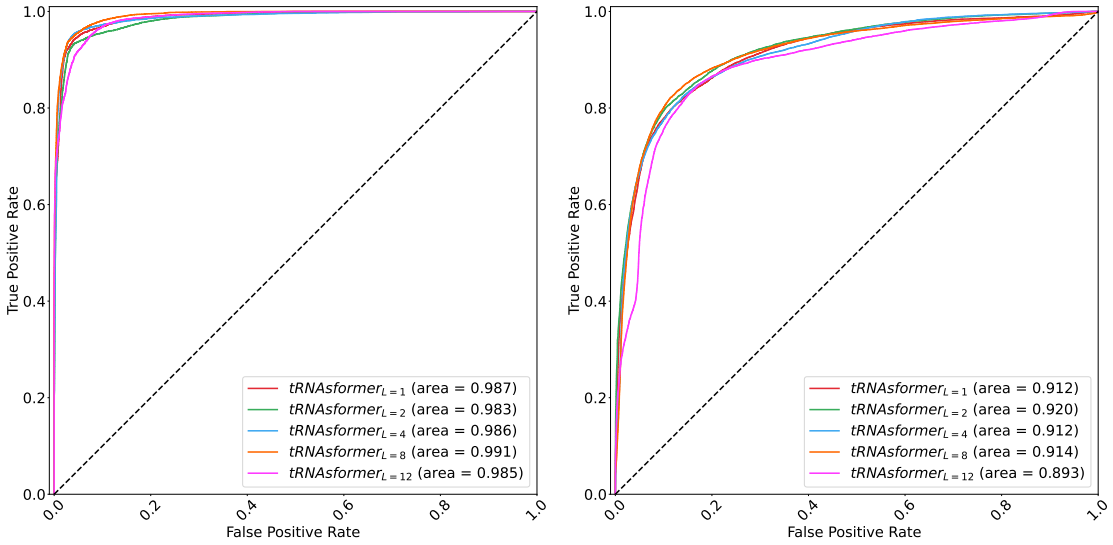


(d) tRNAsformer$_{L=8}$



(e) tRNAsformer$_{L=12}$

Figure 4.7: The confusion matrices for different models applied on 14,200 bags created from the external dataset WSIs. (a)-(d) are for tRNAsformer$_L$, $L = (1, 2, 4, 8, 12)$, respectively.

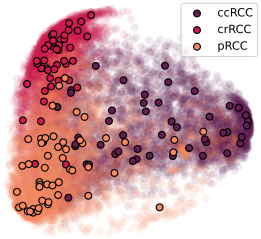


(a) TCGA test dataset. (b) External dataset.

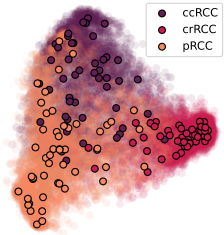
Figure 4.8: The micro ROC curve of different models applied on (a) TCGA test set and (b) the external dataset. The AUC is reported in the legend for all models.

Table 4.5: The accuracy, macro, and weighted F1 scores for classification on TCGA test set and the external dataset for all classification models.

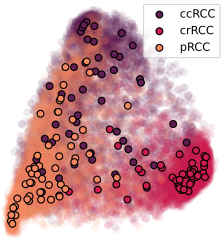
	TCGA			External dataset		
		F1 score			F1 score	
Model	Accuracy	macro	weighted	Accuracy	macro	weighted
tRNAsformer$_{L=1}$	93.75%	0.9488	0.9366	82.39%	0.8241	0.8223
tRNAsformer$_{L=2}$	95.00%	0.9406	0.9496	81.69%	0.8161	0.8145
tRNAsformer$_{L=4}$	96.25%	0.9511	0.9625	78.87%	0.7899	0.7871
tRNAsformer$_{L=8}$	95.00%	0.9414	0.9502	82.39%	0.8251	0.8227
tRNAsformer$_{L=12}$	92.50%	0.9392	0.9243	80.28%	0.8072	0.8034
Low power method (Ch. 3)	93.75%	0.9488	0.9366	73.76%	0.7388	0.7385
CLAM [64]	58.75%	0.5452	0.6537	72.34%	0.7198	0.7190



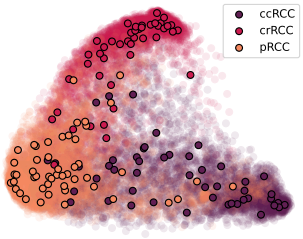
(a) tRNAsformer$_{L=1}$



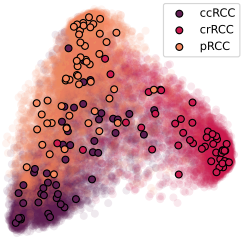
(b) tRNAsformer$_{L=2}$



(c) tRNAsformer$_{L=4}$



(d) tRNAsformer$_{L=8}$



(e) tRNAsformer$_{L=12}$

Figure 4.9: The two-dimensional PCA projection of the external dataset WSI features. (a)-(f) are for tRNAsformer$_L$, $L = (1, 2, 4, 8, 12)$, respectively. Each external test WSI is represented by 100 bags of features. All bags of features associated with the test set are shown with transparent circles. The average of PCA projection of each WSI (average of 100 bags associated with each WSI) is shown in bold circles with black edges.

According to Fig. 3.11, the GP classifier (suggested method in Chapter 3) outperformed all tile-based approaches and CLAM [64]. The GP classifier's accuracy, F1 score (macro and weighted), and AUC were 73.76%, 0.7388, 0.7385, and 0.893, respectively. As it is demonstrated in Table 4.5 and Fig. 4.8, all **tRNAsformer** models surpass the GP classifier in all measures, namely accuracy, F1 score (macro and weighted), and AUC. As a result, **tRNAsformer** surpasses CLAM in WSI classification task on the external dataset as well. Additionally, as it is depicted in Fig. 4.7, the **tRNAsformer** models tend to have more balanced correct predictions for all classes as there is crisp diagonal line highlighted in confusion matrices. To put it another way, **tRNAsformer** models are good at distinguishing between all classes.

4.5.3 Experiment Series 3: WSI Search

WSI search experiments were conducted to assess the quality of the internal representation of the **tRNAsformer**. The model is tested on both TCGA and the external dataset. As it was mentioned earlier in Section 4.2.2 and 4.3 and Subsection 4.5.2, 100 instances were created from each WSI in TCGA dataset; TCGA test set contained 8,000 instances associated with 80 slides. To quantify the performance of **tRNAsformer** in WSI search, first, 100 subsets of instances were created from 8,000 TCGA test instances. Next, a pairwise distance matrix is computed using the WSI embeddings for each subset. The Pearson correlation is employed as the distance metric. Following the *leave-one-patient-out* procedure, the top-k samples were determined for each instance (WSI). Later, P@K and AP@K were computed for each subset. Finally, the Mean Average Precision@K (mAP@K) value was computed by taking average of 100 queries associated with 100 search subsets.

Similarly, 100 instances were created for each WSI in the external dataset. Overall, 100 subsets of 142 WSIs generated for the WSI search in the external dataset. As a result, mAP@K values were evaluated by taking an average from 100 different search experiments. The summary of mAP@K values for both TCGA test and the external dataset are shown in Table 4.6.

To compare **tRNAsformer**'s search results with Yottixel [46], the state-of-the-art in WSI search, the mAP@5 and mAP@10 for Yottixel were calculated. The mAP@5 and mAP@10 for 10 independent Yottixel runs were 0.7416 and 0.7092, respectively. **tRNAsformer** outperforms Yottixel in both mAP@5 and mAP@10 measures. Furthermore, **tRNAsformer** models provide more stability because the mAP@K value does not drop as steeply as other search algorithms while the k increases.

Table 4.6: The mAP@5 and mAP@10 values for all WSI search models applied on TCGA test and the external dataset.

	TCGA		External dataset	
Model	mAP@5	mAP@10	mAP@5	mAP@10
tRNAsformer$_{L=1}$	0.8966	0.8985	0.8026	0.8035
tRNAsformer$_{L=2}$	0.8831	0.8800	0.7988	0.7976
tRNAsformer$_{L=4}$	0.9150	0.9124	0.7819	0.7781
tRNAsformer$_{L=8}$	0.9031	0.8996	0.7674	0.7628
tRNAsformer$_{L=12}$	0.8762	0.8751	0.7262	0.7257
Yottixel [47]	0.764	0.717	0.7416	0.7092

4.6 Conclusions

In this chapter, a multitask MIL framework based on **tRNAsformer** model is proposed for learning WSI representation by learning to predict gene expression from H&E slides. By incorporating the attention mechanism and the Transformer design, **tRNAsformer** can provide more precise predictions for gene expressions from a WSI. Meanwhile, **tRNAsformer** surpassed benchmarks for bulk RNA-seq prediction while having fewer hyperparameters. Additionally, **tRNAsformer** learns exclusive[5] and compact representation for a WSI using molecular signature of the tissue sample. As a result, the proposed technique learns a diagnostically relevant representation from an image by integrating gene information in a multimodal approach.

Furthermore, the Transformer design allowed for more efficient and precise processing of a collection of samples. This property eliminates the need for costly and time-consuming pixel-by-pixel human annotations. Finally, sampling and embedding image tiles using pretrained CNN models offers several advantages:

- Trained on large image datasets, deep CNNs can be exploited to create rich intermediate embeddings from image samples.
- Working with embedded sampled instances[6] is computationally less expensive in comparison with treating each WSI as an instance. According to Table 4.1, the smallest **tRNAsformer** model can have about 60% less hyperparameter in comparison with

[5] A dedicated paradigm for distilling the bag information into a feature vector for WSI representation.
[6] The tile deep features instead of tiles.

MLP-based model. Additionally, they can be about 72% and 15% faster than *MLP*-based model during training and validation, respectively.

- By augmenting data, bootstrapping meets the requirement for big datasets for training deep models.
- By diversifying the instances in a bag, bootstrapping at test time reduces noise.

In contrast to [36] where the spatial transcriptomics dataset was available, the proposed approach in this chapter uses bulk RNA-seq data. As a result, the model described in this chapter employs a weaker type of supervision, as it learns internal representation using a combination of a primary diagnosis and a bulk RNA-seq associated with a WSI. Furthermore, **tRNAsformer** handles the problem by treating a WSI in its entirety, whereas the method explained in [36] separates each tile and estimates the gene expression value for it. Therefore, the method described in[36] ignores the dependencies between tiles. Comparing to [100], the proposed technique in this dissertation processes a considerably smaller set of samples with a larger field of view. In particular, the proposed technique samples bags of 49 instances of $224 \times 224 \times 3$ while the other technique [100] deployed several sampling options with at least 2,500 tiles of size $32 \times 32 \times 3$ per bag. In addition, **tRNAsformer** learns exclusive WSI representation by learning the pixel-to-gene translation. On the other hand, none of the methodologies have an independent representation learning paradigm [84, 36, 100].

In conclusion, the proposed framework can learn reliable internal representations for massive archives of pathology slides that match or outperform the performance of cutting-edge classification and search algorithms developed [46, 64]. It can also predict gene expressions from H&E slides better than other methods [84]. By employing a balanced architecture, the proposed model outperforms existing topologies in both tasks simultaneously.

Chapter 5

Summary and Conclusions

5.1 Thesis Summary

This thesis put forward two techniques to represent gigapixel glass slides utilizing slide-level annotation. In Chapter 3, a novel paradigm for learning visual patterns at low power was proposed that outperformed state-of-the-art WSI representation learning approaches for WSI search and classification in histopathology. Aside from performance, the suggested algorithm is computationally less expensive for analyzing a single WSI. Additionally, the proposed method embeds gigapixel WSIs into a compact fixed-length vector regardless of the original size of the slide.

In Chapter 4, a new pipeline was proposed to learn WSI representations by translating morphological features into the bulk RNA sequences associated with WSIs. The new **tRNAsformer** model, built on the existing Transformer design, allows processing labelled bags of instances instead of requiring exhaustive labelling of a large number of instances. The **tRNAsformer** learns a diagnostically relevant representation of a slide while constrained on learning gene expressions. Not only can the proposed model generate a compact and discriminative representation from a gigapixel slide, but it also connects visual characteristics of the tissue morphology to the gene information. This capability of **tRNAsformer** provides an affordable solution to initially estimate the gene profile of patients from H&E slides. Finally, both suggested algorithms were applied on an independent test cohort for external validation to investigate the generalization of the techniques during domain shift.

Both of the proposed methods need more examination and validation before they can be used in clinical practice. In order to further understand the low-power technique's ability

to treat a wide range of cancers, additional research is needed. As well, the tRNAsformer performance should also be thoroughly analyzed with respect to specific genes related to distinct cancer types. Finally, before clinical application, it is necessary to investigate misclassified cases and irrelevant retrievals to shed light on any model limitations.

5.2 Future Research

In this dissertation, all methods were trained on RCC cases from TCGA dataset. Therefore, pan-cancer research with additional body parts and cancer types might be an excellent place to start [47, 17]. The diversification of samples will help the generalization. Furthermore, a bulk RNA-seq contains genetic information related to all activities in the human body. As a result, incorporating more samples linked with various organs and illnesses provides the prediction model with additional relevant data. Similar to training data, expansion and diversification of the external test dataset is critical to assess potential domain shifts. Also, an external dataset of paired slides and bulk RNA-seq would be useful for gene prediction assessment when it comes to larger datasets.

As it was explained in Chapter 2 Section 2.1 digitized WSIs exhibit heterogeneous patterns of tissue morphology at different magnification levels. As a result, analyzing tissue patterns at different magnifications at the same time and fusing the knowledge can help to improve the WSI representation [105]. The proposed paradigm in Chapter 3 can also be utilized to learn and embed large regions at any magnification. As a result, the size of the instances can vary depending on the tasks, tissue type and primary site. Also, the bag of instances in Chapter 4 can be diversified by including instances from various magnifications.

The size of the database, the retrieval technique, and the length of the embeddings can all add to the complexity of image search. One of the approaches to reduce the search complexity is an initial refinement of the results. As a result, the search may be divided into two stages; First, a rapid approach may be used to pick a large number of candidates. The outcomes can then be filtered and modified for the next stage. Even for circumstances that may require further investigation at higher magnifications, the paradigm described in Chapter 3 is a viable solution for the early refining stage.

Several techniques were tested in this dissertation on both the TCGA data set as well as a private kidney dataset from the Ohio State University (external data set). Even though all algorithms generalized to the external dataset to some extent, it appears that there is still a performance gap as a result of the domain shift. One approach to dealing with the

problem of generalization is to train models on more extensive and more diverse datasets. However, Patients' privacy and a time-consuming and expensive data-gathering process are significant obstacles to generating massive clinical datasets in digital pathology. In order to solve this type of problem, domain adaptation would be a good candidate for consideration.

Last but not least, a comprehensive study of biological findings of **tRNAsformer** model. The Gene Ontology (GO) study, for example, may be used to validate the model in order to discover relationships between visual patterns and genes. The GO describes our understanding of the biological world in three ways; molecular-level, cellular components, and biological process. This investigation would need clinical expertise and oversight.

References

[1] deroneriksson/python-wsi-preprocessing: Python whole slide image preprocessing. https://github.com/deroneriksson/python-wsi-preprocessing. (Accessed on 09/24/2021).

[2] Gdc. https://portal.gdc.cancer.gov/. (Accessed on 10/14/2021).

[3] Gene. https://www.genome.gov/genetics-glossary/Gene. (Accessed on 09/18/2021).

[4] H&e and special staining - histology services - research cro custom services. https://www.abmgood.com/H-and-E-and-Special-Staining.html. (Accessed on 10/25/2021).

[5] Philips intellisite pathology solutions - media library — philips. https://www.philips.com/a-w/about/news/media-library/20181010-Philips-IntelliSite-Pathology-Solutions.html. (Accessed on 10/24/2021).

[6] Statquest: A gentle introduction to rna-seq - youtube. https://www.youtube.com/watch?v=tlf6wYJrwKY. (Accessed on 10/15/2021).

[7] Types of biopsies used to look for cancer. https://www.cancer.org/treatment/understanding-your-diagnosis/tests/testing-biopsy-and-cytology-specimens-for-cancer/biopsy-types.html. (Accessed on 10/24/2021).

[8] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval: The Concepts and Technology Behind Search*. Addison-Wesley Professional, 2nd edition, 2011.

[9] Laura Barisoni, Charlotte Gimpel, Renate Kain, Arvydas Laurinavicius, Gloria Bueno, Caihong Zeng, Zhihong Liu, Franz Schaefer, Matthias Kretzler, Lawrence B Holzman, et al. Digital pathology imaging as a novel platform for standardization and globalization of quantitative nephropathology. *Clinical kidney journal*, 10(2):176–187, 2017.

[10] Babak Ehteshami Bejnordi, Mitko Veta, Paul Johannes Van Diest, Bram Van Ginneken, Nico Karssemeijer, Geert Litjens, Jeroen AWM Van Der Laak, Meyke Hermsen, Quirine F Manson, Maschenka Balkenhol, et al. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *Jama*, 318(22):2199–2210, 2017.

[11] Babak Ehteshami Bejnordi, Guido Zuidhof, Maschenka Balkenhol, Meyke Hermsen, Peter Bult, Bram van Ginneken, Nico Karssemeijer, Geert Litjens, and Jeroen van der Laak. Context-aware stacked convolutional neural networks for classification of breast carcinomas in whole-slide histopathology images. *Journal of Medical Imaging*, 4(4):044504, 2017.

[12] Aicha Ben Taieb. *Analyzing cancers in digitized histopathology images*. PhD thesis, Applied Sciences: School of Computing Science, 2018.

[13] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[14] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1):289–300, 1995.

[15] Wouter Bulten, Hans Pinckaers, Hester van Boven, Robert Vink, Thomas de Bel, Bram van Ginneken, Jeroen van der Laak, Christina Hulsbergen-van de Kaa, and Geert Litjens. Automated deep-learning system for gleason grading of prostate cancer using biopsies: a diagnostic study. *The Lancet Oncology*, 21(2):233–241, 2020.

[16] Gabriele Campanella, Matthew G Hanna, Luke Geneslaw, Allen Miraflor, Vitor Werneck Krauss Silva, Klaus J Busam, Edi Brogi, Victor E Reuter, David S Klimstra, and Thomas J Fuchs. Clinical-grade computational pathology using weakly supervised deep learning on whole slide images. *Nature medicine*, 25(8):1301–1309, 2019.

[17] Anika Cheerla and Olivier Gevaert. Deep learning with multimodal representation for pancancer prognosis prediction. *Bioinformatics*, 35(14):i446–i454, 2019.

[18] Yijiang Chen, Jarcy Zee, Abigail Smith, Catherine Jayapandian, Jeffrey Hodgin, David Howell, Matthew Palmer, David Thomas, Clarissa Cassol, Alton B Farris III, et al. Assessment of a computerized quantitative quality control tool for whole slide images of kidney biopsies. *The Journal of Pathology*, 253(3):268–278, 2021.

[19] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*, 2021.

[20] Dan C Cireşan, Alessandro Giusti, Luca M Gambardella, and Jürgen Schmidhuber. Mitosis detection in breast cancer histology images with deep neural networks. In *International conference on medical image computing and computer-assisted intervention*, pages 411–418. Springer, 2013.

[21] Nicolas Coudray, Paolo Santiago Ocampo, Theodore Sakellaropoulos, Navneet Narula, Matija Snuderl, David Fenyö, Andre L Moreira, Narges Razavian, and Aristotelis Tsirigos. Classification and mutation prediction from non–small cell lung cancer histopathology images using deep learning. *Nature medicine*, 24(10):1559–1567, 2018.

[22] Angel Cruz-Roa, Ajay Basavanhally, Fabio González, Hannah Gilmore, Michael Feldman, Shridar Ganesan, Natalie Shih, John Tomaszewski, and Anant Madabhushi. Automatic detection of invasive ductal carcinoma in whole slide images with convolutional neural networks. In *Medical Imaging 2014: Digital Pathology*, volume 9041, page 904103. International Society for Optics and Photonics, 2014.

[23] Angel Cruz-Roa, Hannah Gilmore, Ajay Basavanhally, Michael Feldman, Shridar Ganesan, Natalie Shih, John Tomaszewski, Anant Madabhushi, and Fabio González. High-throughput adaptive sampling for whole-slide histopathology image analysis (hashi) via convolutional neural networks: Application to invasive breast cancer detection. *PloS one*, 13(5):e0196828, 2018.

[24] Angel Cruz-Roa, Hannah Gilmore, Ajay Basavanhally, Michael Feldman, Shridar Ganesan, Natalie NC Shih, John Tomaszewski, Fabio A González, and Anant Madabhushi. Accurate and reproducible invasive breast cancer detection in whole-slide images: A deep learning approach for quantifying tumor extent. *Scientific reports*, 7(1):1–14, 2017.

[25] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[26] Ugljesa Djuric, Gelareh Zadeh, Kenneth Aldape, and Phedias Diamandis. Precision histology: how deep learning is poised to revitalize histomorphology for personalized cancer care. *NPJ precision oncology*, 1(1):1–5, 2017.

[27] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[28] Kevin Faust, Sudarshan Bala, Randy Van Ommeren, Alessia Portante, Raniah Al Qawahmed, Ugljesa Djuric, and Phedias Diamandis. Intelligent feature engineering and ontological mapping of brain tumour histomorphologies by deep learning. *Nature Machine Intelligence*, 1(7):316–321, 2019.

[29] Kevin Faust, Quin Xie, Dominick Han, Kartikay Goyle, Zoya Volynskaya, Ugljesa Djuric, and Phedias Diamandis. Visualizing histopathologic deep learning classification and anomaly detection using nonlinear feature space dimensionality reduction. *BMC Bioinformatics*, 19(1):173, 2018.

[30] Andrew H Fischer, Kenneth A Jacobson, Jack Rose, and Rolf Zeller. Hematoxylin and eosin staining of tissue and cell sections. *Cold spring harbor protocols*, 2008(5):pdb–prot4986, 2008.

[31] Jacob Gildenblat and Eldad Klaiman. Self-supervised similarity learning for digital pathology. *arXiv preprint arXiv:1905.08139*, 2019.

[32] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[33] Simon Graham, Muhammad Shaban, Talha Qaiser, Navid Alemi Koohbanani, Syed Ali Khurram, and Nasir Rajpoot. Classification of lung cancer histology images using patch-level summary statistics. In *Medical Imaging 2018: Digital Pathology*, volume 10581, page 1058119. International Society for Optics and Photonics, 2018.

[34] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *arXiv preprint arXiv:2103.00112*, 2021.

[35] Yehudit Hasin, Marcus Seldin, and Aldons Lusis. Multi-omics approaches to disease. *Genome biology*, 18(1):1–15, 2017.

[36] Bryan He, Ludvig Bergenstråhle, Linnea Stenbeck, Abubakar Abid, Alma Andersson, Åke Borg, Jonas Maaskola, Joakim Lundeberg, and James Zou. Integrating spatial gene expression and breast tumour morphology via deep learning. *Nature biomedical engineering*, 4(8):827–834, 2020.

[37] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[38] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[39] Markus D Herrmann, David A Clunie, Andriy Fedorov, Sean W Doyle, Steven Pieper, Veronica Klepeis, Long P Le, George L Mutter, David S Milstone, Thomas J Schultz, et al. Implementing the dicom standard for digital pathology. *Journal of pathology informatics*, 9, 2018.

[40] Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70, 1979.

[41] Le Hou, Dimitris Samaras, Tahsin M Kurc, Yi Gao, James E Davis, and Joel H Saltz. Patch-based convolutional neural network for whole slide tissue image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2424–2433, 2016.

[42] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[43] Tim Hubbard, Daniel Barker, Ewan Birney, Graham Cameron, Yuan Chen, L Clark, Tony Cox, J Cuff, Val Curwen, Thomas Down, et al. The ensembl genome database project. *Nucleic acids research*, 30(1):38–41, 2002.

[44] Andrew Janowczyk and Anant Madabhushi. Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases. *Journal of pathology informatics*, 7, 2016.

[45] David Juan, Gabriela Alexe, Travis Antes, Huiqing Liu, Anant Madabhushi, Charles Delisi, Shridhar Ganesan, Gyan Bhanot, and Louis S Liou. Identification of a microrna panel for clear-cell kidney cancer. *Urology*, 75(4):835–841, 2010.

[46] Shivam Kalra, Hamid R Tizhoosh, Charles Choi, Sultaan Shah, Phedias Diamandis, Clinton JV Campbell, and Liron Pantanowitz. Yottixel–an image search engine for large archives of histopathology whole slide images. *Medical Image Analysis*, 65:101757, 2020.

[47] Shivam Kalra, Hamid R Tizhoosh, Sultaan Shah, Charles Choi, Savvas Damaskinos, Amir Safarpoor, Sobhan Shafiei, Morteza Babaie, Phedias Diamandis, Clinton JV Campbell, et al. Pan-cancer diagnostic consensus through searching archival histopathology images using artificial intelligence. *NPJ digital medicine*, 3(1):1–15, 2020.

[48] Rick Kamps, Rita D Brandão, Bianca J Bosch, Aimee DC Paulussen, Sofia Xanthoulea, Marinus J Blok, and Andrea Romano. Next-generation sequencing in oncology: genetic diagnosis, risk prediction and cancer classification. *International journal of molecular sciences*, 18(2):308, 2017.

[49] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[50] Daisuke Komura and Shumpei Ishikawa. Machine learning methods for histopathological image analysis. *Computational and structural biotechnology journal*, 16:34–42, 2018.

[51] Bin Kong, Xin Wang, Zhongyu Li, Qi Song, and Shaoting Zhang. Cancer metastasis detection via spatially structured deep network. In *International Conference on Information Processing in Medical Imaging*, pages 236–248. Springer, 2017.

[52] LF Kozachenko and Nikolai N Leonenko. Sample estimate of the entropy of a random vector. *Problemy Peredachi Informatsii*, 23(2):9–16, 1987.

[53] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical review E*, 69(6):066138, 2004.

[54] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[55] Kimberly R Kukurba and Stephen B Montgomery. Rna sequencing and analysis. *Cold Spring Harbor Protocols*, 2015(11):pdb–top084970, 2015.

[56] Eric S Lander. Array of hope. *Nature genetics*, 21(1):3–4, 1999.

[57] Thomas J Lawton, Geza Acs, Pedram Argani, Gelareh Farshid, Michael Gilcrease, Neal Goldstein, Frederick Koerner, J Jordi Rowe, Melinda Sanders, Sejal S Shah, et al. Interobserver variability by pathologists in the distinction between cellular fibroadenomas and phyllodes tumors. *International journal of surgical pathology*, 22(8):695–698, 2014.

[58] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

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

[60] Alona Levy-Jurgenson, Xavier Tekpli, Vessela N Kristensen, and Zohar Yakhini. Spatial transcriptomics inferred from pathology whole-slide images links tumor heterogeneity to survival in breast and lung cancer. *Scientific reports*, 10(1):1–11, 2020.

[61] Huangjing Lin, Hao Chen, Qi Dou, Liansheng Wang, Jing Qin, and Pheng-Ann Heng. Scannet: A fast and dense scanning framework for metastastic breast cancer detection from whole-slide image. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 539–546. IEEE, 2018.

[62] Geert Litjens, Peter Bandi, Babak Ehteshami Bejnordi, Oscar Geessink, Maschenka Balkenhol, Peter Bult, Altuna Halilovic, Meyke Hermsen, Rob van de Loo, Rob Vogels, et al. 1399 h&e-stained sentinel lymph node sections of breast cancer patients: the camelyon dataset. *GigaScience*, 7(6):giy065, 2018.

[63] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[64] Ming Y Lu, Drew FK Williamson, Tiffany Y Chen, Richard J Chen, Matteo Barbieri, and Faisal Mahmood. Data-efficient and weakly supervised computational pathology on whole-slide images. *Nature Biomedical Engineering*, 5:555–570, 2021.

[65] Marc Macenko, Marc Niethammer, James S Marron, David Borland, John T Woosley, Xiaojun Guan, Charles Schmitt, and Nancy E Thomas. A method for normalizing histology slides for quantitative analysis. In *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1107–1110. IEEE, 2009.

[66] Danial Maleki, Mehdi Afshari, Morteza Babaie, and Hamid R Tizhoosh. Ink marker segmentation in histopathology images using deep learning. In *International Symposium on Visual Computing*, pages 359–368. Springer, 2020.

[67] Michael T McCann, John A Ozolek, Carlos A Castro, Bahram Parvin, and Jelena Kovacevic. Automated histology analysis: Opportunities for signal processing. *IEEE Signal Processing Magazine*, 32(1):78–87, 2014.

[68] Ultan McDermott, James R Downing, and Michael R Stratton. Genomics and the continuum of cancer care. *New England Journal of Medicine*, 364(4):340–350, 2011.

[69] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Icml*, 2010.

[70] Peter Naylor, Marick Laé, Fabien Reyal, and Thomas Walter. Segmentation of nuclei in histopathology images by deep regression of the distance map. *IEEE transactions on medical imaging*, 38(2):448–459, 2018.

[71] Muhammad Khalid Khan Niazi, Anil V Parwani, and Metin N Gurcan. Digital pathology and artificial intelligence. *The Lancet Oncology*, 20(5):253–261, 2019.

[72] Christopher Olah. Understanding lstm networks – colah's blog. https://colah.github.io/posts/2015-08-Understanding-LSTMs/. (Accessed on 09/19/2021).

[73] Fatih Ozsolak and Patrice M Milos. Rna sequencing: advances, challenges and opportunities. *Nature reviews genetics*, 12(2):87–98, 2011.

[74] Liron Pantanowitz. Digital images and the future of digital pathology. *Journal of pathology informatics*, 1, 2010.

[75] Jonas Pichat, Juan Eugenio Iglesias, Tarek Yousry, Sébastien Ourselin, and Marc Modat. A survey of methods for 3d histology reconstruction. *Medical image analysis*, 46:73–105, 2018.

[76] Talha Qaiser and Nasir M Rajpoot. Learning where to see: A novel attention model for automated immunohistochemical scoring. *IEEE Transactions on Medical Imaging*, 38(11):2620–2631, 2019.

[77] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[78] Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, et al. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv:1711.05225*, 2017.

[79] JA Ramos-Vara and MA Miller. When tissue antigens and antibodies get along: revisiting the technical aspects of immunohistochemistry—the red, brown, and blue technique. *Veterinary pathology*, 51(1):42–87, 2014.

[80] Abtin Riasatian, Morteza Babaie, Danial Maleki, Shivam Kalra, Mojtaba Valipour, Sobhan Hemati, Manit Zaveri, Amir Safarpoor, Sobhan Shafiei, Mehdi Afshari, et al. Fine-tuning and training of densenet for histopathology image representation using tcga diagnostic slides. *Medical Image Analysis*, 70:102032, 2021.

[81] Brian C Ross. Mutual information between discrete and continuous data sets. *PloS one*, 9(2):e87357, 2014.

[82] Santanu Roy, Alok kumar Jain, Shyam Lal, and Jyoti Kini. A study about color normalization methods for histopathology images. *Micron*, 114:42–61, 2018.

[83] Andrew J Schaumberg, Mark A Rubin, and Thomas J Fuchs. H&e-stained whole slide image deep learning predicts spop mutation state in prostate cancer. *BioRxiv*, page 064279, 2017.

[84] Benoît Schmauch, Alberto Romagnoni, Elodie Pronier, Charlie Saillard, Pascale Maillé, Julien Calderaro, Aurélie Kamoun, Meriem Sefta, Sylvain Toldo, Mikhail Zaslavskiy, et al. A deep learning model to predict rna-seq expression of tumours from whole slide images. *Nature communications*, 11(1):1–15, 2020.

[85] Eran Segal, Nir Friedman, Naftali Kaminski, Aviv Regev, and Daphne Koller. From signatures to models: understanding cancer using microarrays. *Nature genetics*, 37(6):S38–S45, 2005.

[86] Muhammad Shaban, Ruqayya Awan, Muhammad Moazam Fraz, Ayesha Azam, Yee-Wah Tsang, David Snead, and Nasir M. Rajpoot. Context-aware convolutional neural network for grading of colorectal cancer histology images. *IEEE Transactions on Medical Imaging*, 39(7):2395–2405, 2020.

[87] Brian Shuch, Ali Amin, Andrew J Armstrong, John N Eble, Vincenzo Ficarra, Antonio Lopez-Beltran, Guido Martignoni, Brian I Rini, and Alexander Kutikov. Understanding pathologic variants of renal cell carcinoma: distilling therapeutic opportunities from biologic complexity. *European Urology*, 67(1):85–97, 2015.

[88] Milad Sikaroudi, Amir Safarpoor, Benyamin Ghojogh, Sobhan Shafiei, Mark Crowley, and Hamid R Tizhoosh. Supervision and source domain impact on representation learning: A histopathology case study. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pages 1400–1403. IEEE, 2020.

[89] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[90] Fabio A Spanhol, Luiz S Oliveira, Paulo R Cavalin, Caroline Petitjean, and Laurent Heutte. Deep features for breast cancer histopathological image classification. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1868–1873. IEEE, 2017.

[91] Eleftherios Spyromitros-Xioufis, Grigorios Tsoumakas, William Groves, and Ioannis Vlahavas. Multi-target regression via input space expansion: treating targets as inputs. *Machine Learning*, 104(1):55–98, 2016.

[92] Chetan L Srinidhi, Ozan Ciga, and Anne L Martel. Deep neural network models for computational histopathology: A survey. *Medical Image Analysis*, 67:101813, 2021.

[93] Aravind Srinivas, Tsung-Yi Lin, Niki Parmar, Jonathon Shlens, Pieter Abbeel, and Ashish Vaswani. Bottleneck transformers for visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16519–16529, 2021.

[94] Karin Stacke, Gabriel Eilertsen, Jonas Unger, and Claes Lundström. A closer look at domain shift for deep learning in histopathology. *arXiv preprint arXiv:1909.11575*, 2019.

[95] Hyuna Sung, Jacques Ferlay, Rebecca L Siegel, Mathieu Laversanne, Isabelle Soerjomataram, Ahmedin Jemal, and Freddie Bray. Global cancer statistics 2020: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: A Cancer Journal for Clinicians*, 71(3):209–249, 2021.

[96] Zaneta Swiderska-Chadaj, Thomas de Bel, Lionel Blanchet, Alexi Baidoshvili, Dirk Vossen, Jeroen van der Laak, and Geert Litjens. Impact of rescanning and normalization on convolutional neural network performance in multi-center, whole-slide classification of prostate cancer. *Scientific Reports*, 10:14398, 2020.

[97] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[98] Sairam Tabibu, PK Vinod, and CV Jawahar. Pan-renal cell carcinoma classification and survival prediction from histopathology images using deep learning. *Scientific Reports*, 9:10509, 2019.

[99] A Jamil Tajik. Machine learning for echocardiographic imaging: embarking on another incredible journey, 2016.

[100] Thomas E Tavolara, MKK Niazi, Adam C Gower, Melanie Ginese, Gillian Beamer, and Metin N Gurcan. Deep learning predicts gene expression as an intermediate data modality to identify susceptibility patterns in mycobacterium tuberculosis infected diversity outbred mice. *EBioMedicine*, 67:103388, 2021.

[101] David Tellez, Geert Litjens, Jeroen van der Laak, and Francesco Ciompi. Neural image compression for gigapixel histopathology image analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(2):567–578, 2021.

[102] M Titford. The long history of hematoxylin. *Biotechnic & histochemistry*, 80(2):73–78, 2005.

[103] Hamid R Tizhoosh, Phedias Diamandis, Clinton JV Campbell, Amir Safarpour, Shivam Kalra, Danial Maleki, Abtin Riasatian, and Morteza Babaie. Searching images for consensus: Can ai remove observer variability in pathology? *The American Journal of Pathology*, 2021.

[104] Hamid Reza Tizhoosh and Liron Pantanowitz. Artificial intelligence and digital pathology: challenges and opportunities. *Journal of pathology informatics*, 9, 2018.

[105] Hiroki Tokunaga, Yuki Teramoto, Akihiko Yoshizawa, and Ryoma Bise. Adaptive weighting multi-field-of-view cnn for semantic segmentation in pathology. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12597–12606, 2019.

[106] Thaína A Azevedo Tosta, Paulo Rogério de Faria, Leandro Alves Neves, and Marcelo Zanchetta do Nascimento. Computational normalization of h&e-stained histological images: Progress, challenges and future potential. *Artificial Intelligence in Medicine*, 95:118–132, 2019.

[107] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.

[108] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. *arXiv preprint arXiv:2103.17239*, 2021.

[109] Martin J van den Bent. Interobserver variation of the histopathological diagnosis in clinical trials on glioma: a clinician's perspective. *Acta neuropathologica*, 120(3):297–304, 2010.

[110] Jeroen AWM Van der Laak, Martin MM Pahlplatz, Antonius GJM Hanselaar, and Peter CM de Wilde. Hue-saturation-density (hsd) model for stain recognition in digital images from transmitted light microscopy. *Cytometry: The Journal of the International Society for Analytical Cytology*, 39(4):275–284, 2000.

[111] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[112] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

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

[114] Xi Wang, Hao Chen, Caixia Gan, Huangjing Lin, Qi Dou, Efstratios Tsougenis, Qitao Huang, Muyan Cai, and Pheng-Ann Heng. Weakly supervised deep learning for whole slide lung cancer image analysis. *IEEE Transactions on Cybernetics*, 50(9):3950–3962, 2020.

[115] JD Webster and RW Dunstan. Whole-slide imaging and automated image analysis: considerations and opportunities in the practice of pathology. *Veterinary pathology*, 51(1):211–223, 2014.

[116] Sean R Williamson, Priya Rao, Ondrej Hes, Jonathan I Epstein, Steven C Smith, Maria M Picken, Ming Zhou, Maria S Tretiakova, Satish K Tickoo, Ying-Bei Chen, et al. Challenges in pathologic staging of renal cell carcinoma. *The American journal of surgical pathology*, 42(9):1253–1261, 2018.

[117] D Wittekind. Traditional staining for routine diagnostic pathology including the role of tannic acid. 1. value and limitations of the hematoxylin-eosin stain. *Biotechnic & histochemistry*, 78(5):261–270, 2003.

[118] Weidi Xie, J Alison Noble, and Andrew Zisserman. Microscopy cell counting and detection with fully convolutional regression networks. *Computer methods in biomechanics and biomedical engineering: Imaging & Visualization*, 6(3):283–292, 2018.

[119] Jun Xu, Lei Xiang, Qingshan Liu, Hannah Gilmore, Jianzhong Wu, Jinghai Tang, and Anant Madabhushi. Stacked sparse autoencoder (ssae) for nuclei detection on breast cancer histopathology images. *IEEE transactions on medical imaging*, 35(1):119–130, 2015.

[120] Yu Zhang, Peter Tiňo, Aleš Leonardis, and Ke Tang. A survey on neural network interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2021.