# Learning Discriminative Representations for Gigapixel Images

by

Shivam Kalra

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, 2022

**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: | Nasir Rajpoot<br>Director, Tissue Image Analytics (TIA) Centre,<br>University of Warwick |
| Supervisor: | Hamid Tizhoosh<br>Professor, Artificial Intelligence and Informatics,<br>Mayo Clinic |
| Internal-External Member: | Otman Basir<br>Professor, Dept. of Electrical and Computer Engineering,<br>University of Waterloo |
| Internal Member: | Parsin Haji Reza<br>Assistant Professor, Dept. of Systems Design Engineering,<br>University of Waterloo |
| Internal Member: | Siby Samuel<br>Assistant Professor, Dept. of Systems Design Engineering,<br>University of Waterloo |

## Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the 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.

## Statement of Contributions

All experimental results, figures, and text are generated from my own work during this Ph.D. research. The thesis is based on the following papers that I have published.

A. **S. Kalra**, et al. *Yottixel–an image search engine for large archives of histopathology whole slide images.* Medical Image Analysis 65 (2020)

B. **S. Kalra**, et al. *Pan-cancer diagnostic consensus through searching archival histopathology images using artificial intelligence.* NPJ digital medicine 3.1 (2020).

C. **S. Kalra**, et al. *Learning permutation invariant representations using memory networks.* European Conference on Computer Vision (ECCV) (2020).

D. **S. Kalra**, et al. *Pay Attention with Focus: A Novel Learning Scheme for Classification of Whole Slide Images.* International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI) (2021)

E. M. Adnan, **S. Kalra**, et al. *Representation learning of histopathology images using graph neural networks.* Conference on Computer Vision and Pattern Recognition Workshops (CVPR-W) (2020)

F. M. Adnan, **S. Kalra**, et al. *Federated learning and differential privacy for medical image analysis.* Nature Scientific Reports (2022)

G. **S. Kalra**, et al. *ProxyFL: Decentralized Federated Learning through Proxy Model Sharing.* [Under Review] (Available at Arxiv 2111.11343).

# Abstract

Digital images of tumor tissue are important diagnostic and prognostic tools for pathologists. Recent advancement in digital pathology has led to an abundance of digitized histopathology slides, called *whole-slide images*. Computational analysis of whole-slide images is a challenging task as they are generally gigapixel files, often one or more gigabytes in size. However, these computational methods provide a unique opportunity to improve the objectivity and accuracy of diagnostic interpretations in histopathology. Recently, deep learning has been successful in characterizing images for vision-based applications in multiple domains. But its applications are relatively less explored in the histopathology domain mostly due to the following two challenges. Firstly, there is difficulty in scaling deep learning methods for processing large gigapixel histopathology images. Secondly, there is a lack of diversified and labeled datasets due to privacy constraints as well as workflow and technical challenges in healthcare sector. The main goal for this dissertation is to explore and develop deep models to learn discriminative representations of whole slide images while overcoming the existing challenges. A three-staged approach was considered in this research. In the first stage, a framework called "Yottixel" is proposed. It represents a whole-slide image as a set of multiple representative patches, called *mosaic*. The mosaic enables convenient processing and compact representation of an entire high-resolution whole-slide image. Yottixel allows faster retrieval of similar whole-slide images within a large archives of digital histopathology images. Such retrieval technology enables pathologists to tap into the past diagnostic data on demand. Yottixel is validated on the largest public archive of whole-slide images (*The Cancer Genomic Atlas*), achieving promising results. Yottixel is an unsupervised method that limits its performance on specific tasks especially when the labeled (or partially labeled) dataset can be available. In the second stage, *multi-instance learning* (MIL) is used to enhance the cancer subtype prediction through weakly-supervised training. Three MIL methods have been proposed, each improving upon the previous one. The first one is based on memory-based models, the second uses attention-based models, and the third one uses graph neural networks. All three methods are incorporated in Yottixel to classify entire whole-slide images with no pixel-level annotations. Access to large-scale and diversified datasets is a primary driver of the advancement and adoption of machine learning technologies. However, healthcare has many restrictive rules around data sharing, limiting research and model development. In the final stage, a *federated learning* scheme called "ProxyFL" is developed that enables collaborative training of Yottixel among the multiple healthcare organizations without centralization of the sensitive medical data. The combined research in all the three stages of the PhD has resulted into the development of a holistic and practical framework for learning discriminative and compact representations of whole-slide images in digital pathology.

# Acknowledgements

Writing this thesis has been fascinating and extremely rewarding. I'd like to thank many people who have contributed to my thesis in several ways.

First and foremost, thanks to my supervisor, Professor Hamid R. Tizhoosh for his constant support, encouragement, and patience throughout the PhD. I'm glad be part of his lab, Kimia Lab, where so many researchers are free to explore creative ideas.

I'd like to thank my co-authors Adnan, Sobhan, Jesse, Junfeng, Morteza for all their collaboration in conducting the research.

I owe a special thanks to Prof. Otmon Basir, Prof. Parsin H. Reza, Prof. Siby Samuel to be part of my thesis committee and for taking out their time to review my thesis and providing their valuable suggestions. I'd also like to thank Prof. Nasir Rajpoot to be an external committee member for my thesis, and to his valuable insights.

I would like to acknowledge the scholarship programs offered by Govt. of Canada (NSERC), University of Waterloo (PGS), MITACS (Accelerate), Waterloo.ai (AI Scholarship), Vector Institute (Post-graduate affiliation scholarship) for providing the much needed financial support throughout my PhD.

I'd like to thank my parents, Dr. Naveen and Renu Kalra for raising me to value the education. I deeply appreciate all family, friends, and Jasneet for their infinite support and guidance. Finally, I pay my obeisance to god, the almighty to have bestowed upon me the good health, courage, inspiration, and the light.

# Dedication

*To my beloved parents.*

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1

# Introduction

Histopathology is the gold standard for diagnosing cancer and assessing its prognosis. One of the major obstacles in reaching diagnostic consensus is *observer variability*. It is described as the degree of variation between the diagnostic interpretations when a set of cases are examined by two or more independent clinicians [3]. Cancer diagnoses tend to be highly variable especially as the number of diagnostic criteria continues to evolve in the era of modern medicine [4, 5]. The digitization of histopathology has created a unique opportunity to improve the objectivity and accuracy of diagnostic interpretations through machine learning, particularly deep learning [6]. In this context, a critical question addressed in this dissertation is—*whether the fundamental challenge of diagnostic imaging can be resolved using deep learning. Yottixel* (a portmanteau for *one-yotta-pixel*), is an assistive image search technology for histopathology images developed during this Ph.D. research. It uses deep learning and other machine-learning methods to extract compact and discriminative representations of high-resolution histopathology images. The compact representations enable faster retrieval, and offer lower computational and storage overhead, therefore more feasible in clinical settings. Compared to other computer-vision algorithms, the image search offers an alternative way of building a computational consensus to assist pathologists with "virtual peer review". This PhD research is premised on a hypothesis that the image search technology can potentially remedy the high intra-and inter-observer variability in diagnosis through the search in a large archive of previously (and evidently) diagnosed cases. In other words, an image search technology can assist pathologists by allowing them to tap into the collective wisdom of pathologists that have previously (and evidently) diagnosed similar cases. Learning compact representations of histopathology images exhibits many challenges, especially because these images are gigapixel files, often one or more gigabytes in size. Existing deep learning methods such as convolution neu-

ral networks (CNNs) are insufficient to handle even a single histopathology image in its original resolution. Furthermore, there is a lack of diversified histopathology datasets that contain pixel-level annotations, or delineations, by experts. For this dissertation, a three staged approach is adopted to develop a complete and practical framework for learning discriminative representations of histopathology images. The three stages of PhD research is as follows:

(i) **Stage 1** explores ways to resolve the challenge of processing of gigapixel histopathology images. Existing deep learning methods are unable to process these images in their entirety. Yottixel, a proposed solution divides a histopathology image into a set of representative patches (called *mosaic*). Representing a histopathology image as a mosaic enables the incorporation of existing deep learning methods without the requirement of massive computational and storage overhead.

(ii) **Stage 2** explores strategies to incorporate label information of histopathology images to learn more discriminative representations. Generally, a label (e.g., a cancer subtype) is associated with an entire histopathology image without access to any regional- or pixel-level annotations. The problem at hand is to develop a supervised algorithm that operates on multiple instances (i.e., a mosaic) with a single target label (a cancer subtype). The problem is different from traditional supervised machine learning methods that operate on a single instance and its associated target label. The weak-supervision through multi-instance learning (MIL) is used for training Yottixel using a mosaic, and a target label pairs.

(iii) **Stage 3** explores federated learning (FL) as a distributed and collaborative learning framework to train Yottixel across multiple hospitals while respecting patient privacy. Slide preparation, fixation, and staining techniques utilized at histopathology labs, among other things, cause significant variations in tissue slides. Because of these variations, histopathology images must be integrated across numerous organizations for achieving robustness in the deep models. However, such type of data integration is not possible in medical domain due to regulations and restrictions around data sharing. Existing FL methods are not well-suited for institutionalized applications. Hence, a new scheme for FL is proposed called ProxyFL, especially curated for institutional collaborations in training deep models.

## 1.1 Motivation

The motivation for this PhD research necessitates from two different perspectives. First, the innovation required in machine learning to overcome the challenges of processing and representing the high-resolution histopathology images. Secondly, the feasibility and adoption of machine learning methods in clinical setting. This research is conducted to maintain the balance between these two, i.e., the clinical feasibility and adoption are as important as the innovation in machine learning.

### 1.1.1 Machine Learning Motivations

**Representation of Gigapixel Images.** Traditional approaches to image analysis involved handcrafted and domain-specific features to describe the color, shape, or texture of images. However, handcrafted features are difficult to develop and to transfer to new applications. Naturally, these approaches have been recently overtaken by deep learning. Convolutional neural networks are powerful tools to characterize an image, and such they have gained considerable success recently. However, most recent advances have focused on processing rather small images (i.e., natural images) using deep learning. Extensions of these methods are necessary to handle gigapixel histopathology images and to find subtle differences in diagnostic interpretations.

**Multiple Instance Learning.** Dividing large images into small patches for making class predictions is a common first step in accommodating gigapixel histopathology images. Multiple instance learning (MIL) is a form of weakly supervised learning where training instances are arranged in sets, called bags, and a label is provided for the entire bag. This formulation is gaining interest histopathology domain [7, 8, 9] as it naturally fits various problems and allows to leverage weakly labeled data. A more general MIL approach that contains a full pipeline of processing – mosaicking, and classifying image data in histopathology – is still missing in the literature, and subsequently in clinical utility.

**Distributed, Collaborative, and Privacy-preserving Training.** Tight rules generally govern data sharing in highly regulated industries such as healthcare. Institutions in these disciplines are unable to openly aggregate and communicate their data, limiting research and model development progress. More robust and accurate models would result from sharing information between institutions while maintaining individual data privacy. Federated learning is a learning technique that trains a model across multiple edge devices without sharing the data. However, there is limited reach of federated learning algorithms

for histopathology domain. A research in this area is still required to fill the gap in the literature.

### 1.1.2 Motivations in Cancer Research

**Diagnosis and Prognosis.** Pathologists examine biopsy specimens to identify the presence of a tumor and to characterize multiple features in order to assess tumor aggressiveness. Improving the accuracy of diagnostic interpretations can reduce over-treatment of benign lesions and under-treatment of malignant ones. The visual assessment by pathologist is a complex task that involves years of experiences, and sub-speciality expertise. Machine learning can make these assessments more repeatable and objective through image search. Searching for evidently diagnosed cases similar to a given new case can provide insights to a pathologist into factors driving tumor progression.

**Interpretation.** Visualizing features and locating regions of tumor that most contributes to a prediction can create a teaching mechanism for pathologists. Furthermore, it helps in validating the decision making of a learning algorithm through a human expert. Although for this dissertation, the focus is on H&E-stained histology datasets, the techniques described in this work are not specific to this type of image modality.

## 1.2 Thesis Objectives and Contributions

The central objective of the thesis is to develop a learning framework for extracting compact and discriminative representations of whole-slide images in digital pathology. These representations can be used to develop specialized tools, such as image search for assisting clinicians in histo-diagnosis. The experiments are designed to quantify the quality of the representations in their abilities to search histopathology slides to fetch a slide with the correct primary diagnosis in a large archive. To some extent, this thesis contributes to the ambitious and long-term goal of biomedical community to integrate machine learning as assisting technologies for diagnosis.

The three significant contributions of the thesis are as follows:

1. Yottixel, a framework for learning compact and discriminative representations of whole-slide images (WSIs). The major novelty of Yottixel is the way it internally represents WSIs. Each WSI is converted to a set of representative patches (mosaic) that are then converted to "barcodes" for the compact and efficient retrieval and storage. The details are discusses in Chapter 3.

2. Three novel MIL methods to train Yottixel's backbone deep network to extract more discriminative features. The proposed MIL methods are based on weak supervision on the existing WSI labels (such as anatomic sites and primary diagnoses). MIL facilitates the training of deep models in the histopathology domain, since most of the time regional- or pixel- level annotations are not available, expensive, or time-consuming to obtain. The three proposed approaches are discussed in Chapter 4, Chapter 5, Chapter 6 respectively.

3. ProxyFL, a proxy-based federated learning framework that enables distributed training of Yottixel across multiple institutions while protecting patient privacy. The distributed and private training can facilitate deep learning research in healthcare, especially histopathology since it can accelerate model training, imrpove model's performance without compromising regulations and privacy. A popular approach for federated learning called FedAvg is discussed in Chapter 7. The ProxyFL is discussed in Chapter 8.

## 1.3 Thesis Organization

The thesis is organized in nine Chapters and is structured as follows:

1. Chapter 2 introduces the necessary definitions, concepts, related work including various current approaches for applications of machine learning in histopathology, multi-instance learning, and federated/distributed machine learning.

2. Chapter 3 presents the proposed framework for representing and searching a histopathology image — Yottixel.

3. Chapter 4, Chapter 5, Chapter 6 presents different weakly-supervised (multi-instance learning) methods to enhance the discriminative capabilities of Yottixel for cancer subtyping.

4. Chapter 7 and Chapter 8 present two federated learning methods for training Yottixel distributively among multiple hospitals without explicitly sharing the private (local) data.

5. A general conclusion is presented in Chapter 9.

# Chapter 2

# Background and Related Work

As noted in Chapter 1, the main contribution of this dissertation is the design and development of a representation learning framework for histopathology images, called Yottixel. The PhD research is conducted in three stages—(i) laying the foundation of Yottixel, (ii) enhancing Yottixel with weak-supervision through multi-instance learning, and (iii) adding distributed learning capabilities to Yottixel through federated learning. This chapter covers the literature review of topics associated with understanding of Yottixel framework. These topics are fundamentals of digital pathology, whole-slide images (WSIs), current state of machine learning in histopathology (applications and challenges), some common computation tools/libraries and open datasets used by researchers in the field of computational histopathology. Further, the multi-instance learning is described and its applications in histopathology domain. After that, the federated learning and differential privacy are explained with their recent applications in digital histopathology.

## 2.1 Digital Pathology

Digital pathology refers to digitization of traditional pathology routines. In this context, an equivalent of light microscope is digital scanner, and for a tissue-containing glass slide (biopsy sample) is a *whole-slide image (WSI)*. A WSI is a digitized and self-contained version of a glass slide (processed specimen in the laboratory) that can be viewed through a specialized software system. The WSIs have been long recognized as a research and education tool [10, 11] since its introduction in the early 1980s. However, only very recently (after almost 35 years), the healthcare industry has indicated an increased levels of interest in the total or partial adoption of WSIs for diagnostic purposes [12], attributed to recent

advancements in image acquisition and user-interface technologies [13, 14], ans most likely due to the effects of the COVID-19 pandemic (CITE).

### 2.1.1 WSI File and Format

The WSIs are often much larger than other typical modalities of medical images [10]. Generally speaking, resolution of the base layer of a WSI is more than $50,000 \times 50,0000$ pixels. Even with proprietary encryption and compression, an average size of a WSI file is $\approx$ 1–4 GB. Unlike a conventional digital image file, which usually contains a single static view, a WSI is composed of multiple layers of image-data arranged in a pyramid structure [11, 15], as shown below. The bottom layer has the highest resolution whereas

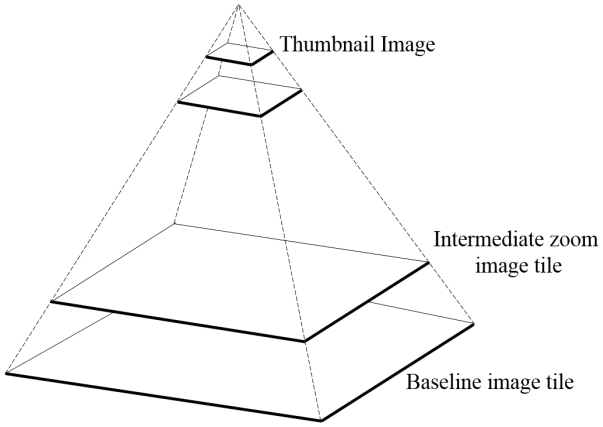



Figure 2.1: A pyramid like arrangement of image data in a WSI file.

the top layer is usually a thumbnail of ordinary sized image (e.g., several hundred pixels in each dimension). The magnification of $20\times$ is the most commonly used as the base layer [16, 17], with about 4 levels of reduction going up in the pyramid. Generally, each WSI file contains various properties that are encoded in its headers, including the information about the physical resolution covered per pixel known as MPP or (microns $\mu$ per pixel). The $20\times$ magnification is usually around 0.5 MPP, however it may be specified by the scanner's vendor. The magnification levels of $40\times$, $60\times$, and even $80\times$ are available [18] but the real-world usage of such high level of magnification is sporadic [17].

## 2.2 Machine Learning for Digital Pathology

The rapid adoption of digital pathology has resulted in accumulation of large number of WSIs. Many attempts have been made to analyze WSIs using digital image analysis based on machine learning (ML) algorithms to assist with various tasks including diagnosis [19, 20, 21, 22]. Researchers in both image analysis and pathology fields recognize and promote the importance of computer-driven analysis of pathology images [22]. Digital pathological image analysis often uses general image recognition technology (e.g., facial recognition) as a baseline [19]. However, since WSIs have unique properties, customized processing techniques need to be designed.

ML techniques for digital pathology are divided into supervised learning, and unsupervised learning. The goal of supervised learning is to infer a function that can map the input images to their appropriate labels. The input for such methods could be either an entire WSI or a regional WSI patch. The tasks related to supervised learning in digital pathology can involve identifying the type of cancer, Gleason grading of a tumor, mortality prediction, segmentation of areas of interest (e.g., tumor), and many others. On the other hand, the goal of unsupervised learning, such as, clustering, image-based search, and dimensionality reduction, is to infer hidden structures and relationships from the unlabeled data. Due to abundance of unlabeled data in histopathology, unsupervised learning may be a natural choice in many clinical and research applications.

There are various challenges concerning WSIs processing through existing ML techniques. However, ML offers many opportunities to improve objectivity and accuracy of diagnostic interpretations in histopathology. These challenges and opportunities are discussed in the following subsection.

### 2.2.1 Challenges

**Large dimensionality.** WSIs are gigapixel digital images of extremely large dimensions. Image sizes larger than 50,000 $\times$ 50,000 pixels are quite common in digital pathology. However, AI models (e.g., deep networks) trained to classify natural images of animals, objects and buildings, use much smaller sized images such as 250$\times$250 pixels as an input. Directly feeding WSIs to these AI models can easily exhaust resources of even the most powerful GPU clusters[1]. Therefore, WSIs are commonly divided into smaller image regions known as "patches" [19, 23]. Each patch is analyzed independently, and aggregated together into the final prediction. Different schemes could be used for aggregating the result,

[1]https://www.nvidia.com/en-us/geforce/graphics-cards/compare/

sophisticated schemes such as Multi Instance Learning (MIL), or much simpler paradigms such as majority voting.

Patching is a potential solution for not just AI models but also for general computer vision methods. However, even for patches, one may need to downsample them in order to be able to feed them into a deep network. A region smaller than 1.5 $\mu m^2$ may not be suitable for many diagnostic purposes [23], most of the time, at least 1000 $\times$ 1000 pixels at 20$\times$ resolution is required to inspect minute visual clues. Downsampling these patches may result in loss of crucial information. On the other hand, deep nets with larger input sizes would need higher computational resources for their training and inference.

**Multi magnification nature of histopathological images.** Tissues are usually composed of cells with distinct features [19]. Information regarding cell shape is captured at higher magnification levels, whereas structural information is composed of multiple cells, captured at lower magnification levels (see Figure 2.2) [24]. A pathologist diagnoses a disease by investigating a tissue sample at various magnification levels to see both cellular and structural (e.g., glandular) patterns. Strictly from a computer vision perspective, the manifestations of visual patterns at different magnification levels are very distinct. The image analysis for histopatholoigcal images can be improved by incorporating information at multiple resolutions. This, however, would add to the complexity of building image analysis tools. Recent studies propose reinforcement-learning agents to automatically identify the best magnification level for the given ML task [25]. The advantage of incorporating different magnification levels is a new research topic, and is dependent on types of diseases and tissues, and machine learning algorithms [24].

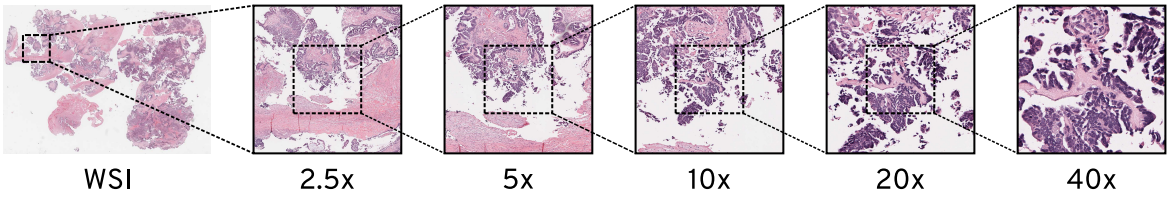



Figure 2.2: Illustration of patches from different magnification levels exhibit different visual patterns. All patches are centered around the same coordinates (images re-scaled for convenient visualization).

**Lack of labeled data.** Deep learning is highly successful when it is avail of large quantity of labeled training images, but is often impeded when the data set is small. Unlike natural images, crowd-sourced labeling of pathology images is usually not an option, as it requires experts (i.e., pathologists) to manually delineate the region of interest (i.e., anomalies or malignancies). Beside the time constraint, manual annotations often pose a financial

bottleneck to a research organization. Even for the labeled datasets, majority of the time labels are usually available at WSI or case-level, whereas the requirement of labels for training a deep network is generally at patch-level. There two areas of machine learning that can be applied to alleviate these problem. Firstly, few-shot learning can be used to organize the training of a deep network with fewer samples. A few shot-learning framework often employs some domain-related prior knowledge on a learning agent, thereby enabling it to learn quicker with few samples [26]. For digital pathology, such prior knowledge could include rotation invariance (rotating a WSI should result in the same label), and staining invariance [27, 28]. Secondly, multi-instance learning can help in managing the problem of the labels associated with WSI or case-level instead of the patch level. The multi-instance learning enables the training with a bag of instances, instead of a single instance [29].

**Lack of diversified dataset.** The actual number of patterns derived from different cancers and malignancies from a visual perspective is nearly infinite [23]. A single (sub)type of cancer can manisfest itself in various specific patterns. This extreme polymorphism makes recognizing malignancy by image algorithms exceptionally challenging [23, 30]. Apart from the variability from the tissue morphology perspective, even slide preparation site, among other things, may cause significant variations. The sources of variation in WSIs include different manufacturers of staining reagents, thickness of tissue sections, and scanner calibrations [19]. Learning without considerations of these variations can seriously affect the performance of ML algorithms. Because of this variability, medical data must be integrated across numerous organizations to increase the generalization of deep models. On the other hand, medical data centralization involves regulatory constraints as well as workflow and technical challenges, such as managing and distributing the data. Because each histopathology image is often a gigapixel file, the latter is very important in digital pathology. There are a few diverse and large scale datasets, a fact that limits research and model development progress. Federated learning is a distributed learning approach that allows multi-institutional collaborations on decentralized data while protecting the data privacy rules of each collaborator. Federated learning can enable histopathology labs to aggregate and communicate their data resulting in robust and accurate models while maintaining patient privacy.

### 2.2.2 Opportunities

**Computer-assisted Diagnosis (CAD).** CAD is an actively researched area in image analysis for digital pathology. It involves assisting the physicians by automating some of the basic tasks performed by a pathologist. CAD is a form of supervised learning, i.e., mapping WSIs to some label, such as recognizing a cancer subtype, classifying a tissue

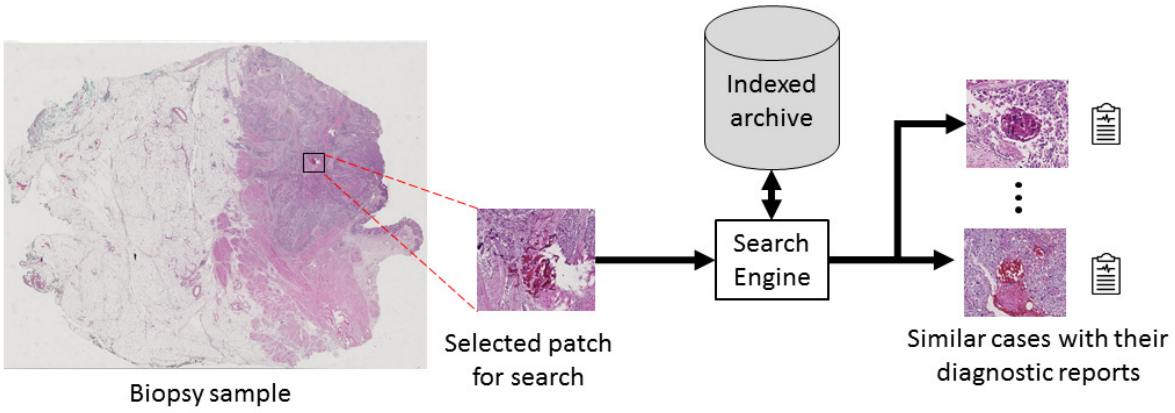



Figure 2.3: The general overview of CBIR systems for digital pathology. The selected region for search could be the entire WSI as well.

sample as either begin or malignant, or estimating tumor grade. CAD may also lead to the reduce variability in interpretations and prevent overlooking by investigating all pixels within WSIs [19]. It may facilitate some of the "boringly" repeated and mundane tasks performed by a pathologist thereby reducing human errors. Various diagnosis-related tasks reported in literature include detection or segmentation of region of interest (ROI) such as tumor region in a WSI [31], cancer staging [32], tissue segmentation [33], and nuclei density estimation [34].

**Content-based image retrieval (CBIR) for WSIs.** In CBIR systems a search tool takes an image as an input and returns similar images (shown in Figure 2.3) by macthing it against other images in an indexed archive. Whilst CBIR systems of medical images have been well researched [35, 36], only with the emergence of digital pathology and deep learning has research begun to focus on image search and analysis in histopathology [37]. However, there are two major drawbacks of CBIR systems that limit their integration into digital pathology. Firstly, most conventional CBIR proposals use basic image features that capture low-level characteristics of an image such as color, edges, textures, or shapes. This approach generally fails to capture high-level patterns corresponding to the semantic content of histopathology images. Secondly, WSIs are gigapixel digital images. However, most proposed CBIR technologies are designed for much smaller image dimensions (i.e., smaller than $300 \times 300$ pixels). In addition to the large dimensions, pathology images exhibit an intractable level of variability in visual features that makes their identification, compared with that of natural images, even more challenging.

The majority of recent studies in computational pathology have reported the success of supervised AI algorithms for classification and segmentation [38, 31]. This overrepresentation compared to other AI algorithms is related to the ease of design and in-lab validation to generate highly accurate results. However, compared to other methods of computer-vision algorithms, CBIR offers a new approach to computational pathology. To facilitate image search, CBIR algorithms essentially describe the content of an image with non-textual attributes, generally with a vector of real numbers known as a *feature vector*. An AI agent could be trained to transform an image into a feature vector as its representation. This learning process is known as *representation learning*. If a feature vector encompasses the descriptive visual properties of an image, then searching for similar images becomes a nearest-neighbour matching problem. Images with similar content could be retrieved based on a comparison of their feature vectors and not based on direct pixel comparison, or indirectly through associated textual metadata. This is generally possible if a feature vector encodes the semantic structures of an image invariant to scale, rotation, translation, and even to some degree, to deformation [39]. Such rich and descriptive features can numerically *represent* images for the purpose of identification, which is the core task of any CBIR system.

In literature, there are two main points of view for processing whole-slide images [40]. First one is called *sub-setting methods* which considers a small section of large pathology image as essential part such that processing of small subset substantially reduces processing time. Secondly, a *tiling* approach that breaks images into smaller and controllable patches and tries to process them against each other [41] which naturally requires more care in design and is more expensive in execution. However, tiling approach is a distinct approach toward full automation. The majority of research works in literature prefers the *sub-setting* method because of its advantage of speed and accuracy. However, it needs expert knowledge and intervention to extract proper subsets. Mehta et al. [42] proposed an offline CBIR system which utilizes sub-images rather than entire digital slide. Using scale-invariant feature transform (SIFT) [43] to search for similar structures by indexing each sub-image, experimental results suggested, when compared to manual search, 80% accuracy for the top-5 results retrieved from the database that holds 50 IHC (immunohistochemistry) stained pathology images, consisting of 8 resolution levels. Akakin and Gurcan [44] developed a multi-tiered CBIR system based on WSI, which is capable of classifying and retrieving digital slides using both multi-image query and images at slide-level. Authors test proposed system on $1,666$ whole-slide images extracted from 57 follicular lymphoma (FL) tissue slides containing three subtypes and 44 neuroblastoma (NB) tissue slides comprised of 4 sub-types. Experimental results suggested 93% and 86% average classification accuracy for FL and NB diseases, respectively. More recently, Zhang et al. [45]

developed an scalable CBIR method to cope with WSIs by using supervised kernel hashing technique which compresses a 10,000-dimensional feature vector into only ten binary bits, which is observed to preserve the concise representation of the image. These short binary codes are then used to index all existing images for quick retrieval for of new query images. The proposed framework is validated on breast histopathology data set comprised of 3,121 WSIs from 116 patients; experiments report accuracy levels of 88.1% for processing at a speed of 10ms for all 800 testing images.

### 2.2.3 Common Tools and Datasets

**Common Tools.** The OpenSlide library offers a vendor-agnostic API for reading WSI files [46]. QuPath is a powerful and extensible tool for viewing and analyzing WSIs [47]. The Openseadragon[1] library provides an API to create custom GUI components for interacting and displaying WSIs. Apart from that, standard data science, deep learning, and imaging libraries stack for Python or any other computer language can be utilized for building image analysis algorithms for WSIs.

**Benchmark Datasets.** Some public datasets used as benchmarks for testing ML techniques are reported in Table 2.1. These datatsets are especially useful for testing newer ML techniques where researchers may not be necessarily interested in developing the entire processing pipeline for WSI analysis.

| Dataset Name | Image Size | # Images | Staining | ML Application |
|---|---|---|---|---|
| KIMIA960 [48] | 308×168 | 960 | H&E, IHC | Classification |
| BreakHis [49] | 700×460 | 7909 | H&E | Classification |
| PCAM [27] | 96×96 | 327,680 | H&E | Classification |
| BreastPathQ [50] | - | 2579 | H&E | Regression, Segmentation, MIL |

Table 2.1: Various different datasets used as benchmark for testing ML techniques.

**Public WSI Archives.** Some public archives of WSIs along with their case- and WSI-level metadata are reported in Table 2.2. The WSIs in the real clinical settings are stored in the same way as these public archives, enabling ML researchers to validate their algorithms in an environment similar to the real-world. However, these archives are difficult to work with for validating new ML ideas. Processing an entire WSI requires developing the complete pipeline of preprocessing, patching, and more.

[1] https://openseadragon.github.io/

| Dataset Name | # WSI | Staining | Disease | ML Applications |
|---|---|---|---|---|
| TCGA [51] | $\approx$ 30,000 | H&E | Normal/Cancer | Classification |
| GTEx [52] | 25,380 | H&E | Normal | - |
| TUPAC16 [53] | 821 | H&E | Breast Cancer | Classification |
| Camelyon17 [54] | 1000 | H&E | Breast cancer | Segmentation |
| KIMIA Path24 [55] | 24 | H&E/IHC | Various | Classification, Search |

Table 2.2: Public archives of histopathology WSIs.

## 2.3 Multi-Instance Learning

In a typical supervised machine learning problem, every input instance is assigned to a label. However, in many real-life applications a label might be associated with a set of instances (called a bag). In such scenarios, a learning agent must observe all instances in a set at once to learn its associated label. This type of learning is known as multiple instance learning (MIL) or, learning from weakly annotated data. Whereas MIL has many applications in medical imaging [56, 57], there is a growing interest for the usage of MIL for histopathological image analysis [7, 58, 59]. This growing interest is mainly due to the limitations of the regional or pixel-level WSI annotations. The majority of the available WSI datasets have labels associated with either each patient (having multiple WSIs), or with each WSI itself. For computational tractability, a WSI is usually dismantled into multiple patches before processed by any ML technique. All these patches together are assigned the same label, thus making MIL as natural candidate for image analysis of WSIs.

**Problem formulation.** In case of a MIL problem, instead of a single instance, there is a bag containing multiple instances, $X = \{x_1, x_2, \ldots, x_K\}$, e.g., a set of patches in a WSI. The instances in a bag are assumed to be "orderless", i.e., arranging instances in a different order should not affect the outcome of a learning technique. Furthermore, the number of instances in each bag $K$ could be different. Given this information, there are many different tasks to be solved through MIL. For instance in binary classification, a single binary label $Y$ is associated with a bag $X$ (WSI is "benign" or "malignant"). In this case, even if a single instance in a bag is "1" then the whole bag is labeled as "1" (i.e., even if a single patch is malignant then the entire WSI is malignant, otherwise benign). Now, assume that there are labels associated with each instance $y_k \in [0, 1]$ for $k = 1, \ldots, K$, even though the instance-level labels remain unknown during the training process. We can re-write the

MIL problem as follows,

$$Y = \begin{cases} 0, & \text{iff } \sum_k y_k = 1 \\ 1, & \text{otherwise.} \end{cases}$$

In order to train a MIL model for the binary classification, the idea is to use a weak classifier $h_\theta$ (weak because it does not make the final prediction) to classify each instance $y_k^{'} = h_\theta(x_k)$, then the final prediction is computed by taking the maximum over all predictions, i.e.,

$$Y^{'} = \max_k \{y_k^{'}\} = \max_k \{h_\theta(x_k)\}, \tag{2.1}$$

to minimize the discrepancy between $Y^{'}$ and $Y$ by perturbing the parameters $\theta$ of the weak classifier $h_\theta$. The variable $h_\theta$ can be a neural network, or any trainable technique. To optimize the loss (2.1), a pressing caveat for gradient-based method is the *vanishing gradients* in very deep networks due to the maximum operator. However, it can be solved by using a "soft" approximation of maximum, such as softmax.

Now, a question arises: How can the MIL problem stated above be formulated for multi-class labels, or for the representation learning of WSIs? Simply based on the intuition, similar to the binary classification, the weak classifier $h_\theta$ can be replaced with a feature extractor (e.g., a deep network), and *max* could be replaced with some permutation-invariant pooling operation. In fact, Zaheer et al. mathematically verified this intuition [60] by stating the following theorem:

**Theorem 1.** *A function $f(x)$ operating on a set $X = \{x_1,\ldots,x_K\}$ having elements from a countable universe, is a valid set function, i.e., invariant to the permutation of instances in $X$, if it can be decomposed to*

$$f(x) = \rho\left(\sum_{x \in X} \phi(x)\right), \tag{2.2}$$

*for any suitable transformations $\phi$ and $\rho$.*

This theorem provides a general strategy to approximate any arbitrary set function $f(x)$ by decomposing it to (2.2).

**Exchangeability.** The exchangeability is an important topic from statistics that is directly applicable in MIL. A sequence of random variables $X = x_1, \ldots, x_K$, i.e. a bag in MIL, is *exchangeable* if the joint probability of the distribution does not change on permutation of indices $1, 2, \ldots, K$. Mathematically, if

$$P(x_1, \ldots, x_n) = P(x_{\pi(1)}, \ldots, x_{\pi(K)})$$

for a permutation function $\pi$, then the sequence $X = x_1, \ldots, x_K$ is exchangeable. Similarly, a machine learning model (e.g., a deep network) is said to be *exchangeable model* if the output of the model is invariant to the permutation of its inputs. Exchangeable models can be of two types depending on the application, (i) permutation invariant, and (ii) permutation equivariant.

A model represented by a function $f : X \to Y$ where $X$ is a set, is said to be *permutation equivariant* if permutation of input instances permutes the output labels with the same permutation $\pi$. Mathematically, a permutation-equivariant model is represented as

$$f(x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(n)}) = [f_{\pi(1)}, f_{\pi(2)}, \ldots, f_{\pi(n)}].$$

Similarly, a function is permutation-invariant if permutation of input instances does not change the output of the model. Mathematically,

$$f(x_1, x_2, \ldots, x_n) = f(x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(n)})$$

**Exchangeability and MIL.** With all the discussion so far, it can be inferred that an MIL problem is a special case of exchangeability that involves training a permutation-invariant exchangeable model. The Equation 2.2 from Theorem 1 can be used to develop a general strategy for training such models, as follows:

1. Apply the transformation $\phi$ on all instances of a bag.
2. Aggregate the transformed instances using a permutation-invariant pooling operation (e.g., sum), and
3. Apply the final transformation $\rho$ on the combined instances transformed by $\phi$. These two transformations, $\phi$ and $\rho$, can be deep neural networks, thereby enabling the end-to-end training of an MIL task.

**MIL for histopathological image analysis.** MIL is particularly useful for digital pathology. The ground-truth labeling is expensive whereas labels are generally available at WSI or case level as opposed to regional or pixel level (required by conventional techniques). As a standard diagnostic protocol, a pathologist analyzes a given case and renders a pathology report. A case often contains several tissue specimens (WSIs) from a single patient. Therefore data produced from regular clinical practices, without any extra effort is well suited in MIL framework. On average, a small-sized pathology laboratory processes $\approx$ 60,000 cases per year [61], producing a vast amount of data, presenting an opportunity for

MIL methods to exploit the fine-grained information with minimum efforts from human-supervised annotations. Dismantling a WSI into smaller patches is a common practice for processing it. These patches can be grouped as a bag for an MIL approach. Isle et al. [58] used attention-based pooling with MIL to infer patches of higher importance, for a disease classification task. It is interesting to note, the weak supervision of class labels at the bag-level allowed their model to comprehend the "importance" of patches (instance level). This opens many possibilities, a large amount of partially labeled training data (already available), a MIL method can deduce instance-level attributes from a bag-level supervision. This has potentials to discover hidden patterns of clinical importance [19]. Sudarshan et al. used MIL for histopathological breast cancer image classification [7]. A permutation-invariant operator for MIL was introduced by Tomczak et al. and successfully applied to digital pathology images [8].

## 2.4 Distributed and Private Machine Learning

### 2.4.1 Federated Learning (FL)

In contrast to conventional learning algorithms, federated learning (FL) algorithms learn from decentralized data distributed across various client devices. In most examples of FL, there is a *centralized server* which facilitates training a shared model and addresses critical issues such as data privacy, security, access rights, and heterogeneity [62]. In FL, every client locally trains a copy of the centralized model, represented by the model weights $\omega$, and reports its updates back to the server for aggregation across clients, without disclosing local private data. Mathematically, FL can be formulated as

$$\min_{\omega \in \mathbb{R}^d} f(\omega) \qquad \text{with} \qquad f(\omega) = \frac{1}{n} \sum_{i=1}^{n} f_i(\omega), \tag{2.3}$$

where $f(\omega)$ represents the total loss function over $n$ clients, and $f_i(\omega)$ is the loss function with respect to client $i$'s local data. The objective is to find weights $\omega$ that minimize the overall loss. McMahan et al. [62] introduced federated averaging, or *FedAvg* (Algorithm **??**), in which each client receives the current model $\omega_t$ from the server, and computes $\nabla f_i(\omega_t)$, the average gradient of the loss function over its local data. The gradients are used to update each client's model weights using stochastic gradient descent (SGD) as $\omega_{t+1}^i \leftarrow \omega_t - \eta \nabla f_i(\omega_t)$ using the learning rate $\eta$. Next, the central server receives the updated weights $\omega_{t+1}^i$ from all participating clients and averages them to update the central model,

$\omega_{t+1} \leftarrow \sum_{i=1}^{n} \frac{n_i}{n} \omega_{t+1}^{i}$, where $n_i$ is the number of data points used by client $i$. To reduce the communication costs, several local steps of SGD can be taken before communication and aggregation, however, this affects the convergence properties of FedAvg [63].

Other methods for FL have also been proposed. Yurochkin et al. [64] proposed a Bayesian framework for FL. Claici et al. [65] used KL divergence to fuse different models. Much work has also been done to improve the robustness of FL algorithms. Pillutla et al. [66] proposed a robust and secure aggregation oracle based on the geometric median using a constant number of calls to a regular non-robust secure average oracle. Andrychowicz et al. [67] proposed a meta-learning approach to coordinate the learning process in client/server distributed systems by using a recurrent neural network in the central server to learn how to optimally aggregate the gradients from the client models. Li et al. [68] proposed a new framework for robust FL where the central server learns to detect and remove malicious updates using a spectral anomaly detection model, leading to targeted defense. Most of the algorithms cannot be directly compared or benchmarked as they address different problems in FL such as heterogeneity, privacy, or adversarial robustness. FedAvg is most commonly used because of it's scalability to large datasets and comparable performance to other FL algorithms.

**Federated learning (FL) in histopathology.** FL is especially important for histopathology as it facilitates collaboration among institutions without sharing private patient data. Histopathology images are too large for centralized machine learning algorithms, distributed machine learning are more effective as they can share the processing cost among multiple nodes. One prominent challenge when applying FL to medical images, and specifically histopathology, is the problem of *domain adaptation*. Since hospitals have diverse imaging methods and devices, images from a group of hospitals will be markedly different, and machine learning methods risk overfitting to non-semantic differences between them. Models trained using FL can suffer from serious performance drops when applied to images from previously unseen hospitals [69]. Several recent works have explored applications of FL in histopathology, and grapple with this problem. Lu et al. [70] demonstrated the feasibility and effectiveness of FL for a large-scale computational pathology studies. FedDG proposed by Liu et al. [71] is a privacy-preserving solution to learn a generalizable FL model through an effective continuous frequency space interpolation mechanism across clients. Sharing frequency domain information enables the separation of semantic information from noise in the original images. Li et al. [72] address the problem of domain adaptation with a physics-driven generative approach to disentangle the information about model and geometry from the imaging sensor.

### 2.4.2 Differential Privacy

While FL attempts to protect privacy by keeping private data on client devices, it does not provide a quantitative privacy guarantee. Updated model parameters are still sent from the clients to a centralized server, and these can contain private information [73], such that even individual data points can be reconstructed [74]. *Differential privacy* is a formal framework for quantifying the privacy that a protocol provides [75]. The core idea of DP is that privacy should be viewed as a resource, something that is used up as information is extracted from a dataset. The goal of private data analysis is to extract as much useful information as possible while consuming the least private content. To formalize this concept, consider a *database* $\mathcal{D}$, which is simply a set of datapoints, and a probabilistic function $M$ acting on databases, called a *mechanism*. The mechanism is said to be $(\epsilon, \delta)$-*differentially private* if for all subsets of possible outputs $\mathcal{S} \subseteq \text{Range}(M)$, and for all pairs of databases $\mathcal{D}$ and $\mathcal{D}'$ that differ by one element,

$$\Pr[M(\mathcal{D}) \in \mathcal{S}] \leq \exp(\epsilon) \Pr[M(\mathcal{D}') \in \mathcal{S}] + \delta. \tag{2.4}$$

When both $\epsilon$ and $\delta$ are small positive numbers, Equation 2.4 implies that the outcomes of $M$ will be almost unchanged in distribution if one data point is changed in the database. In other words, adding one patient's data to a differentially private study will, with high probability, not affect the outcomes.

The advantage of DP is that it is quantitative. It yields a numerical guarantee on the amount of privacy that can be expected, in the stochastic sense, where lower $\epsilon$ and $\delta$ implies that the mechanism preserves more privacy. The framework also satisfies several useful properties. When multiple DP-mechanisms are composed, the total operation is also a DP-mechanism with well defined $\epsilon$ and $\delta$[76]. Also, once the results of a DP-mechanism are known, no amount of post-processing can change the $(\epsilon, \delta)$ guarantee [77]. Hence, while FL alone does not guarantee privacy, we can apply FL in conjunction with DP to give rigorous bounds on the level of privacy afforded to clients and patients who participate in the collaboration.

The simplest way to create a DP-mechanism is by adding Gaussian noise to the outcomes of a deterministic function with bounded sensitivity [78]. This method can be used in the context of training a machine learning model by clipping the norm of gradients to bound them, then adding noise, a process called *differentially private stochastic gradient descent* (DP-SGD) [79]. McMahan et al. [80] applied this at scale to FL.

**Differential privacy for medical image analysis.** Past works have noted the potential solution DP provides for machine learning in the healthcare domain. Kaissis et al. [81]

surveyed privacy-preservation techniques to be used in conjunction with machine learning, which were then implemented for classifying chest X-rays and segmenting CT scans [82, 83]. In histopathology, Lu et al. [70] reported DP guarantees for a neural network classifier trained with FL, following Li et al. [84]. Their treatment involved adding Gaussian noise to trained model weights. However, neural networks weights do not have bounded sensitivity making their differential privacy guarantee vacuous. A meaningful guarantee would require clipping the model weights before adding noise, thereby restricting the sensitivity. The more standard approach of DP-SGD, which clips gradient updates and adds noise, for use in federated learning.

## 2.5 Summary

This chapter introduced the necessary definitions and concepts that readers should familiarized themselves for understanding the content presented in later chapters. The next chapter goes into details of Yottixel framework, that lays the foundation for representing histopathology images through a mosaic (a set of patches). The Yottixel is validated on the largest public archive of whole-slide images.

# Part I

# Yottixel - A Framework for Representing Histopathology Images

Yottixel is the proposed framework for creating compact and discriminative representations of gigapixel whole-slide images. This is the first stage of the PhD research that lays the foundation of a framework for processing the gigapixel histopathology images through deep learning. Yottixel is validated on a content-based image retrieval task using one of the largest public archive of whole-slide images—The Cancer Genomic Atlas (TCGA). The results are encouraging and establish the baselines for methods developed later during the PhD research.

# Chapter 3

# Yottixel

## 3.1 Prologue

The content of this chapter is based on two articles published during the Ph.D. research:

A. **S. Kalra**, et al. *Yottixel–an image search engine for large archives of histopathology whole slide images.* Medical Image Analysis 65 (2020)

B. **S. Kalra**, et al. *Pan-cancer diagnostic consensus through searching archival histopathology images using artificial intelligence.* NPJ digital medicine 3.1 (2020).

This chapter discusses the details of a framework, called Yottixel, to extract compact and discriminative representations of histopathology images. The validation study (Paper B) of Yottixel has been accepted in *Nature's Partnered Journal,* npj Digital Medicine and has received considerable attention (cited 44 times as of March 16, 2022). The experiments, and data collected for the Yottixel project have laid down a solid foundation to validate other methods discussed in this dissertation.

## 3.2 Introduction

*Yottixel* is a portmanteau for *one yotta pixel* alluding to the big-data nature of pathology images. It is a framework to extract compact and discriminative representations of gigapixel histopathology images. The underlying technology behind Yottixel consists of a

series of machine learning algorithms including clustering techniques, deep networks, and gradient barcoding. The Yottixel represents a whole-slide image (WSI) by extracting the set of representative patches called *mosaic*. The patches of mosaic are then converted into a set of barcodes, a process that is both storage-friendly and computationally efficient. This set of barcode is called "bunch of barcode"(BoB) that forms the representation of the WSI. For validating Yottixel, WSIs from The Cancer Genome Atlas (TCGA) [51] repository provided by the National Cancer Institute (NCI)/National Institutes of Health (NIH) were used. Almost 30,000 WSI files of 25 primary anatomic sites and 32 cancer subtypes were processed by dismantling these large slides into almost 20,000,000 image patches (also called tiles) that were then individually indexed employing approximately 3,000,000 barcodes. The validation results proves the efficacy of Yottixel as a competitive image search engine for digital pathology, achieving >90% accuracy for predicting the cancer sub-type among some of the anatomical sites.

**Design motivation.** The Yottixel is designed while keeping few major observations in mind. (i) Not many works have developed a representation learning and retrieval solutions for entire high-resolution WSIs; the focus is generally on patch processing (for instance, [85, 86]). (ii) Much research has been dedicated to process *labeled* repositories where malignant regions in WSI files have been delineated by trained pathologists (for instance, [87, 40]). (iii) Many approaches index images with real-valued features, a requirement that would be hard to meet in reality because of the storage and computational requirements [88, 89]. (iv) Some works use hashing for fast search to increase the feasibility of retrieval, but hash codes may not easily facilitate data exchange among repositories (for instance, [90, 91]). (v) Histopathology images contain several diversely shaped edges, intricate and irregular structures, and high gradient changes that create an inconceivable complexity for most computer vision algorithms [19, 23, 20].

## 3.3 Method

This section describes the design and implementation of Yottixel (Figure 3.1). This work has two main practical contributions. First, we propose a method for representing an entire WSI with a small set of patches, referred to as *mosaic*. The concept of mosaicking is fundamental for the feasibility of processing such large images. Secondly, we construct and test an end-to-end ensemble framework that indexes and retrieves WSIs based on their content.

The distinguishing aspect of Yottixel is the utilization of barcodes for image represen-

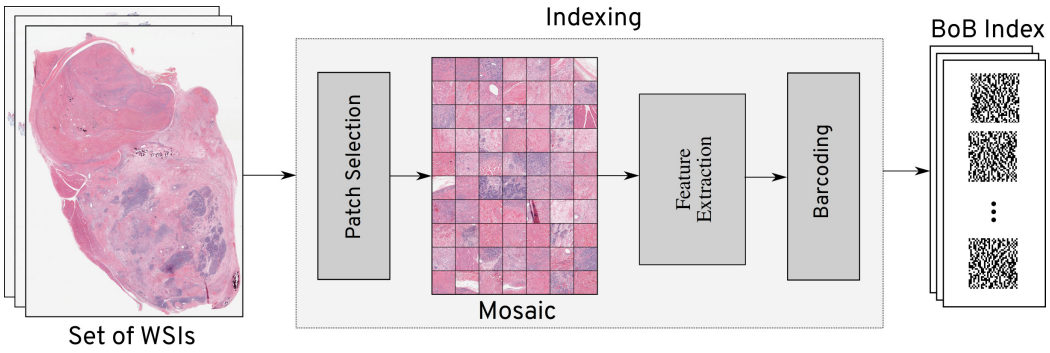



Figure 3.1: Overview of Yottixel's indexing framework to generate the BoB index. Patch selection generates the mosaic. Individual barcodes may be used for patch search. All barcodes of any given scan can be used for searching WSI.

tation and characterization. A WSI is indexed by converting its associated mosaic to a set of barcodes. This set of barcodes constitutes an index for the given WSI, referred to as "**bunch of barcodes**" (BoB) index. The BoB index accelerates the retrieval process and alleviates the computation and storage burden on the deployment infrastructure for laboratories and clinics. Yottixel is a complete and functioning search engine for indexing WSIs for CBIR systems with major emphasis on performance and scaling for laboratory and hospital requirements.

Yottixel has two major phases of operation: (i) offline indexing and (ii) run-time search. During the initial deployment of Yottixel, offline indexing consumes the maximum computation resources to index the available WSI files. Once a sufficient number of images are indexed, the two phases are activated simultaneously. However, offline indexing is set to run preemptively allowing runtime search to acquire higher precedence over the available resources.

### 3.3.1 Offline Indexing Phase

The crux of a search engine platform for a large archive of medical images of high dimensionality is its *indexing*. The structure of the index determines the speed, reliability, and robustness of search results. Yottixel indexes a WSI by (i) computing its mosaic (a representative set of patches) and then (ii) converting the mosaic to a BoB index. The design choices of the indexing algorithm are influenced by real-world scenarios in a mid-size pathology laboratory or clinic, where hundreds of thousands of WSI files are generated every year. However, the computing and storage infrastructure are generally not sufficient

for hosting a sophisticated image search engine on-site. The first-time indexing of large existing archives cannot be implemented in most hospitals and laboratories because of the requirement of high storage and computational resources amid a sluggish transition to digital pathology that requires an expensive IT infrastructure. This will be particularly apposite if the indexing technology is not designed with efficiency in mind.

**STEP 1: Computing the mosaic.** Yottixel receives a set of WSIs queued for indexing. For each queued WSI, a representative set of patches, or *mosaic* is computed. Employing a mosaic considerably reduces the computation burden. Instead of operating over an entire WSI, all the subsequent image processing operations are applied on the mosaic of WSI. The algorithm for creating a mosaic is outlined in Algorithm 2 (in Appendix A), Lines 8–26. Firstly, a WSI is segmented into $K_{CH}$ different regions based on their colour (staining) composition by using the k-means algorithm. The variable $K_{CH}$ is a parameter set based on visual inspection of different WSIs. We found that a typical WSI exhibits not more than nine visually distinct regions. Segmenting these regions captures the pattern variability from a computer-vision perspective but may not have relevance strictly from a histopathology point-of-view (we do not look at epithelium versus connective tissue, for instance). The colour-based segmentation is outlined in Algorithm 2(in Appendix A) from Lines 10–20. However, colour-based segmentation frequently resulted in the separation of different tissue types within a WSI, such as blood stains from muscles, fat, and in some cases, even cancerous regions. In a second stage of segmentation, a small percentage of patches are randomly selected by preserving the spatial diversity from the colour-segmented regions. Again, we used the k-means algorithm for grouping the patches based on their location (Appendix A, Algorithm 2, Lines 21–26). Currently, this small percentage is fixed to 5% as suggested by empirical evidence. However, ideally it should vary depending on the complexity and variations within a given slide. The primary reason for doing k-means clustering second time is to sample the different patches from the same group by taking patches from diverse locations. One may use the random sampling, however it may result in inconsistent results. Since k-means algorithm converges to a solution quite fast, it is more appropriate than random sampling when applied to both color and location. The patches are collected from all segmented regions constituting the mosaic of the given WSI (**??**). Patch clustering may be performed at a lower resolution (e.g., 5× magnification) because a higher resolution does not offer any superiority for many tasks performed on histopathlogy whole slide images ([92, 93, 94]). With these settings, a typical mosaic obtained is ≈20 times smaller than the specimen area depicted in the WSI.

**STEP 2: Creating the BoB index.** The patches in a mosaic are converted to a set of barcodes. These barcodes constitute the index for a single WSI file. The algorithm for creating the BoB index from a given WSI is provided in 2, lines 27–35. First, a patch is con-

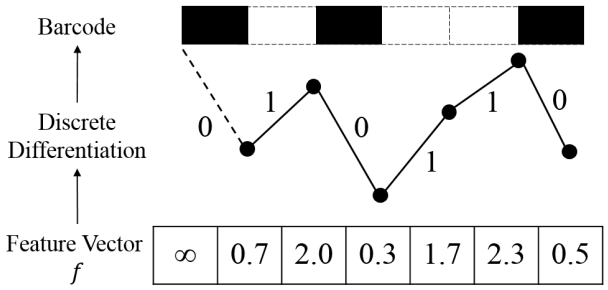



Figure 3.2: Visual depiction of the *MinMax algorithm* used to convert a feature vector into a barcode for single patch in a mosaic.

verted to a feature vector using the last convolution layer of the DenseNet [95]. We applied the Global Average Pooling (GAP) over the feature maps from the last convolution layer to extract a feature vector of size 1024. The network used for the feature extraction was pre-trained on natural images from ImageNet dataset [**?**]. Although pre-trained networks have learned from natural images, they still offer robust image characterization properties for histopathology images [96]. We decided to use DenseNet after visually inspecting the search results using features from the last average pooling layer of the VGG19 [**?**], Inception [97], and in-house trained and fine-tuned solutions [55, 96]. DenseNet allows capturing more compound/complex patterns and structures within histopathology patches. This is useful for many problems. One of the examples is searching for glomeruli in Kidney scans. The glomeruli form complex structures that cannot be interpreted from simple features like cellularity/fat. After feature extraction, we used the discrete differentiation (see Figure 3.2) to convert the feature vector to a binary representation called "barcode" which is light-weight and enables a fast Hamming distance search. For an average WSI file of size ≈700 MB, the BoB index can be as small as ≈10 KB, i.e., 70,000 smaller than the original file.

**STEP 3: Binarization using MinMax algorithm.** Although deep features can be used directly to measure the similarity between images via distance metrics such as $\ell_2$, computational efficiency is a serious issue, especially for searches in large databases across all primary sites (i.e., exhaustively searching k-nearest neighbors). Therefore, we employed a binarization method to convert these features into binary codes. Binary features allow for fast real-time search. During a run-time query, high-dimensional features are extracted from the query image and converted to barcodes. We used accelerated CPU commands to calculate the Hamming distance for the nearest neighbors queries. It has been stated that the MinMax algorithm for binarization is particularly useful for the retrieval and indexing of histopathology scans in terms of both speed and storage [98].

**The algorithm summary.** The algorithm is summarized in Appendix A, §A.1.

### 3.3.2 Runtime Search Phase

Once a sufficiently large index is created, Yottixel provides users with an interactive interface to perform search queries on their WSIs. There are two modes of searching—vertical and horizontal. In the *vertical search* mode, image matching in the archive is confined to the same anomotical site as the query patch/WSI for all patients, whereas in the *horizontal search*, the entire index is searched across all anatomies for all patients.

In summary, Yottixel assigns "a bunch of barcodes" to each WSI to index an entire digital slide. The BoB indexing enables Yottixel to search a large archive of histopathology images very efficiently. The index can be easily shared among institutions if necessary. The similarity in two BoBs are calculated by computing the median of minimums of Hamming distances of all pairwise barcodes between the query BoB and the another (see Algorithm 3 in Appendix A).

## 3.4 Dataset

The publicly available dataset of 30,072 WSIs from the TCGA project [51] (Genomic Data Commons GDC) was used to conduct the validation study of Yottixel. The total size of data is ≈16 TB in the compressed form; thereby requiring a massive computational and storage capacity to operate any algorithm on it.

The WSIs are tagged with a primary diagnosis. The 952 WSIs were removed due to the following reasons—poor staining, low resolution, lack of all magnification levels in the WSI pyramid, large presence of out-of-focus regions, and/or presence of unreadable regions within an image. Most WSIs had a magnification of 20× or 40×, some at lower magnifications. In total, the 29,120 number of WSIs were processed at 20× magnification for this study. The dataset contains 25 anatomic sites with 32 cancer subtypes. Ten tumor types (brain, endocrine, gastrointestinal tract, gynecological, hematopoietic, liver/pancreaticobiliary, melanocytic, prostate/testis, pulmonary, urinary tract) had more than one primary diagnoses. From the 29,120 WSIs, 26,564 specimens were neoplasms, and 2,556 were non-neoplastic (containing only normal tissue). A total of 17,425 files comprised of frozen section digital slides, and 11,579 files were of permanent hematoxylin and eosin (H&E) sections. For the remaining 116 WSIs, the tissue section preparation was unspecified. We did not remove manual pen markings from the slides when present. The TCGA

codes for all 32 cancer subtypes are provided in Table A.2 (Appendix A). The TCGA dataset has a number of shortcomings [99]. Many of the cases are of frozen section in which tissue morphology may be compromised by frozen artifacts. Available cases may also reflect research bias in institutional biorepository collections. Furthermore, WSIs are distribution across the primary diagnosis is imbalanced (common for real-world datasets). In spite of the shortcomings, the TCGA is the largest public dataset that can support a pan-cancer validation of AI solutions for digital pathology.

## 3.5 Results and Experiments

Three major series of experiments were conducted to validate the Yoittixel search engine. These experiment series have been designed after consultation with pathologists and reviewing the existing literature.

**Parameters.** A set of parameters for indexing was set empirically. The number of color clusters $k_{CH}$ was set to 9. The percentage of patches $p_M$ to build the mosaic was set to 5%. Clustering was performed in $m_x^c = 5\times$ whereas indexing was performed in $m_x^{idx} = 20\times$. The patch size at low magnification was $s_l = 250$ pixels (equivalent to $2mm$) and $s_h = 1000$ pixels (equivalent to $500\mu m$).

**Experiment series.** For the first two series of experiments, the "accuracy" of image search was calculated through "leave-one-patient-out" samplings. Whereas the literature of computer vision focuses on top-$n$ accuracy (if any one of the $n$ search results is correct, then the search is considered be to be successful), however for this experiment majority-$n$ accuracy was calculated (only if the majority among $n$ search results were correct, the search was considered correct). Specifically, "correct" means that the tumor type (horizontal search) or tumor subtype within a specific diagnostic category (vertical search) was recognized correctly and matched by the majority of identified and retrieved cases. In order to avoid falsification of results through anatomic duplicates, we excluded all WSIs of the patient when one of the WSIs was the query. In the last series of experiments, a study was developed to validate the quality of Yottixel search results directly by the experts (three pathologists).

### 3.5.1 Horizontal Search: Cancer Type Recognition

The first series of experiments undertaken for all anatomic sites was *horizontal search*. The query WSI is compared against all other cases in the repository, regardless of anatomic site categorization. Of course, the primary anatomic site is generally known, and, in many cases, the cancer type may also be known to the pathologist. Thus, the purpose of the horizontal search (which is for either organ or cancer type recognition) is principally a fundamental algorithmic validation that may also have applications like searching for origin of malignancy in case of metastatic cancer. The results of the horizontal search are depicted in Figure 3.3. All experiments were conducted via "*leave-one-patient-out*" validation.

**Observations.** Provided there are *sufficient* number of patients, we observed that the more we retrieve the more likely it was to achieve the right diagnosis. General top-n accuracy that is common in the computer vision literature show high values but may not be suitable in the medical domain as it considers the search to be a success if at least one of the search results has the same cancer type as the query image. The majority vote among top n search results appears to be much more conservative and perhaps more appropriate. With some exceptions, a general trend is observable that the more images/patients are available the higher the search- based consensus accuracy. The number of cases positively correlated with the majority vote accuracy for both frozen sections and permanent diagnostic slides.

### 3.5.2 Vertical Search: Correctly Subtyping Cancer

In the second series of experiments, we performed *vertical search*. Given the primary site of the query slide we confined the search only to WSIs from that organ. Hence, the goal of the vertical search was to recognize the cancer subtype. For this purpose, only those primary anatomic sites in the dataset with at least two possible subtypes were selected. Sample retrievals are illustrated in Figure A.5. The results for "leave-one-patient-out" validation are depicted in Figure 3.4 and Figure 3.5.

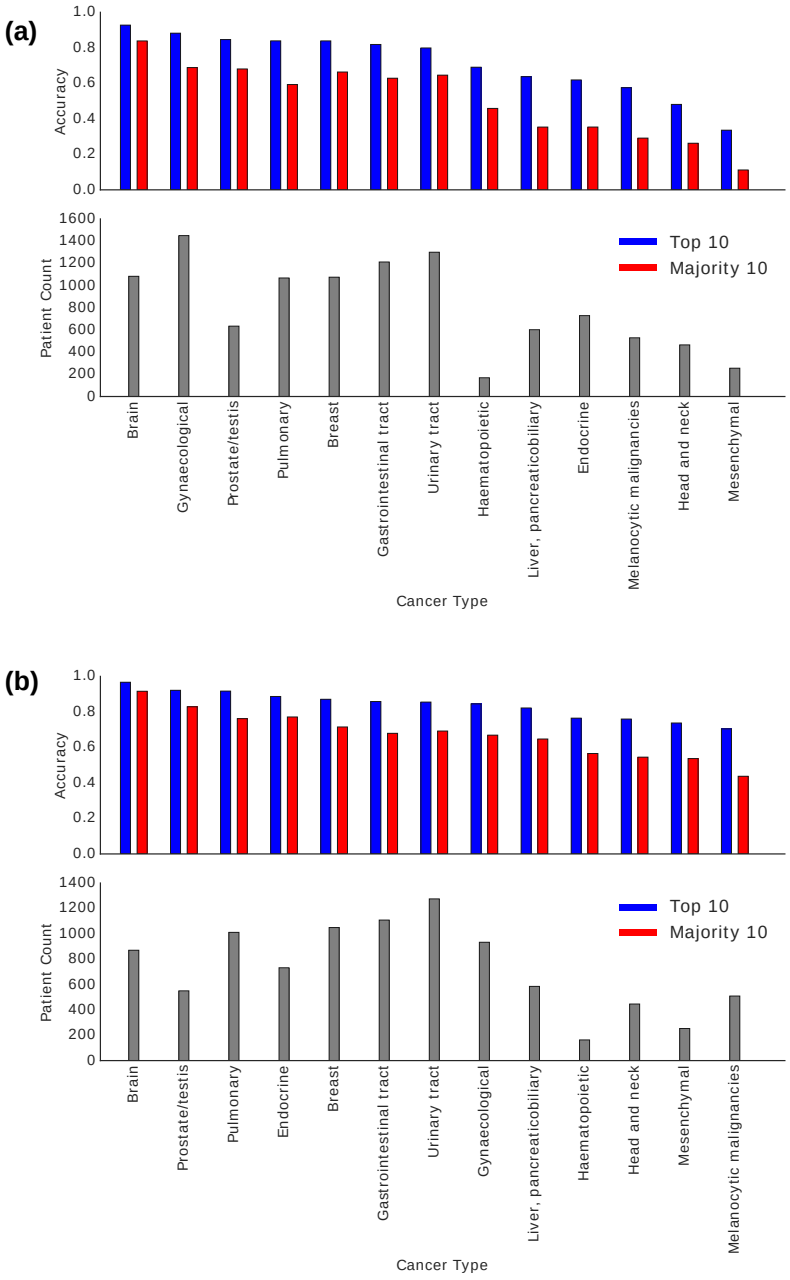
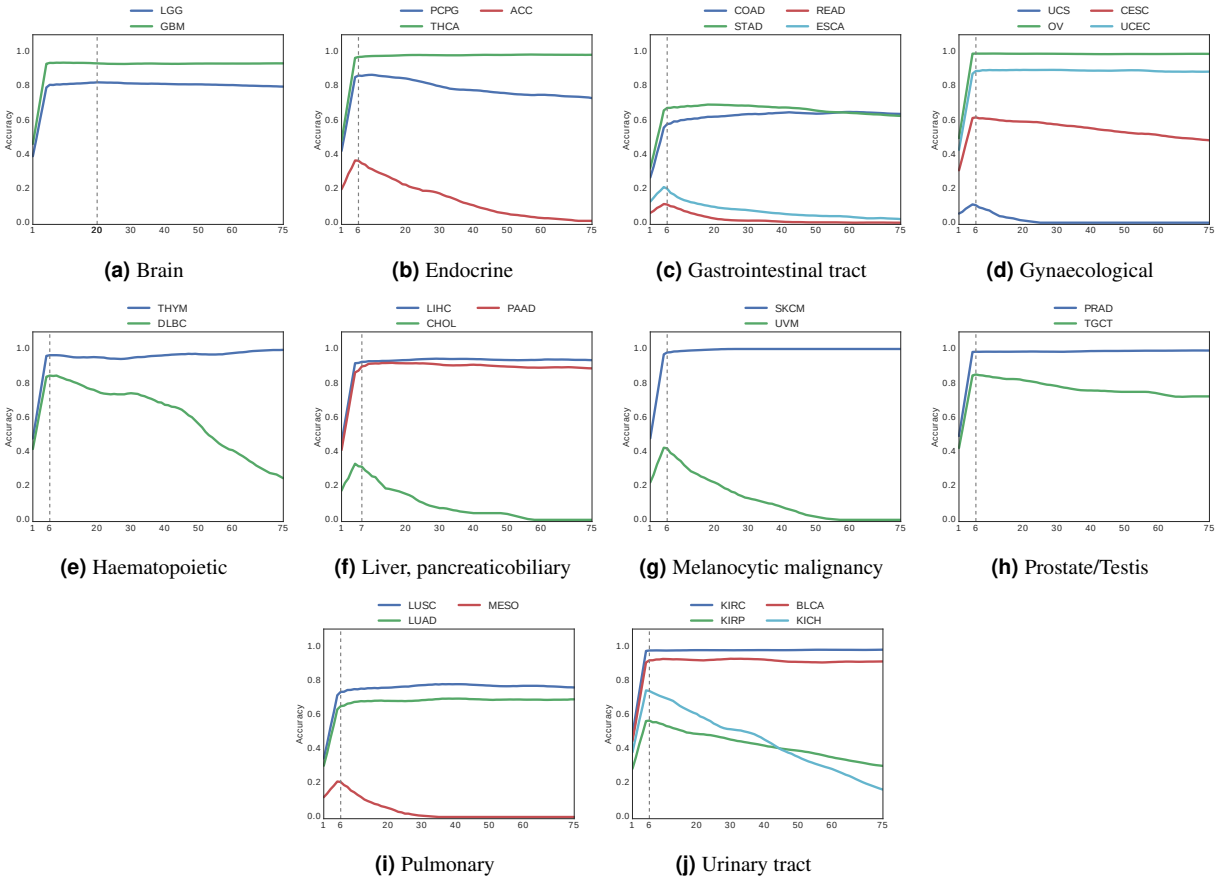



Figure 3.3: Horizontal search for frozen sections (top) and permanent diagnostic slides (bottom).



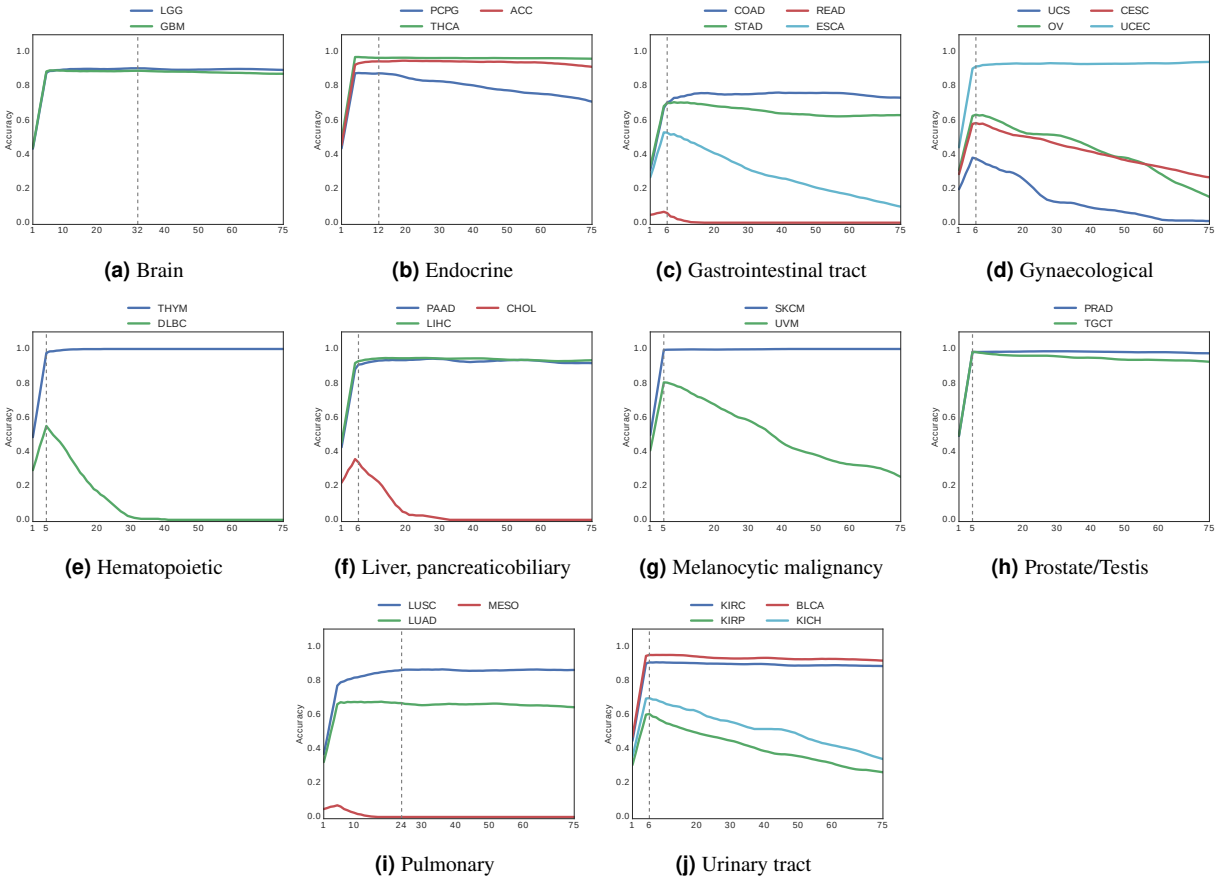

**(a)** Brain **(b)** Endocrine **(c)** Gastrointestinal tract **(d)** Gynaecological

**(e)** Haematopoietic **(f)** Liver, pancreaticobiliary **(g)** Melanocytic malignancy **(h)** Prostate/Testis

**(i)** Pulmonary **(j)** Urinary tract

Figure 3.4: Vertical search in frozen sections slides from anatomic sites with at least two cancer subtypes.

**Observations.** For both frozen sections and permanent diagnostic slides we continue to see a general trend whereby "*the more patients the better*". With majority-vote accuracy values reaching above 90% in many cases for both frozen and diagnostic slides shows that a search-based computational consensus appear to be possible when a large number of evidently diagnosed patients are available. In most cases, it appeared that taking the majority of the top-7 search results provided the highest accuracy in most cases. However, the accuracy dropped drastically for subtypes with a small number of patients as we retrieved more and more images beyond 6 slides as the majority in such cases were taken from incorrect cases (we do not filter any result; no threshold is used; hence, all search results are considered as valid results). Based on all observations, it seems that there is a direct relationship between the number of diagnosed WSIs in the dataset and achievable consensus accuracy.




**(a)** Brain **(b)** Endocrine **(c)** Gastrointestinal tract **(d)** Gynaecological

**(e)** Hematopoietic **(f)** Liver, pancreaticobiliary **(g)** Melanocytic malignancy **(h)** Prostate/Testis

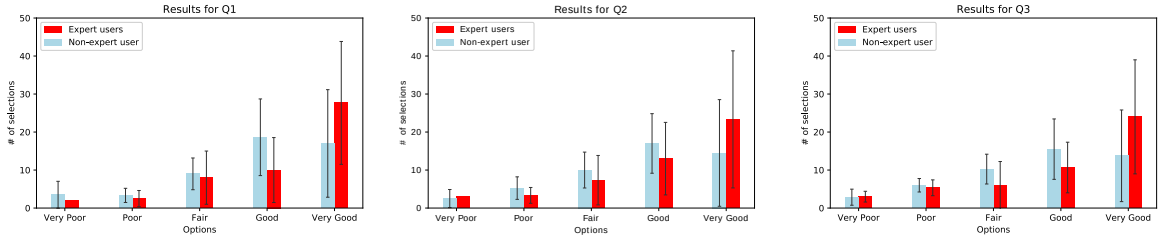
**(i)** Pulmonary **(j)** Urinary tract

Figure 3.5: Vertical search in permanent diagnostic slides from anatomic sites with at least two cancer subtypes.

For vertical search we calculated positive correlations of 0.5456 for frozen sections and 0.5974 for permanent diagnostic slides. This trend was more pronounced for horizontal search with positive correlation of 0.7780 for frozen sections slides and 0.7201 for permanent diagnostic slides.

### 3.5.3 Testing by Pathologists

We measured the effectiveness of search and retrieval through feedbacks of pathologists, evaluating how well the search results align with the subjective perception of its end-users. The Yottixel search results were evaluated by three pathologists and 4 non-experts




(a) Responses for $Q_1$. (b) Responses for $Q_2$ (c) Responses for $Q_3$

Figure 3.6: Response frequency for each option among the top three search results. There are more selections of `Poor` and `Very poor` for $Q_3$ compared with $Q_1$ and $Q_2$.

(see Figure 3.6, Table A.1 in Appendix A). We created a web application to gather the pathologist's evaluations about the search results. The web application presented a user with query images and their top three search results. The pathologists were not aware of the order in which top-3 results were shown to them. For each session, there were total of 48 queries presented to them. All pathologists answered the same questions, but we reshuffled the questions and ordering in each session to counteract any biases.

For each query result, pathologists would provide their feedback from five discrete values–Very Poor, Poor, Fair, Good, and Very Good. After gathering the data for the study, we sorted the pathologists' feedback in the original top-3 ordering, referring $Q_1$, $Q_2$, and $Q_3$ as the top $1^{st}$, $2^{nd}$, and $3^{rd}$ result, respectively.

**Observations.** The general summary of the participant's feedback is presented in Figure 3.6. Both expert and non-expert users, ranked $Q_1$ more positively than $Q_3$. It is interesting to note that, on an average, non-expert users ranked a higher number of `Very Poor` to $Q_1$ compared with $Q_2$ and $Q_3$. However, this was not true for pathologists. The trends for pathologists are very concrete and reflect positively on our approach. For instance, $Q_1$ has the highest number of `Very Good` compared with others, $Q_2$ has higher number of `Very Good` and `Good` than $Q_3$. Similarly, $Q_3$ has the highest number of `Very Poor`.

## 3.6 Summary

This chapter discussed Yottixel, a framework for representing a whole-slide image as a mosaic then eventually as a set of barcodes (BoB). One downside of Yottixel is that it is

completely unsupervised even when the labels could be made available. These labels can enable Yottixel in extracting more discriminative features. The next three chapters will discuss three different weakly-supervised methods for training Yottixel's backbone (feature extractor).

# Part II

# Weakly-Supervised Methods

Yottixel is an unsupervised framework. Its performance is governed by the quality of features extracted from the underlying pre-trained backbones. Weakly supervised methods allow training the Yottixel's backbone model, thereby enhancing its performance for the cancer subtype prediction. Multi-instance learning (MIL) approaches are used as a form of weak-supervision. Three different MIL methods have been proposed that not only improve the Yottixel for the cancer subtype classification, but also enable visualizing the patches that are deemed important for the given prediction. The proposed methods do not require extensive regional- or pixel-level information, and work with information usually available along WSIs, such as anatomical site (i.e., the organ), and primary diagnosis.

# Chapter 4

# Learning Permutation-invariant Representations using Memory Networks

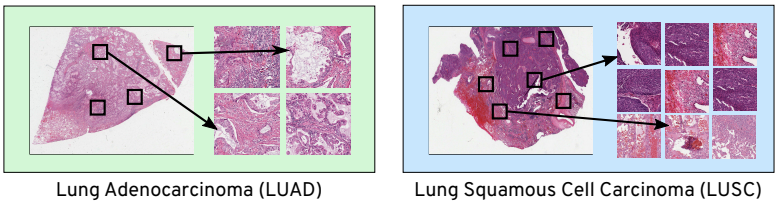
## 4.1 Prologue

This chapter is based on the following paper published during this Ph.D. research:

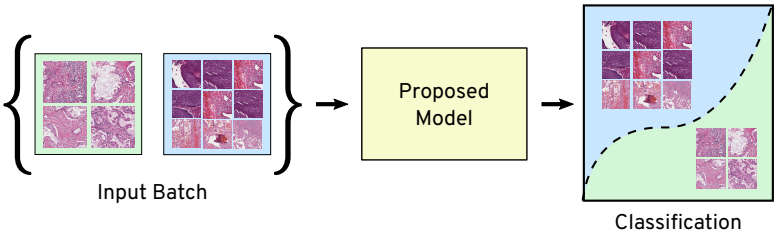
A. **S. Kalra**, et al. *Learning permutation-invariant representations using memory networks.* European Conference on Computer Vision (ECCV) (2020).

Generally, a WSI is accompanied with information such as its anatomic site of origin and/or the primary diagnosis. This information provides an opportunity to improve discriminative capabilities of Yottixel's backbone (feature extractor). As noted in the previous chapter (Chapter 3), Yottixel represents a WSI as a set of patches called mosaic. Therefore, all patches in mosaic are associated to a single label (e.g., primary diagnosis). Henceforth, multi-instance learning (MIL) methods are required to train on such data. This chapter proposes a MIL method called Memory-based Exchangeable Model (MEM).

**Summary.** Many real world tasks such as classification of digital histopathological images and 3D object detection involve learning from a set of instances. In these cases, only a group of instances or a set, collectively, contains meaningful information and therefore only the sets have labels, and not individual data instances. In this chapter, a permutation invariant neural network called *Memory-based Exchangeable Model (MEM)* for learning universal set




(a) Patch Extraction




(b) Set Classification

Figure 4.1: An exemplar application of learning permutation invariant representation for disease classification of Whole-Slide Images (WSIs). (a) A set of patches are extracted from each WSI of patients with lung cancer. (b) The sets of patches are fed to the proposed model for classification of the sub-type of lung cancer—LUAD versus LUSC. The model classifies on a per set basis. This form of learning is known as Multi Instance Learning (MIL).

functions is proposed. The MEM model consists of memory units which embed an input sequence to high-level features enabling it to learn inter-dependencies among instances through a self-attention mechanism. MEM is evaluated on various toy datasets, point cloud classification, and classification of whole slide images (WSIs) into two subtypes of lung cancer—Lung Adenocarcinoma, and Lung Squamous Cell Carcinoma. A new dataset of containing only lung slides are created from the dataset used in the last chapter. The proposed approach achieves a competitive accuracy of 84.84% for classification of two sub-types of lung cancer which is 15% improvement over Yottixel. The results on other datasets are promising as well, and demonstrate the efficacy of the proposed model.

## 4.2 Introduction

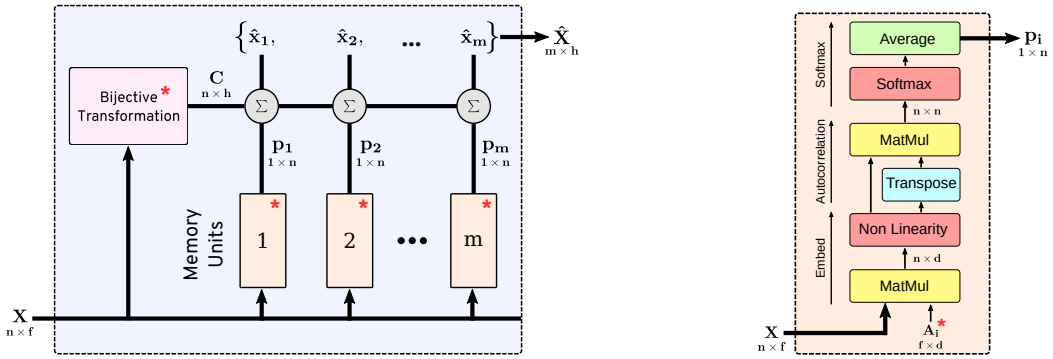
Deep artificial neural networks have achieved impressive performance for representation learning tasks. The majority of these deep architectures take a single instance as an input. Recurrent Neural Networks (RNNs) are a popular approach to learn representations from

sequential ordered instances. However, the lack of permutation invariance renders RNNs ineffective for exchangeable or unordered sequences. We often need to learn representations of unordered sequential data, or exchangeable sequences in many practical scenarios such as Multiple Instance Learning (MIL). In the MIL scenario, a label is associated with a set, instead of a single data instance. One of the application of MIL is classification of high resolution histopathology images, called whole slide images (WSIs). Each WSI is a gigapixel image with size $\approx$ 50,000 $\times$ 50,000 pixels. The labels are generally associated with the entire WSI instead of patch, region, or pixel level. MIL algorithms can be used to learn representations of these WSIs by disassembling them into multiple representative patches as discussed in the last chapter.

MEM is a novel architecture for exchangeable sequences incorporating attention over the instances to learn inter-dependencies. We use the results from Deep Sets [60] to construct a permutation invariant model for learning set representations. The main contribution is a sequence-to-sequence permutation invariant layer called **Memory Block**. The proposed model uses a series of connected memory block layers, to model complex dependencies within an input set using a self attention mechanism. The model is validated using a toy datasets and two real-world applications. The real world applications include, i) point cloud classification, and ii) classification of WSI into two sub-type of lung cancers—Lung Adenocarcinoma (LUAD)/ Lung Squamous Cell Carcinoma (LUSC) (see Figure 4.1).

## 4.3 Related Work

The majority of the related work and background topics are covered in Chapter 2, §2.3. MEM is based on memory networks, an idea of using an external memory for relational learning tasks was introduced by Weston et al. [100]. Later, an end-to-end trainable model was proposed by Sukhbaatar et al. [101]. Memory networks enable learning of dependencies among instances of a set by providing an explicit memory representation for each instance in the sequence. The idea of self attention is popularized by [102], these models are known as *transformers*, widely used in NLP applications. The proposed MEM model uses the self-attention (similar to transformers) within memory vectors, aggregated using a pooling operation (weighted averaging) to form a permutation-invariant representation (based on Theorems 1 and 2 in Chapter 2, §2.3).




(a) memory block (b) memory unit

Figure 4.2: $X$ is an input sequence containing $n$ number of $f$-dimensional vectors. (a) The **memory block** is a sequence-to-sequence model that takes $X$ and returns another sequence $\hat{X}$. The output $\hat{X}$ is a permutation-invariant representation of $X$. A bijective transformation model (an autoencoder) converts the input $X$ to a permutation-equivariant sequence $C$. The weighted sum of $C$ is computed over different probability distributions $p_i$ from memory units. The hyper-parameters of a memory block are i) dimensions of the bijective transformation $h$, and ii) number of memory units $m$. (b) The **memory unit** has $A_i$, an embedding matrix (trainable parameters) that transforms elements of $X$ to a $d$-dimensional space (memories). The output $p_i$ is a probability distribution over the input $X$, also known as attention. The memory unit has a single hyper-parameter $d$, i.e. the dimension of the embedding space. (* represents learnable parameters.)

## 4.4 Proposed Approach

This section discusses the motivations, components, and offers an analysis of the proposed **M**emory-based **E**xchangeable **M**odel (MEM) capable of learning permutation invariant representation of sets and unordered sequences.

### 4.4.1 Motivation

In order to learn an efficient representation for a set of instances, it is important to focus on instances which are "important" for a given task at hand, i.e., we need to attend to specific instances more than other instances. We therefore use the memory network to learn an attention mapping for each instance. Memory networks are conventionally used for NLP for mapping questions posted in natural language to an answer [100, 101]. We exploit the

idea of having *memories* which can learn *key* features shared by one or more instances. Through these *key* features, the model can learn inter-dependencies using transformer style self-attention mechanism. As inter-dependencies are learnt, a set can be condensed into a compact vector such that a MLP can be used for a classification or regression learning.

### 4.4.2 Model Components

MEM is composed of four sequentially connected units: i) a feature extraction model, ii) memory units, iii) memory blocks, and iv) fully connected layers to predict the output.

A *memory block* is the main component of MEM and learns a permutation invariant representation of a given input sequence. Multiple memory blocks can be stacked together for modeling complex relationships and dependencies in exchangeable data. The memory block is made of memory units and a bijective transformation unit shown in Figure 4.2

**Memory Unit.** A memory unit transforms a given input sequence to an attention vector. The higher attention value represents the higher "importance" of the corresponding element of the input sequence. Essentially, it captures the relationships among different elements of the input. Multiple memory units enable the memory block to capture many complex dependencies and relationships among the elements. Each memory unit consists of an embedding matrix $\mathbf{A_i}$ that transforms a $f$-dimensional input vector $x_j$ to a $d$-dimensional memory vector $u_{ij}$, as follows:

$$u_{ij} = \rho(x_j \mathbf{A_i}),$$

where $\rho$ is some non-linearity. The memory vectors are stacked to form a matrix $\mathbf{U_i} = [u_{i0}, \ldots, u_{in}]$ of the shape $(n \times d)$. The relative degree of correlations among the memory vectors are computed using cross-correlation followed by a column-wise softmax and then taking a row-wise average, as follows:

$$\begin{aligned} S_i &= \text{column-wise-softmax}(\mathbf{U_i U_i^T}), \\ p_i &= \text{row-wise-average}(S_i), \end{aligned} \tag{4.1}$$

The $p_i$ is the final output vector $(1 \times n)$ from the $i^{th}$ memory unit $\mathbf{U_i}$, as shown in Figure 4.2. The purpose of memory unit is to embed feature vectors into another space that could correspond to a distinct "attribute" or "characteristic" of instances. The cross correlation or the calculated attention vector represents the instances which are highly suggestive of those "attributes" or "characteristic". We do not normalize memory vectors as magnitude of these vectors may play an important role during the cross correlation.

**Memory Block.** A memory block is a sequence-to-sequence model, i.e., it transforms a given input sequence $X = x_1, \ldots, x_n$ to another representative sequence $\hat{X} = \hat{x}_1, \ldots, \hat{x}_m$. The output sequence is invariant to the element-wise permutations of the input sequence. A memory block contains $m$ number of memory units. In a memory block, each memory unit takes a sequential data as an input and generates an attention vector. These attention vectors are subsequently used to compute the final output sequence. The schematic diagram of a memory block is shown in Figure 4.2a.

The final output sequence $\hat{X}$ of a memory block is computed as a weighted sum of $\mathbf{C}$ with the probability distributions $p_1, \ldots, p_m$ from all the $m$ memory units where $\mathbf{C}$ is a bijective transformation of $X$ learned using an autoencoder. Each memory block has its own autoencoder model to learn the bijective mapping. The $i^{th}$ element $\hat{x}_i$ of the output sequence $\hat{X}$ is computed as matrix multiplication of $p_i$ and $\mathbf{C}$, as follows:

$$\hat{x}_i = p_i \mathbf{C},$$

where, $p_i$ is the output of $i^{th}$ memory unit given by (4.1).

The bijective transformation from $X \mapsto C$ enables equivariant correspondence between the elements of the two sequences $X$ & $\hat{X}$, and maps two different elements in the input sequence to different elements in the output sequence. It must be noted that bijective transformation is permutation equivariant not invariant. The reconstruction maintains one-to-one mapping between $X$ and $C$. The final output sequence from a memory block is permutation invariant as it uses matrix multiplication between $p_i$ (attention) and $C$.

### 4.4.3 Model Architecture

1. Each element of a given input sequence $X = x_1, \ldots, x_n$ is passed through a feature extraction model to produce a sequence of feature vectors $F = f_1, \ldots, f_n$.

2. The feature sequence $F$ is then passed through a memory block to obtain another sequence $\hat{X}$ which is a permutation-invariant representation of the input sequence. The number of elements in the sequence $\hat{X}$ depends on the number of memory unit in the memory block layer.

3. Multiple memory blocks can be stacked in series. The output from the last memory block is either vectorized or pooled, which is subsequently passed to a MLP layer for classification or regression.

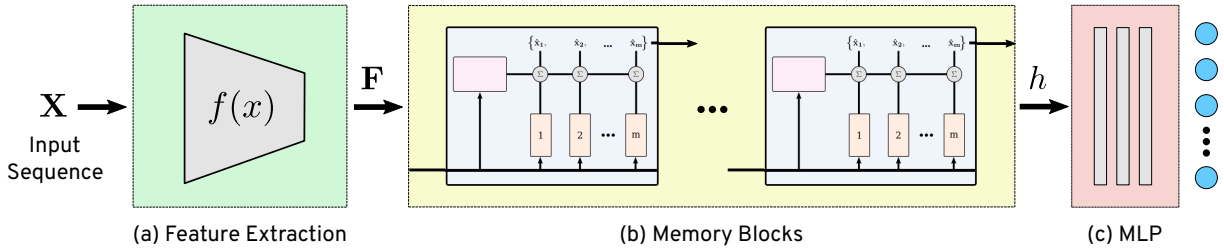



Figure 4.3: The overall architecture of the proposed Memory-based Exchangeable Model (MEM). The input to the model is a sequence, for e.g., a sequence of images or vectors. Each element of the input sequence $X$ is passed through **(a)** feature extractor (CNN or MLP) to extract a sequence of feature vectors $F$, which is passed to **(c)** sequentially connected memory blocks. A memory block outputs another sequence which is a permutation-invariant representation of the input sequence. The output from the last memory block is vectorized and given to **(c)** MLP layers for classification/regression.

### 4.4.4 Analysis

This section discusses the mathematical properties of our model. We use theorems from Deep Sets [60] to prove that our model is permutation invariant and universal approximator for arbitrary set functions.

**Property 1.** Memory units are permutation equivariant.

Consider an input sequence $X = x_1 \dots x_n$. Since, for each memory unit,

$$\mathbf{U_i} = [\rho(x_o\mathbf{A_i}), \rho(x_1\mathbf{A_i}), \dots, \rho(x_n\mathbf{A_i})]$$

By Equation (**??**), $\mathbf{U_i}$ is permutation equivariant and thus $S_i$ in (4.1) is permutation equivariant. Finally, the attention vector $p_i$ is calculated by averaging all rows, therefore the final output of memory unit $p_i$ is permutation equivariant.

**Property 2.** Memory Blocks are permutation invariant.

A memory block layer consisting of $m$ memory units generates a sequence $\hat{X} = \hat{x}_1, \dots, \hat{x}_m$ where $\hat{x}_i$ can be written as:

$$\hat{x}_i = p_i\mathbf{C}$$

Since both $\mathbf{C}$ and $p_i$ are permutation equivariant, therefore, $\hat{x}_i$, which is calculated by matrix multiplication of $p_i$ and $\mathbf{C}$, is permutation invariant.

| Methods | Sum of Even Digits | | Prime Sum | Counting Unique Images | | Maximum of Set | | Gaussian Clustering |
|---|---|---|---|---|---|---|---|---|
| | Accuracy | MAE | Accuracy | Accuracy | MAE | Accuracy | MAE | NLL |
| FF + MEM + MB1 (ours) | 0.9367 ± 0.0016 | 0.2516 ± 0.0105 | 0.9438 ± 0.0043 | 0.7108 ± 0.0084 | 0.3931 ± 0.0080 | 0.9326 ± 0.0036 | 0.1449 ± 0.0068 | **1.348** |
| FF + MEM + Mean (ours) | 0.9355 ± 0.0015 | 0.2437 ± 0.0087 | 0.7208 ± 0.0217 | 0.4264 ± 0.0062 | 0.9525 ± 0.0109 | 0.9445 ± 0.0035 | 0.1073 ± 0.0067 | 1.523 |
| FF + MEM + Max (ours) | **0.9431 ± 0.0020** | **0.2295 ± 0.0098** | 0.9361 ± 0.0060 | 0.6888 ± 0.0066 | 0.4140 ± 0.0079 | 0.9498 ± 0.0022 | 0.1086 ± 0.0060 | 1.388 |
| FF + MEM + Dotprod (ours) | 0.8411 ± 0.0045 | 0.3932 ± 0.0065 | **0.9450 ± 0.0086** | **0.7284 ± 0.0055** | **0.3664 ± 0.0037** | **0.9517 ± 0.0041** | **0.0999 ± 0.0097** | 1.363 |
| FF + MEM + Sum (ours) | 0.9353 ± 0.0022 | 0.2739 ± 0.0081 | 0.6652 ± 0.0389 | 0.3138 ± 0.0094 | 1.3696 ± 0.0151 | 0.9430 ± 0.0031 | 0.1318 ± 0.0058 | 1.611 |
| FF + Mean (DS) | 0.9159 ± 0.0019 | 0.2958 ± 0.0049 | 0.5280 ± 0.0078 | 0.3140 ± 0.0071 | 1.2169 ± 0.0136 | 0.3223 ± 0.0075 | 1.0029 ± 0.0155 | 2.182 |
| FF + Max (DS) | 0.6291 ± 0.0047 | 1.3292 ± 0.0211 | 0.9257 ± 0.0033 | 0.7088 ± 0.0060 | 0.3933 ± 0.0059 | **0.9585 ± 0.0012** | **0.0742 ± 0.0032** | 1.608 |
| FF + Dotprod (DS) | 0.1503 ± 0.0015 | 1.8015 ± 0.0016 | 0.9224 ± 0.0028 | **0.7254 ± 0.0063** | **0.3726 ± 0.0054** | **0.9548 ± 0.0017** | 0.1355 ± 0.0027 | 8.538 |
| FF + Sum (DS) | 0.6333 ± 0.0043 | 0.5763 ± 0.0069 | 0.5264 ± 0.0050 | 0.2982 ± 0.0042 | 1.3415 ± 0.0169 | 0.3344 ± 0.0038 | 0.9645 ± 0.0111 | 12.05 |

Table 4.1: Results on the toy datasets for different configurations of MEM and feature pooling. It must be noted that for Maximum of Set, the configuration FF + Max (DS) achieves the best accuracy but it may predict the output perfectly by learning the identity function therefore we highlighted second best configuration FF + Dotprod (DS) as well.

## 4.5 Experiments

We performed two series of experiments comparing MEM against the simple pooling operations proposed by Deep Sets [60]. In the first series of experiments, we established the learning ability of the proposed model using toy datasets. For the second series, we used two real-world dataset, i) classification of subtypes of lung cancer against the largest public dataset of histopathology whole slide images (WSIs) [103], and ii) 3-D object classification using Point Cloud Dataset [104].

**Model Comparison.** We compared the performance of MEM against Deep Sets [60]. We use same the feature extractor for both Deep Sets and MEM, and experimented with different choices of pooling operations—max, mean, dot product, and sum. MEM also has a special pooling "$mb1$", which is a memory block with a single memory unit in the last hidden layer. Therefore, we tested 9 different models for each experiment—five configurations of our model, and four configurations of Deep Sets. We tried to achieve the best performance by varying the hyper-parameters for each of the configuration of both MEM and Deep Sets. We found that MEM had higher learning capacity, therefore higher number of parameters resulted in better accuracy for MEM but not necessarily for Deep Set. We denote the common feature extractor as **FF** and Deep Sets as **DS** in the discussion below. The other approaches that are compared have been appropriately cited.

### 4.5.1 Toy Datasets

To demonstrate the advantage of MEM over simple pooling operations, we consider four toy problems, involving regression and classification over sets. We constructed these toy

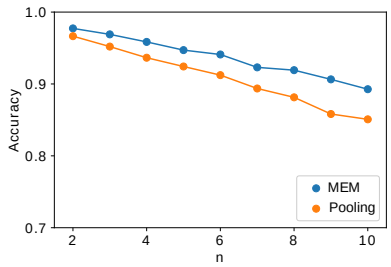



Figure 4.4: Comparison of MEM and feature pooling on a regression problem involving finding the sum of even digits within a set of MNIST images. Each point corresponds to the best configurations for the two models.

datasets using the MNIST dataset.

**Sum of Even Digits.** Sum of even digits is a regression problem over the set of images containing handwritten digits from MNIST. For a given set of images $X = \{x_1, \ldots, x_n\}$, the goal is to find the sum of all even digits. We used the Mean Absolute Error (MAE). We split the MNIST dataset into 70-30% training, and testing data-sets, respectively. We sampled 100,000 sets of 2 to 10 images from the training data. For testing, we sampled 10,000 sets of images containing $m$ number of images per set where $m \in [2, 10]$. Figure 4.4 shows the performance of MEM against simple pooling operations with respect to the number of images in the set.

**Prime Sum.** Prime Sum is a classification problem over a set of MNIST images. A set is labeled positive if it contains any two digits such that their sum is a prime number. We constructed the dataset by randomly sampling five images from the MNIST dataset. We constructed the training data with 20,000 sets randomly sampled from the training data of MNIST. For testing, we randomly sampled 5,000 sets from the testing data of MNIST. The results are reported in the second column of Table 4.1 that shows the robustness of memory block.

**Maximum of a Set.** Maximum of a set is a regression problem to predict the highest digit present in a set of images from MNIST. We constructed a set of five images by randomly selecting samples from MNIST dataset. The label for each set is the largest number present in the set. For example, images of $\{2, 5, 3, 3, 6\}$ is labeled as 6. We constructed 20,000 training sets and for testing we randomly sampled 5,000. The detailed comparison

of accuracy and MAE between different models is given in the second last column of Table 4.1. We found that FF+Max learns the identity mapping and thus results in a very high accuracy. In all the training sessions, we consistently obtained the training accuracy of 100% for the FF+Max configuration, whereas MEM generalizes better than the Deep Sets.

**Counting Unique Images.** Counting unique images is a regression problem over a set. This task involves counting unique objects in a set of images from fashion MNIST dataset [105]. We constructed the training data by selecting a set, as follows:

1. Let n be the number of total images and u be the number of unique image in the set.
2. Randomly select an integer $n$ between 2 and 10.
3. Randomly select another integer $u$ between 1 and $n$.
4. Select $u$ number of unique objects from fashion-MNIST training data.
5. Then add $n$-$u$ number of randomly selected objects selected in the previous step.

The task is to count unique objects $u$ in a given set. The results are shown in the third column of Table 4.1.

**Amortized Gaussian Clustering.** Amortized Gaussian clustering is a regression problem that involves estimating the parameters of a population of Mixture of Gaussian (MoG). Similar to Set Transformer [106], we test our model's ability to learn parameters of a Gaussian Mixture with $k$ components such that the likelihood of the observed samples is maximum. This is in contrast to the EM algorithm which updates parameters of the mixture recursively until the stopping criterion is satisfied. Instead, we use MEM to directly predict parameters of a MoG i.e. $f(x;\theta) = \{\pi(x), (\mu(x), \sigma(x))_{j=1}^{k}\}$. For simplicity we sample from MoG with only four components. The Generative process for each training dataset is as follows

1. Mean of each Gaussian is selected from a uniform distribution i.e. $\mu_{j=1}^{k} \sim \text{Unif}(0, 8)$.
2. Select a cluster for each instance in the set, i.e.,
$$\pi \sim \text{Dir}([1, 1]^T); z_i \sim \text{Categorical}(\pi)$$
3. Generate data from an univariate Gaussian $\sim \mathcal{N}(\mu_{z_i}, 0.3)$.

We created a dataset of 20,000 sets each consisting of 500 points sampled from different MoGs. Results in Table 4.1 show that MEM is significantly better than Deep Sets.

### 4.5.2 Real World Datasets

To show the robustness and scalability of the model for the real-world problems, we have validated MEM on two larger datasets. Firstly, we tested our model on a point cloud dataset for predicting the object type from the set of 3D coordinates. Secondly, we used the largest public repository of histopathology images (TCGA) [103] to differentiate between two main sub-types of lung cancer. Without any significant effort in extracting histologically relevant features and fine-tuning, we achieved a remarkable accuracy of 84.84% on 5-fold validation.

#### 4.5.2a. Point Cloud Classification

We evaluated MEM on a more complex classification task using ModelNet40 [104] point cloud dataset. The dataset consists of 40 different objects or classes embedded in a three dimensional space as points. We produce point-clouds with 100 points (x, y, z-coordinates) each from the mesh representation of objects using the point-cloud library's sampling routine [107][1]. We compare the performance against various other models reported in Table 4.2. We experimented with different configurations of our model and found that FF+MB1 works best for 100 points cloud classification. We achieves the classification accuracy of 85.21% using 100 points. Our model performs better than Deep Sets and Set Transformer for the same number of instances, showing the effectiveness of having attention from memories.

#### 4.5.2b. Lung Cancer Subtype Classification

Lung Adenocarcinoma (LUAD) and Lung Squamous Cell Carcinoma (LUSC) are two main types of non-small cell lung cancer (NSCLC) that account for 65-70% of all lung cancers [112]. Classifying patients accurately is important for prognosis and therapy decisions. Automated classification of these two main subtypes of NSCLC is a crucial step to build computerized decision support and triaging systems. We present a two-staged method to differentiate LUAD and LUSC for whole slide images, short WSIs, that are very large images. Firstly, we implement a method to systematically sample patches/tiles from WSIs. Next, we extract image features from these patches using Densenet [95]. We then use MEM to learn the representation of a set of patches for each WSI.

[1] We obtained the training and test datasets from Zaheer et al. [60]

| Configuration | Instance Size | Accuracy |
|---|---|---|
| 3DShapeNet [104] | $30^3$ | 0.77 |
| Deep set [60] | 100 | 0.8200 |
| VoxNet [108] | $32^2$ | 0.8310 |
| 3D GAN [109] | $64^3$ | 0.833 |
| Set Transformer [106] | 100 | 0.8454 |
| Set Transformer [106] | 1000 | 0.8915 |
| Deep set [60] | 5000 | 0.9 |
| MVCNN [110] | $164 \times 164 \times 12$ | 0.901 |
| Set Transformer [106] | 5000 | 0.9040 |
| VRN Ensemble [111] | $32^3$ | 0.9554 |
| **FF + MEM + MB1 (Ours)** | **100** | **0.8521** |

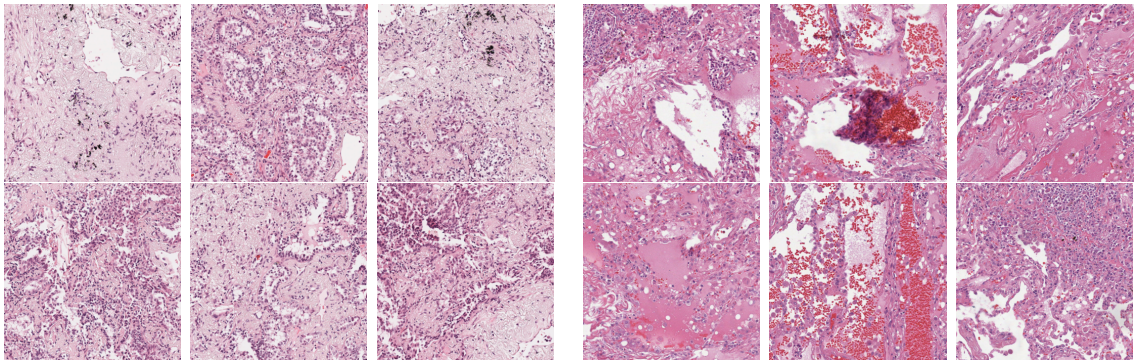
Table 4.2: Test accuracy for the point cloud classification on different instance sizes using various methods. MEM with configuration FF + MEM + MB1 achieves 85.21% accuracy for the instance size of 100 which is best compared to others.

To the best of our knowledge, this is the first ever study conducted on all the lung cancer slides in TCGA dataset (comprising of 2 TB of data consisting of 2.5 million patches of size 1000×1000 pixels). All research works in literature use a subset of the WSIs with their own test-train split instead of cross validation, making it difficult to compare against them. However, we have achieved greater than or similar to all existing research works without utilizing any expert's opinions (pathologists) or domain-specific techniques. We used 2,580 WSIs from TCGA public repository [103] with 1,249, and 1,331 slides for LUAD and LUSC, respectively. We process each WSI as follows.

1. **Tissue Extraction.** Every WSI contains a bright background that generally contains irrelevant (non-tissue) pixel information. We removed non-tissue regions using color thresholds.

2. **Selecting Representative Patches.** Segmented tissue is then divided into patches. All the patches are then grouped into a pre-set number of categories (classes) via a clustering method. A 10% of all clustered patches are uniformly randomly selected distributed within each class to assemble *representative patches*. Six of these representative patches for each class (LUAD and LUSC) is shown in Figure 4.5.

3. **Feature Set.** A set of features for each WSI is created by converting its representative patches into image features. We use DenseNet [95] as the feature extraction model. There are a different number of feature vectors for each WSI.



(a) Lung Adenocarcinoma (b) Lung Squamous Cell Carcinoma

Figure 4.5: The patches extracted from two WSIs of patients with (a) LUAD and (b) LUSC. Each slide roughly contains 500 patches.

The results are shown in Table 4.3. We achieved the maximum accuracy of 84.84% with FF + MEM + Sum configuration. It is difficult to compare our approach against other approaches in literature due to non-standardization of the dataset. Coudray et al. [92] used the TCGA dataset with around 1,634 slides to classify LUAD and LUSC. They achieved AUC of 0.947 using patches at 20×. We achieved a similar AUC of 0.94 for one of the folds and average AUC of **0.91**. In fact, without any training they achieved the similar accuracy as our model (around 85%). It is important to note that we did not do any fine-tuning or utilize any form of input from an expert/pathologist. Instead, we extracted diverse patches and let the model learn to differentiate between two sub-types by "attending" relevant ones. Another study by Jaber et al. [113] uses cell density maps, achieving an accuracy of 83.33% and AUC of 0.9068. However, they used much smaller portion of the TCGA, i.e., 338 TCGA diagnostic WSIs (164 LUAD and 174 LUSC) were used to train, and 150 (71 LUAD and 79 LUSC).

## 4.6 Summary

This chapter introduced a Memory-based Exchangeable Model (MEM) for learning permutation invariant representations. The proposed method uses attention mechanisms over "memories" (higher order features) for modeling complicated interactions among elements of a set. It is proven that the model is universal approximation of set functions. One limitation of the approach is that "attention" is not individually computed for a patch. It hard to visualize the patches that deemed important for the prediction providing limited

| Methods | Accuracy |
|---|---|
| **Coudray et al. [92]** | **0.85** |
| Jabber et al. [113] | 0.8333 |
| Khosravi et al. [114] | 0.83 |
| Yu et al. [115] | 0.75 |
| **FF + MEM + Sum (ours)** | **0.8484 ± 0.0210** |
| FF + MEM + Mean (ours) | 0.8465 ± 0.0225 |
| FF + MEM + MB1 (ours) | 0.8457 ± 0.0219 |
| FF + MEM + Dotprod (ours) | 0.6345 ± 0.0739 |
| FF + sum (DS) | 0.5159 ± 0.0120 |
| FF + mean (DS) | 0.7777 ± 0.0273 |
| FF + dotprod (DS) | 0.4112 ± 0.0121 |

Table 4.3: Accuracy for LUAD vs LUSC classification for various methods. For our experiments, we conducted comprehensive 5-fold cross validation accuracy whereas other methods have used non-standardized test set.

interoperability. In the next chapter, improve the interpretations of a model by explicitly learning an "attention" value for each mosaic's patch.

# Chapter 5

# Pay Attention with Focus: A Novel Learning Scheme for Classification of Whole Slide Images

## 5.1 Prologue

This chapter is based on the following paper published during this Ph.D. research:

A. **S. Kalra**, et al. *Pay Attention with Focus: A Novel Learning Scheme for Classification of Whole Slide Images*. International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI) (2021)

Chapter 4 introduced a new method to predict through mosaic data obtained from Yottixel. However, the limitation of the method was to infer the "importance" of each patch for the final classification. In this chapter, an attention-based models are used to find the relative contribution of each patch towards the final predictions. The attention-based model can be inferred across all patches of a WSI to find regions of interest (ROIs). These ROIs can be determined autonomously by the deep network.

**Summary.** The feature extractor backbone of Yottixel is fine-tuned using hierarchical target labels of WSIs, i.e., anatomic site and primary diagnosis. The set of encoded patch-level features from a WSI is then used to compute the primary diagnosis probability through the proposed *Pay Attention with Focus* scheme, an attention-weighted averaging of predicted

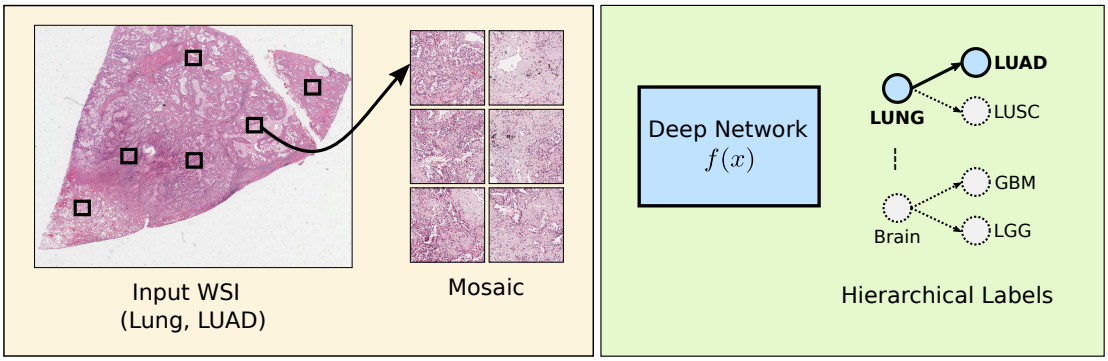
probabilities for all patches of a mosaic modulated by a trainable focal factor. Experimental results show that the proposed method can be as robust, and effective as the previously proposed MEM model (Chapter 4), however, with an advantage of being more transparent, explainable, computationally efficient.

## 5.2 Background

The majority of literature has been well discussed in Chapter 2. The proposed method utilizes an idea of training with multi-target labels arranged in hierarchy. A WSI usually contains at least two target labels, anatomic site, and primary diagnosis that are arranged in a hierarchy. The simplest way to deal with multi-label classification with $k$ labels is to treat this as $k$ independent binary classification. Although this approach may be helpful, it does not capture label dependencies. This limitation can degrade the performance in many applications where there is strong dependency among labels, for example, in WSI classification. To address this limitation, two different approaches, i.e., transformation and algorithm adaption methods, have been proposed [116]. In transformation-based methods, multi-label data is converted to new single label data to apply regular single-label classification. On the other hand, in the adaptation-based category, this is attempted to modify the basic single-label algorithm to handle multi-label data [117]. The FocAtt falls into the adaption-based category for handling the multi-target labels.

## 5.3 Method

There are two stages in the proposed method (i) bag preparation, and (ii) multi-instance learning with FocAtt-MIL. In the first stage, representative patches (called mosaic) are extracted from a WSI using Yottixel. The mosaic's patches are encoded to a set of feature vectors (called bag) using a deep network. The feature extraction model can be a pre-trained network, or can be fined-tuned to increase its effectiveness as shown in Figure 5.1. In the second stage, the proposed MIL technique (called FocAtt-MIL) is trained to predict the primary diagnosis for a given bag (a WSI). The schematic for the second stage is shown in Figure 5.2.




(a) Mosaic Preparation (b) Fine-tune using Heirarchical Labels

Figure 5.1: **Training a Feature Extractor.** A feature extractor is trained with hierarchical target labels of a WSI. (a) A set of representative WSI patches (called mosaic) is extracted [2]. (b) The patches are used to fine-tune a deep network; each patch is assigned the parent WSI's labels, i.e., anatomic site and primary diagnosis.

### 5.3.1 Model Components

**Bag Preparation.** Yottixel's patch selection method is used to extract the representative patches from a WSI, called mosaic. The mosaic is transformed into a bag $X = \{x_1, \ldots, x_n\}$, where $x_i$ is the feature vector of $i^{th}$ patch, obtained through a deep network (a feature extractor). The Figure 5.2 shows the bag preparation stage, the frozen network $f(x)$ represents a non-trainable deep network used as a feature extractor.

**Fine-tune a Feature Extractor using Hierarchical Labels.** In MIL, robust features enable weak learners to make better predictions thus improving the final aggregated prediction. A WSI is generally associated with the following two labels—anatomic site and primary diagnosis. These two labels are arranged in hierarchy as shown in Figure 5.1. Consider, $y_{as}$ and $y_{pd}$ represent anatomic site and primary diagnosis respectively. Then, instead of predicting these labels independently, we predict $P(y_{as})$, and $P(y_{pd}|y_{as})$. The conditional probability $P(y_{pd}|y_{as})$ helps in modelling the dependent relationship. Using Bayes theorem, we get, $P(y_{as}|y_{pd}) = P(y_{pd}|y_{as})P(y_{as})/P(y_{pd})$, where $P(y_{as}|y_{pd}) = 1$, because of the dependence. We simplify $P(y_{pd}) = P(y_{pd}|y_{as})P(y_{as})$, and compute cross entropy losses for the predictions of both $y_{as}$ and $y_{pd}$. We equally weight both the losses towards the final loss of the network.

**WSI Context Learning.** A single vector representation of a WSI (or a bag $X$) is

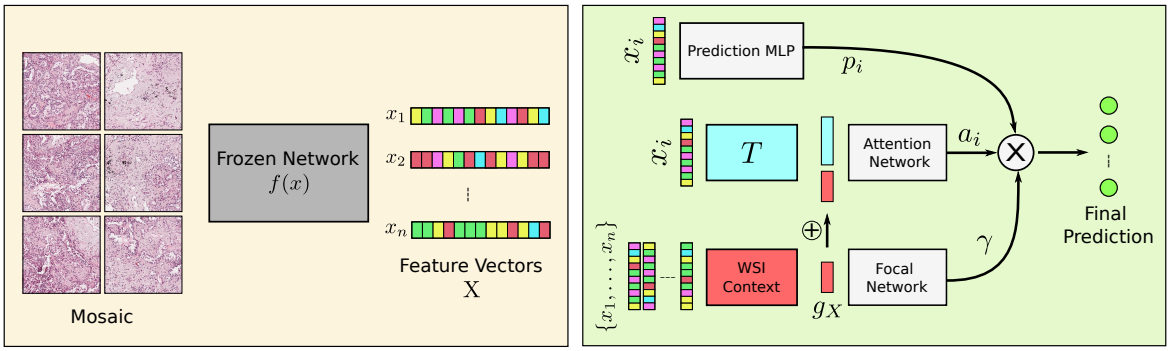



Figure 5.2: **Classification of WSIs with FocAtt-MIL.** The two-stage method for the classification of WSI. (a) The mosaic of a WSI is converted to a bag $X$ containing a set of feature vectors $\{x_1, \ldots, x_n\}$. (b) The feature vectors in a bag $X$ are transformed to the primary diagnosis probability through FocAtt-MIL. The prediction probability $p_i$ is computed for an individual feature vector $x_i$. A WSI context $g_X$ is computed for the entire bag $X$ using (5.1). The WSI context $g_X$ is used to compute the attention value $a_i$ and the focal factor $\gamma$. The final prediction is computed using (5.2).

computed as,

$$g_X = \phi(\theta(x_1), \ldots, \theta(x_n)), \tag{5.1}$$

where, $\theta$ is a neural network and $\phi$ is a pooling function, such as sum, mean, and max. It has been proven in [118] that (5.1) can approximate any set function. The vector $g_X$ is used by the attention module and the focal network.

**The FocAtt-MIL Approach.** The FocAtt-MIL is a permutation-invariant model that learns to predict a target label (primary diagnosis) $y_{pd}$ from a bag $X$ (a WSI). The approach is composed of four major components (Figure 5.2):

1. *Prediction MLP.* A prediction $p_i$ is computed for each item $x_i$ in the bag $X$, using a trainable deep network called Prediction MLP.

2. *WSI Context.* It a deep network that computes a single vector representing an entire bag $X$ using (5.1).

3. *Attention Module.* The attention module is composed of two networks, a transformation network $T$, and the Attention Network. The attention module takes the $i^{th}$ patch $x_i \in X$, and the WSI context $g_X$ to compute an attention value $a_i \in [0, 1]$ for that patch.

4. *Focal Network.* Another deep network that uses WSI context $g_X$ to compute a focal factor $\gamma$ (a vector) that modulates the final prediction. The length of $\gamma$ is same as the number of discrete values in the target label, thus allowing the per dimension modulation.

**The Final Prediction.** The final output from the FocAtt-MIL is computed by aggregating individual attention-weighted predictions modulated by the learned focal factor, as follows

$$y(j) = \sum_{i=1}^{n} \mathbf{p_i}(j)^{\gamma(j)} a_i. \tag{5.2}$$

The $\mathbf{p_i}$, and $\boldsymbol{\gamma}$ in (5.2) are both vectors. The $y$ is converted to a probability distribution by dividing with $sum(y)$.

## 5.4 Experiments

We evaluated the proposed approach for two different WSI classification tasks. All experiments are conducted with 4 Nvidia V100 GPUs (32 GB vRAM each). The code has been written using the Tensorflow library [119].

### 5.4.1 LUAD vs LUSC Classification

For this task, we utilized the dataset created in the study proposed in the last chapter. We establish the efficacy of FocAtt-MIL to differentiate between LUAD and LUSC. Similar to the last chapter, we obtained mosaic for each WSI using Yottixel (Chapter 3), and subsequently converted the mosaic to a bag $X$ of features using a pre-trained DenseNet [95]. We

Table 5.1: Performance comparison for LUAD/LUSC classification via transfer learning.

| **Algorithm** | **Accuracy** |
|---|---|
| Coudray et al. [92] | 0.85 |
| MEM (Chapter 4) | 0.85 |
| Khosravi et al. [114] | 0.83 |
| Yu et al. [115] | 0.75 |
| **FocAtt-MIL (proposed method)** | **0.88** |

did not fine-tune the feature extraction model for this task in order have a fair comparison against MEM (Chapter 4) and other approaches. We trained the FocAtt-MIL to classify bags between the two sub-types of lung cancer. We achieved the accuracy of 88% on test WSIs (AUC of 0.92). The accuracy has been reported in Table 5.1.

We conducted an **ablation study** to understand the effect of different model parameters. Removing the WSI context $g_X$ from the attention module, resulted in 4% reduction of the accuracy. Excluding the focal factor $\gamma$ and the global context $g_X$ from the final prediction, resulted in 6% reduction in the accuracy. The ablation suggests that the model's performance is the most optimal by (i) incorporating the WSI context $g_X$ in the attention computation, and (ii) allowing the focal factor to modulate the final aggregated prediction.

We used the attention module of the trained model to **visualize the attention heatmap** on the unseen WSIs (Figure 5.3). The visual inspection of these two WSIs reveals that the model made its decision based on regions containing malignant tissue and ignored non-cancerous regions. In the LUSC WSI (right), regions with squamous formations are deemed the most important ones. For the LUAD WSI (left), the salient regions are solely coming from the malignant area, implying that the model differentiates between normal lung alveolar tissue and LUAD. Therefore, one could say that attention heatmaps are histopathologically meaningful. For LUAD samples, regions where contrast makes cancerous glandular structures easier to recognize. However, this phenomenon cannot be seen in LUSC samples, as the model is responsive to regions that are completely composed of malignant squamous carcinoma.

### 5.4.2 Pan-cancer Analysis

In the second experiment series, we evaluated the approach against a large-scale pan-cancer classification of WSIs. The **dataset** used for this task has been proposed by Riasatian et al. [120]. It comprises more than 7 TB data, consisting of 7,097 training, and 744 test WSIs, distributed across 24 different anatomic sites, and 30 different primary diagnoses. All WSIs in the dataset are taken from a public repository of WSIs, TCGA [51]. We obtained a mosaic for each WSI, and then applied a cellularity filter [120] to further reduce the number of patches in each mosaic. Subsequently, we obtained 242,202 patches for training WSIs and 116,088 patches for testing WSIs. Each patch is of the size 1000×1000, but we resized them to 256×256 pixels.

We used three different **feature extractors** to validate the FocAtt-MIL. We prepared a separate "bag" for each feature extractor. These three feature extractors are:

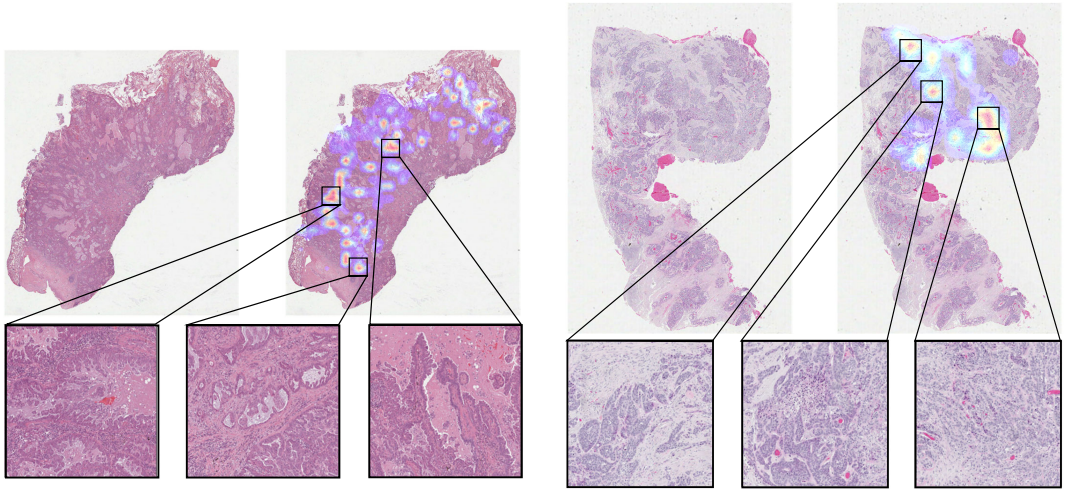


Figure 5.3: **Attention Visualization.** The attention values augmented on the two exemplar WSIs. **Left Image (LUAD):** Regions of the highest importance come from the cancerous regions while sparing normal lung tissue, fibrosis, and mucin deposition. Additionally, by inspecting important regions at a higher magnification, it is noticeable that the malignant glandular formations border with non-malignant areas. **Right Image (LUSC):** Regions that are considered to be important for classification are composed of malignant squamous cells. However, unlike LUAD, the attention model seems to be responsive to regions with solid malignant structures.

DenseNet (DN) [95], KimiaNet [120], and the fine-tuned DenseNet (FDN). We fined-tuned the DenseNet on training patches using weak labels obtained from their respective WSIs. The weakly labelled fine-tuning has shown to be effective [120]. In our case, the weak labels are anatomic site, and primary diagnosis, arranged in a hierarchy. This hierarchical arrangement of labels is incorporated during the training using the approach outlined earlier in the Section 8.4. For the fine-tuning, we used Adam optimizer [121] and a learning rate of $10^{-5}$ were used for 20 epochs.

We **trained the FocAtt-MIL** model with the same architecture for all the three different bags. We tested three different configurations of FocAtt-MIL, i.e., FocAtt-MIL-DN, FocAtt-MIL-KimiaNet, and FocAtt-MIL-FDN. For all the three configurations, we used the SGD optimizer with a learning rate of 0.01, weight decay of $10^{-6}$, and momentum of 0.9. We applied *gradient clipping* of 0.01 and dropout between layers to prevent the exploding gradients. We trained models for 45 epochs. **??** shows the validation loss and accuracy while training the three different configurations. It is evident that FocAtt-MIL-FDN is

outperforming from the very early epochs. It is interesting to note that, both FocAtt-MIL-FDN, and FocAtt-MIL-KimiaNet (feature extractors specialized for histopathology) seems to have converged to an optimal validation accuracy around 20-25 epochs.

The 30 unique primary diagnoses in the dataset can be further grouped into 13 tumor types. The type of tumour is generally known at the inference time, and the objective is to predict the cancer sub-type. To **validate the efficacy of our model**, we computed the cancer sub-type classification (i.e., primary diagnosis) accuracy for the given tumour type. This type of classification is called *vertical classification*. The vertical classification results are reported in Table 5.2[1]. The results show that FocAtt-MIL can elevate the accuracy of pre-trained features; DenseNet features have shown to under-perform compared to KimiaNet features [120, 122]. However, within the proposed FocAtt-MIL scheme, DenseNet features become quite competitive. This applies to the fine-tuned DenseNet (FocAtt-MIL-FDN) as well, whose results are on par with the highly customized KimiaNet features when used within the FocAtt-MIL framework.

## 5.5 Summary

This chapter introduced a MIL method with three-fold contributions (i) a novel attention-based MIL approach for the classification of WSIs, (ii) fine-tuning a feature extractor model using multiple and hierarchically arranged target labels of WSIs, and (iii) inferring the insights of the model's decision making by visualizing attention values. The limitation of a method is that it does not have capacity to model complex relationships within the patches of mosaic. Even though, it performed equivalent to MEM (Chapter 4), the next natural step is to combine strengths of both approaches into one method. A single method that can model complex relationship among patches, and allow attention inference. The next chapter discuss one such approach developed during the Ph.D.

[1]For abbreviations GBM, LGG, ACC,..., see Appendix A

Table 5.2: Pan-cancer vertical classification accuracy of FocAtt-MIL for features from regular DenseNet (FocAtt-MIL-DN), KimiaNet (FocAtt-MIL-KimiaNet), and DenseNet fine-tuned with hierarchical labels (FocAtt-MIL-FDN).

| **Tumor Type** | **Primary Diagnosis** | **FocatAtt-MIL-DN** | **FocAtt-MIL-KimiaNet** | **FocAtt-MIL-FDN** |
|---|---|---|---|---|
| Brain | GBM | **0.9714** | 0.9429 | 0.8571 |
| | LGG | 0.6410 | 0.7692 | **0.8205** |
| Endocrine | ACC | 0.6667 | 0.6667 | 0.6667 |
| | PCPG | **1.0000** | **1.0000** | **1.0000** |
| | THCA | 0.9608 | **1.0000** | **1.0000** |
| Gastrointestinal tract | COAD | **0.6875** | 0.4375 | 0.5000 |
| | ESCA | 0.5000 | **0.8571** | 0.5714 |
| | READ | 0.0833 | 0.5000 | **0.6667** |
| | STAD | **0.8333** | 0.7333 | **0.8333** |
| Gynaecological | CESC | 0.8824 | **0.9412** | 0.7647 |
| | OV | 0.5000 | 0.8000 | **1.0000** |
| | UCS | 0.6667 | **1.0000** | 0.3333 |
| Liver, pancreaticobiliary | CHOL | 0.2500 | 0.0000 | **0.5000** |
| | LIHC | 0.8857 | **0.9143** | 0.8571 |
| | PAAD | **1.0000** | 0.7500 | 0.8333 |
| Melanocytic malignancies | SKCM | **0.9167** | 0.8750 | **0.9167** |
| | UVM | **1.0000** | 0.2500 | **1.0000** |
| Prostate/testis | PRAD | **1.0000** | 0.9500 | **1.0000** |
| | TGCT | **1.0000** | **1.0000** | **1.0000** |
| Pulmonary | LUAD | 0.5789 | 0.8158 | **0.8947** |
| | LUSC | **0.9302** | 0.6977 | 0.7442 |
| | MESO | 0.6000 | **1.0000** | **1.0000** |
| Urinary tract | BLCA | 0.9118 | **1.0000** | 0.8529 |
| | KICH | 0.5455 | 0.6364 | **0.7273** |
| | KIRC | 0.9200 | 0.9000 | **0.9600** |
| | KIRP | 0.5714 | 0.6786 | **0.7143** |

# Chapter 6

# Representation Learning of Histopathology Images using Graph Neural Networks

## 6.1 Prologue

This chapter is based on the following paper published during this Ph.D. research:

A. M. Adnan, **S. Kalra**, et al. *Representation learning of histopathology images using graph neural networks.* Conference on Computer Vision and Pattern Recognition Workshops (CVPR-W) (2020)

Two MIL-based methods (Chapter 4 and Chapter 5) for classification of have been discussed. MEM (Chapter 4) enables capturing inter-dependence among patches, where as FocAtt (Chapter 5) allows to capture importance of individual patch. This chapter proposes a solution based on graph neural network that provides the strength of both the previously proposed approaches.

**Summary.** The proposed method models a WSI as a fully-connected graph, where each node represents a patch, and it is connected to every other node within the graph (full connected). The inter-relationship among nodes (adjacency matrix), and "attention" or importance of each node is learned during the training phase. The graph pooling operations is used to automatically infer patches with higher relevance. The performance of the

approach is validated for discriminating two sub-types of lung cancers, Lung Adenocarcinoma (LUAD) & Lung Squamous Cell Carcinoma (LUSC), achieving 88.8% and AUC of 0.89. This is the best results achieved so far for the LUSC/LUAD classification among the three proposed MIL approaches.

## 6.2 Background

This section covers the important topics related to graph neural networks that have been utilized for development of the proposed approach.

### 6.2.1 Deep Learning with Graphs

**Graph Representation.** A graph can be fully represented by its node list $V$ and adjacency matrix $\mathbf{A}$. Graphs can model many types of relations and processes in physical, biological, social, and information systems. A connection between two nodes $V_i$ and $V_j$ is represented using an edge weighted by $a_{ij}$.

**Graph Convolution Neural Networks (GCNNs).** GCNNs generalize the operation of convolution from grid data to graph data. A GCNN takes a graph as an input and transforms it into another graph as the output. Each feature node in the output graph is computed by aggregating features of the corresponding nodes and their neighboring nodes in the input graph. Like CNNs, GCNNs can stack multiple layers to extract high-level node representations. Depending upon the method for aggregating features, GCNNs can be divided into two categories, namely spectral-based and spatial-based. Spectral-based approaches define graph convolutions by introducing filters from the perspective of graph signal processing. Spectral convolutions are defined as the multiplication of a node signal by a kernel. This is similar to the way convolutions operate on an image, where a pixel value is multiplied by a kernel value. Spatial-based approaches formulate graph convolutions as aggregating feature information from neighbors. Spatial graph convolution learns the aggregation function, which is permutation invariant to the ordering of the node.

**ChebNet.** It was introduced by Defferrard et al. [123]. Spectral convolutions on graphs are defined as the multiplication of a signal $x \in R^N$ (a scalar for every node) with a filter $g(\theta) = diag(\theta)$ parameterized by $\theta \in R^N$ in the Fourier domain, i.e.,

$$g_\theta \circledast x = U g_\theta U^T x,$$

where $U$ is the matrix of eigenvectors of the normalized graph Laplacian $L = I_N - D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$. This equation is computationally expensive to calculate as multiplication with the eigenvector matrix U is $O(N^2)$. Hammond et al. [124] suggested that that $g_\theta$ can be well-approximated by a truncated expansion in terms of Chebyshev polynomials $T_k(x)$, i.e,

$$g_{\theta'}(\Lambda) \approx \sum_{k=0}^{K} \theta' T_k(\Lambda).$$

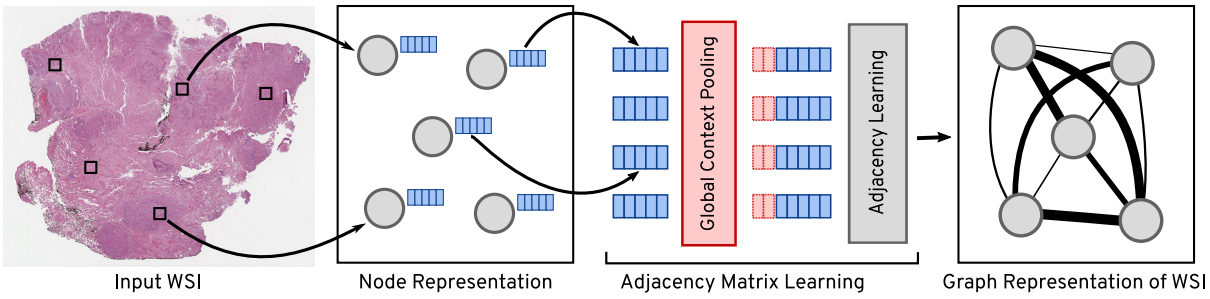



Figure 6.1: **Transforming a WSI to a fully-connected graph.** A WSI is represented as a graph with its nodes corresponding to distinct patches from the WSI. A node feature (a blue block beside each node) is extracted by feeding the associated patch through a deep network. A single context vector, summarizing the entire graph is computed by pooling all the node features. The context vector is concatenated with each node feature, subsequently fed into adjacent learning block. The adjacent learning block uses a series of dense layers and cross-correlation to calculate the adjacency matrix. The computed adjacency matrix is used to produce the final fully-connected graph. In the figure, the thickness of the edge connecting two nodes corresponds to the value in the adjacency matrix.

The kernels used in ChebNet are made of Chebyshev polynomials of the diagonal matrix of Laplacian eigenvalues. ChebNet uses kernel made of Chebyshev polynomials. Chebyshev polynomials are a type of orthogonal polynomials with properties that make them very good at tasks like approximating functions.

**GraphSAGE.** It was introduced by Hamilton et al. [125]. GraphSAGE learns aggregation functions that can induce the embedding of a new node given its features and neighborhood. This is called inductive learning. GraphSAGE is a framework for inductive representation learning on large graphs that can generate low-dimensional vector representations for nodes

and is especially useful for graphs that have rich node attribute information. It is much faster to create embeddings for new nodes with GraphSAGE.

**Graph Pooling Layers.** Similar to CNNs, pooling layers in GNNs downsample node features by pooling operation. We experimented with Global Attention Pooling, Mean Pooling, Max Pooling, and Sum Pooling. Global Attention Pooling [126] was introduced by Li et al. and uses soft attention mechanism to decide which nodes are relevant to the current graph-level task and gives the pooled feature vector from all the nodes.

### 6.2.2 Set Representation

**Universal Approximator for Sets.** We use results from Deep Sets [60] to get the global context of the set of patches representing WSI. Zaheer et al. proved in [60] that any set can be approximated by $\rho \sum(\phi(x))$ where $\rho$ and $\phi$ are some function, and $x$ is the element in the set to be approximated.

## 6.3 Method

The proposed method for representing a WSI has two stages, i) sampling important patches and modeling them into a fully-connected graph, and ii) converting the fully-connected graph into a vector representation for classification or regression purposes. These two stages can be learned end-to-end in a single training loop. The major novelty of our method is the learning of the adjacency matrix that defines the connections within nodes. The overall proposed method is shown in Figure 6.1 and Figure 6.2. The method can be summarized as follows.

1. The important patches are sampled from a WSI using a color-based method described in [127]. A pre-trained CNN is used to extract features from all the sampled patches.
2. The given WSI is then modeled as a fully-connected graph. Each node is connected to every other node based on the adjacency matrix. The adjacency matrix is learned end-to-end using Adjacency Learning Layer.
3. The graph is then passed through a Graph Convolution Network followed by a graph pooling layer to produce the final vector representation for the given WSI.

The main advantage of the method is that it processes entire WSIs. The final vector representation of a WSI can be used for various tasks—classification (prediction cancer type), search (KNN search), or regression (tumor grading, survival prediction) and others.

### 6.3.1 Model Components

**Patch Selection and Feature Extraction.** We used the Yottixel method for patch selection proposed in Chapter 3. Every WSI contains a bright background that generally contains irrelevant (non-tissue) pixel information. We removed non-tissue regions using color thresholds. Segmented tissue is then divided into patches. All patches are grouped into a pre-set number of categories (classes) via a clustering method. A portion of all clustered patches (e.g., 10%) are randomly selected within each class. Each patch obtained after patch selection is fed into a pre-trained DenseNet [95] for feature extraction. We further feed these features to trainable fully connected layers and obtain final feature vectors each of dimension 1024 representing patches.

**Graph Representation of WSI.** We propose a novel method for learning WSI representation using GCNNs. Each WSI is converted to a fully-connected graph, which has the following two components.

1. Nodes $V$: Each patch feature vector represents a node in the graph. The feature for each node is the same as the feature extracted for the corresponding patch.

2. Adjacency Matrix $\mathbf{A}$: Patch features are used to learn the $\mathbf{A}$ via adjacency learning layer.

**Adjacency Learning Layer.** Connections between nodes $V$ are expressed in the form of the adjacency matrix $\mathbf{A}$. Our model learns the adjacency matrix in an end-to-end fashion in contrast to the method proposed in [128] that thresholds the $\ell_2$ distance on pre-computed features. Our proposed method also uses global information about the patches while calculating the adjacency matrix. The idea behind using the global context is that connection between two same nodes/patches can differ for different WSIs; therefore, elements in the adjacency matrix should depend not only on the relation between two patches but also on the global context of all the patches.

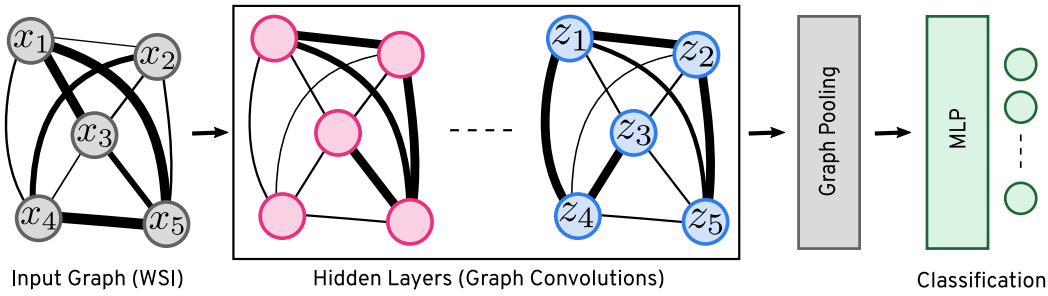



Figure 6.2: **Classification of a graph representing a WSI.** A fully connected graph representing a WSI is fed through a graph convolution layer to transform it into another fully-connected graph. After a series of transformations, the nodes of the final fully-connected graph are aggregated to a single condensed vector, which is fed to an MLP for classification purposes.

1. Let $W$ be a WSI and $w_1, w_2, \ldots w_n$ be its patches. Each patch $w_i$ is passed through a feature extraction layer to obtain corresponding feature representation $x_i$.

2. We use the theorem by Zaheer et al. [60] to obtain the global context from the features $x_i$. Feature vectors from all patches in the given WSI are pooled using a pooling function $\phi$ to get the global context vector $c$. Mathematically,

$$c = \phi(x_1, x_2, \ldots, x_n). \tag{6.1}$$

   Zaheer et al. showed that such a function can be used as an universal set approximator.

3. The global context vector $c$ is then concatenated to each feature vector $x_i$ to obtain concatenated feature vector $x_i'$ which is passed through MLP layers to obtain new feature vector $x_i^* \cdot x_i^*$ are the new features that contain information about the patch as well as the global context.

4. Features $x_i^*$ are stacked together to form a feature matrix $\mathbf{X}^*$ and passed through a cross-correlation layer to obtain adjacency matrix denoted by $\underset{n\times n}{\mathbf{A}}$ where each element $a_{ij}$ in $\mathbf{A}$ shows the degree of correlation between the patches $w_i$ and $w_j$. We use $a_{ij}$ to represent the edge weights between different nodes in the fully connected graph representation of a given WSI.

**Graph Convolution Layers.** Once we implemented the graph representation of the WSI, we experimented with two types of GCNN: ChebNets and GraphSAGE Convolution,

which are spectral and spatial methods, respectively. Each hidden layer in GCNN models the interaction between nodes and transforms the feature into another feature space. Finally, we have a graph pooling layer that transforms node features into a single vector representation. Thus, a WSI can now be represented by a condensed vector, which can be further used to do other tasks such as classification, image retrieval, etc.

### 6.3.2 Method Summary

Our proposed method can be used in any MIL framework. The general algorithm for solving MIL problems is as follows:

1. Consider each instance as a node and its corresponding feature as the node features.
2. The global context of the bag of instances is learned to calculate the adjacency matrix **A**.
3. A fully connected graph is constructed with each instance as a node and $a_{ij}$ in $A$ representing the edge weight between $V_i$ and $V_j$.
4. Graph convolution network is used to learn the representation of the graph, which is passed through a graph pooling layer to get a single feature vector representing the bag of instances.
5. The single feature vector from the graph can be used for classification or other learning tasks.

## 6.4 Experiments

We evaluated the performance of our model on two datasets i) a popular benchmark dataset for MIL called MUSK1 [129], and ii) LUAD vs LUSC dataset (introduced in Chapter 4). Our proposed method achieved a state-of-the-art accuracy of 92.6% on the MUSK1 dataset. We further used our model to discriminate between two sub-types of lung cancer—Lung Adenocarcinoma (LUAD) and Lung Squamous Cell Carcinoma (LUSC).

We used PyTorch Geometric library to implement graph convolution networks [130]. We used pre-trained DenseNet [95] to extract features from histopathology patches. We further feed DenseNet features through three dense layers with dropout ($p = 0.5$).

### 6.4.1 Toy Dataset - MUSK1 Dataset

It has 47 positive bags and 45 negative bags. Instances within a bag are different conformations of a molecule. The task is to predict whether new molecules will be musks or non-musks. We performed 10 fold cross-validation five times with different random seeds. We compared our approach with various other works in literature, as reported in Table 6.1. The miGraph [9] is based on kernel learning on graphs converted from the bag of instances. The latter two algorithms, MI-Net [131], and Attention-MIL [58], are based on DNN and use either pooling or attention mechanism to derive the bag embedding.

| **Algorithm** | **Accuracy** |
|---|---|
| mi-Graph [9] | 0.889 |
| MI-Net [131] | 0.887 |
| MI-Net with DS [131] | 0.894 |
| Attention-MIL [58] | 0.892 |
| Attention-MIL with gating [58] | 0.900 |
| Ming Tu et al. [128] | 0.917 |
| **Proposed Method** | **0.926** |

Table 6.1: Evaluation on MUSK1. The method achieved the highest among other MIL methods in literature.

### 6.4.2 LUAD vs LUSC Classification

We used the same dataset containing lung slides that has been used for this research as utilized in the last two chapters. We obtained the features through Yottixel (Chapter 3). We trained our model to classify bags as two cancer subtypes. The highest 5-fold classification AUC score achieved was 0.92, and the average AUC across all folds was 0.89. We performed cross-validation across different patients, i.e., training was performed using WSIs from a totally different set of patients than the testing. The results are reported in Table 6.2. We achieved state-of-the-art accuracy using the transfer learning scheme. In other words, we extracted patch features from an existing pre-trained network, and the feature extractor was not re-trained or fine-tuned during the training process. The Figure 6.5 shows the receiver operating curve (ROC) for one of the folds.

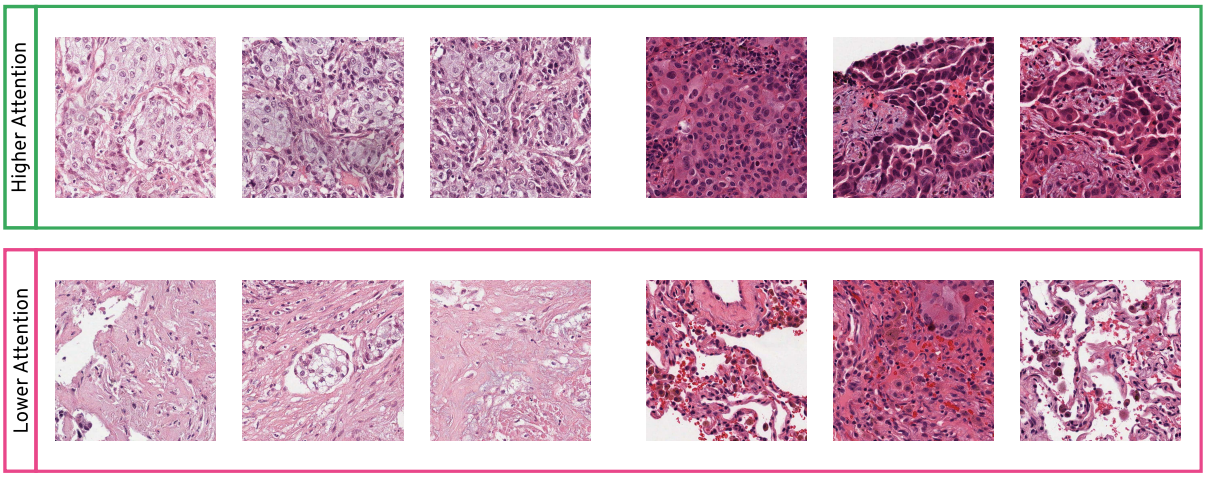



Figure 6.3: Inferring the attention values of the learned model. Six patches from two WSIs diagnosed with LUSC and LUAD, respectively. The six patches are selected, such that the first three (top row) are highly "attended" by the network, whereas the last three (bottom row) least attended. The first patch in the upper row is the most attended patch (more important) and the first patch in the lower row in the least attended patch (less important).

### 6.4.3 Model Inference

One of the primary obstacles for real-world application of deep learning models in computer-aided diagnosis is the black-box nature of the deep neural networks. Since our proposed architecture uses Global Attention Pooling [126], we can visualize the importance that our network gives to each patch for making the final prediction. Such visualization can provide more insight to pathologists regarding the model's internal decision making. The global attention pooling layer learns to map patches to "attention" values. The higher attention values signify that the model focuses more on those patches. We visualize the patches with high and low attention values in Figure 6.3. One of the practical applications of our approach would be for triaging. As new cases are queued for an expert's analysis, the CAD system could highlight the regions of interests and sort the cases based on the diagnostic urgency. We observe that patches with higher attention values generally contain more nuclei. As morphological features of nuclei are vitals for making diagnostic decisions [132], it is interesting to note this property is learned on its own by the network. Figure 6.4 shows the t-SNE plot of features vectors for some of the WSIs. It shows the clear distinction between the two cancer subtypes, further favoring the robustness of our method.

| Algorithm | AUC |
|---|---|
| Coudray et al. [92] | 0.85 |
| Khosravi et al. [114] | 0.83 |
| Yu et al. [115] | 0.75 |
| MEM (Chapter 4) | 0.85 |
| FocAtt (Chapter 5) | 0.87 |
| **Proposed method** | **0.89** |

Table 6.2: Performance of various methods for LUAD/LUSC predictions using transfer learning. Our results report the average of 5-fold accuracy values.

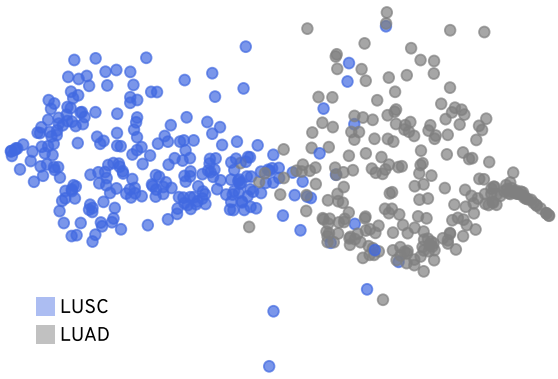



Figure 6.4: t-SNE visualization of feature vectors extracted after the Graph Pooling layer from different WSIs. The two distinct clusters for LUAD and LUSC demonstrate the efficacy of the proposed model for disease characterization in WSIs. The overlap of two clusters contain WSIs that are morphologically and visually similar.

### 6.4.4 Ablation Study

We tested our method with various different configurations for the TCGA dataset. We used two layers in Graph Convolution Network—ChebNet and SAGE Convolution. We found that ChebNet outperforms SAGE Convolution and also results in better generalization. Furthermore, we experimented with different numbers of filters in ChebNet, and also different pooling layers—global attention, mean, max, and sum pooling. We feed the pooled representation to two fully connected Dense layers to get the final classification between LUAD and LUSC. All the different permutations of various parameters result in 32 different configurations, the results for all these configurations are provided in Table 6.3. It should be noted that the results reported in the previous sections are based on Cheb-7 with mean pooling.

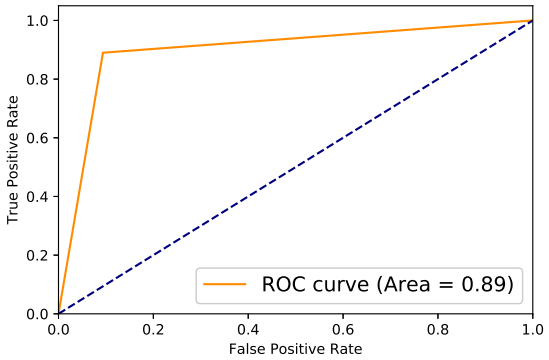



Figure 6.5: The ROC curve of prediction.

| configuration | **mean** | attention | max | add |
|---|---|---|---|---|
| **Cheb-7** | **0.8889** | 0.8853 | 0.7891 | 0.4929 |
| Cheb 3_BN | 0.8771 | 0.8635 | 0.8471 | 0.5018 |
| Cheb 5 | 0.8762 | 0.8830 | 0.8750 | 0.5082 |
| Cheb 3 | 0.8752 | 0.8735 | 0.8702 | 0.5090 |
| Cheb 5_BN | 0.8596 | 0.8542 | 0.7179 | 0.4707 |
| Cheb 7_BN | 0.7239 | 0.6306 | 0.5618 | 0.4930 |
| SAGE CONV_BN | 0.6866 | 0.5848 | 0.6281 | 0.5787 |
| SAGE CONV | 0.5784 | 0.6489 | 0.5389 | 0.5690 |

Table 6.3: Comparison of different network architecture and pooling method (attention, mean, max and sum pooling). **BN** stands for BatchNormalization [1], **Cheb** stands for Chebnet with corresponding filter size and **SAGE** stands for SAGE Convolution. The best performing configuration is Cheb-7 with mean pooling.

## 6.5 Summary

This chapter proposed a technique for representing a WSI as a fully-connected graph. The graph convolution networks are used to extract the features for classifying the lung WSIs into LUAD/LUSC. The results suggests that the proposed method achieves the better performance compared to the other two MIL approaches proposed in the last two chapters.

Furthermore, the proposed method is explainable and transparent. The potential of weak supervision for Yottixel is limited due to two factors (i) the computational and storage overhead, and (ii) restrictions around integration of medical data from diversified sources. The next part of thesis discusses the methods overcome these challenges. The proposed methods allow scaling Yottixel training over distributed nodes while maintaining patients' privacy.

# Part III

# Distributed & Privacy-Preserving Methods

Training of Yottixel is limited due the restricted scaling and access to the diversified dataset. Training Yottixel quickly becomes cumbersome and impractical as dataset grows bigger. This challenge necessitates to scale the training of Yottixel over distributed computers. However, algorithms dealing with medical data cannot be easily distributed due to confidentiality and privacy concerns around sharing medical data. Histopathology data cannot be aggregated and communicated across multiple institutions limiting research and model development progress. More robust and accurate models would result from sharing information among institutions while maintaining individual data privacy. Federated learning is a distributed learning system that allows multi-institutional collaborations on decentralized data while protecting the data privacy of each collaborator. The next two chapters discuss two different federated learning methods as reliable frameworks for distributed training of Yottixel on the decentralized data (protecting the privacy of each collaborator).

# Chapter 7

# Federated Averaging (FedAvg) for Histopathology Image Analysis

## 7.1 Prologue

This chapter is based on the following paper published during this Ph.D. research:

A. M. Adnan, **S. Kalra**, et al. *Federated learning and differential privacy for medical image analysis*. Nature Scientific Reports (2022)

Thus far, we have developed methodology for extracting patches (Chapter 3) from whole-slide images, and using weak-supervision methods to train Yottixel's deep networks (Chapter 4, Chapter 5, Chapter 6). However, the major limitation of Yottixel are (i) computational and storage overhead during training, and (ii) accesibility of diversified dataset. This chapter introduces a case-study on distributed training of Yottixel through federated learning that allows preserving the data privacy of involved participants, while scaling the training over distributed nodes. The distributed training enables sharing the resources efficiently among distributed participants thus enables efficient scaling.

**Summary.** This chapter conducts a case study of applying a differentially private federated learning framework for analysis of histopathology images. The most popular and common federated leaning scheme called *FedAvg* (Federated Averaging) has been utilized. The effects of IID and non-IID distributions along with the number of healthcare providers, i.e., hospitals and clinics, and the individual dataset sizes, using The Cancer Genome Atlas (TCGA) dataset, a public repository, to simulate a distributed environment. The empirical

comparison of the performance of private, distributed training to conventional training and demonstrate that distributed training can achieve similar performance with strong privacy guarantees. The effect of different source domains for histopathology images by evaluating the performance using external validation is also studied. The work indicates that differentially private federated learning is a viable and reliable framework for the collaborative development of machine learning models in medical image analysis.

## 7.2 Introduction & Background

Deep learning models are data-intensive, i.e., they often require millions of training examples to learn effectively. Medical images may contain confidential and sensitive information about patients that often cannot be shared outside the institutions of their origin, especially when complete de-identification cannot be guaranteed. The European General Data Protection Regulation (GDPR) and the United States Health Insurance Portability and Accountability Act (HIPAA) enforce guidelines and regulations for storing and exchanging personally identifiable data and health data. Ethical guidelines also encourage respecting privacy, that is, the ability to retain complete control and secrecy about one's personal information [81]. As a result, large archives of medical data from various consortia remain widely untapped sources of information. For instance, histopathology images cannot be collected and shared in large quantities due to the aforementioned regulations, as well as due to data size constraints given their high resolution and gigapixel nature. Without sufficient and diverse datasets, deep models trained on histopathology images from one hospital may fail to generalize well on data from a different hospital (out-of-distribution) [133, 69]. The existence of bias or the lack of diversity in images from a single institution brings about the need for a collaborative approach which does not require data centralization. One way to overcome this problem is by collaborative data sharing (CDS) or federated learning among different hospitals [134].

This chapter explores federated learning (FL) as a collaborative learning paradigm, in which models can be trained across several institutions without explicitly sharing patient data. The readers can refer to Chapter 2, §2.4 for more discussion on FL and differential privacy. This chapter shows that using federated learning with additional privacy preservation techniques can improve the performance of histopathology image analysis compared to training without collaboration. The benefits, drawbacks, potential weaknesses, as well as technical implementation considerations are discussed. Finally, lung cancer images from The Cancer Genome Atlas (TCGA) dataset [103] is to construct a simulated environment of several institutions to validate our approach.

## 7.3 Method

The proposed method (*local* to each client) consists of two steps, *bag preparation* and *Multiple-Instance Learning (MIL)*. In the first step, representative patches called mosaic are extracted from a full-resolution WSI using Yottixel's approach (Chapter 3). In the second step, we formulate the representation learning of WSIs as a set learning problem by applying a MEM model Chapter 4. The MEM model is locally trained through DP-SGD to provide quantitative privacy bounds, and the local MEM models are centrally aggregated through FedAvg. In this section, we discuss the bag preparation step and MIL. An overview of the proposed method is visualized in **??**.

### 7.3.1 Model Components

**Bag Preparation.** A patch selection method of Yottixel is used to extract representative patches (called *mosaics*) from each WSI. A sample WSI and its mosaic is illustrated in **??**. The steps involved in creations of a mosaic are: (i) removal of non-tissue regions using colour thresholding; (ii) grouping the remaining tissue-containing patches into a pre-set number of categories through a clustering algorithm; and (iii) randomly selecting a portion of all clustered patches (e.g., 10%) within each cluster, yielding a *mosaic*. The mosaic is transformed into a bag $X = \{x_1, \ldots, x_n\}$ for MIL, where $x_i$ is the feature vector of the $i^{\text{th}}$ patch, obtained through a pre-trained feature extractor network. We use a DenseNet model for the feature extractor [95]. Each patch in the mosaic has size 1000 $\times$ 1000 pixels at 20x magnification (0.5 mpp resolution). The complete approach for patch-extraction is discussed in Chapter 3.

**MIL Method.** The MEM model is used (Chapter 4) as a weak-supervision to extract feature vectors from mosaic. MEM consists of memory units composed within a memory block. A *memory block* is the main component of MEM and produces a permutation invariant representation from a input sequence. Multiple memory blocks can be stacked together for modeling complex relationships and dependencies in set data. The memory block is made of memory units and a bijective transformation (details in Chapter 4).

## 7.4 Experiments and Discussion

We validated the performance of FL for the classification of histopathology images using a simulated distributed environment and also using real-world hospital data. Previous studies

have mostly experimented with a fixed number of clients having similar distributions of data [81, 70, 135]. Since real-world data is not necessarily IID, it is important to study the effect of non-IID data on the performance of FL, specifically *FedAvg*. Furthermore, we provide a privacy analysis of the method through the differential privacy framework, suggesting that FL can outperform non-collaborative training while maintaining a strong privacy guarantee.

In the *first experiment series*, we vary the number of clients, with each client representing one hospital. To make our simulated environment better approach the non-IID real-world data, each client can have a different number of patients and a different distribution of cancer sub-types. In the *second experiment series*, we calculate the privacy bound of differentially private FL using real-world hospital data. We used the available attributes in TCGA to divide the dataset across the tissue origin site (hospital) and created four client datasets as shown in Table 7.4.

### 7.4.1 Dataset

We obtained 2,580 hematoxylin and eosin (H&E) stained WSIs of lung cancer from TCGA [51], comprising about 2 TB of data. The images were split into two groups of 1,806 training, and 774 testing samples WSIs (see Chapter 4). We transformed each raw image into a mosaic (see Chapter 3), and then into a bag of features $X$ using a pre-trained DenseNet [95]. From the data, we carried out two experiment series by varying the parameters of FedAvg, or by varying the data distributions across clients. These experiment series are discussed as follows.

### 7.4.2 Experiment Series 1 - Effect of Number of Clients and Data Distributions

We studied the effect of IID and non-IID distributions on the performance of FedAvg by randomly dividing the training images without replacement among different clients (hospitals). We also varied the number of clients ($n$) while keeping the total number of images fixed. IID data is generated by uniformly dividing each cancer sub-type, i.e. LUAD and LUSC, among different clients. For each cancer sub-type, a probability distribution is created by assigning a random value to each client and then dividing it by the total sum. Subsequently, images are divided among different clients by sampling from the probability distribution. FL achieves superior performance for both IID and non-IID distributions of data compared to non-collaborative training. FL performs comparably to centralized

| Data Distribution | Number of Clients $n$ | Accuracy | | |
|---|---|---|---|---|
| | | **Without FL** | **With FL** | **Centralized** |
| IID | 4 | $0.731 \pm 0.03$ | $0.824 \pm 0.02$ | $0.848 \pm 0.02$ |
| | 8 | $0.620 \pm 0.06$ | $0.780 \pm 0.05$ | |
| | 16 | $0.570 \pm 0.03$ | $0.726 \pm 0.06$ | |
| | 32 | $0.527 \pm 0.02$ | $0.641 \pm 0.09$ | |
| Non IID | 4 | $0.682 \pm 0.10$ | $0.824 \pm 0.01$ | $0.848 \pm 0.02$ |
| | 8 | $0.561 \pm 0.08$ | $0.823 \pm 0.05$ | |
| | 16 | $0.524 \pm 0.03$ | $0.750 \pm 0.06$ | |
| | 32 | $0.520 \pm 0.03$ | $0.550 \pm 0.20$ | |

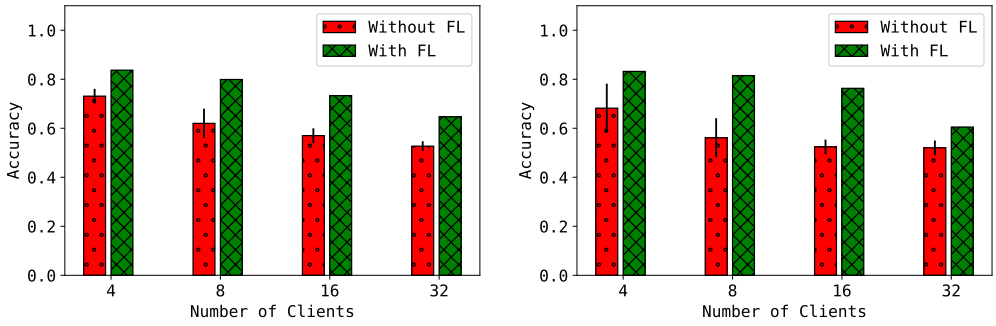
Table 7.1: Evaluation on different data distributions. Centralized accuracy denotes the accuracy when the data is centralized. The accuracy without FL is the mean and standard deviation of accuracy values across multiple clients without any collaboration. The accuracy with FL is the mean and standard deviation of the central model trained at the end of FL evaluated on each client dataset.

training for reasonably sized datasets ($n = 4, 8$). Interestingly, FL can achieve slightly better accuracy when trained on a non-IID data distribution. Results are summarized in Table 7.1 and Figure 7.1. The number of training samples for each client model is in Figure B.1 (Appendix B).

We compared the performance with and without FedAvg for each setting. In total we tested 16 experimental settings in Table 7.1. In each of the experiments, the server model trained using FedAvg outperformed the models trained using local client datasets, showing the advantage of collaboration. As the total dataset is divided into smaller partitions for more clients, both client and server model performances deteriorate. We used SGD optimizer with learning rate = 0.01. The local epoch for each client was set to 1 and the server model was trained for 250 communication rounds. We visualize the relative improvement of FedAvg in Table 7.1.

### 7.4.3 Experiment Series 2 - Real-World Dataset

In this experiment, we use Differential Private Federated Learning (DP-FL) to ensure data privacy. Differential Privacy (DP) was not considered in experiment series 1 since the objective was to study the effects of data size, distribution, and the number of clients on the performance of distributed learning/federated learning in general. In the second experiment series, we considered the effect of distributional differences from different source




(a) IID Data (b) Non-IID Data

Figure 7.1: Comparison of the mean accuracy across clients versus the accuracy of the central model trained with FL for the fabricated clients (not the real hospitals). The accuracy is computed on two types of data distribution settings across clients—IID and Non-IID.

hospitals, and a requirement to preserve privacy. Histopathology images can differ greatly, among others depending on the staining and imaging protocols of the source hospital. We selected seven hospitals from the TCGA dataset, four to act as clients in FL, and an additional three to provide externally collected data for model robustness testing. The distribution of images by hospital is described in Table 7.4. For each of the four clients, we divided their available images in an $80:20$ ratio for training and internal testing datasets, respectively. Then we combined the images from the remaining three hospitals into a single external validation dataset to study the effects of distributions shifts on FedAvg.

In this experiment, we use Differential Private Federated Learning (DP-FL) to ensure data privacy. Differential Privacy (DP) was not considered in experiment series 1 since the objective was to study the effects of data size, distribution, and the number of clients on the performance of distributed learning/federated learning in general. In experiment series 2, we compared the performance of privacy-preserving FL training with both centralized training and non-collaborative training. In the FL training, the four hospitals act as clients collaborating to train one central model. Performance is evaluated on each client's internal test set, as well as the external validation set. For comparison, we train a single model on the combined (centralized) training datasets which gives an upper bound on what could be achieved in the absence of privacy regulations. Finally, in the non-collaborative setting each client hospital trains their own model on only their own training dataset. We used DP-SGD to train the FL and combined models and computed the privacy guarantees $(\epsilon, \delta)$ using a Rényi DP accountant [136]. It was observed that the MEM model was sensitive to

| Gradient Clipping | Noise Multiplier | Privacy Budget ($\epsilon$) | Test Accuracy | External Accuracy |
|---|---|---|---|---|
| 1.0 | 4 | 2.90 | 0.815 | 0.740 |
| 1.5 | 4 | 3.26 | 0.759 | 0.719 |
| 2.0 | 4 | 3.89 | 0.765 | 0.732 |
| 1.0 | 6.0 | 2.34 | 0.832 | 0.737 |
| 1.0 | 2.0 | 10.01 | 0.782 | 0.748 |

Table 7.2: Ablation Study of DP hyperparameters (gradient clipping and noise multiplier)

| | Non-collaborative Training | | DP-FL Training | | FL Training | | Combined Training | |
|---|---|---|---|---|---|---|---|---|
| Source Hospital | Test | External | Test | External | Test | External | Test | External |
| International Genomics Consortium | 0.654 | 0.631 | 0.823 ± 0.01 | 0.707 ± 0.01 | 0.823 ± 0.01 | 0.741 ± 0.01 | 0.839 ±0.01 | 0.768 ± 0.003 |
| Indivumed | 0.648 | 0.556 | | | | | | |
| Asterand | 0.709 | 0.701 | | | | | | |
| John Hopkins | 0.681 | 0.600 | | | | | | |

Table 7.3: Evaluation of collaborative and non-collaborative learning on Test and External Datasets using DP-SGD, achieving privacy parameter $\epsilon = 2.90$ for $\delta = 0.0001$. For FL and Combined training we report the mean accuracy and standard deviation across the client's test datasets. On the external dataset we ran the experiments using three random initialization, and report the mean accuracy and standard deviation across them.

DP-SGD hyper parameters. We used a vectorized Adam optimizer[137] with the following hyper-parameter values[79]: epochs = 180, training set size = 705, batch size = 32, gradient clipping norm = 1.0, Gaussian noise standard deviation = 4.0, number of microbatches = 32, learning rate = $2 \times 10^{-5}$. Ablation study is provided in the Table 7.2.

As shown in Table 7.3, FL training achieves strong privacy bounds ($\epsilon = 2.90$ at $\delta = 0.0001$) with better performance than non-collaborative training, comparable to centralized training. This demonstrates that FL could be effectively used in clinical settings to ensure data privacy with no significant degradation in performance. Results are shown in Table 7.3. FedAvg achieves comparable performance to centralized training without explicitly sharing private data with strong privacy guarantees. Due to distribution shifts, accuracy decreases on external validation for both Federated Learning and centralized training. Therefore, we experimentally demonstrate the Federated Learning can be used for medical image analysis in real-world setting without explicitly sharing data, while achieving similar performance to centralized training with data sharing.

| Dataset Type | Source Hospital (Clients) | LUAD Images | LUSC Images | Total |
|---|---|---|---|---|
| Train/Test | International Genomics Consortium | 189 | 78 | 267 |
| | Indivumed | 94 | 117 | 211 |
| | Asterand | 90 | 117 | 207 |
| | Johns Hopkins | 121 | 78 | 199 |
| External | Christiana Healthcare | 169 | 54 | 223 |
| | Roswell Park | 35 | 75 | 110 |
| | Princess Margaret Hospital (Canada) | 0 | 52 | 52 |

Table 7.4: Source hospitals for test/train and external dataset and their data distribution.

## 7.5 Summary

There are two major limitations of Fedvg (i) assumption of a central entity during the training, and (ii) no personalization for individual client. The FedAvg requires a central entity for managing training states, and aggregating gradients, however a central entity is difficult to assign in the collaborations among hospitals or medical institutions. Furthermore, the incentive for the collaboration is the improvement of model's performance on the local test data (local to the participating hospital). However, in the case of FedAvg, one a single global model is learned which may not provide personalized improvement for individual participants. The research in the chapter suggests that private federated learning achieves a comparable result compared to conventional centralized training, and hence it could be considered for distributed training on medical data. But further improvements are required to overcome the discussed limitations. These limitations are resolved through a scheme proposed in the next chapter, called ProxyFL or a Proxy-based Federated Learning.

# Chapter 8

# ProxyFL: Decentralized Federated Learning through Proxy Model Sharing

## 8.1 Prologue

This chapter is based on the following paper published during this Ph.D. research:

A. **S. Kalra**, et al. *ProxyFL: Decentralized Federated Learning through Proxy Model Sharing*. Under Review (March 2022)

The last chapter presented a case-study on a federated learning method (FedAvg) for analysis of histopathology images. The experiments suggests that private federated learning achieves a comparable result compared to conventional centralized training, and hence it could be considered for distributed training on medical data. However, there are two major limitations of the FedAvg (i) assumption of a central entity during the training, and (ii) no personalization for individual client. To resolve these limitations, a decentralized version of federated learning is proposed in this chapter (i.e., no central entity and only peer-to-peer communication).

**Summary.** This chapter introduces a communication-efficient scheme for decentralized federated learning called *ProxyFL*, or proxy-based federated learning. Each participant in ProxyFL maintains two models, a private model, and a publicly shared proxy model designed to protect the participant's privacy. Proxy models allow efficient information

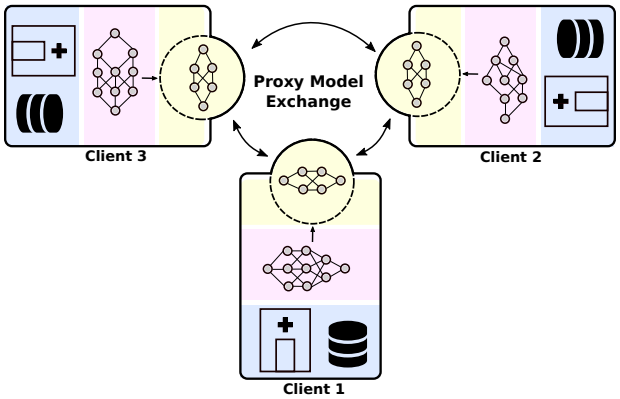



Figure 8.1: *ProxyFL* is a communication-efficient, decentralized federated learning method where each client (e.g., hospital) maintains a private model, a proxy model, and private data. During distributed training, the client communicates with others only by exchanging their proxy model which enables data and model autonomy. After training, a client's private model can be used for inference.

exchange among participants using the *PushSum method* without the need of a centralized server. The proposed method eliminates a significant drawback of canonical federated learning by allowing model heterogeneity; each participant can have a private model with any architecture. Furthermore, the proposed protocol for communication by proxy leads to stronger privacy guarantees using differential privacy analysis. Experiments on popular image datasets, and a pan-cancer diagnostic problem using over 30,000 high-quality gigapixel histology whole slide images, show that ProxyFL can outperform existing alternatives with much less communication overhead and stronger privacy.

## 8.2 Introduction

FedAvg (FL) is a distributed learning framework that was designed to train a model on data that could not be centralized [**?**]. It trains a model in a distributed manner directly on client devices where data is generated, and gradient updates are communicated back to the centralized server for aggregation. However, the canonical FL setting is not suited to the multi-institutional collaboration problem, as it involves a centralized third party that controls a single model. Considering a collaboration between hospitals, creating one central model may be undesirable. Each hospital may seek autonomy over its own model for regulatory compliance and tailoring to its own specialty.

While it is often claimed that FL provides improved privacy since raw data never leaves the client's device [**?**], it does not provide the guarantee of security that regulated institutions require. FL involves each client sending unaudited gradient updates to the central server, which is problematic since deep neural networks are capable of memorizing individual training examples, which may completely breach the client's privacy [138].

In contrast, meaningful and quantitative guarantees of privacy are provided by the differential privacy (DP) framework [75]. In DP, access to a database is only permitted through randomized queries in a way that obscures the presence of individual data points. More formally, let $\mathcal{D}$ represent a set of data points, and $M$ a probabilistic function, or *mechanism*, acting on databases. We say that the mechanism is $(\epsilon, \delta)$-*differentially private* if for all subsets of possible outputs $\mathcal{S} \subset \text{Range}(M)$, and for all pairs of databases $\mathcal{D}$ and $\mathcal{D}'$ that differ by one element,

$$\Pr[M(\mathcal{D}) \in \mathcal{S}] \leq \exp(\epsilon) \Pr[M(\mathcal{D}') \in \mathcal{S}] + \delta. \tag{8.1}$$

The spirit of this definition is that when one individual's data is added or removed from the database, the outcomes of a private mechanism should be largely unchanged in distribution. This will hold when $\epsilon$ and $\delta$ are small positive numbers. In this case an adversary would not be able to learn about the individual's data by observing the mechanism's output, hence, privacy is preserved. DP mechanisms satisfy several useful properties, including strong guarantees of privacy under composition, and post-processing [77, 76]. These properties make DP a suitable solution for ensuring data privacy in a collaborative FL setting.

In this chapter, we propose proxy-based federated learning, or *ProxyFL*, for decentralized collaboration between institutions which enables training of high-performance and robust models, without sacrificing data privacy or communication efficiency. The contributions are: (i) a method for decentralized FL in multi-institutional collaborations that is adapted to heterogeneous data sources, and preserves model autonomy for each participant; (ii) incorporation of DP for rigorous privacy guarantees; (iii) analysis and improvement of the communication overhead required to collaborate.

## 8.3 Related Work

**Decentralized FL for highly regulated domains.** Unlike centralized FL [**?**, 139] where federated clients coordinate to train a centralized model that can be utilized by everyone as a service, decentralized FL is more suitable for multi-institutional collaborations due to regulatory constraints. The main challenge of decentralized FL is to develop a protocol

that allows information passing in a peer-to-peer manner. Gossip protocols [140] can be used for efficient communication and information sharing [141, 142]. There are different forms of information being exchanged in the literature, including model weights [143, 144], knowledge representations [145] or model outputs [146, 147]. However, unlike our method, none of these protocols provides a guarantee of privacy for participants, and therefore cannot be used safely in highly regulated domains.

**Mutual learning.** Each client in our ProxyFL has two models that serve different purposes. They are trained using a DP variant of deep mutual learning (DML) [148] which is an approach for mutual knowledge transfer. DML compares favourably to knowledge distillation between a pre-trained teacher and a typically smaller student [149] since it allows training both models simultaneously from scratch, and provides beneficial information to both models. Federated Mutual Learning (FML) [150] introduces a meme model that resembles our proxy model, which is also trained mutually with each client's private model, but is aggregated at a central server. However, FML is not well-suited to the multi-institutional collaboration setting as it is centralized and provides no privacy guarantee to clients.

**Differential privacy in FL.** Although raw data never leaves client devices, FL is still susceptible to breaches of privacy [74, 73]. DP has been combined with FL to train centralized models with a guarantee of privacy for all clients that participate [80]. By ensuring that gradient updates are not overly reliant on the information in any single training example, gradients can be aggregated centrally with a DP guarantee [151]. We take inspiration from these ideas for ProxyFL.

**Computational pathology.** The main application domain considered in this work is computational pathology. Various articles have emphasized the need for privacy-preserving FL when facing large-scale computational pathology workloads. [152] and [153] used FL for medical image augmentation and segmentation. Their method used a centralized server to aggregate selective weight updates that were treated in a DP framework, but they did not account for the total privacy budget expended over the training procedure. [84] and [**?**] built medical image classification models with FL, and added noise to model weights for privacy. However, model weights have unbounded sensitivity, so no meaningful DP guarantee is achieved with these techniques.

## 8.4 Method - ProxyFL

*ProxyFL*, or proxy-based federated learning is our proposed approach for decentralized federated learning. It is designed for multi-institutional collaborations in highly-regulated

domains, and as such incorporates quantitative privacy guarantees with efficient communication.

### 8.4.1 Problem Formulation & Overview

We consider the decentralized FL setting involving a set of clients $\mathcal{K}$, each with a local data distribution $\mathcal{D}_k, \forall k \in \mathcal{K}$. Every client maintains a *private model* $f_{\phi_k} : \mathcal{X} \to \mathcal{Y}$ with parameters $\boldsymbol{\phi}_k$, where $\mathcal{X}, \mathcal{Y}$ are the input/output spaces respectively. In this work, we assume that all private models have the same input/output specifications, but may have different structures[1]. The goal is to train the private models collectively so that each generalizes well on the joint data distribution.

There are three major challenges in this setting: (i) The clients may not want to reveal their private model's structure and parameters to others. Revealing model structure can expose proprietary information, increase the risk of adversarial attacks [154], and can leak private information about the local datasets [155]. (ii) In addition to model heterogeneity, the clients may not want to rely on a third party to manage a shared model, which precludes centralized model averaging schemes. (iii) Information sharing must be efficient, robust, and peer-to-peer. To address the above challenges, we introduce an additional *proxy model* $h_{\boldsymbol{\theta}_k} : \mathcal{X} \to \mathcal{Y}$ for each client with parameters $\boldsymbol{\theta}_k$. It serves as an interface between the client and the outside world. As part of the communication protocol, all clients agree on a common proxy model architecture for compatibility.

In every round of ProxyFL, each client trains its private and proxy models jointly so that they can benefit from one another. With differentially private training, the proxy can extract useful information from private data, ready to be shared with other clients without violating privacy constraints. Then, each client sends its proxy to its out-neighbors and receives new proxies from its in-neighbors according to a communication graph, specified by an adjacency matrix $P$ and de-biasing weights $\mathbf{w}$. Finally, each client aggregates the proxies they received, and *replaces* their current proxy. The overall procedure is shown in Figure 8.1 and Algorithm 1. We discuss each step in detail in the subsequent subsections.

### 8.4.2 Training Objectives

For concreteness, we consider classification tasks. To train the private and proxy models at the start of each round of training, we apply a variant of DML [148]. Specifically, when

[1]This can be further relaxed by including client-specific input/output adaptation layers.

**Algorithm 1** ProxyFL

**Require**: Proxy parameters $\boldsymbol{\theta}_k^{(0)}$, private parameters $\boldsymbol{\phi}_k^{(0)}$, de-biasing weight $w_k^{(0)}$ for client $k$, DML weights $\alpha, \beta \in (0,1)$, learning rate $\eta > 0$, adjacency matrix $P^{(t)}$

**for** each round $t = 0, \ldots, T-1$ at client $k \in \mathcal{K}$ **do**
  **for** each local optimization step **do**
    Sample mini-batch $\mathcal{B}_k = \{(\mathbf{x}_i, y_i)\}_{i=1}^{B}$ from $\mathcal{D}_k$
    Update local proxy and private models:

$$\boldsymbol{\theta}_k^{(t)} \leftarrow \boldsymbol{\theta}_k^{(t)} - \eta \widetilde{\nabla} \widehat{\mathcal{L}}_{\boldsymbol{\theta}_k}(\mathcal{B}_k) \quad \text{\# DP update}$$

$$\boldsymbol{\phi}_k^{(t)} \leftarrow \boldsymbol{\phi}_k^{(t)} - \eta \nabla \widehat{\mathcal{L}}_{\boldsymbol{\phi}_k}(\mathcal{B}_k) \quad \text{\# non-DP update}$$

  **end for**
  $\boldsymbol{\phi}_k^{(t+1)} \leftarrow \boldsymbol{\phi}_k^{(t)}$
  Send $\left(P_{k',k}^{(t)} \boldsymbol{\theta}_k^{(t)}, P_{k',k}^{(t)} w_k^{(t)}\right)$ to out-neighbors;
  receive $\left(P_{k,k'}^{(t)} \boldsymbol{\theta}_{k'}^{(t)}, P_{k,k'}^{(t)} w_{k'}^{(t)}\right)$ from in-neighbors
  Update local proxy $\boldsymbol{\theta}_k^{(t+1)} \leftarrow \sum_{k'} P_{k,k'}^{(t)} \boldsymbol{\theta}_{k'}^{(t)}$
  Update de-bias weight $w_k^{(t+1)} \leftarrow \sum_{k'} P_{k,k'}^{(t)} w_{k'}^{(t)}$
  De-bias $\boldsymbol{\theta}_k^{(t+1)} \leftarrow \boldsymbol{\theta}_k^{(t+1)} / w_k^{(t+1)}$
**end for**
**return** $\boldsymbol{\theta}_k^{(T)}$, $\boldsymbol{\phi}_k^{(T)}$

training the private model for client $k$, in addition to the cross-entropy loss (CE)

$$\mathcal{L}_{\mathrm{CE}}(f_{\boldsymbol{\phi}_k}) := \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_k} \mathrm{CE}[f_{\boldsymbol{\phi}_k}(\mathbf{x}) \| y], \tag{8.2}$$

DML adds a KL divergence loss (KL)

$$\mathcal{L}_{\mathrm{KL}}(f_{\boldsymbol{\phi}_k}; h_{\boldsymbol{\theta}_k}) := \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_k} \mathrm{KL}[f_{\boldsymbol{\phi}_k}(\mathbf{x}) \| h_{\boldsymbol{\theta}_k}(\mathbf{x})], \tag{8.3}$$

so that the private model can also learn from the current proxy model. The objective for learning the private model is given by

$$\mathcal{L}_{\boldsymbol{\phi}_k} := (1-\alpha) \cdot \mathcal{L}_{\mathrm{CE}}(f_{\boldsymbol{\phi}_k}) + \alpha \cdot \mathcal{L}_{\mathrm{KL}}(f_{\boldsymbol{\phi}_k}; h_{\boldsymbol{\theta}_k}), \tag{8.4}$$

where $\alpha \in (0,1)$ balances between the two losses. The objective for the proxy model is similarly defined as

$$\mathcal{L}_{\boldsymbol{\theta}_k} := (1-\beta) \cdot \mathcal{L}_{\mathrm{CE}}(h_{\boldsymbol{\theta}_k}) + \beta \cdot \mathcal{L}_{\mathrm{KL}}(h_{\boldsymbol{\theta}_k}; f_{\boldsymbol{\phi}_k}). \tag{8.5}$$

where $\beta \in (0, 1)$. As in DML, we alternate stochastic gradient steps between the private and proxy models.

In our context, mini-batches are sampled from the client's private dataset. Releasing the proxy model to other clients risks revealing that private information. Therefore, each client uses differentially private stochastic gradient descent (DP-SGD) [151] when training the proxy (but not the private model). Let $\mathcal{B}_k = \{(\mathbf{x}_i, y_i)\}_{i=1}^{B}$ denote a mini-batch sampled from $\mathcal{D}_k$. The stochastic gradient is $\nabla\widehat{\mathcal{L}}_{\phi_k}(\mathcal{B}_k) := \frac{1}{B}\sum_{i=1}^{B} \mathbf{g}_{\phi_k}^{(i)}$ where

$$\begin{aligned}\mathbf{g}_{\phi_k}^{(i)} := (1-\alpha)\nabla_{\phi_k}\mathrm{CE}[f_{\phi_k}(\mathbf{x}_i)\|y_i] \\ + \alpha\nabla_{\phi_k}\mathrm{KL}[f_{\phi_k}(\mathbf{x}_i)\|h_{\boldsymbol{\theta}_k}(\mathbf{x}_i)].\end{aligned} \tag{8.6}$$

$\nabla\widehat{\mathcal{L}}_{\boldsymbol{\theta}_k}(\mathcal{B}_k)$ and $\mathbf{g}_{\boldsymbol{\theta}_k}^{(i)}$ are similarly defined for the proxy. To perform DP training for the proxy, the per-example gradient is clipped, then aggregated over the mini-batch, and finally Gaussian noise is added [151]:

$$\begin{aligned}\overline{\mathbf{g}}_{\boldsymbol{\theta}_k}^{(i)} &:= \mathbf{g}_{\boldsymbol{\theta}_k}^{(i)} \;/\; \max\left(1, \|\overline{\mathbf{g}}_{\boldsymbol{\theta}_k}^{(i)}\|_2/C\right), \\ \widetilde{\nabla}\widehat{\mathcal{L}}_{\boldsymbol{\theta}_k}(\mathcal{B}_k) &:= \frac{1}{B}\left(\sum_{i=1}^{B}\overline{\mathbf{g}}_{\boldsymbol{\theta}_k}^{(i)} + \mathcal{N}(0, \sigma^2 C^2 I)\right),\end{aligned} \tag{8.7}$$

where $C > 0$ is the clipping threshold and $\sigma > 0$ is the noise level (see Lines 2–5 in 1).

### 8.4.3 Privacy Guarantee

The proxy model is the only entity that a client reveals, so each client must ensure this sharing does not compromise the privacy of their data. Since arbitrary post-processing on a DP-mechanism does not weaken its $(\epsilon, \delta)$ guarantee [77], it is safe to release the proxy as long as it was trained via a DP-mechanism. DP-SGD as defined in Equation 8.7 is based on the Gaussian mechanism [78] which meets the requirement of Equation 8.1 by adding Gaussian noise to the outputs of a function $f$ with bounded sensitivity $C$ in $L_2$ norm,

$$M(x) = f(x) + \mathcal{N}(0, \sigma^2 C^2 I). \tag{8.8}$$

DP-SGD simply takes $f(x)$ to be the stochastic gradient update, with clipping to ensure bounded sensitivity.

Every application of the DP-SGD step incurs a privacy cost related to the clipping threshold $C$, and noise level $\sigma$. A strong bound on the total privacy cost over many applications of DP-SGD is obtained by using the framework of Rényi differential privacy [136, 156]

to track privacy under compositions of DP-SGD, then convert the result to the language of $(\epsilon, \delta)$-DP as in Equation 8.1 [157].

Finally, privacy guarantees are tracked on a per-client basis. In a multi-institutional collaboration, every client has an obligation to protect the privacy of the data it has collected. Hence, each client individually tracks the parameters ($\epsilon$, $\delta$) for its own proxy model training, and can drop out of the protocol when its prespecified privacy budget is reached. Throughout the chapter we specify $\delta$ based on the dataset size, and compute $\epsilon$.

### 8.4.4 Communication Efficiency & Robustness

The proxies serve as interfaces for information transfer and must be locally aggregated in a way that facilitates efficient learning among clients. One may use a central parameter server to compute the average of the proxies, similar to [150]. However, this will incur a communication cost that grows linearly in the number of clients, and is not decentralized. We propose to apply the PushSum scheme [140, 142] to exchange proxies among clients.

Let $\Theta^{(t)} \in \mathbb{R}^{|\mathcal{K}| \times d_\theta}$ represent the stacked proxies at round $t$, where the rows are the proxy parameters $\boldsymbol{\theta}_k^{(t)}, \forall k \in \mathcal{K}$. We use $P^{(t)} \in \mathbb{R}^{|\mathcal{K}| \times |\mathcal{K}|}$ to denote the weighted adjacency matrix representing the graph topology at round $t$, where $P_{k,k'}^{(t)} \neq 0$ indicates that client $k$ receives the proxy from client $k'$. Note that $P^{(t)}$ needs to be column-stochastic, but need not be symmetric (bidirectional communication) nor time-invariant (across rounds). Such a $P^{(t)}$ will ensure efficient communication when it is sparse. The communication can also handle asymmetrical connections such as different upload/download speeds, and can adapt to clients joining or dropping out since it is time-varying.

With these notations, every round of communication can be concisely written as $\Theta^{(t+1)} = P^{(t)}\Theta^{(t)}$. Under certain mixing conditions [158], it can be shown that $\lim_{T\to\infty} \prod_{t=0}^{T} P^{(t)} = \boldsymbol{\pi}\mathbf{1}^\top$, where $\boldsymbol{\pi}$ is the limiting distribution of the Markov chain and $\mathbf{1}$ is a vector of all ones. Suppose for now that there is no training for the proxies between rounds, i.e., updates to the proxies are due to communication and replacement only. In the limit, $\boldsymbol{\theta}_k$ will converge to $\boldsymbol{\theta}_k^{(\infty)} = \pi_k \sum_{k' \in \mathcal{K}} \boldsymbol{\theta}_{k'}^{(0)}$. To mimic model averaging (i.e., computing $\frac{1}{|\mathcal{K}|} \sum_{k' \in \mathcal{K}} \boldsymbol{\theta}_{k'}^{(0)}$), the bias introduced by $\pi_k$ must be corrected. This can be achieved by having the clients maintain another set of weights $\mathbf{w} \in \mathbb{R}^{|\mathcal{K}|}$ with initial values $\mathbf{w}^{(0)} = \mathbf{1}$. By communicating $\mathbf{w}^{(t+1)} = P^{(t)}\mathbf{w}^{(t)}$, we can see that $\mathbf{w}^{(\infty)} = \boldsymbol{\pi}\mathbf{1}^\top\mathbf{w}^{(0)} = |\mathcal{K}|\boldsymbol{\pi}$. As a result, the de-biased average is given by $\boldsymbol{\theta}_k^{(\infty)}/w_k^{(\infty)} = \frac{1}{|\mathcal{K}|} \sum_{k' \in \mathcal{K}} \boldsymbol{\theta}_{k'}^{(0)}$.

Finally, recall that the proxies are trained locally in each round. Instead of running the communication to convergence for proxy averaging, we alternate between training (Lines 2–5 in 1) and communicating (Lines 7–10) proxies, similar to [159].

| Client | # Slides | | |
|---|---|---|---|
| | Training | Test | Total |
| C1: University of Pittsburgh | 1,310 | 562 | 1,872 |
| C2: Indivumed | 1,004 | 431 | 1,435 |
| C3: Asterand | 818 | 351 | 1,169 |
| C3: MSKCC | 798 | 342 | 1,140 |
| **Total** | | | 5,616 |

Table 8.1: Distribution of WSIs across 4 different participating clients. The total of 5,616 WSIs accounts for around 6 TB of imaging data.

## 8.5 Experiments

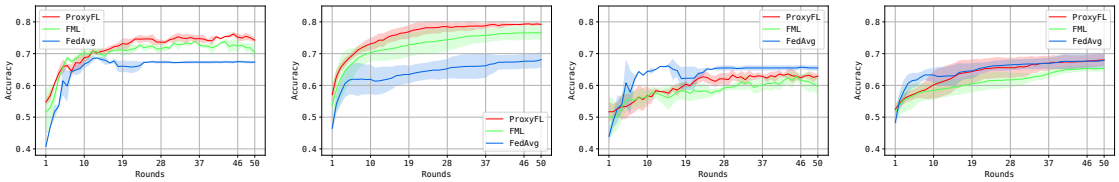
### 8.5.1 Dataset

In this experiment, we evaluated ProxyFL on a multi-origin real-world dataset. We considered the largest public archive of whole-slide images (WSIs), namely The Cancer Genome Atlas (TCGA) [103]. TCGA provides about 30,000 H&E stained WSIs originating from various institutions, distributed across multiple primary diagnoses. The client data for this study was derived from TCGA by splitting it across four major institutions: i) University of Pittsburgh, ii) Indivumed, iii) Asterand, and iv) Memorial Sloan Kettering Cancer Center (MSKCC). The data splits for each participating client are described in Table 8.1. The total data size is around 6 TB for all hospitals.

**WSI pre-processing & model setup.** Each WSI is an extremely large image (more than 50,000 x 50,000 pixels with a size often much larger than several hundred MBs), and cannot be directly processed by a CNN. In order to classify a WSI, we divided it into a small number of representative patches called a mosaic, using the techniques from [2]. The mosaic patches were then converted into feature vectors using a pre-trained DenseNet [95]. Each WSI corresponds to a set of features; these sets are then used for training a classifier based on the DeepSet architecture [60]. In the context of ProxyFL, both the private and proxy models are DeepSet-based.

### 8.5.2 Results

**Experimental setup.** The experiments were conducted using four V100 GPUs. Three FL methods were compared: ProxyFL, FML, and FedAvg. In each scenario, training was

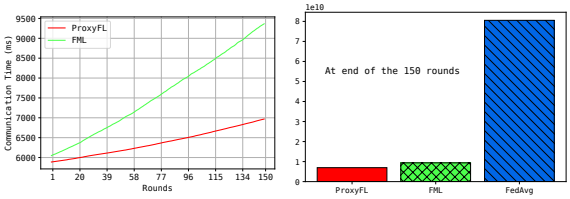



(a) Internal test data (Left: $\sigma = 1.4$; Right: $\sigma = 0.7$)

(b) External test data (Left: $\sigma = 1.4$; Right: $\sigma = 0.7$)

| Client | Privacy Guarantees ($C = 0.7, \delta = 10^{-4}$) | |
|---|---|---|
| | Strong ($\sigma = 1.4$) | Weak ($\sigma = 0.7$) |
| C1 | $\epsilon = 1.56$ | $\epsilon = 4.52$ |
| C2 | $\epsilon = 1.81$ | $\epsilon = 5.31$ |
| C3 | $\epsilon = 2.04$ | $\epsilon = 6.02$ |
| C4 | $\epsilon = 2.07$ | $\epsilon = 6.11$ |

(c) Privacy Guarantees




(d) Communication time

Figure 8.2: Performance of ProxyFL, FML, and FedAvg on the histopathology dataset involving four hospitals. The mean accuracy and standard deviation of clients on both internal and external data is recorded at the end of each round for two DP settings, and is presented in (a) and (b) respectively. Three random seeds were used. As expected, stronger privacy results in the lower overall accuracy for the internal dataset, but ProxyFL and FML show commensurate changes. Privacy gurantees for each method are listed in (c), computed based on the training set sizes in Table 8.1. The communication time per client for 150 rounds of training is shown in (d). FedAvg has less efficient communication because it exchanges the larger private model whereas ProxyFL and FML exchange the lightweight proxy models.

conducted for 50 rounds with a mini-batch size of 16. All methods were tested with two DP settings, one with strong privacy $\sigma = 1.4$, and the other with comparatively weak privacy $\sigma = 0.7$, both with $C = 0.7$. The client-level privacy guarantees for the two DP settings are provided in Figure 8.2c, computed based on the training set sizes in Table 8.1. FedAvg and the proxy models used the DP-SGD optimizer with learning rate 0.01, whereas the private models used the Adam with learning rate 0.001. For ProxyFL and FML the private models are used to compute the accuracy values, whereas the central model is used in the case of FedAvg.

Performance was computed based on two test datasets – internal and external. Both datasets are local to the clients. Internal test data is sampled from the same distribution as the client's private training data, whereas external test data comes from other clients involved in the federated training, and hence a different institution entirely[2]. The 32 unique primary diagnoses in the dataset can be further grouped into 13 tumor types[3]. The tumor type of a WSI is generally known at inference time, so the objective is to predict the cancer sub-type. We evaluated our method by its accuracy of classifying a cancer sub-type (primary diagnosis) of a WSI given that its tumor type is already known.

**Discussion.** The sub-type classification results for internal and external data on two different DP settings (strong and weak privacy) for each method are reported in Figure 8.2a and Figure 8.2b. ProxyFL achieves overall higher accuracy compared to FML and FedAvg on the internal test data for both privacy settings. For the external test data, all three methods perform similar to each other with FedAvg slightly ahead when using stronger privacy. ProxyFL has noticeably better convergence compared to FML as shown by the lower variance in both privacy settings. When strong privacy is used, the FedAvg central model has converged by around the $25^{\text{th}}$ round showing no improvement in the performance across both test datasets. Both ProxyFL and FML are more communication efficient than FedAvg because they exchange lightweight proxy models rather than the larger private models (Figure 8.2d), but ProxyFL has the lowest communication overhead due to using fewer model exchanges.

[2]For the external test data, we only use examples with a primary diagnosis present in the client's local data. If access to this type of external data is not possible due to privacy concerns, model performance could be validated on public external data.

[3]Tumor types are from Tables 3 and 4 of [122].

## 8.6 Summary

This chapter proposed a novel decentralized federated learning scheme, *ProxyFL*, for multi-institutional collaborations without revealing participants' private data. ProxyFL preserves data and model privacy, and provides a decentralized and communication-efficient mechanism for distributed training. Experiments suggest that ProxyFL is competitive compared to other baselines in terms of model accuracy, communication efficiency, and privacy preservation. ProxyFL makes Yottixel a collaborative, and holistic framework for analysis and representation of histopathology images.

# Chapter 9

# Conclusions & Future Work

The goal of this research was to develop a representation learning framework for histopathology images that enables image searching in a large archive (with a reasonable speed and accuracy). The major challenge in processing pathology images is their extremely high dimensionality, often one or more gigabytes in size. In this regard, in Chapter 3, the challenge of processing the large dimensionality was addressed through a proposed framework, Yottixel— that systematically divided a large histopathology image into a set of representative patches, called *mosaic*. Chapter 4, Chapter 5, Chapter 6, contributed towards resolving a major shortcoming of Yottixel by incorporating different ways of weak-supervision to train the feature extraction backbone of Yottixel. The weak-supervision in form of multi-instance learning enabled Yottixel to compute more discriminative representations thereby improving its performance in cancer sub-type classification. As Yottixel became a trainable approach, in the last two chapters (Chapter 7, Chapter 8), methods were developed to enable private and distributed training of Yottixel through federated learning. The private and distribute training allows us to integrate diverse medical datasets. Furthermore, it enables scaling Yottixel capabilities over multiple hospitals. In this final chapter of the thesis, main contributions are summarized and some promising directions for future research in the field of representation of histopathology images are put forward.

## 9.1 Highlights of Thesis Contributions

The main contributions of this thesis can be summarized as follows:

- **Representing a histopathology image as a set of representative patches.**

Chapter 3 discussed a method, called Yottixel that resolved a challenge of processing gigapixel histopathology images by dividing them into a mosaic of patches. Existing deep learning methods are unable to process these images in their entirety. Yottixel divides a histopathology image into a set of representative patches (called *mosaic*) which enables the incorporation of existing deep learning methods without the requirement of massive computational and storage overhead. The initial concept of Yottixel was completely unsupervised, and achieved good retrieval accuracy in a large archive of whole-slide images.

- **Incorporating slide-level label information for more discriminative features.** Chapter 4, Chapter 5, Chapter 6 proposed three different multi-instance learning methods to incorporate label information of histopathology images to learn more discriminative representations. Generally, a label (e.g., a cancer subtype) is associated with an entire histopathology image without access to any regional- or pixel-level annotations. The problem at hand is different from traditional supervised machine learning that operates on a single instance and its associated target label. The weak-supervision through multi-instance learning (MIL) is used for training Yottixel using a mosaic, and a target label pairs (cancer sub-type). Three different MIL methods have been proposed that not only improve the Yottixel for the cancer subtype classification, but also enable visualizing the patches that are deemed important for the given prediction.

- **Private and distributed training of Yottixel.** Yottixel can be trained through weakly-supervised, multi-instance learning based approaches. However, its functionality is limited due the restricted scaling of its training and the access to the diversified dataset. Training Yottixel on a large dataset is a time-consuming task, and it becomes quickly cumbersome and impractical as dataset grows in size. Chapter 7, Chapter 8 represent two different federated learning (FL) methods as a paradigm for distributed and collaborative learning framework to train Yottixel across multiple hospitals while respecting patient privacy. The experimental results suggest that private federated learning achieves a comparable result compared to conventional centralized training, and hence it could be considered for distributed training on medical data. Furthermore, a new scheme for FL is proposed called ProxyFL (Chapter 8), especially curated for institutional collaborations in training deep models.

### 9.1.1 Limitations

The experimental results demonstrated that the proposed methods produce discriminative representations of histopathology images. However, there are some limitations accompanied with these methods which should be considered.

- **Yottixel.** One of the major limitations of Yottixel is its inability to learn to produce mosaic patches. The mosaic patches are obtained in an completely unsupervised manner. Sometime, thge mosaic of a histopathology image misses important regions thus adversely affecting the final representation. Performance of Yottixel is heavily dependent on its hyperparameters, however Chapter 3 provides good default values that have been determined empirically.
- **ProxyFL.** The work focused on the privacy aspects of the ProxyFL protocol, but not on its security. We have assumed that all clients collaborate in good faith. It provides limited handling of malicious participants. The participants may not have malicious intent, however their local data distribution may be very different from other clients, comprising the model's performance for others.

## 9.2 Future Work

The proposed methods in this thesis open several new directions for future work. Below, the main topics are described.

### 9.2.1 Reinforcement Learning for Mosaic Extraction

The mosaic of a whole-slide image is extracted in a completely unsupervised manner as described in Chapter 3. It uses color, and spatial clustering to extract the mosaic. The extracted mosaic becomes an independent representation of a given histopathology image, and it is kept disengaged from the weakly supervised learning methods presented in the Chapter 4, Chapter 5, Chapter 6. If the diagnostically relevant patches are missing in the mosaic, the learning approach will suffer in performance or will overfit on non-generalizing or irrelevant features. An important future direction to expand Yottixel would be develop a trainable mosaic extraction method. A mosaic extraction approach that can be tied into the end-to-end learning. In the other words, the weakly supervised methods are not only able to enhance the feature extraction capabilities of Yottixel but also alter the mosaic

extraction policy. In this context, reinforcement learning (RL) is a relevant avenue to explore. Qaiser & Rajpoot have proposed a RL-based method for automated scoring of IHC stained HER2 slides of breast cancer [160]. Unlike fully supervised models that process all the regions of a given input image, the proposed model treats IHC scoring as a sequential selection task and effectively localizes diagnostically relevant regions by deciding "where to see". The proposed model carries the potential to solve other histology image analysis problems where it is difficult to get precise pixel-level annotations. A similar approach may also eventually assist the pathologist in automated localization and classification of potential ROIs in both H&E and IHC stained histology images. In RL, a reward function guides the learning, an learning agent performs the actions that maximize the future rewards from the system. In case of Yottixel, reward function can be modeled as the "correct" retrievals (the same cancer sub-type).

### 9.2.2 Multi-Modal Deep Learning

In Chapter 4, Chapter 5, and Chapter 6, methods were developed to enhance the Yottixel features through the weakly supervised training on the primary diagnosis labels associated with histopathology images. In Chapter 5, a fine-tuning method was developed that utilized the hierarchical relationship among anatomical site and primary diagnosis. We notice that exposing more data, their relationships to a deep model results in the better performance. A histopathology image has vast amount of data associated to it, the most important one is the pathology report that summarizes the opinion of a given case by an expert. A relevant future direction would be exploring methods for unification of different data modalities, such as pathology reports, genome sequence data, clinical data for training the Yottixel. Nevertheless, data from different modalities potentially have complementary information since they reflect the same patient (case) from different perspectives and may influence each other (i.e., the gene alternation may induce the cell morphologic changes in tumor regions [161]). Therefore, a method that can process the misaligned information from multiple modalities—image, text, gene data, tabular clinical data, can make full use of the potential complementary information to generalize well on a given task. The attention-based multi-instance learning as developed in this thesis are weakly supervised learning approaches that effectively learns to distinguish between unrelated patches and discriminative patches. At present, MIL is mainly based on the guidance of the knowledge from the imaging modality alone [161] and overlooks useful supplementary knowledge from real scenarios to guide the instance-level attention. It's worth exploring how to borrow useful knowledge from another modality (i.e., genome/exome data, or pathologists' reports

data) to guide the MIL in imaging modality to optimally distribute the instance-level attention. One such method in literature is MMMI learning [161] to overcome the challenges of fusing histopathological images and tabular clinical data to predict breast cancer. A strategy for multimodal data integration with application to biomarkers identification in spinocerebellar ataxia is presented in [162].

### 9.2.3 Semi-Supervised Deep Learning

The most recent success of deep learning is in the area of supervised learning and is made possible by data sets with millions of labeled images [163, 164]. While unsupervised feature learning is an active area of research, its use in a classification model does not yet compare with supervised feature learning. This is particularly unfortunate in medical applications for which gathering that much finely-annotated data is cost and time prohibitive. In the case of histopathology, there is a great deal of unlabeled data available, making unsupervised or semi-supervised feature learning attractive. This thesis explored weakly-supervised methods where each input instance would have some indirect relationship to the output or target label. In the case of the problem explored in this thesis, multiple input instances were associated to a single target or output label. A class of weakly supervised methods called multi-instance learning (MIL) was utilized to approach the given problem. However, many times, labels are simply not available for given whole-slide images, perhaps these images are created as a part of digitization effort from an institution or images are in public domain where label information was never released. With limited annotation, deep learning model training often covers only a limited fraction of the histopathology data space. There could exist considerable discrepancy (in appearance) between the labeled and unlabeled sets. Thus, the trained deep models are at risk of over-fitting and do not generalize well to unseen data. To allow better generalization, an array of methods based on semi-supervised learning (SSL) are utilized. The assumption is that unlabeled images are commonly from the original data distribution and contain useful information. In practice, there is often a large amount of unlabeled data available which are free to use. Some powerful SSL methods used the feature distribution of unlabeled images to reduce the need for labeling. For example, images are projected to low-dimensional feature space and pseudo-labels are assigned to unlabeld images based on clustering features [165]. In [166], images were intentionally perturbed to explore the decision boundary for adversarial training. These methods can be incorporated in Yottixel training to take advantage of large quantity of unlabeled data in histopathology.

### 9.2.4 Personalized Federated Learning

Chapter 8 discussed a framework for institutionalized federated learning in highly regulated domains, such as medicine. The proposed method eliminates a significant drawback of canonical federated learning by allowing — (i) model heterogeneity; each participant can have a private model with any architecture, and (ii) independence from a central entity; decentralization of federated learning. However, a major limitation of ProxyFL was limited objective personalization for involved participants. In federated learning, each participant has different objective, and different distribution of the local dataset. For example, among participating hospitals, some hospitals may specialize in lung diseases and would only expect their models to improve their performance on lung-related cases. The performance improvement over general cases may not be significant for these hospitals. Recently, achieving the idea of personalization in federated learning has gained lot of attention [167, 168, 169]. The personalized federated learning is under-explored not just in histopathology domain, but in the entire medical imaging domain. A future direction would be to extend ProxyFL to achieve the goal of personalized federated learning.

# References


[1] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[2] Shivam Kalra, HR 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.

[3] Omar Kujan, Ammar Khattab, Richard J Oliver, Stephen A Roberts, Nalin Thakker, and Philip Sloan. Why oral histopathology suffers inter-observer variability on grading oral epithelial dysplasia: an attempt to understand the sources of variation. *Oral oncology*, 43(3):224–231, 2007.

[4] Hamid R Tizhoosh, Phedias Diamandis, Clinton JV Campbell, Amir Safarpoor, 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*, 191(10):1702–1708, 2021.

[5] I-Jun Chou, Huei-Shyong Wang, William P Whitehouse, and Cris S Constantinescu. Paediatric multiple sclerosis: Update on diagnostic criteria, imaging, histopathology and treatment choices. *Current Neurology and Neuroscience Reports*, 16(7):1–12, 2016.

[6] Jeroen Van der Laak, Geert Litjens, and Francesco Ciompi. Deep learning in histopathology: the path to the clinic. *Nature medicine*, 27(5):775–784, 2021.

[7] PJ Sudharshan, Caroline Petitjean, Fabio Spanhol, Luiz Eduardo Oliveira, Laurent Heutte, and Paul Honeine. Multiple instance learning for histopathological breast cancer image classification. *Expert Systems with Applications*, 117:103–111, 2019.

[8] Jakub M Tomczak, Maximilian Ilse, and Max Welling. Deep learning with permutation-invariant operator for multi-instance histopathology classification. *arXiv preprint arXiv:1712.00310*, 2017.

[9] Zhi-Hua Zhou, Yu-Yin Sun, and Yu-Feng Li. Multi-instance learning by treating instances as non-iid samples. In *Proceedings of the 26th annual international conference on machine learning*, pages 1249–1256, 2009.

[10] Liron Pantanowitz, Janusz Szymas, David Wilbur, and Yukako Yagi. Whole slide imaging for educational purposes. *Journal of Pathology Informatics*, 3(1):46.

[11] Shaimaa Al-Janabi, André Huisman, and Paul J. Van Diest. Digital pathology: Current status and future perspectives. 61(1):1–9.

[12] Bethany Jill Williams, David Bottoms, and Darren Treanor. Future-proofing pathology: The case for clinical adoption of digital pathology. 70(12):1010–1018.

[13] Marcial García Rojo, Gloria Bueno García, Carlos Peces Mateos, Jesús González García, and Manuel Carbajo Vicente. Critical comparison of 31 commercially available digital slide systems in pathology. *International Journal of Surgical Pathology*, 14(4):285–305.

[14] Michael Thrall, Walid Khalbuss, and Liron Pantanowitz. Telecytology: Clinical applications, current challenges, and future benefits. *Journal of Pathology Informatics*, 2(1):51.

[15] Seung Park, Liron Pantanowitz, and Anil Vasdev Parwani. Digital Imaging in Pathology. 32(4):557–584.

[16] Shaimaa Al-Janabi, André Huisman, Peter G. J. Nikkels, Fiebo J. W. ten Kate, and Paul J. van Diest. Whole slide images for primary diagnostics of paediatric pathology specimens: A feasibility study. 66(3):218–223.

[17] Nikolas Stathonikos, Mitko Veta, André Huisman, and PaulJ van Diest. Going fully digital: Perspective of a Dutch academic pathology lab. *Journal of Pathology Informatics*, 4(1):15.

[18] TiffanyL Sellaro, Robert Filkins, Chelsea Hoffman, JeffreyL Fine, Jon Ho, AnilV Parwani, Liron Pantanowitz, and Michael Montalto. Relationship between magnification and resolution in digital pathology systems. *Journal of Pathology Informatics*, 4(1):21.

[19] Daisuke Komura and Shumpei Ishikawa. Machine learning methods for histopathological image analysis. *Computational and Structural Biotechnology Journal*, 2018.

[20] Anant Madabhushi and George Lee. Image analysis and machine learning in digital pathology: Challenges and opportunities. *Medical Image Analysis*, 33:170–175, October 2016.

[21] Anant Madabhushi, Shannon Agner, Ajay Basavanhally, Scott Doyle, and George Lee. Computer-aided prognosis: Predicting patient and disease outcome via quantitative fusion of multi-scale, multi-modal data. 35(7):506–514, 2011.

[22] Metin N. Gurcan, Laura Boucheron, Ali Can, Anant Madabhushi, Nasir Rajpoot, and Bulent Yener. Histopathological Image Analysis: A Review. 2:147–171, 2009.

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

[24] Neslihan Bayramoglu, Juho Kannala, and Janne Heikkilä. Deep learning for magnification independent breast cancer histopathology image classification. In *2016 23rd International conference on pattern recognition (ICPR)*, pages 2440–2445. IEEE, 2016.

[25] Nanqing Dong, Michael Kampffmeyer, Xiaodan Liang, Zeya Wang, Wei Dai, and Eric Xing. Reinforced auto-zoom net: towards accurate and fast breast cancer segmentation in whole-slide images. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 317–325. Springer, 2018.

[26] Yaqing Wang, QUANMING Yao, James Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. In *arXiv: 1904.05046*. 2019.

[27] Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation equivariant cnns for digital pathology. In *International Conference on Medical image computing and computer-assisted intervention*, pages 210–218. Springer, 2018.

[28] Jelica Vasiljević, Friedrich Feuerhake, Cédric Wemmert, and Thomas Lampert. Towards histopathological stain invariance by unsupervised domain augmentation using generative adversarial networks. *Neurocomputing*, 460:277–291, 2021.

[29] James Foulds and Eibe Frank. A review of multi-instance learning assumptions. *The knowledge engineering review*, 25(1):1–25, 2010.

[30] Hamed Erfankhah, Mehran Yazdi, Morteza Babaie, and Hamid R Tizhoosh. Heterogeneity-aware local binary patterns for retrieval of histopathology images. *IEEE Access*, 7:18354–18367, 2019.

[31] Zichao Guo, Hong Liu, Haomiao Ni, Xiangdong Wang, Mingming Su, Wei Guo, Kuansong Wang, Taijiao Jiang, and Yueliang Qian. A fast and refined cancer regions segmentation framework in whole-slide breast pathological images. *Scientific reports*, 9(1):882, 2019.

[32] Emad A Rakha, Mohamed Aleskandarani, Michael S Toss, Andrew R Green, Graham Ball, Ian O Ellis, and Leslie W Dalton. Breast cancer histologic grading using digital microscopy: concordance and outcome association. *Journal of clinical pathology*, 71(8):680–686, 2018.

[33] Jeffrey J Nirschl, Andrew Janowczyk, Eliot G Peyster, Renee Frank, Kenneth B Margulies, Michael D Feldman, and Anant Madabhushi. Deep learning tissue segmentation in cardiac histopathology images. In *Deep Learning for Medical Image Analysis*, pages 179–195. Elsevier, 2017.

[34] Akito Nagase, Masanobu Takahashi, and Masayuki Nakano. Automatic calculation and visualization of nuclear density in whole slide images of hepatic histological sections. *Bio-medical materials and engineering*, 26(s1):S1335–S1344, 2015.

[35] Henning Müller, Nicolas Michoux, David Bandon, and Antoine Geissbuhler. A review of content-based image retrieval systems in medical applications—clinical benefits and future directions. *International journal of medical informatics*, 73(1):1–23, 2004.

[36] Md Mahmudur Rahman, Prabir Bhattacharya, and Bipin C Desai. A framework for medical image retrieval using machine learning and statistical similarity matching techniques with relevance feedback. *IEEE transactions on Information Technology in Biomedicine*, 11(1):58–69, 2007.

[37] Anant Madabhushi and George Lee. Image analysis and machine learning in digital pathology: Challenges and opportunities. *Medical image analysis*, 33:170–175, 2016.

[38] 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.

[39] Karel Lenc and Andrea Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 991–999, 2015.

[40] Jocelyn Barker, Assaf Hoogi, Adrien Depeursinge, and Daniel L Rubin. Automated classification of brain tumor type in whole-slide digital pathology images using local representative tiles. *Medical image analysis*, 30:60–71, 2016.

[41] David A Gutman, Jake Cobb, Dhananjaya Somanna, Yuna Park, Fusheng Wang, Tahsin Kurc, Joel H Saltz, Daniel J Brat, Lee AD Cooper, and Jun Kong. Cancer digital slide archive: an informatics resource to support integrated in silico analysis of tcga pathology data. *Journal of the American Medical Informatics Association*, 20(6):1091–1098, 2013.

[42] Neville Mehta, Alomari Raja'S, and Vipin Chaudhary. Content based sub-image retrieval system for high resolution pathology images using salient interest points. In *IEEE International Conference of the Engineering in Medicine and Biology Society*, pages 3719–3722, 2009.

[43] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee, 1999.

[44] Hatice Cinar Akakin and Metin N Gurcan. Content-based microscopic image retrieval system for multi-image queries. *IEEE transactions on information technology in biomedicine*, 16(4):758–769, 2012.

[45] Xiaofan Zhang, Wei Liu, Murat Dundar, Sunil Badve, and Shaoting Zhang. Towards large-scale histopathological image analysis: Hashing-based image retrieval. *IEEE Transactions on Medical Imaging*, 34(2):496–506, 2015.

[46] Adam Goode, Benjamin Gilbert, Jan Harkes, Drazen Jukic, and Mahadev Satyanarayanan. Openslide: A vendor-neutral software foundation for digital pathology. *Journal of pathology informatics*, 4, 2013.

[47] Peter Bankhead, Maurice B Loughrey, José A Fernández, Yvonne Dombrowski, Darragh G McArt, Philip D Dunne, Stephen McQuaid, Ronan T Gray, Liam J Murray, Helen G Coleman, et al. Qupath: Open source software for digital pathology image analysis. *bioRxiv*, page 099796, 2017.

[48] Meghana Dinesh Kumar, Morteza Babaie, Shujin Zhu, Shivam Kalra, and Hamid R Tizhoosh. A comparative study of cnn, bovw and lbp for classification of histopathological images. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–7. IEEE, 2017.

[49] Fabio Alexandre Spanhol, Luiz S Oliveira, Caroline Petitjean, and Laurent Heutte. Breast cancer histopathological image classification using convolutional neural networks. In *2016 international joint conference on neural networks (IJCNN)*, pages 2560–2567. IEEE, 2016.

[50] Mohammad Peikari, Sherine Salama, Sharon Nofech-Mozes, and Anne L Martel. Automatic cellularity assessment from post-treated breast surgical specimens. *Cytometry Part A*, 91(11):1078–1087, 2017.

[51] Katarzyna Tomczak, Patrycja Czerwińska, and Maciej Wiznerowicz. The cancer genome atlas (tcga): an immeasurable source of knowledge. *Contemporary oncology*, 19(1A):A68, 2015.

[52] John Lonsdale, Jeffrey Thomas, Mike Salvatore, Rebecca Phillips, Edmund Lo, Saboor Shad, Richard Hasz, Gary Walters, Fernando Garcia, Nancy Young, et al. The genotype-tissue expression (gtex) project. *Nature genetics*, 45(6):580, 2013.

[53] Tumor Proliferation Assessment Challenge. Tupac16-miccai grand challenge, 2016.

[54] 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.

[55] Morteza Babaie, Shivam Kalra, Aditya Sriram, Christopher Mitcheltree, Shujin Zhu, Amin Khatami, Shahryar Rahnamayan, and H. R. Tizhoosh. Classification and Retrieval of Digital Pathology Scans: A New Dataset. *Conference on Computer Vision and Pattern Recognition Workshops*, 2017.

[56] Murat Dundar, Balaji Krishnapuram, RB Rao, and Glenn M Fung. Multiple instance learning for computer aided diagnosis. In *Advances in neural information processing systems*, pages 425–432, 2007.

[57] Yan Xu, Tao Mo, Qiwei Feng, Peilin Zhong, Maode Lai, I Eric, and Chao Chang. Deep learning of feature representation with multiple instance learning for medical

image analysis. In *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 1626–1630. IEEE, 2014.

[58] Maximilian Ilse, Jakub Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *International conference on machine learning*, pages 2127–2136. PMLR, 2018.

[59] Yan Xu, Jun-Yan Zhu, I Eric, Chao Chang, Maode Lai, and Zhuowen Tu. Weakly supervised histopathology cancer image segmentation and classification. *Medical image analysis*, 18(3):591–604, 2014.

[60] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in neural information processing systems*, pages 3391–3401, 2017.

[61] Michael Bonert, Uzma Zafar, Raymond Maung, Ihab El-Shinnawy, Ipshita Kak, Jean-Claude Cutz, Asghar Naqvi, Rosalyn A Juergens, Christian Finley, Samih Salama, et al. Evolution of anatomic pathology workload from 2011 to 2019 assessed in a regional hospital laboratory via 574,093 pathology reports. *PloS one*, 16(6):e0253876, 2021.

[62] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[63] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the Convergence of FedAvg on Non-IID Data. In *International Conference on Learning Representations*, 2020.

[64] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Trong Nghia Hoang, and Yasaman Khazaeni. Bayesian Nonparametric Federated Learning of Neural Networks. *arXiv:1905.12022 [cs, stat]*, May 2019.

[65] Sebastian Claici, Mikhail Yurochkin, Soumya Ghosh, and Justin Solomon. Model Fusion with Kullback–Leibler Divergence. *arXiv:2007.06168 [cs, stat]*, July 2020.

[66] Krishna Pillutla, Sham M. Kakade, and Zaid Harchaoui. Robust Aggregation for Federated Learning. *arXiv:1912.13445 [cs, stat]*, December 2019.

[67] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W. Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando de Freitas. Learning to learn by gradient descent by gradient descent, 2016.

[68] Suyi Li, Yong Cheng, Wei Wang, Yang Liu, and Tianjian Chen. Learning to Detect Malicious Clients for Robust Federated Learning, 2020.

[69] Taher Dehkharghanian, Azam Asilian Bidgoli, Abtin Riasatian, Pooria Mazaheri, Clinton JV Campbell, Liron Pantanowitz, HR Tizhoosh, and Shahryar Rahnamayan. Biased data, biased ai: Deep networks predict the acquisition site of tcga images. *under review*, 2021.

[70] Ming Y Lu, Dehan Kong, Jana Lipkova, Richard J Chen, Rajendra Singh, Drew FK Williamson, Tiffany Y Chen, and Faisal Mahmood. Federated learning for computational pathology on gigapixel whole slide images. *arXiv preprint arXiv:2009.10190*, 2020.

[71] Quande Liu, Cheng Chen, Jing Qin, Qi Dou, and Pheng-Ann Heng. FedDG: Federated Domain Generalization on Medical Image Segmentation via Episodic Learning in Continuous Frequency Space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1013–1023, June 2021.

[72] Daiqing Li, Amlan Kar, Nishant Ravikumar, Alejandro F Frangi, and Sanja Fidler. Fed-Sim: Federated Simulation for Medical Imaging, 2020.

[73] Abhishek Bhowmick, John Duchi, Julien Freudiger, Gaurav Kapoor, and Ryan Rogers. Protection against reconstruction and its applications in private federated learning. *arXiv preprint arXiv:1812.00984*, 2018.

[74] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 691–706. IEEE, 2019.

[75] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography*, pages 265–284, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[76] Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 51–60. IEEE, 2010.

[77] Cynthia Dwork and Aaron Roth. The Algorithmic Foundations of Differential Privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, August 2014.

[78] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our Data, Ourselves: Privacy Via Distributed Noise Generation. In *Advances in Cryptology (EUROCRYPT 2006)*, volume 4004 of *Lecture Notes in Computer Science*, pages 486–503. Springer Verlag, May 2006.

[79] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep Learning with Differential Privacy. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, October 2016.

[80] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning Differentially Private Recurrent Language Models. In *International Conference on Learning Representations*, 2018.

[81] Georgios A. Kaissis, Marcus R. Makowski, Daniel Rückert, and Rickmer F. Braren. Secure, privacy-preserving and federated machine learning in medical imaging. *Nature Machine Intelligence*, 2(6):305–311, June 2020. Number: 6 Publisher: Nature Publishing Group.

[82] Georgios Kaissis, Alexander Ziller, Jonathan Passerat-Palmbach, Théo Ryffel, Dmitrii Usynin, Andrew Trask, Ionésio Lima, Jason Mancuso, Friederike Jungmann, Marc-Matthias Steinborn, et al. End-to-end privacy preserving deep learning on multi-institutional medical imaging. *Nature Machine Intelligence*, 3(6):473–484, 2021.

[83] Alexander Ziller, Dmitrii Usynin, Rickmer Braren, Marcus Makowski, Daniel Rueckert, and Georgios Kaissis. Medical imaging deep learning with differential privacy. *Scientific Reports*, 11(1):1–8, 2021.

[84] Xiaoxiao Li, Yufeng Gu, Nicha Dvornek, Lawrence H. Staib, Pamela Ventola, and James S. Duncan. Multi-site fMRI analysis using privacy-preserving federated learning and domain adaptation: ABIDE results. *Medical Image Analysis*, 65:101765, 2020.

[85] Joseph Galaro, Alexander R Judkins, David Ellison, Jennifer Baccon, and Anant Madabhushi. An integrated texton and bag of words classifier for identifying anaplastic medulloblastomas. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 3443–3446. IEEE, 2011.

[86] Harshita Sharma, Alexander Alekseychuk, Peter Leskovsky, Olaf Hellwich, RS Anand, Norman Zerbe, and Peter Hufnagl. Determining similarity in histological

images using graph-theoretic description and matching methods for content-based image retrieval in medical diagnostics. *Diagnostic pathology*, 7(1):134, 2012.

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

[88] Ji Wan, Dayong Wang, Steven Chu Hong Hoi, Pengcheng Wu, Jianke Zhu, Yongdong Zhang, and Jintao Li. Deep learning for content-based image retrieval: A comprehensive study. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 157–166. ACM, 2014.

[89] Lin Yang, Xin Qi, Fuyong Xing, Tahsin Kurc, Joel Saltz, and David J. Foran. Parallel content-based sub-image retrieval using hierarchical searching. *Bioinformatics*, 30(7):996–1002, November 2013.

[90] Menglin Jiang, Shaoting Zhang, Junzhou Huang, Lin Yang, and Dimitris N Metaxas. Scalable histopathological image analysis via supervised hashing with multiple features. *Medical image analysis*, 34:3–12, 2016.

[91] Huei-Fang Yang, Kevin Lin, and Chu-Song Chen. Supervised learning of semantics-preserving hash via deep convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 40(2):437–451, 2018.

[92] 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.

[93] Mark D Zarella, Matthew R Quaschnick, David E Breen, and Fernando U Garcia. Estimation of fine-scale histologic features at low magnification. *Archives of pathology & laboratory medicine*, 142(11):1394–1402, 2018.

[94] Mesut Toğaçar, Kutsal Baran Özkurt, Burhan Ergen, and Zafer Cömert. Breastnet: A novel convolutional neural network model through histopathological images for the diagnosis of breast cancer. *Physica A: Statistical Mechanics and its Applications*, page 123592, 2019.

[95] 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.

[96] Brady Kieffer, Morteza Babaie, Shivam Kalra, and Hamid R Tizhoosh. Convolutional neural networks for histopathology image classification: Training vs. using pre-trained networks. In *2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–6. IEEE, 2017.

[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] Meghana Dinesh Kumar, Morteza Babaie, and Hamid R Tizhoosh. Deep barcodes for fast retrieval of histopathology scans. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.

[99] Lee AD Cooper, Elizabeth G Demicco, Joel H Saltz, Reid T Powell, Arvind Rao, and Alexander J Lazar. Pancancer insights from the cancer genome atlas: the pathologist's perspective. *The Journal of pathology*, 244(5):512–524, 2018.

[100] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2014.

[101] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448, 2015.

[102] 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.

[103] John N Weinstein, Eric A Collisson, Gordon B Mills, Kenna R Mills Shaw, Brad A Ozenberger, Kyle Ellrott, Ilya Shmulevich, Chris Sander, and Joshua M Stuart. The cancer genome atlas pan-cancer analysis project. *Nature genetics*, 45(10):1113–1120, 2013.

[104] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[105] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[106] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer. *CoRR*, abs/1810.00825, 2018.

[107] R.B. Rusu and S. Cousins. 3d is here: Point cloud library (pcl). In *2011 IEEE international conference on robotics and automation*, pages 1–4, May 2011.

[108] Daniel Maturana and Sebastian Scherer. VoxNet: A 3D Convolutional Neural Network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928.

[109] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T. Freeman, and Joshua B. Tenenbaum. Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling.

[110] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik G. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proc. ICCV*, 2015.

[111] Andrew Brock, Theodore Lim, J. M. Ritchie, and Nick Weston. Generative and Discriminative Voxel Modeling with Convolutional Neural Networks.

[112] 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.

[113] Mustafa I Jaber, Liudmila Beziaeva, Christopher W Szeto, John Elshimali, Shahrooz Rabizadeh, and Bing Song. Automated adeno/squamous-cell nsclc classification from diagnostic slide images: A deep-learning framework utilizing cell-density maps, 2019.

[114] Pegah Khosravi, Ehsan Kazemi, Marcin Imielinski, Olivier Elemento, and Iman Hajirasouliha. Deep convolutional neural networks enable discrimination of heterogeneous digital pathology images. *EBioMedicine*, 27:317–328, 2018.

[115] Kun-Hsing Yu, Ce Zhang, Gerald J Berry, Russ B Altman, Christopher Ré, Daniel L Rubin, and Michael Snyder. Predicting non-small cell lung cancer prognosis by fully automated microscopic pathology image features. *Nature communications*, 7:12474, 2016.

[116] Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering*, 26(8):1819–1837, 2013.

[117] Vaishali S Tidake and Shirish S Sane. Multi-label classification: a survey. *International Journal of Engineering and Technology*, 7(1045), 2018.

[118] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander Smola. Deep sets. *arXiv preprint arXiv:1703.06114*, 2017.

[119] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, pages 265–283, 2016.

[120] 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. *arXiv preprint arXiv:2101.07903*, 2021.

[121] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2014.

[122] Shivam Kalra, HR 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.

[123] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016.

[124] David K Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011.

[125] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, pages 1024–1034, 2017.

[126] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015.

[127] S Kalra, C Choi, S Shah, L Pantanowitz, and HR Tizhoosh. Yottixel–an image search engine for large archives of histopathology whole slide images. *arXiv preprint arXiv:1911.08748*, 2019.

[128] Ming Tu, Jing Huang, Xiaodong He, and Bowen Zhou. Multiple instance learning with graph neural networks. *arXiv preprint arXiv:1906.04881*, 2019.

[129] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[130] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[131] Xinggang Wang, Yongluan Yan, Peng Tang, Xiang Bai, and Wenyu Liu. Revisiting multiple instance neural networks. *Pattern Recognition*, 74:15–24, 2018.

[132] Shivang Naik, Scott Doyle, Shannon Agner, Anant Madabhushi, Michael Feldman, and John Tomaszewski. Automated gland and nuclei segmentation for grading of prostate and breast cancer histopathology. In *2008 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 284–287. IEEE, 2008.

[133] Ravi Aggarwal, Viknesh Sounderajah, Guy Martin, Daniel SW Ting, Alan Karthikesalingam, Dominic King, Hutan Ashrafian, and Ara Darzi. Diagnostic accuracy of deep learning in medical imaging: a systematic review and meta-analysis. *NPJ digital medicine*, 4(1):1–23, 2021.

[134] Micah J. Sheller, Brandon Edwards, G. Anthony Reina, Jason Martin, Sarthak Pati, Aikaterini Kotrotsou, Mikhail Milchenko, Weilin Xu, Daniel Marcus, Rivka R. Colen, and Spyridon Bakas. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. *Scientific Reports*, 10(1):12598, July 2020. Number: 1 Publisher: Nature Publishing Group.

[135] Ken Chang, Niranjan Balachandar, Carson Lam, Darvin Yi, James Brown, Andrew Beers, Bruce Rosen, Daniel L Rubin, and Jayashree Kalpathy-Cramer. Distributed deep learning networks among institutions for medical imaging. *Journal of the American Medical Informatics Association*, 25(8):945–954, 2018.

[136] Ilya Mironov. Rényi differential privacy. *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, August 2017.

[137] Pranav Subramani, Nicholas Vadivelu, and Gautam Kamath. Enabling fast differentially private sgd via just-in-time compilation and vectorization, 2020.

[138] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The Secret Sharer: Evaluating and Testing Unintended Memorization in Neural Networks. In *Proceedings of the 28th USENIX Conference on Security Symposium*, SEC'19, pages 267–284, USA, 2019. USENIX Association.

[139] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020.

[140] David Kempe, Alin Dobra, and Johannes Gehrke. Gossip-based computation of aggregate information. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 482–491. IEEE, 2003.

[141] Angelia Nedić and Alex Olshevsky. Stochastic gradient-push for strongly convex functions on time-varying directed graphs. *IEEE Transactions on Automatic Control*, 61(12):3936–3947, 2016.

[142] Angelia Nedić, Alex Olshevsky, and Michael G Rabbat. Network topology and communication-computation tradeoffs in decentralized optimization. *Proceedings of the IEEE*, 106(5):953–976, 2018.

[143] Chengxi Li, Gang Li, and Pramod K Varshney. Decentralized federated learning via mutual knowledge transfer. *IEEE Internet of Things Journal*, 2021.

[144] Yutao Huang, Lingyang Chu, Zirui Zhou, Lanjun Wang, Jiangchuan Liu, Jian Pei, and Yong Zhang. Personalized cross-silo federated learning on non-iid data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35-9, pages 7865–7873, 2021.

[145] Thorsten Wittkopp and Alexander Acker. Decentralized federated learning preserves model and data privacy. In *International Conference on Service-Oriented Computing*, pages 176–187. Springer, 2020.

[146] Tao Lin, Lingjing Kong, Sebastian U Stich, and Martin Jaggi. Ensemble distillation for robust model fusion in federated learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 2351–2363. Curran Associates, Inc., 2020.

[147] Jiaxin Ma, Ryo Yonetani, and Zahid Iqbal. Adaptive distillation for decentralized learning from heterogeneous clients. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 7486–7492. IEEE, 2021.

[148] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4320–4328, 2018.

[149] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[150] Tao Shen, Jie Zhang, Xinkang Jia, Fengda Zhang, Gang Huang, Pan Zhou, Kun Kuang, Fei Wu, and Chao Wu. Federated mutual learning. *arXiv preprint arXiv:2006.16765*, 2020.

[151] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[152] Wenqi Li, Fausto Milletarì, Daguang Xu, Nicola Rieke, Jonny Hancox, Wentao Zhu, Maximilian Baust, Yan Cheng, Sébastien Ourselin, M Jorge Cardoso, et al. Privacy-preserving federated brain tumour segmentation. In *International workshop on machine learning in medical imaging*, pages 133–141. Springer, 2019.

[153] Jing Ke, Yiqing Shen, and Yizhou Lu. Style normalization in histology with federated learning. In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pages 953–956. IEEE, 2021.

[154] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, pages 1322–1333, New York, NY, USA, 2015. Association for Computing Machinery.

[155] Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Lei Yu, and Wenqi Wei. Demystifying membership inference attacks in machine learning as a service. *IEEE Transactions on Services Computing*, pages 1–1, 2019.

[156] Ilya Mironov, Kunal Talwar, and Li Zhang. Rényi Differential Privacy of the Sampled Gaussian Mechanism. *arXiv preprint arXiv:1908.10530*, 2019.

[157] Borja Balle, Gilles Barthe, Marco Gaboardi, Justin Hsu, and Tetsuya Sato. Hypothesis Testing Interpretations and Renyi Differential Privacy. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2496–2506. PMLR, 26–28 Aug 2020.

[158] E. Seneta. *Non-negative Matrices and Markov Chains*. Springer Series in Statistics. Springer New York, 2006.

[159] Mahmoud Assran, Nicolas Loizou, Nicolas Ballas, and Mike Rabbat. Stochastic gradient push for distributed deep learning. In *International Conference on Machine Learning*, pages 344–353. PMLR, 2019.

[160] 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.

[161] Hang Li, Fan Yang, Xiaohan Xing, Yu Zhao, Jun Zhang, Yueping Liu, Mengxue Han, Junzhou Huang, Liansheng Wang, and Jianhua Yao. Multi-modal multi-instance learning using weakly correlated histopathological images and tabular clinical information. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 529–539. Springer, 2021.

[162] Imene Garali, Isaac M Adanyeguh, Farid Ichou, Vincent Perlbarg, Alexandre Seyer, Benoit Colsch, Ivan Moszer, Vincent Guillemot, Alexandra Durr, Fanny Mochel, et al. A strategy for multimodal data integration: application to biomarkers identification in spinocerebellar ataxia. *Briefings in bioinformatics*, 19(6):1356–1369, 2018.

[163] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors.

[164] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *ImageNet Large Scale Visual Recognition Competition (ILSVRC)*, September 2014.

[165] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5070–5079, 2019.

[166] Wenyuan Li, Zichen Wang, Yuguang Yue, Jiayun Li, William Speier, Mingyuan Zhou, and Corey Arnold. Semi-supervised learning using adversarial training with good and bad samples. *Machine Vision and Applications*, 31(6):1–11, 2020.

[167] Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning. *arXiv preprint arXiv:2003.13461*, 2020.

[168] Rui Hu, Yuanxiong Guo, Hongning Li, Qingqi Pei, and Yanmin Gong. Personalized federated learning with differential privacy. *IEEE Internet of Things Journal*, 7(10):9530–9539, 2020.

[169] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948*, 2020.

[170] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[171] Hamid R Tizhoosh, Shujin Zhu, Hanson Lo, Varun Chaudhari, and Tahmid Mehdi. Minmax radon barcodes for medical image retrieval. In *International Symposium on Visual Computing*, pages 617–627. Springer, 2016.

[172] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[173] Abraham Bookstein, Vladimir A Kulyukin, and Timo Raita. Generalized hamming distance. *Information Retrieval*, 5(4):353–375, 2002.

[174] Leland Wilkinson and Michael Friendly. The history of the cluster heat map. *The American Statistician*, 63(2):179–184, 2009.

# APPENDICES

# Appendix A

Addition content for Chapter 3.

## A.1 Yottixel Algorithm Overview

Yottixel framework incorporates clustering, transfer learning, and barcodes. In general, before any search is performed, all images in the repository have to be "indexed", i.e., every WSI is catalogued utilizing a "bunch of barcodes" (BoB indexing). These barcodes are stored for later use and generally not visible to users. This process contains several steps (Figure A.1):

1. **Tissue Extraction.** Every WSI contains a bright (white) background that generally contains irrelevant (non-tissue) pixel information. In order to process the tissue, we need to *segment* the tissue region(s) and generate a black and white image (binary mask) that provides the location of all tissue pixels as "1" (white). Such a binary mask is depicted in the top row of Figure A.1.

2. **Mosaicking.** Segmented tissue now gets *patched* (divided into patches/tiles). These patches have a fixed size at a fixed magnification (e.g., 500×500 $\mu m^2$ at 20× scan resolution). All patches of the WSI get grouped into a pre-set number of categories (classes) via a clustering method (we used $k$-means algorithm). A clustering algorithm is an unsupervised method that automatically groups WSI patches into clusters (i.e., groups) that contain similar tissue patterns. A small percentage (5%-20%) of all clustered patches are selected uniformly distributed within each class to assemble a **mosaic**. This mosaic represents the entire tissue region within the WSI. A sample mosaic consisting of 4 patches is depicted in the second row of Figure A.1. Most WSIs we processed had a mosaic with around 70-100 patches.

3. **Feature Mining.** All patches of the mosaic of each WSI are now pushed through pre-trained artificial neural networks (generally trained with *natural* images using datasets such as ImageNet [170]). The output of the network is ignored and the last pooling layers or the first connected layers are generally used as "features" to represent each mosaic patch. There could be approximately 1000-4000 features. The third row of Figure A.1 shows this process where the features (colored squares) are passed on to the next stage, namely BoB indexing.

4. **Bunch of Barcodes.** All feature vectors of each mosaic are subsequently converted into binary vectors using the *MinMax* algorithm [171]. This bunch of barcodes is the final index information for every query/input WSI that will be stored in the Yottixel index for future or immediate search. This is illustrated at the bottom of Figure A.1.

## A.2 Yottixel Extended Results

**Visualization of Search Results.** Examining best, average, and worst cases for diagnostic slides, we randomly selected 3,000 slides and visualized them using the T-distributed Stochastic Neighbor Embedding (t-SNE) method [172] (see Figure A.4). From this visualization we can observe that several subtype groups have been correctly extracted through search (see groups $a$ to $f$ ). We can also observe the presence of outliers (e.g., DLBC in groups $a$ and $b$). The outliers may be a product of the resolution of these scans, at least in part. At 20x magnification, for example, recognizing a diffuse large B-cell lymphoma (DLBC) from other large cell, undifferentiated non-hematopoietic tumors may not always be immediately possible for pathologists. This typically requires serial sections examined at multiple magnifications with ancillary studies such as immunohistochemistry.

**The challenge of validating histologic similarity.** One of the major benefits of using classification methods is that they can easily be validated; every image belongs to a class or not, a binary concept that can be conveniently quantified by counting the number of correctly/incorrectly categorized cases. It should be noted that through treating the image search as a classifier, we have not only used the primary diagnosis for "objective" evaluation of search results but also we are most likely ignoring some performance aspects of image search as search is a technology inherently suitable for looking at border cases and fuzziness of histologic similarity. The concept of similarity in image search is intrinsically a gradual concept (i.e., cannot be answered with a simple yes/no in many cases) and mostly a matter of degree (*very similar*, *quite dissimilar*, etc.). Additionally, the similarity (or dissimilarity)

between images is generally calculated using a distance metric/measure (in our case the Hamming distance [173]). The histologic similarity as perceived by pathologists may not correspond to tests where we used distance as a classification criterion. In other words, the classification-based tests that we run may be too harsh for search results and ignorant toward anatomic similarities among different organs.

One of the possible ways of examining the performance of the search is to look at the *heatmap* [174] of the confusion matrix. The values to construct the heatmap can be derived from the relative frequency of every subtype among the top 10 search results for a given subtype. A perfect heatmap would exhibit a pronounced diagonal with other cells being insignificant. Figure A.2 shows the generated heatmap for all diagnostic subtypes in the dataset. The ordering of subtypes along the $y$-axis was done manually. It should be noted that our matching heatmap is not symmetrical like a correlation-based heatmap.

**Analysis of the heatmap.** The pronounced diagonal in Figure A.2 shows that most disease subtypes have been correctly *classified* as they were very frequently retrieved among the top 10 horizontal search results. We find that MESO is a difficult diagnosis with almost absent diagonal values. READ and COAD build a confusion region of 4 squares; they are confused with each other frequently. The same observation can be made for LUAD and LUSC. The vertical values for LUAD and LUSC also show that they are present in many other searches, for instance, when we search for UESC, HNSC and ESCA. Of note, the observational analysis of the heatmap alone may be limited. If we cluster (group) the search result frequencies and construct the dendrograms for the relationships in order to create an advanced heatmap, we might more easily discover the benefits of the search (see **??**). We observe that some relations, that otherwise are considered misclassifications are actually histologically meaningful. Such as, LGG and GBM are both glial tumors of the central nervous system, rectum and colon cancer are gland forming tumors of the colon, and both uterine and ovarian carcinoma are grouped under gynecological. The errors (i.e., misclassifications) identified were still within the general grouping that the tumor originated from. Hence, from an image search perspective, it suggests that is it good at being close to the site of origin when it makes "classification" errors.

**Chord diagram of image search.** We used a chord diagram to further explore retrieved results. A chord diagram is the graphic display of the inter-relationships between numbers in a matrix. The numbers are arranged radially around a circle with the relationships between the data points generally visualized as arcs connecting the numbers/labels. In Figure A.3a, the chord diagram of horizontal search(cancer type recognition) for 11,579 permanent diagnostic slides of the TCGA dataset is illustrated. It can be observed that

certain tumors derived from the same organ are related (e.g. LGG and GBM, UCEC and CESC, and Kidney RCC and KIRP). Even tumors from different anatomic locations appear to match (e.g. GBM and sarcoma). This may be attributed to the fact that such high-grade tumors likely display similar morphologic findings.

**Algorithm 2** Pseudo-code for creating the index or bunch of barcodes (BoB) for a given WSI $I$

Set $k_{CH}$ (number of color clusters)
Set $p_M$ (percentage of patches to build the mosaic)
Set $m_x^c$ (clustering magnification)
Set $m_x^{idx}$ (indexing magnification)
Set patch sizes $s_l/s_h$ in low/high magnifications
**procedure** CREATE_INDEX($I$)
    ▷ Extract the tissue regions
    $T \leftarrow$ TissueSegmentation($I$)
    ▷ Select a low magnification within the WSI pyramid
    $I_{m_x^c} \leftarrow$ SelectMagnification($I, m_x^c$)
    ▷ Perform dense patching for patch size $s_l \times s_l$
    $P \leftarrow$ DensePatching($I_{m_x^c}, s_l$)
    ▷ Isolate patches containing tissue regions
    $P_T \leftarrow T \cap P$
    **for** $i \in [1, Len(P_T)]$ **do**
        ▷ Calculate the histogram of $i^{\text{th}}$ patch
        $H_{P_T}[i,:] \leftarrow$ RGBHistogram($P_T[i]$)
    **end for**
    ▷ Perform k-means clustering on histograms
    $C_1, C_2, ..., C_{k_{CH}} \leftarrow KMeans(H_{P_T}, k_{CH})$
    **for** $i \in [1, k_{CH}]$ **do**
        ▷ Cluster the location of patches in $C_i$
        $C_{i_M} \leftarrow$ KMeans($H_{P_T}(i,:), p_M \times |C_i|$)
        ▷ Construct the Mosaic
        $M \leftarrow C_{i_M}$
    **end for**
    $BoB_I \leftarrow$ Empty array to store BoB index for $I$
    **for** $j \in [1, length(M)]$ **do**
        ▷ Get a patch ( $s_h \times s_h$) at $m_x^{idx}$ magnification
        $P_{m_x^{idx}}[j] \leftarrow$ GetPatch($I, M[j]$)
        ▷ Extract the feature from a deep network
        $F \leftarrow$ DeepNet($P_{m_x^{idx}}[j]$)
        ▷ Convert the feature to a barcode
        $B \leftarrow$ MinMaxBarcode($F$)
        Append $B$ to a BoB array $BoB_I$
    **end for**
    Return $BoB_I$
**end procedure**

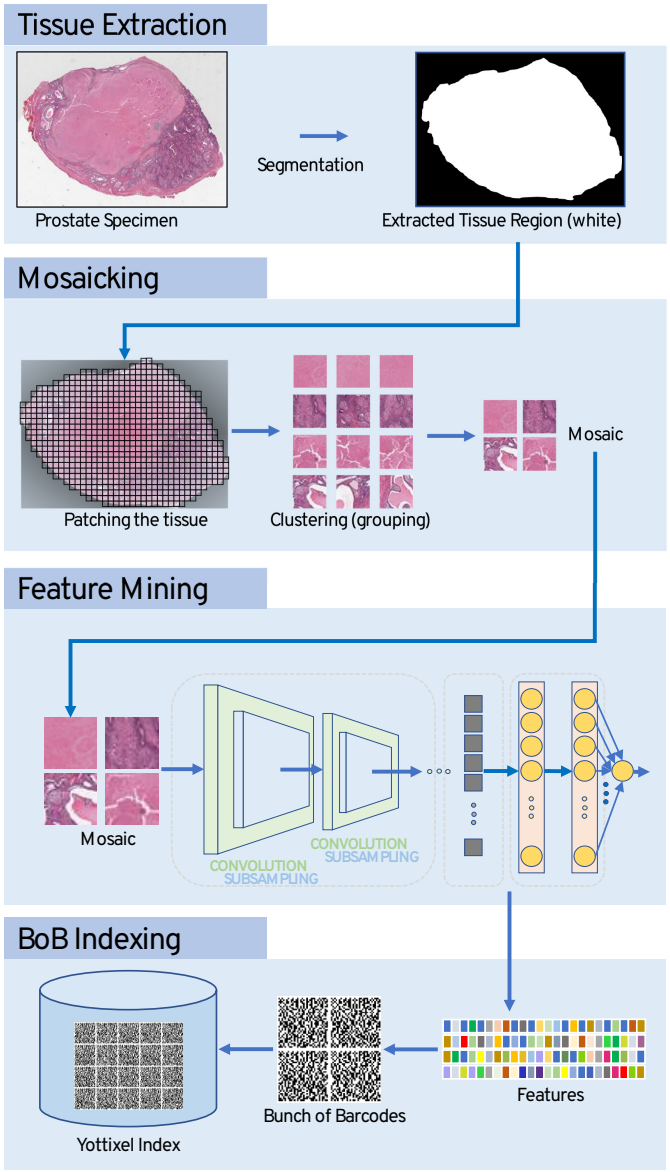



Figure A.1: Yottixel Image Search Engine: Whole-slide images are segmented first to extract the tissue region by excluding the background (top block). A mosaic of representative patches (tiles) is assembled through grouping of all patches of the tissue region using an unsupervised clustering algorithm (second block from the top). All patches of the mosaic are fed into a pre-trained artificial neural network for feature mining (third block from the top). Finally, a bunch of barcodes is generated and added to the index of all WSI files in the archive (bottom block).

**Algorithm 3** Distance between two given WSIs $I_q$ and $I$

```
1: procedure SCAN_DISTANCE(I_q, I)
2:     D_I ← ∅
3:     for b_{I_q} ∈ I_q.bob do
4:         H_min ← ∞
5:         for b_I ∈ I.bob do
6:             d ← getHammingDistance(b_I, b_{I_q})
7:             if d < H_min then
8:                 H_min ← d
9:             end if
10:        end for
11:        D_I = D_I ∪ {H_min}
12:    end for
13:    D ← findMedian(D_I)
14:    return D
15: end procedure
```

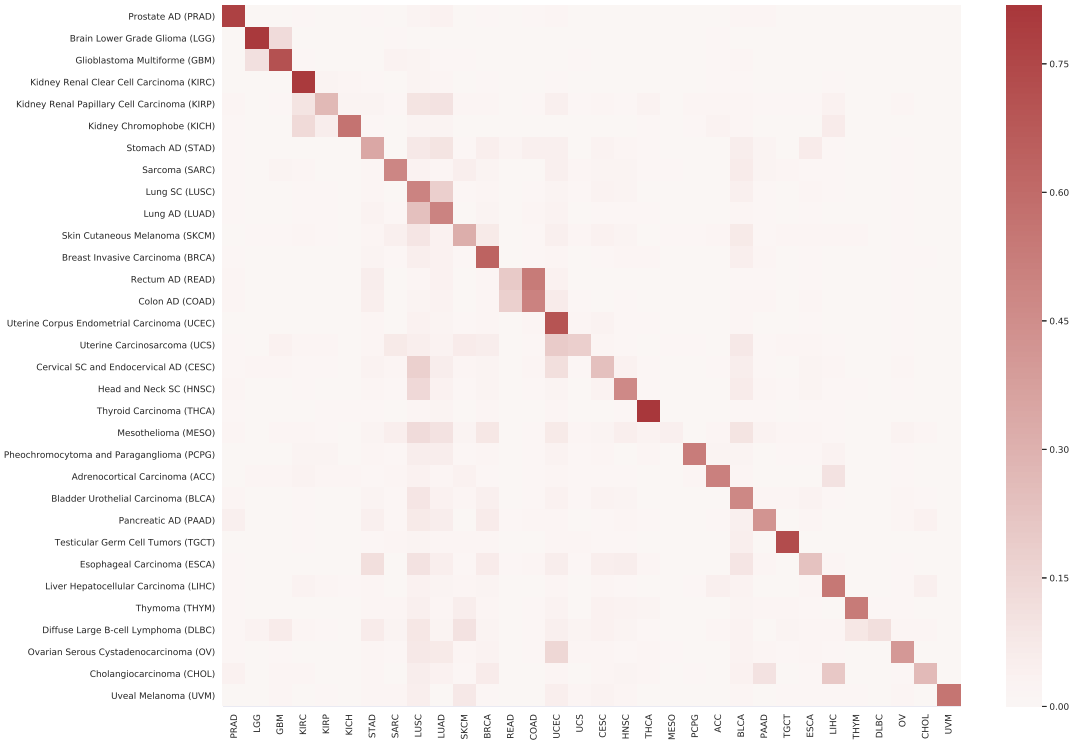
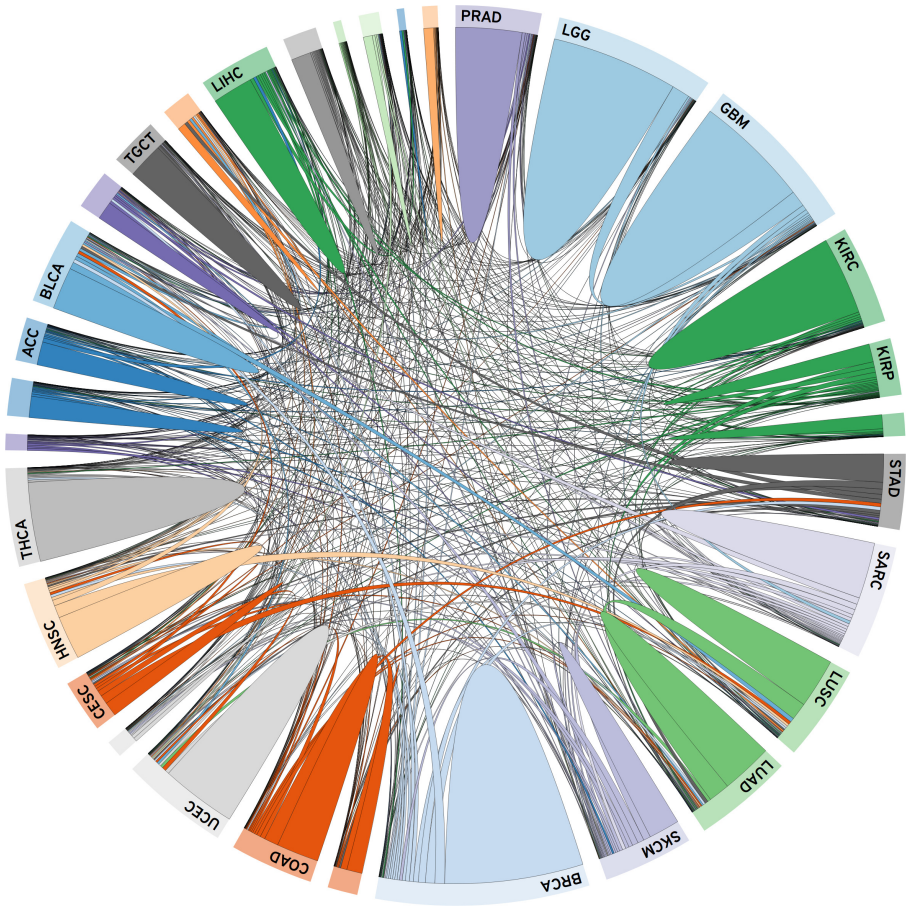
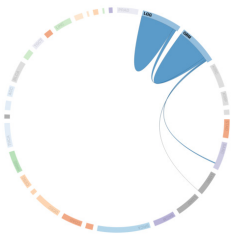


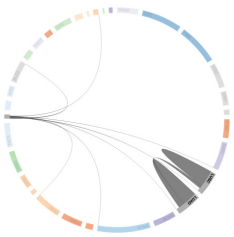

Figure A.2: Heatmap of re-scaled relative frequency of matched (red) and mismatched (pale) search results for each diagnosis from permanent diagnostic slides. Re-scaling of frequencies was done through dividing each frequency by the total number of slides for each subtype.




**(a)** Chord Diagram



**(b)** Brain



**(c)** Pulmonary

**(d)** Gynaecological

Figure A.3: Chord diagram of horizontal image search for diagnostic slides of the TCGA dataset (a). Sample relations for brain (LGG and GBM), pulmonary (LAUD, LUSC and MESO) and gynecological (UCEC, UCS and CESC). The chord diagram can be interactively viewed online: https://bit.ly/2k6g3k1.

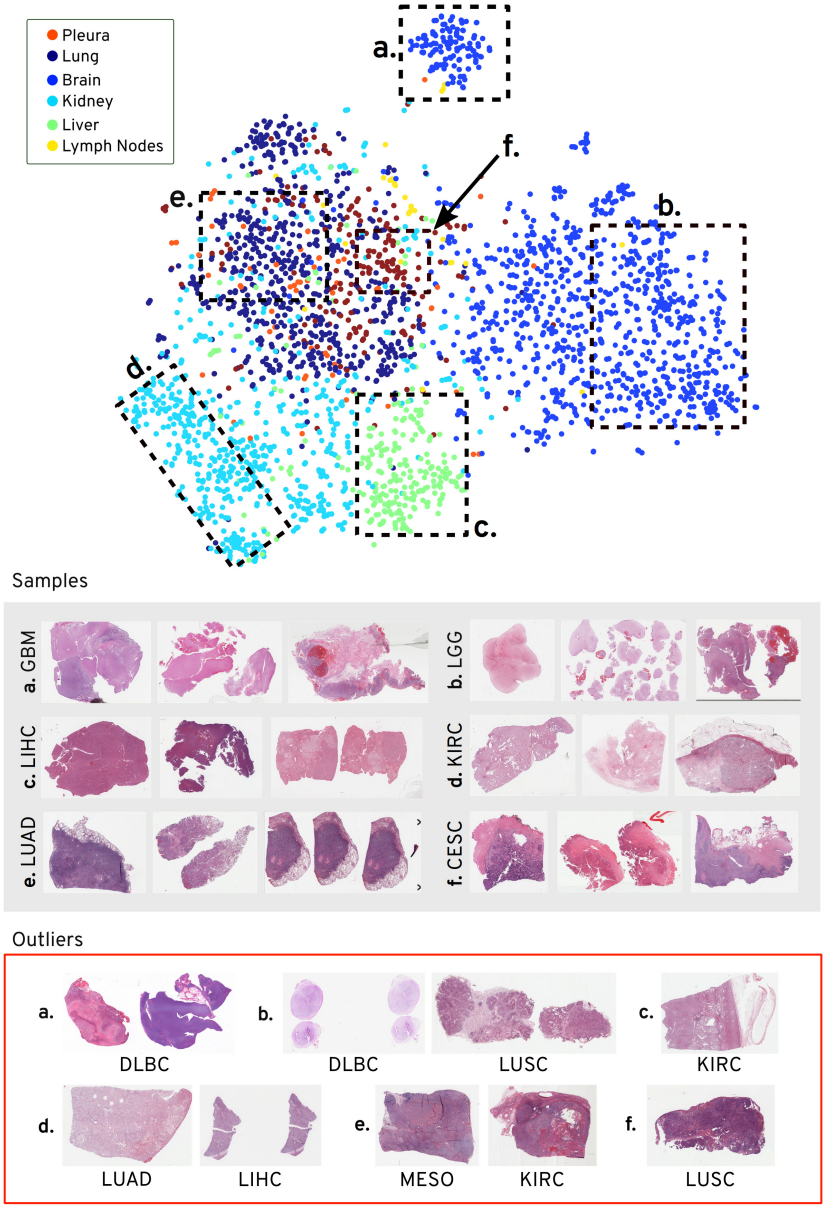



Figure A.4: T-distributed Stochastic Neighbor Embedding (t-SNE) visualization of pairwise distances of 3000 randomly selected diagnostic slides from six different primary sites. Six different cluster formation can be seen, labelled with alphabets. The random slides from the majority cancer sub-type within each of the assigned areas are shown in *Samples* box (gray background). The outliers (not belonging to the majority cancer sub-type or the primary site) are shown in the Outliers box (red outline).

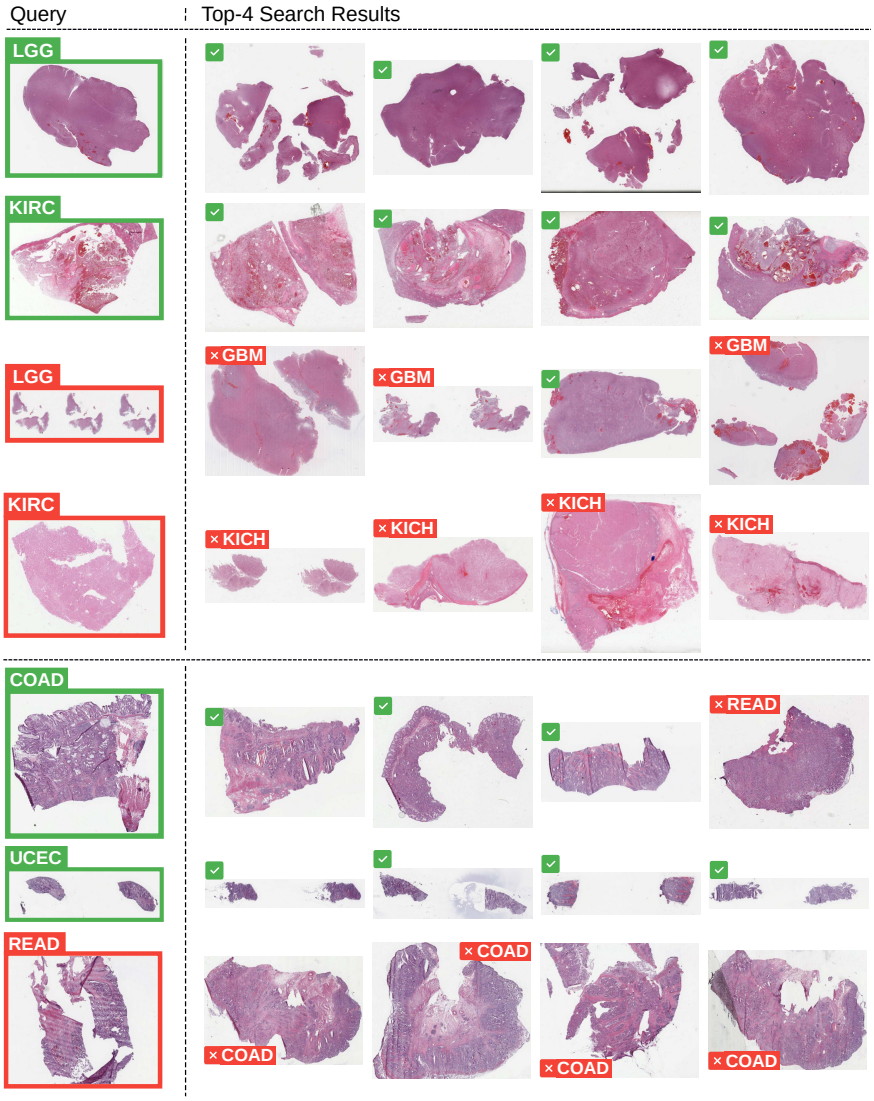



Figure A.5: Sample retrievals for cancer subtype categorization through majority votes. The top four slides are of permanent diagnostic slides whereas the bottom three slides are of frozen section slides. The misclassified and successful queries are marked with red and green boundaries, respectively. (for abbreviations see Table A.2)

| # | $\rho_{sp}$ | $Q_1$ | | $Q_2$ | | $Q_3$ | |
|---|---|---|---|---|---|---|---|
| | | MOS | $d_{t_1}$ | MOS | $d_{t_1}$ | MOS | $d_{t_1}$ |
| 1. | -1.00 | 3.33 | 72.0 | 4.33 | 76 | 4.67 | 77 |
| 2. | NA | 4.67 | 55.0 | 4.67 | 58 | 4.67 | 59 |
| 3. | -1.00 | 2.00 | 68.0 | 2.33 | 71 | 3.67 | 72 |
| 4. | -0.50 | 4.00 | 68.0 | 4.67 | 77 | 4.33 | 80 |
| 5. | -1.00 | 4.00 | 81.0 | 4.67 | 86 | 4.67 | 86 |
| 6. | -0.50 | 3.67 | 62.0 | 4.67 | 63 | 4.33 | 71 |
| 7. | -0.87 | 4.00 | 73.0 | 4.33 | 75 | 4.33 | 77 |
| 8. | NA | 4.67 | 66.0 | 4.67 | 67 | 4.67 | 69 |
| 9. | -0.87 | 3.33 | 76.0 | 4.00 | 77 | 4.00 | 79 |
| 10. | 1.00 | 4.00 | 102.0 | 3.33 | 104 | 2.67 | 105 |
| 11. | 0.87 | 3.33 | 70.0 | 3.33 | 80 | 3.00 | 81 |
| 12. | -0.87 | 4.00 | 51.0 | 4.67 | 52 | 4.67 | 55 |
| 13. | NA | 4.33 | 74.0 | 4.67 | 74 | 4.67 | 74 |
| 14. | 0.87 | 4.67 | 58.0 | 4.67 | 65 | 4.33 | 69 |
| 15. | 0.00 | 3.00 | 77.0 | 2.67 | 78 | 4.33 | 78 |
| 16. | 1.00 | 5.00 | 87.0 | 4.33 | 93 | 1.33 | 97 |
| 17. | -0.50 | 4.67 | 62.0 | 5.00 | 62 | 5.00 | 64 |
| 18. | 0.87 | 5.00 | 91.0 | 4.67 | 95 | 4.33 | 95 |
| 19. | -1.00 | 4.00 | 68.0 | 4.67 | 73 | 5.00 | 75 |
| 20. | 0.50 | 4.00 | 71.0 | 3.33 | 76 | 3.67 | 79 |
| 21. | 0.50 | 5.00 | 70.0 | 3.33 | 71 | 4.00 | 78 |
| 22. | 1.00 | 5.00 | 58.0 | 3.00 | 74 | 2.00 | 77 |
| 23. | 0.00 | 4.33 | 79.0 | 4.67 | 83 | 3.33 | 83 |
| 24. | 1.00 | 5.00 | 57.0 | 4.33 | 60 | 1.67 | 77 |
| 25. | -0.50 | 4.67 | 66.0 | 4.00 | 78 | 5.00 | 79 |
| 26. | -0.87 | 4.00 | 55.0 | 4.33 | 55 | 5.00 | 60 |
| 27. | 0.87 | 4.33 | 72.0 | 3.33 | 73 | 3.33 | 75 |
| 28. | 1.00 | 5.00 | 77.0 | 4.00 | 79 | 3.00 | 82 |
| 29. | 1.00 | 4.67 | 50.0 | 4.33 | 70 | 4.00 | 74 |
| 30. | 0.00 | 4.33 | 75.0 | 4.00 | 76 | 4.67 | 76 |
| 31. | 1.00 | 4.67 | 65.0 | 4.33 | 70 | 4.00 | 76 |
| 32. | 1.00 | 5.00 | 85.0 | 4.33 | 90 | 4.00 | 91 |
| 33. | 0.87 | 5.00 | 58.0 | 4.67 | 62 | 4.67 | 65 |
| 34. | -0.87 | 3.00 | 76.0 | 3.00 | 81 | 4.67 | 83 |
| 35. | 0.87 | 4.67 | 75.0 | 4.67 | 78 | 4.33 | 79 |
| 36. | NA | 5.00 | 52.0 | 5.00 | 56 | 5.00 | 56 |
| 37. | NA | 5.00 | 60.0 | 5.00 | 61 | 5.00 | 62 |
| 38. | 1.00 | 4.67 | 71.0 | 3.33 | 76 | 3.00 | 83 |
| 39. | NA | 5.00 | 65.0 | 5.00 | 69 | 5.00 | 71 |
| 40. | NA | 4.33 | 67.0 | 4.33 | 68 | 4.33 | 68 |
| 41. | 0.50 | 3.67 | 77.0 | 4.33 | 77 | 3.67 | 82 |
| 42. | 0.87 | 4.33 | 89.0 | 3.67 | 96 | 2.67 | 96 |
| 43. | -0.87 | 4.33 | 58.0 | 4.33 | 63 | 4.67 | 68 |
| 44. | 0.50 | 5.00 | 83.0 | 4.33 | 94 | 4.67 | 100 |
| 45. | 0.87 | 5.00 | 54.0 | 4.67 | 60 | 4.67 | 62 |
| 46. | 0.87 | 3.67 | 74.0 | 2.00 | 75 | 2.00 | 76 |
| 47. | NA | 4.67 | 57.0 | 4.67 | 60 | 4.67 | 60 |
| 48. | -1.00 | 3.33 | 60.0 | 3.67 | 63 | 4.00 | 68 |
| $\sum$ | 0.16 ± 0.76 | 4.30 ± 0.96 | | 4.13 ± 1.05 | | 4.028 ± 1.21 | |

Table A.1: Mean-Opinion-Score (MOS) of three pathologists for top three search results. MOS is "a numerical measure of the human-judged overall quality of an event or experience". Shades of green represent positive responses (in favour of Yottixel) and shades of red represent negative responses (against Yottixel). Rank coefficient $\rho_{sp}$ represents the rank correlation of the MOS with respect to the internal ranking of Yottixel based on the Hamming distance.

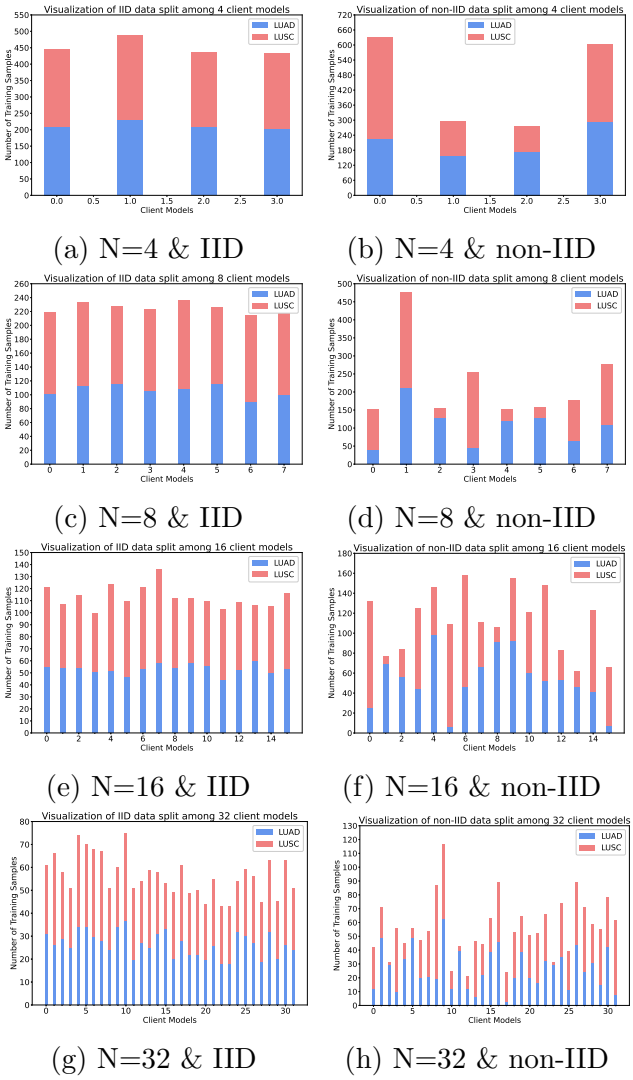
| TCGA Code | Primary Diagnosis | #Patients |
|---|---|---|
| ACC | Adrenocortical Carcinoma | 86 |
| BLCA | Bladder Urothelial Carcinoma | 410 |
| BRCA | Breast Invasive Carcinoma | 1097 |
| CESC | Cervical Squamous Cell Carcinoma and Endocervical Adenocarcinoma | 304 |
| CHOL | Cholangiocarcinoma | 51 |
| COAD | Colon Adenocarcinoma | 459 |
| DLBC | Lymphoid Neoplasm Diffuse Large B-cell Lymphoma | 48 |
| ESCA | Esophageal Carcinoma | 185 |
| GBM | Glioblastoma Multiforme | 604 |
| HNSC | Head and Neck Squamous Cell Carcinoma | 473 |
| KICH | Kidney Chromophobe | 112 |
| KIRC | Kidney Renal Clear Cell Carcinoma | 537 |
| KIRP | Kidney Renal Papillary Cell Carcinoma | 290 |
| LGG | Brain Lower Grade Glioma | 513 |
| LIHC | Liver Hepatocellular Carcinoma | 376 |
| LUAD | Lung Adenocarcinoma | 522 |
| LUSC | Lung Squamous Cell Carcinoma | 504 |
| MESO | Mesothelioma | 86 |
| OV | Ovarian Serous Cystadenocarcinoma | 590 |
| PAAD | Pancreatic Adenocarcinoma | 185 |
| PCPG | Pheochromocytoma and Paraganglioma | 179 |
| PRAD | Prostate Adenocarcinoma | 499 |
| READ | Rectum Adenocarcinoma | 170 |
| SARC | Sarcoma | 261 |
| SKCM | Skin Cutaneous Melanoma | 469 |
| STAD | Stomach Adenocarcinoma | 442 |
| TGCT | Testicular Germ Cell Tumors | 150 |
| THCA | Thyroid Carcinoma | 507 |
| THYM | Thymoma | 124 |
| UCEC | Uterine Corpus Endometrial Carcinoma | 558 |
| UCS | Uterine Carcinosarcoma | 57 |
| UVM | Uveal Melanoma | 80 |

Table A.2: The TCGA codes (in alphabetical order) of all 33 primary diagnoses and corresponding number of evidently diagnosed patients in the dataset (TCGA = The Cancer Genome Atlas)

# Appendix B

Additional content for Chapter 7.

## B.1 FedAvg Extended Results




(a) N=4 & IID (b) N=4 & non-IID

(c) N=8 & IID (d) N=8 & non-IID

(e) N=16 & IID (f) N=16 & non-IID

(g) N=32 & IID (h) N=32 & non-IID

Figure B.1: Visualisation of IID and non-IID distribution of data among client models