Content-based Image Retrieval of Gigapixel Histopathology Scans: A Comparative Study of Convolution Neural Network, Local Binary Pattern, and Bag of visual Words

by

Shivam Kalra

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2018

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 Contribution

The thesis is partly based on the following papers which are referred to in text by their associated alphabet.

A. M. Babaie, **S. Kalra**, A. Sriram, C. Mitcheltree, S. Zhu, A. Khatami, S. Rahnamayan, and H.R. Tizhoosh. *Classification and Retrieval of Digital Pathology Scans: A New Dataset.* The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Honolulu, USA, 2017, pp. 8-16.

B. S. Zhu, Y. Li, **S. Kalra**, H.R. Tizhoosh. *Multiple Disjoint Dictionaries for Representation of Histopathology Images.* Journal of Visual Communication and Image Representation (JVCI), Elsevier.

C. B. Kieffer, M. Babaie, **S. Kalra**, and H.R. Tizhoosh. *Convolutional Neural Networks for Histopathology Image Classification: Training vs. Using Pre-Trained Networks.* International Conference on Image Processing Theory, Tools & Applications (IPTA), Montreal, Canada, 2017, pp. 1-6.

I've contributed to implementation, experimentation, and writing of all the Papers (A, B and C). I'm corresponding author for Paper A with equal contributions as the first author. My contributions for Paper B and C are to lesser extent.

Majority of Chapter 6, Chapter 7, and section on Machine Learning (§2.6) from Chapter 2 are taken from Paper A. Some of the results and figures presented in Chapter 7 are taken from Paper B and Paper C.

Abstract

THE state-of-the-art image analysis algorithms offer a unique opportunity to extract semantically meaningful features from medical images. The advantage of this approach is automation in terms of content-based image retrieval (CBIR) of medical images. Such an automation leads to more reliable diagnostic decisions by clinicians as the direct beneficiary of these algorithms.

Digital pathology (DP), or whole slide imaging (WSI), is a new avenue for image-based diagnosis in histopathology. WSI technology enables the digitization of traditional glass slides to ultra high-resolution digital images (or digital slides). Digital slides are more commonly used for CBIR research than other modalities of medical images due to their enormous size, increasing adoption among hospitals, and their various benefits offered to pathologists (e.g., digital telepathology). Pathology laboratories are under constant pressure to meet increasingly complex demands from hospitals. Many diseases (such as cancer) continue to grow which creates a pressing need to utilize existing innovative machine learning schemes to harness the knowledge contained in digital slides for more effective and efficient histopathology.

This thesis provides a qualitative assessment of three popular image analysis techniques, namely Local Binary Pattern (LBP), Bag of visual Words (BoW), and Convolution Neural Networks (CNN) in their abilities to extract the discriminative features from gigapixel histopathology images. LBP and BoW are well-established techniques used in different image analysis problems. Over the last 5-10 years, CNN has become a frequent research topic in computer vision. CNN offers a domain-agnostic approach for the automatic extraction of discriminative image features, used for either classification or retrieval purposes. Therefore, it is imperative that this thesis gives more emphasis to CNN as a viable approach for the analysis of DP images.

A new dataset, *Kimia Path24* is specially designed and developed to facilitate the research in classification and CBIR of DP images. *Kimia Path24* is used to measure the quality of image-features extracted from LBP, BoW, and CNN; resulting in the best accuracy values of 41.33%, 54.67%, and 56.98% respectively. The results are somewhat surprising, suggesting that the accuracy score of handcrafted feature extraction algorithm, i.e., LBP can reach very close to the deep features extracted from CNN. It is unanticipated, considering that CNN requires much more computational resources and efforts for designing and fine-tuning. One of the conclusions is that CNN needs to be trained for the problem with a large number of training images to realize its comprehensive benefits. However, there are many situations where large, balanced, and the labeled dataset is not available; one such area is histopathology at present.

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

I'd also like to thank my co-supervisor, Professor Shahryar Rahnamayan for his valuable advice and knowledge. I know Prof. Shahryar since my undergraduate degree and his role is the most vital in encouraging me to do the higher studies. He has always steered me in the right direction whenever I needed.

I'd like to thank my co-authors Aditya, Brady, Morteza, Shujin, Aditya, Prof. Tizhoosh, and Prof. Shahryar for all their help in writing papers and conducting the research.

I owe a special thanks to Professor Andrea Scott and Professor Karim Karim for taking out their time to review my thesis and provide the valuable suggestions.

I would like to acknowledge the scholarship programs offered by Govt. of Canada (NSERC-M), Govt. of Ontario (OGS), and University of Waterloo (PGS) for providing the much needed financial support throughout my degree.

I would like to thank Sharcnet's support team for maintaining the software and hardware infrastructure of Copper cluster. I'd like to thank Brady for his help in setting virtual machines and Python environment on Microsoft Azure cloud platform. All these computing resources helped me running my experiments smoothly.

I'd like to thank my parents, Dr. Naveen and Renu Kalra for raising me to value the education. I deeply appreciate my sisters, Shruti Thareja and Neha Kalra, and my brother in-law, Rohan Thereja for their infinite support and guidance.

Many thanks to Jasneet and all my friends for being supportive; their encouragement and laughs kept me going. 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 and sisters.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

CNN Convolution Neural Network xi, 6, 7, 23, 25–32, 35–40, 42, 52, 53, 61–65, 71, 73, 75–77, 91

LBP Local Binary Pattern xi, xiv, 6, 7, 23, 25, 42–45, 47, 49, 52, 61–65, 67–69, 73, 75–77

BoW Bag of visual Words xi, 6, 7, 23, 25, 42, 47–52, 61–65, 69–71, 73, 75–77

WSI Whole Slide Imaging xiii, 3, 4, 9–14, 16–18, 55

FP Forward Pass xiv, 27–29, 31, 33–37

ML Machine Learning 2, 3, 6, 9, 15, 16, 18, 19, 26, 73, 76

AI Artificial Intelligence 2, 6, 18, 19

DL Deep Learning 3, 19, 26

DP Digital Pathology 3, 4, 6, 7, 9–11, 13–18, 22, 55

CBIR Content-based Image Retrieval 6, 7, 18–23, 25, 39, 55, 60, 65

CAD Computer Assisted Diagnosis 10, 16, 17

BP Backward Pass 27–31, 33–36

SGD Stochastic Gradient Descent 28, 40, 41

Chapter 1

Introduction

> “*The last thing that we find in making a book is to know what we must put first.*”
>
> — Blaise Pascal

1.1 Motivation

DURING the late 1999s, Institute of Medicine (IOM) released a report titled “To Err is Human: Building a Safer Health System” [7]. Following is an excerpt from the report

> “*..as many as 98,000 people die in any given year from medical errors.*
> *..that’s more than deaths from motor vehicle accidents, breast cancer, or AIDS.* [7]”

The report establishes “mistake” as the third most significant reason for deaths in the US. For many obvious reasons, it gathered plenty of media attention raising questions regarding the competency of healthcare professionals. However, the original intent of the report was often lost in the blame game that emanated from the rhetoric of media. We, humans, by nature, make mistakes, and healthcare is no different. What is more important is to learn from the mistakes and use available information to prevent or reduce future errors.

The advent of digitization in medicine has opened new horizons for integration of innovative Machine Learning (ML) solutions into clinical practices. It aspires to make healthcare safer, traceable, and of better quality. The motivation for the thesis comes from the desire to apply state-of-the-art ML algorithms on the current subjective task of image-based diagnosis. Computer algorithms, especially based on Artificial Intelligence (AI), offer a unique opportunity to extract useful features from medical images. The advantage of this approach includes automated search of similar medical images in extensive archives of hospitals and laboratories. Displaying the similar images from past patients who have been diagnosed and treated, provides useful information to medical practitioners leading to improved accuracy and precision of diagnostic interpretations.

1.2 Problem Statement

This thesis is concerned with irreducible error rate that comes from the current practices in diagnostic pathology. The errors can occur during different stages of diagnosis; including biopsy, sample preparation, and final interpretation — affecting patient's safety as well as reputation of the healthcare provider.

Diagnostic pathology involves many complex image analysis tasks, such as detecting and counting mitotic events (cells divisions) for identifying breast cancer [8], segmentation of nuclei, and tissue classification (e.g., cancerous vs. non-cancerous) based on complicated patterns and morphology seen under microscopes at different magnification levels. Furthermore, there is high variability in slide preparations (e.g., staining and scanning across different vendors), variance permeating due to different grades of a same disease, and vendor-specific platforms. These variables make pathology-based diagnosis even more challenging.

Majority of the errors in diagnostic pathology are human errors, caused mainly due to the biological limitations of humans in handling complicated image-analysis tasks. Some of these limitations are — **(i)** humans are capable of distinguishing only 30 levels of gray shades [9] whereas electronic standard is 256, **(ii)** human brain cannot comprehend a complex scientific image analysis, systematically and tenaciously, which can be processed in an automated manner through computer algorithms, and **(iii)** humans efficiency can be affected by several factors, such as emotions, stress; however, computer algorithms are objective and operative all times. These factors make the human-related errors irreducible yet unavoidable. At present, the best course of actions to prevent the diagnostic mistakes are re-assessment and re-evaluation of your own work, or to seek a second opinion by consulting with others.

Over the last decade, ML algorithms have evolved considerably. With progress in the field of Deep Learning (DL), machines have become highly efficient in automated image analysis and feature extraction, primarily in recognition of natural scenes, almost approaching the human-level performance [10]. However, ML methods in histopathology image analysis are relatively less prominent and have many unexplored potentials. The principal challenge of the research is to develop a pragmatic ML solution which is capable of understanding the ontological status of histopathology indicators for diagnostic purposes; thus ensuring more confident interpretation in diagnosis by enabling streamlined and efficient workflow for pathologists.

1.3 Background

Histopathology is based on analyzing and interpreting different shapes, sizes, and architectural patterns of cells and tissues can be combined with patient's clinical records, and various other factors in order to study the manifestation of disease. The word "histopathology" originates from the combination of two branches of science "histology" and "pathology." Histology is the study of microscopic structures of tissues whereas pathology involves the diagnosis of diseases through microscopic examinations of surgically removed specimens.

Histopathology is one of the essential disciplines throughout the healthcare delivery system. It is studied and practiced by the medical experts known as *pathologists*. The primary clinical duty of pathologists involves conducting a microscopic analysis of glass slides containing tissue specimens to render pathology reports. The reports created by pathologists are used for many clinical decisions, such as screening for diseases, developing diagnostic plans, monitoring progression of diseases, and managing various therapies and their prognosis.

Interpreting images of tissues and cells at high resolutions is the core of histopathology. Over centuries, the microscope has been the only available instrumentation for this undertaking; providing live images at an increasing resolution through ever improving-optics [11]. With the increased digitization of clinical practices, histopathology is also leading its way in utilizing a digital imaging technology as the "digital-age" alternative to conventional light microscopy. Pathology routines conducted in a digital image based environment, including management, sharing, and interpretation of pathological information is known as Digital Pathology (DP).

Robotic microscopic scanners are used to digitize glass slides into gigapixel images through a process known as Whole Slide Imaging (WSI) or *virtual microscopy*. The gigapixel images obtained from WSI are *digital slides*. WSI technology simulates the light microscopy for pathologists (digital slides combined with software systems provide the same functionality as a microscope but on computer screens) [12]. Figure 1.1 shows a sample digital slide obtained from a WSI scanner, compared with an aerial map of entire city of Waterloo in ON, Canada. The figure establishes an impression of the complexity of tasks involved in diagnostic pathology.

DP is one of the recent significant achievements in integration of modern computational practices within traditional medicines [13]. Digital slides are used for primary diagnosis, telepathology (remote access of glass slides as the digital slides), quality assurance (e.g., proficiency testing and validations), archiving, as a tool for education among pathologists in training, and for digital image analysis [12, 14]. During recent times, WSI technology is rapidly growing due to the continuous improvement in capabilities and throughput of WSI scanners, development of user-friendly software systems for managing and viewing digital slides, and vendor neural storage solutions.

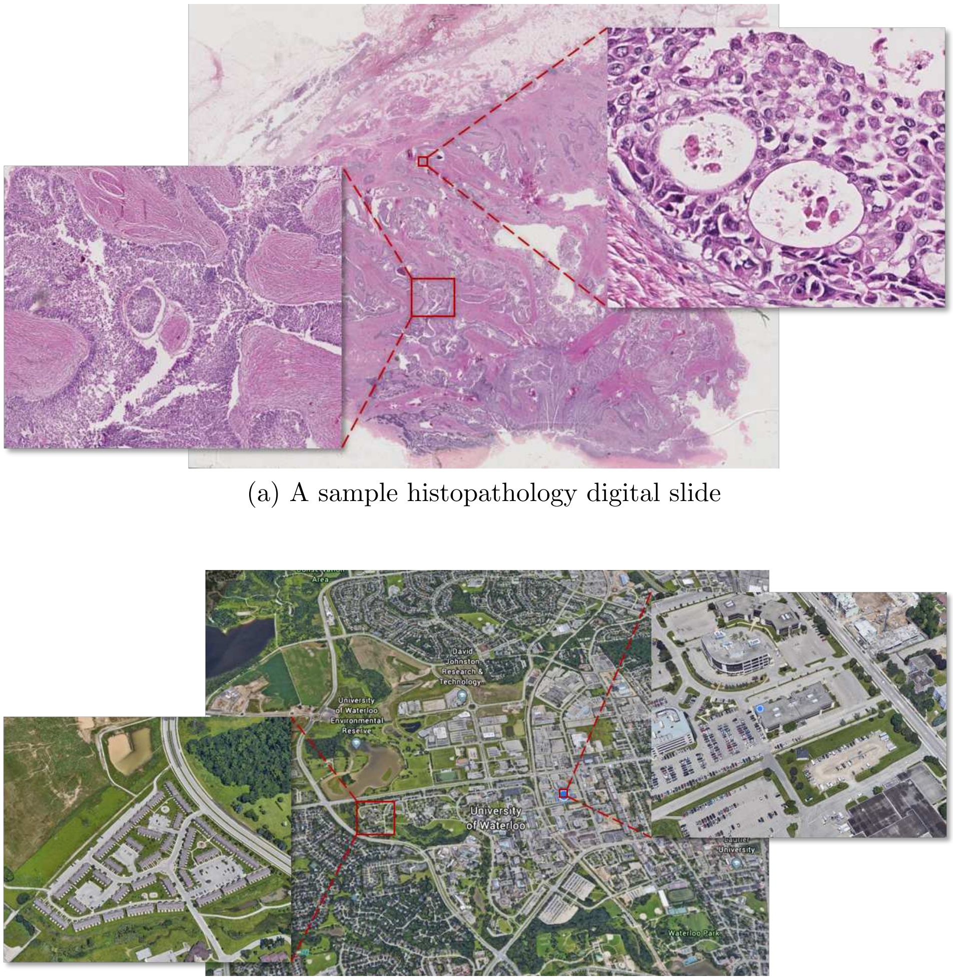
1.3.1 Current State of histodiagnosis

Researchers in histopathology study and identify the correlation between the manifestation of disease and the presence of specific histological patterns. Upon statistical verification of their analysis, researchers present their findings to a professional community as a peer-reviewed publication. Eventually, over the course of time, these *new findings* are established as the "truth." Now, the subsequent task for pathologists around the world (highly disparate regarding their training, practices, and experiences) is to apply these results to their diagnosis routines while carefully following all the guidelines in the peer-reviewed paper. This entire process can have many sources of errors leading to misdiagnosis.

The current practice of histodiagnosis usually involves a pathologist examining tissue slides and rendering a report. Depending on the regulations followed within the pathology facility, another pathologist may or may not verify the observations of the first pathologist. The report from the pathologist facility is given to the referring clinician, who may or may not ask for further verification of the findings. A single mistake in these interrelated events is critical and can cause undesirable harm to the patient.

It is important to mention that pathology laboratories are under tremendous pressure to meet increasingly complex demands from hospitals. As many diseases (such as cancer) continue to grow, complexity and number of pathology tests have simultaneously increased[1]. Therefore, it is needed that pathologists work as efficient as possible to support the consistent quality of patient care.

[1] Cancer is expected to increase 40% by 2030 within Canada, Canadian Cancer Statistics, 2015


David Johnston Research & Technology
University of Waterloo Environmental Reserve
University of Waterloo
Waterloo Park

(a) A sample histopathology digital slide

(b) The City of Waterloo in Ontario, Canada

Figure 1.1: An exemplary to acquaint readers with an enormous resolution of a regular histopathology digital slide. **(a)** is a sample histopathology digital slide with two highlighted regions under different magnification levels; less magnified region shows multiple tissue types and more magnified area shows individual nuclei of a single tissue type, similarly **(b)** shows map of entire city of Waterloo, ON, Canada taken from Google Earth; less magnified area is entire student's residential complex at University of Waterloo and more magnified area is single office building in university. The two images are of not exact same resolutions, insight of the figure is that intricacies within digital slides are at the same scale as locating a single building in city.

One of the important issue that is usually not discussed in quality assurance of histopathology is that errors occur even at the time of the research establishing a particular diagnostic procedure. In fact, critiquing a research work in histopathology can be very difficult if it is one's opinion against other without having a method more objective than the one already available. This discussion brings us to conclude that there is a strong need to quantify the salient histological patterns and structures used by pathologists, whether they are during the research or at the time of actual diagnosis. If machines are taught to recognize the same histological-markers as pathologists then research and diagnosis in histopathology can become more reliable and assertive.

With increase in application of DP for research, teleconsultation, and external quality assurance practices [15]; the amount of digital histopathology data has simultaneously increased. In this context, the thesis research work is pursued to apply computer vision and ML algorithms for the quantitative image-based analysis of DP images. The idea is that a complex arrangement and patterns of pixels in digital slides tie in with the semantic cues used by experts in the field, i.e., pathologists. Since most of the current diagnostic pathology is based on the subjective (but educated) opinion of pathologists, there is an urgent need for quantitative image-based assessment. The machine extracted features are not only important from a diagnostic perspective but they also facilitate the underlying reasoning for rendering a specific diagnosis (driven by existing medical knowledge).

1.4 Thesis Objectives and Contributions

The central objective of the thesis is to provide a qualitative assessment of three different image analysis techniques — Local Binary Pattern (LBP), Bag of visual Words (BoW), and Convolution Neural Network (CNN). The experiments are designed to quantify the quality of the image analysis methods in their abilities to extract the discriminative image-features from digital histopathology slides, suitable for classification and retrieval purposes.

To some extent, this thesis contributes to the ambitious and long-term goal of biomedical community to integrate AI assistants into primary histodiagnosis. For the widespread acceptance of AI in pathology, it is essential that the underlying image analysis algorithms capture the similar level of semantic-knowledge from digital slides as that of a pathologist.

Content-based Image Retrieval (CBIR) is a prime example of assistive technology in medical fields. One of the clinical use of CBIR is illustrated in Figure 1.2. CBIR is the primary focus for evaluating the image analysis algorithms used in this thesis. The two significant contributions of the thesis are **(i)** an assessment of three image analysis approaches commonly used in the literature, and **(ii)** a new dataset, *Kimia Path24*, specially designed and developed to facilitate the research in classification and CBIR of DP images.

The performance of LBP, BoW, and CNN is evaluated using the images from *Kimia Path24* dataset. The results suggest that all the three techniques are capable of extracting significant image-features from histopathology images. Tuning of different hyperparameters for each of the three methods has a compelling effect on the quality of extracted features. Furthermore, the discriminative power of handcrafted algorithms, such as LBP reaches very close to the deep CNN, based on the benchmarks of *Kimia Path24*. This result is particularly surprising as CNN is more complicated from designing and training perspectives. However, the architecture of CNN offers way more flexibility than other two approaches. Presently, CNN is a prevalent topic of research and there are many existing pre-trained CNN models in the literature. Three different types of CNN models are utilized in this thesis, two of them are famous pre-trained models (namely VGG16 and Inception v3), and the third one is trained and designed from scratch. Other popular networks, such as ResNet-51, U-nets were not tested due to time limitation.

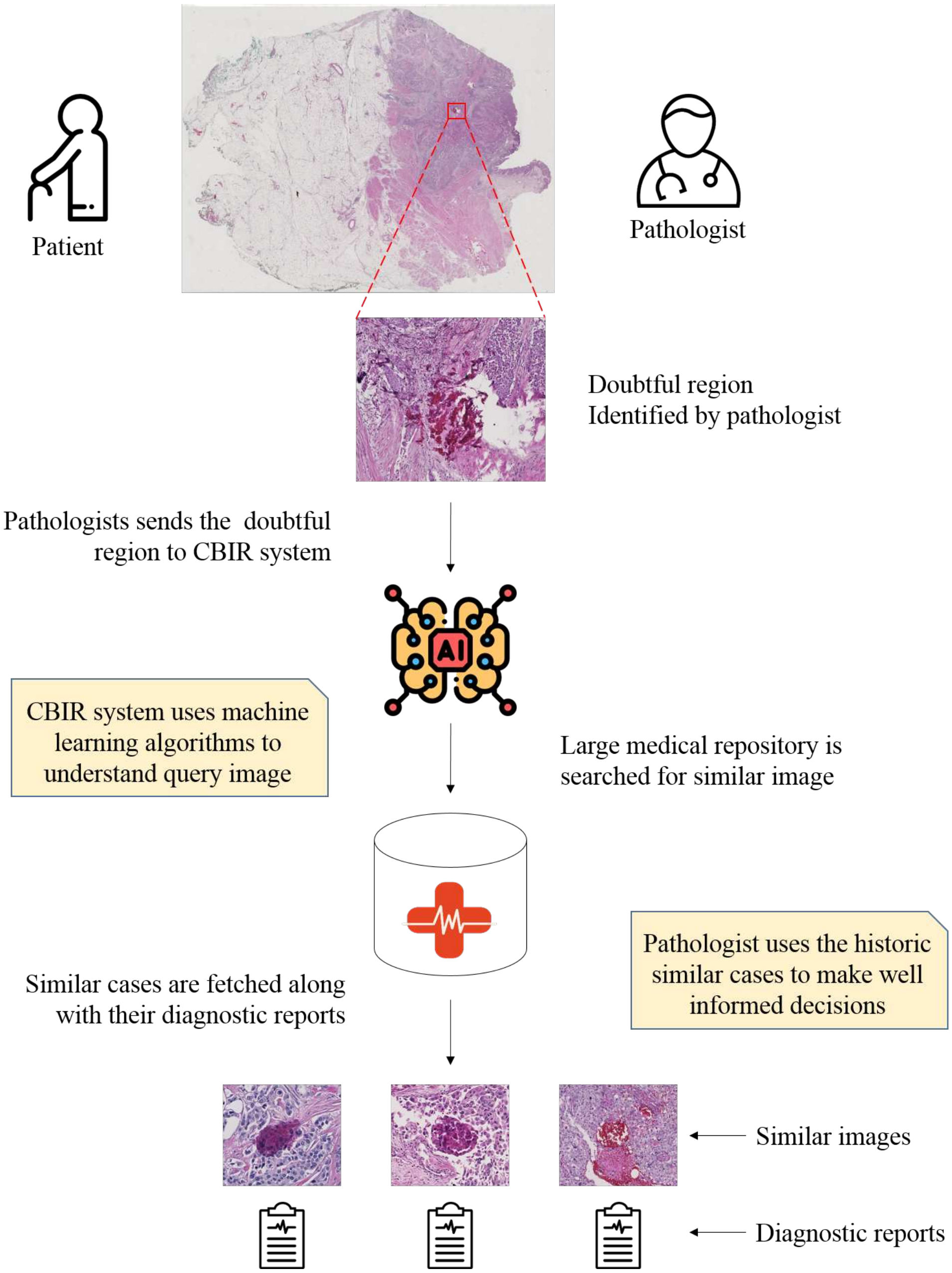

Patient
Pathologist
Doubtful region
Identified by pathologist
Pathologists sends the doubtful
region to CBIR system
AI
CBIR system uses machine
learning algorithms to
understand query image
Large medical repository is
searched for similar image
Pathologist uses the historic
similar cases to make well
informed decisions
Similar cases are fetched along
with their diagnostic reports
Similar images
Diagnostic reports

Figure 1.2: Schematic illustrates a CBIR system provding assistance to a pathologist during a common routine in diagnosis.

Chapter 2

Literature Review

> “*The opposite of a correct statement is a false statement. But the opposite of a profound truth may well be another profound truth.*”
>
> — Neils Bohr

2.1 Introduction

This chapter discusses the four general themes that are essential for the thesis. Firstly, the current state of Digital Pathology (DP) is briefly reviewed from the standpoint of acceptance by pathologists, clinical relevance, as a tool for “future proofing” pathology, precision & reliability, and legality & regulations surrounding its clinical usage. Secondly, to understand different causes of mistakes or misdiagnosis in histopathology, and possible solutions provided by adoption of DP. Thirdly, a literature survey to examine the extent of work done in integrating computer vision algorithms in histopathology fields, and compelling opportunities Machine Learning (ML) brings to the “objective” side of histopathology. Finally, we go back to the “subjective” nature of medical image analysis, and understand the answer for a question like — *can machines replace pathologists?*

These are particularly exciting times for writing this thesis. On April 12th, 2017, USA Food and Drug Administration (FDA) declares the clearance of first ever Whole Slide Imaging (WSI) scanner, Phillips IntelliSite Pathology Solutions, for the primary diagnostic use in the USA[1]. This action of FDA creates the significant milestone in compliance of DP in traditional pathology services in the U.S. It also opens a new horizon in computational pathology for the entire world. With the U.S., many companies and hospitals provide their support to WSI technology, thus, bringing increased funding, industrial collaborations, and higher awareness and interest within research community for computational pathology. In the foreseeable future, diagnostic pathology is expected to undergo a paradigm shift with the focus on comprehensive integration of Computer Assisted Diagnosis (CAD) (just like the current state of CAD in radiology).

2.2 Histopathology Glass Slide Preparation

Before starting a discussion on DP, it is vital to understand the process of preparing pathology glass slides. According to [16], there are four steps involved in the preparation of glass slides before they are digitized or used for diagnosis. These four steps are as follows:

(i) **Collection:** Tissue samples (specimens) are collected from the affected area of a patient using surgery or needle biopsy.

(ii) **Embedding:** Tissues samples are embedded into paraffin wax to allow cutting them into thin sections (sometimes frozen sections are used, e.g., for surgical pathology).

(iii) **Sectioning:** Embedded samples are cut into thin sections with special equipment called "microtome."

(iv) **Staining**[2]**:** Different stains and dyes are used to highlight different components of the "sectioned" tissue. The most common type of staining method is H&E (hematoxylin and eosin)[3].

2.3 Digital Pathology and Whole Slide Imaging

The primary and widely popular imaging technology used in DP is WSI. In fact, the two terms "digital pathology" and "whole slide imaging" have been used interchangeably in

[1] https://www.fda.gov/NewsEvents/Newsroom/PressAnnouncements/ucm552742.htm

[2] Living tissues are colorless, staining aims to reveal cellular components of tissues whereas counter-stains provide contrast for highlighting different biological structures [3].

[3] In H&E staining technique, H is acidophilic and colors cell nuclei in (dark) purple, and E is basophilic and colors extracellular cytoplasm in shades of pink (see §A.1.1 on pp. 89).

the literature. WSI has been long recognized as a research and education tool [17, 11] since its introduction in the early 1980s. However, only very recently (after almost 35 years), the healthcare industry has indicated the increased levels of interest in the total or partial adoption of DP for diagnostic purposes [18]. This section contains the literature survey to provide readers with insight into the current state of WSI in real-world clinical practices, and its potency for replacing the traditional light microscopy.

2.3.1 WSI Scanners

WSI, also commonly referred to as *virtual microscopy*, is an imaging technology used for digitization of a regular glass slide to a *digital slide* or *whole-slide image*. The digital slides are used for viewing by humans through specialized software systems, or for performing digital image analysis. WSI scanner (a device used for WSI) uses robotic microscopes to scan glass slides. It employs a sophisticated software system to stitch different pieces of scanned images from the glass slide into a composite digital image, i.e., whole-slide image or digital slide [17].

Recent advancements in image acquisition and control systems have resulted in significant improvements in WSI technology. The enhanced capabilities of WSI scanners include, reduction in average time for scanning glass slide (about few minutes per slide), and autonomous processing of up to 300 glass slides [19, 20, 21].

Nowadays, WSI scanners are highly portable, generally, set-up on the table-tops within premises of diagnostic centers. The modern WSI scanners produce digital slides in the time efficient manner, often automating all intermediate steps, such as localization of tissue, and focus plane selection [22].

2.3.2 WSI Files and Format

WSI files generated by WSI scanners (digital slides or whole-slide images) are often much larger than other typical modalities of medical images [17]. Generally speaking, resolution of a single WSI is more than $50,000 \times 50,0000$. Even with proprietary encryption and compression, size of each WSI file is around 1–4 GB.

Unlike a conventional digital image file, which usually contains a single static view, the whole-slide image is comprised of multiple "tiles" of image-data arranged in a pyramid-like structure [11, 23], as shown below:

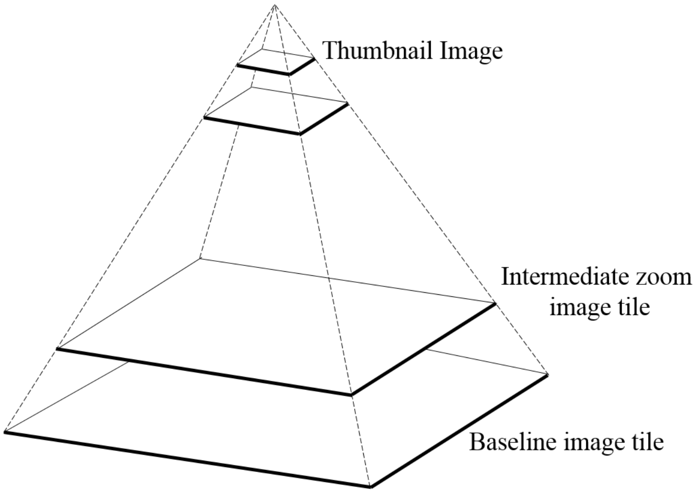


Figure 2.1: Tiles (image data) of different resolutions arranged as "pyramid" in a WSI file. The base tile contains a image data of highest resolution whereas topmost tile is a thumbnail-sized image.

The bottom tile is of the highest resolution (e.g., ×20) whereas the top tile is a thumbnail-sized image for entire whole-slide image. The magnification of ×20 is the most commonly used in WSI, sufficient for most of the diagnostic cases [24, 22]. The WSI scanners with magnification levels of ×40, ×60, and even ×80 are available [25]. However, the real-world usage of WSI scanners beyond ×20 magnification levels is sporadic [22].

The pyramid structuring of tiles within a WSI file makes it apt for simulating virtual microscopy. These files are viewed using the specialized software (usually vendor specific) systems that can appropriately determine the tile based on the magnification level selected by a user. These software systems provide fluid interaction and user-friendly interfaces to visualize whole-slide images [20, 22].

The primary advantage of WSI technology is — WSI files can be accessed remotely, e.g., over the internet [17]. The remote access to WSI files allows pathologists to assess, consult and diagnose cases remotely. The remote consultation of pathology cases is also known as *teleconsultations*. WSI technology is becoming progressively robust with facilities, such as virtual three-dimensional microscopy [26], often necessary for studying and analyzing cell structures [20].

2.3.3 Current State of WSI

This section contains a brief literature survey on the current state of WSI technology.

Acceptance of WSI by pathologists: The literature suggests that there is a widespread reluctance in the adoption of WSI technology by pathologists [20, 27]. This reluctance is acting as a barrier to the broad adoption of DP in real clinics. Various factors come into play for pathologists to prefer traditional glass slides over digital slides, some of the factors include — high setup cost for WSI related equipment and infrastructure, limited control of focus in viewing digital slides (compared to the light microscopy), uncertainty about the quality of digital slides versus traditional glass slides, and inexperience of pathologists in using WSI related software systems [18].

Validation of WSI as a tool for primary diagnosis: The major factor restricting the complete adoption of WSI for primary diagnosis is not legal regulations but rather the suspicion of its quality. *Can digital slides compete with original microscopic glass slides regarding efficiency and reliability for the primary diagnostics?* This question is of the significant concern within pathology community. Several studies demonstrate the non-inferiority of WSI [28, 22, 29, 30]. For many years, WSI technology is used in Canada and Europe for primary diagnosis purposes. Majority of the data on real-world deployment of DP comes from these two countries [31, 32], which further assures the efficacy of WSI technology as the tool for primary diagnosis.

Recently, U.S. Food and Drug Administration (FDA) conducted one of the largest and most comprehensive investigations thus far on the evaluation of WSI scanners developed by Phillips [33]. This inquiry by FDA has resulted in the first-ever clearance of a WSI scanner for primary diagnostic use in the U.S., provides compelling evidence in favor of WSI technology.

Clinical studies on the efficacy of WSI: An article [34] suggests that adoption of DP resulted in up to 13% increase in diagnostic efficiency of pathologists. This increased efficiency is a cumulative outcome due to the efficiency gains from the organization, querying, searching, and consultation of pathology cases digitally rather than handling & managing fragile glass slides [34].

Interestingly, literature also contains experiences of the real pathology laboratories around the world that have adopted "fully digital" workflow, revealing the positive outlook towards DP as "promising" technology [28, 22]. Like any other technology, WSI has its pros and cons — some major ones are listed in Table A.1 on pp. 90 along with their supporting citations from literature. The adoption of DP seems inevitable with the ever-increasing demand of healthcare sector. DP can offer a platform for improving the service efficiency and effectiveness of diagnosis in histopathology. In fact, it has been called vital for “future-proofing” of diagnostic pathology [18].

Remarks: From the literature review on the current state of WSI, it is concluded that WSI technology is not any lesser than traditional light microscopy. Full or even partial adoption of digital workflow in pathology has many gains to hospitals — positive influence from the patient’s security, service quality, training of new pathologists, management, and even strategic (e.g., turnaround time, and increase capacity) perspectives without adversely affecting the diagnostic accuracy and efficiency. However, at present, use of DP is mostly widespread for education, training, and research purposes than for the primary diagnosis in pathology clinics.

2.4 Histopathology Mistakes

Mistakes and errors happen, in pathology as in any other field. There are various sources of errors in diagnostic pathology. According to [35], pathology errors occur during the three phases of diagnosis, as follows:

(i) **Pre-analytic phase:** Test selection by a referring clinician and sending the specimen to a pathology lab.

(ii) **Analytic phase[1]:** Preparation of a glass slide and assessment by a pathologist.

(iii) **Post-analytic phase:** The clinician receives and interprets the report and takes required action.

The tasks involved in diagnostic pathology are very complicated and partly subjective, making mistakes and misjudgments inevitable. There are many reasons other than the complexity of tasks that are accountable for histopathology errors. The following paragraphs discuss various reasons for pathology mistakes and explain how DP acts as a promising platform to prevent such mistakes.

Rigorous nature of work in diagnostic pathology: The goal of a diagnostic pathology is to render the *complete* and *correct* diagnosis in an appropriate and timely manner [35]. Both, punctuality and correctness (though subjective) are two significant attributes of pathologist’s daily conducts; departing from the internal restrictions set by a healthcare institution, pathologists may face legal implications.

[1] For this thesis, the discussion of pathology mistakes is limited only to the *analytic phase*.

Increasing workload pressure: A trend in the literature suggests that the errors in diagnostic pathology occur, not only because of the complicated nature of involved tasks but also due to the ever-increasing demand in working environment of pathologists. Pathology laboratories are under tremendous pressure to manage large workload volume, fast turnaround times, and also to train new pathologists [18]. ML-driven DP is, therefore, an alluring platform to reduce pathologists' workload by making their daily routines more efficient and streamlined.

Lack of classification schemes for pathology errors: Most of the errors in pathology diagnosis remain unreported as they are rectified somewhere down the treatment pipeline. Moreover, the ones that get reported are very challenging to quantify. The two mains reasons that make pathology mistakes challenging to quantify are — associated subjectivity within diagnosis and lack of an appropriate error classification scheme [36]. ML methods enable quantitative analysis of DP images, allows a better understanding of the relationship between of different image features and mechanisms of disease process. Therefore, quantitative assessment of DP images facilitates researchers to develop more objective error-classification schemes.

2.5 Applications of ML in Digital Pathology

Histopathology is the "gold standard" for diagnosis of many different diseases including almost all the types of cancer [37]. Due to the increasing pressure in working environment of pathologists, there is an urgent need to remove inefficiencies existing within the current histopathology practices.

Studies show that strategies, such as double-reading, case conferences, and consultations reduce diagnostic variation and interpretations errors during the *analytic phase*, ranging from 1.2 to 50 errors per 1000 cases [38]. This reduced rate of errors in diagnosis suggests that DP combined with ML algorithms can be a robust technology for the better future of histopathology.

Rendering supporting "digital opinions": Most of the errors in pathology are recognized early during the treatment, either by consultation or re-assessment [36]. However, these errors still add upon the inconvenience of patients, and negatively affect the reputation of a healthcare provider. Survey on DP shows that — out of 5000 referral cases that were reviewed by a second pathologist, 11.3% of the reviews had minor or major differences in diagnosis from the original diagnosis, and 1.2% of all the reviews resulted in a change in management of the patient [18]. Furthermore, DP allows easy access to digital slides, therefore, combining DP with ML methods can further facilitate the solicitation of second "digital opinion" for pathologists resulting in more reliable diagnostic decisions.

Automatic prioritization of pending cases: A review article states that *"approximately 80% of 1 million prostate biopsies performed in the U.S. every year are benign; this suggests that prostate pathologists are spending 80% of their time sieving through benign tissue"* [3]. The time "wasted" by pathologists on "obvious cases" can be reduced if an automated system prioritizes the pending cases. ML techniques utilize the knowledge extracted from the large historical cases (already diagnosed and treated) within archives of hospitals, and can automatically infer the "urgency" of un-examined cases. This automatic priority-based sorting of the pending cases can allow pathologists to focus on more critical matters first.

Capitalizing upon growing digital archives: Utilization of digital platform for handling pathological information eases institution-wide communication among pathologists. Furthermore, it facilitates pathologists to write their comments and annotations on a central digital platform which is integrated with hospital's digital archives [39, 18]. Access to the growing digital information containing the annotations and notes from the real pathologists are highly beneficial for supervised training of ML algorithms. Over the time these ML tools can become "smarter" and better at providing digital assistance to pathologists.

2.6 Machine Learning for Histopathology

This section discusses recent developments in image analysis tools and ML techniques used in DP images from feature extraction, content-based retrieval, segmentation, and tissue classification perspective.

Contrary to popular belief, one of the earliest pursuits in the adoption of digital image analysis was not for the face recognition, but rather for the study of medical images [40]. A survey [3] states that the widespread use of CAD can be traced back to the development of digital mammography during the early 1990s. In fact, CAD is now integral to many clinical routines for diagnostic radiology and recently becoming imminent in diagnostic pathology as well.

With an astounding increase in the workload of pathologists, there is compelling need to integrate the CAD systems for pathology routines [41, 40, 42, 3]. Researchers in both image analysis and pathology fields have recognized the importance of quantitative analysis of pathology images using ML techniques [3]. With continuous advancement of WSI scanners and their proliferation in clinics and laboratories (§2.3.1), this has resulted into a substantial accumulation of histopathology images, justifying the increased demand for their analysis for improvement of the current state of diagnostic pathology [40, 41].

2.6.1 Image Analysis in Digital Pathology

In DP, the large dimensionality of images pose a challenge for both computation and storage; hence, contextually understanding regions of interest in images helps the faster diagnosis and detection for implementing soft-computing techniques [43]. Over the years, traditional image-processing tasks, such as filtering, registration, and segmentation, classification and retrieval have gained more significance.

Particularly for histopathology, cell structures, such as cell nuclei, glands, and lymphocytes are observed to hold prominent characteristics that serve as hallmarks for detecting cancerous cells [44]. Researchers also anticipate that one can correlate histological patterns with protein and gene expression, perform exploratory histopathology image analysis, and perform CAD to provide pathologists with required support for decision making [44]. The idea behind CAD to quantify spatial histopathology structures has been under investigation since the 1990s, as presented by Wiend et al. [45], Bartels et al. [46], and Hamilton et al. [47]. However, due to limited computational resources, implementing such ideas have been overlooked or delayed.

More recently, Bankhead et al. [48] provided open-source bio-imaging software, called *QuPath* that supports WSI by giving tumor identification and biomarker evaluation tools which developers can use to implement new algorithms to improve the further outcome of analyzing complex tissue images.

2.6.2 Image Retrieval

Retrieving similar (visual) semantics of image requires extracting salient features that are descriptive of image content. In its entirety, there are two main points of view for processing whole-slide images [49]. First one is called *sub-setting methods* which consider a small section of large pathology image as essential part such that processing of small subset substantially reduces processing time.

Majority of research-work in literature prefers *sub-setting* method because of its advantage of speed and accuracy. However, it needs expert knowledge and intervention to extract proper subset. On the other hand, *tiling methods* break images into smaller and controllable patches and try to process them against each other [50] which naturally requires more care in design and is more expensive in execution. However, it indeed is a distinct approach toward full automation.

Traditionally, the extensive medical image database is packaged with textual annotations classified by specialists; however, this approach does not perform well against ever demanding growth of DP. In 2003, Zheng et al. [51] developed online Content-based Image Retrieval (CBIR) system wherein the client provides query image and corresponding search parameters to the server side. The server then performs similarity searches based on feature types, such as color histogram, image texture, Fourier coefficients, and wavelet coefficients, while using vector dot-product as a distance metric for retrieval. The server then returns images that are similar to query image along with similarity scores and feature descriptor.

Mehta et al. [52], on the other hand, proposed offline CBIR system which utilizes sub-images rather than entire digital slide. Using scale-invariant feature transform (SIFT) [53] 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 stained pathology images (immunohistochemistry), consisting of 8 resolution levels. In 2012, Akakin and Gurcan [54] developed 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. [55] developed scalable CBIR method to cope with WSI 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 whole-slide images from 116 patients; experimental results state accuracy of 88.1% for processing at a speed of 10ms for all 800 testing images.

2.7 Final Remarks

Authors of [56] hypothesize that continuous developments and innovation in ML combined with advances in raw computing power herald an age where well-designed Artificial Intelligence (AI) can be significantly used for complicated diagnostic interpretations of medical images. With use of advance AI, recently, Google's AlphaGo AI defeated a high-profile player[1], Lee Sedol, with the score of 4-1 in the game of Go (hugely complex ancient strategy game). The AI of AlphaGo established the case that machines are smarter than humans, at least in complex strategy games. The algorithm of AlphaGo's AI is inspired by the design of biological brain; such algorithms belong to the division of ML known as Deep Learning (DL). The original paper describing the architecture of AlphaGo is published in the *Nature* [57].

Besides all the accomplishments of ML, an article by Su. J [58] reveals that there are a lot of shortcomings of these DL models, such as their high sensitive to tiny perturbations; small yet specific changes to input images may "fool" even the most current state-of-the-art DL models, i.e., the optical illusions for machines. Now, the question arises — *can we replace pathologists with AI?* To this question, the answer at present times is — *No, we cannot.* In fact, the question itself is an erroneous comparison between two very dissimilar activities, i.e., high-level cognition (a human forte) versus high-level computation (an AI forte, at least for now) [59]. G. Sharma in [59] states that a diagnosis is well-thought-out cognitive opinion, encompassing years in training and experience subjected to high levels of heuristics and biases. Therefore, it is not moral to leave the crucial decisions in diagnosis entirely to machines. However, it is more favorable to use machines as an assistive tool that leads to increased reliability in outcomes of critical clinical decisions.

In this thesis, CBIR of pathology images are studied as an approach for effective and efficient histodiagnosis. CBIR is a crucial research area that involves analysis and interpretation of medical information (e.g., patterns of cells and tissues) for reliable patient diagnosis. The next chapter explains different components of a CBIR system.

[1] https://deepmind.com/research/alphago/alphago-korea/

Chapter 3

Content-based Image Retrieval

> “ *Sooner or later all things are numbers, yes?* ”
>
> — Terry Pratchett

3.1 Introduction

Content-based Image Retrieval (CBIR) is one of the most critical fields in computer vision since the last decade. It allows a user to query an image and retrieve the similar images from a vast repository of images. CBIR has many practical applications in the real-world, and it is particularly a useful technology for medical images, since textual features extracted from medical reports are often not the adequate representation of the content of the associated medical images [60, 61, 62].

Figure 3.1 shows an interaction among different components a CBIR system. A vast repository containing images $I_1, I_2, \dots I_n$ is fed into an image descriptor yielding feature vectors $F_1, F_2, \dots F_n$. These feature vectors are indexed (i.e., using red-black trees or hashing algorithms) based on the pair-wise distances calculated with a distance metric. Eventually, searching the images similar to a given query image I_q is a two step process. Firstly, the query image I_q is transformed into a feature vector F_q using the same image descriptor. Finally, the feature vector F_q is used to search the “closest” (smallest distance) image within the indexed database, i.e., the most similar image.

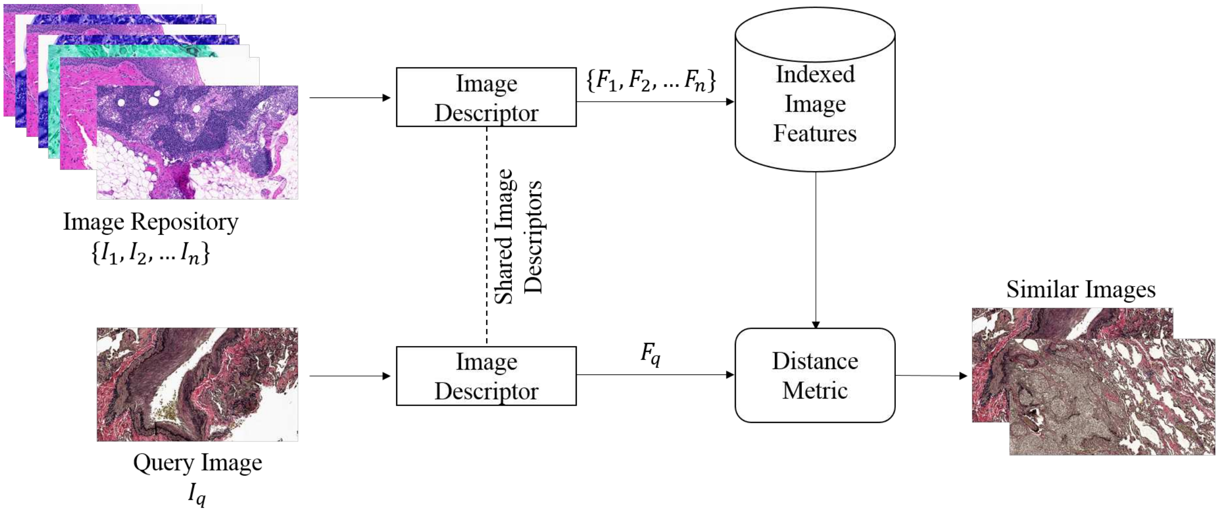


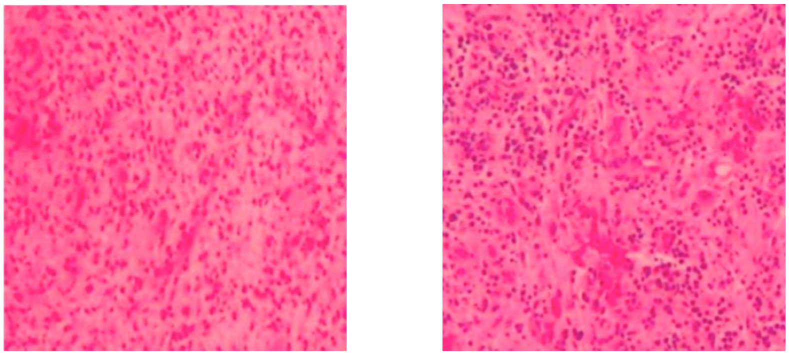
Figure 3.1: Illustration of the steps followed by a CBIR system.

Two main components that determine efficacy of a CBIR system are **(i)** *image descriptor* or technique to extract a vector-representation of an image, and **(ii)** *distance metric* used for comparing the similarity between two vectors (representations of images obtained from the image descriptor). An ideal distance metric should yield a larger value by comparing two different images than by comparing similar ones. At the same time, an ideal image descriptor should capture crucial indicators within an image summarizing its content.

3.1.1 Types of *similarity* between Images

The "similarity" among images can be expressed in two aspects **(i)** visual and **(ii)** semantic. Two images are visually similar if majority of their characteristics look identical to an observer (e.g., color, shape, and texture). On other hand, the semantic similarity between two images captures the similarity from the perspective of an expert in the field (e.g., pathologists for histopathology).

Two semantically similar images are likely visually similar but the opposite case may not be necessarily valid. One of the example is illustrated in Figure 3.2 — shows cancerous and non-cancerous images of the same tissue type (3.2a, 3.2b), visually same but vary a lot in the structural organization of their cells.



(a) Cancerous

(b) Non cancerous

Figure 3.2: Exemplar of the visually similar but semantically different patches from the brain tissue. The image in **(a)** is cancerous (gliomas) whereas, the image in **(b)** is inflamed but non-cancerous. Both the images are adapted from [3].

3.1.2 Properties of useful CBIR

A useful CBIR system must employ an image descriptor capable of extracting the semantically significant visual features from the images. At the same time, selection of a distance metric is equally essential to exploit the synergy between two.

Digital Pathology (DP), in particular, is challenging domain for CBIR as subtle and localized differences, semantically discriminate pathology images. Moreover, pathology images are captured at gigapixel resolutions, therefore, exhibit a considerable variability of visual features which are more difficult to obtain than the natural images (e.g., plenty of edges, intricate structures, and high gradient changes). The semantics of pathology images and natural images are at two opposites ends, the pathology images are discerned only by medical specialists whereas the natural images are seen all around us.

3.2 Image Descriptor

The *image descriptor* is a uni-variate operator that transforms a given image $I_q \in \mathcal{X}$ into a d-dimensional vector $F_q \in \mathbb{R}^d$ such that $d \ll |\mathcal{X}|$ and the feature vector F_q is representative of the image I_q in $\mathbb{R}^d$.

What makes a useful *image descriptor*? An image descriptor plays a crucial role in building an accurate and functional CBIR system. Ideally, an image descriptor should output a feature vector F_q that most precisely discriminates between the various semantic and visual markers of an image. For CBIR in a histopathology domain, it is crucial that an image descriptor should focus on the same bio-markers as used by the pathologist during diagnosis. For example, a "good" image descriptor would be capable of discriminating between two semantically different images (3.2a and 3.2b) in Figure 3.2.

In this thesis, three popular image descriptors are explored — **(i)** Local Binary Pattern (LBP), **(ii)** Bag of visual Words (BoW), and **(iii)** Deep Image Descriptors (i.e., Convolution Neural Network (CNN)). All the three image descriptors are well-established and extensively studied techniques in computer vision fields, and offer the contrasting differences regarding their training and quality of the extracted features.

Feature Extractor	Trainable?	Defined
Local Binary Pattern (LBP)	No	§5.2, pp. 43
Bag of visual Words (BoW)	Yes	§5.3, pp. 47
Deep Descriptors	Partially yes	§5.4, pp. 52

Table 3.1: Image descriptors used in this thesis.

In the table above, CNN approach is denoted as "Partially yes" because CNN allows three type of feature extraction schemes for a given problem **(i)** pre-trained CNN models (already trained on existing datasets not necessarily from the same domain as the given problem), **(ii)** fine-tuned CNN models (already trained but fine-tuned with the images from the current problem), and **(iii)** training a CNN model from scratch.

3.3 Distance Measures

A *distance* $\boldsymbol{d} : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}^+$ is a bi-variate operator (i.e., its takes two arguments, e.g., $x \in \mathcal{X}$ and $y \in \mathcal{X}$) and outputs a value in $\mathbb{R}^+ = [0, \infty]$.

What makes a useful *distance measure*? A useful distance measurement must capture the "right" characteristics of input vectors x and y (e.g., χ^2 distance is better suited for the histograms than any other distance measurements). The distance measurement is the critical choice for optimal performance of a CBIR system. The wrong decision of a distance metric can result in the sub-optimal performance of CBIR (even for the highly optimized image descriptors). For this thesis, three distance metrics ℓ_1, ℓ_2 and χ^2 (chi-square) are used for conducting all the CBIR related experiments. These three distance metrics are explained in the next following section.

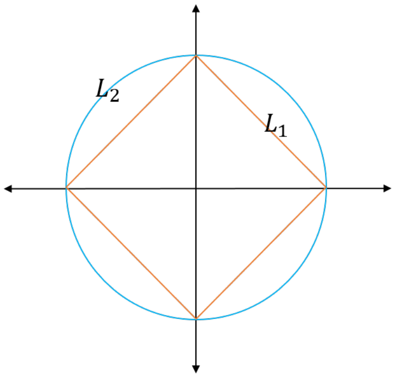


Figure 3.3: Unit balls in $\mathbb{R}^2$ for ℓ_1 (orange) and ℓ_2 (blue) distance metrics

3.3.1 Distance Calculations

3.3.1a. ℓ_1 Distance

Consider two vectors $x = (x_1, x_2, ..., x_d)$ and $y = (y_1, y_2, ..., y_d)$ in $\mathbb{R}^d$ then, ℓ_1 distance $\boldsymbol{d_{\ell_1}}$ between x and y is calculated as

$$\boldsymbol{d_{\ell_1}}(x, y) = \| x - y \|_{\ell_1} = \sum_{i=1}^{d} (x_i - y_i) \tag{3.1}$$

The ℓ_1 distance is also known as "Manhattan" distance since it is a sum of lengths on each coordinate axis; distance for walking in a planned city like Manhattan with straight pedestrian roads forming a virtual coordinate axis system.

3.3.1b. ℓ_2 Distance

ℓ_2 distance ($\boldsymbol{d_{\ell_2}}$) is interpreted as the *Euclidean* or "ordinary" straight-line distance between two vectors. For two d-dimensional vectors (i.e., x, and y), $\boldsymbol{d_{\ell_2}}$ between them is calculated as

$$\boldsymbol{d_{\ell_2}}(x, y) = \| x - y \|_{\ell_2} = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2} \tag{3.2}$$

Figure 3.3 shows the unit balls for ℓ_1 and ℓ_2 in orange and blue color respectively. Both unit balls touch points, a unit distance from the origin along each of the axis. However, the unit ball for ℓ_1 distance is smaller than ℓ_2 distance.

3.3.1c. χ^2 (chi-squared) Distance

Chi-squared distance $\boldsymbol{d_{\chi^2}}$ is a distance between two histograms, $x = [x_1, .., x_d]$ and $y = [y_1, ..., y_d]$, each with d bins, calculated as

$$\boldsymbol{d_{\chi^2}}(x, y) = \sum_{i=1}^{d} \frac{(x_i - y_i)^2}{(x_i + y_i)^2} \tag{3.3}$$

Moreover, both histograms must be normalized such that their entries sum up to one. χ^2 distance is often used in computer vision problems for computing distances between the histogram representations of images. The Name of the distance is derived from Pearson's chi-squared test statistic (used for comparing discrete probability distributions).

3.4 Summary

This chapter reviewed the essential components of a CBIR system, i.e., image descriptor (§3.2) and distance measure (§3.3). An image descriptor convert a given image into a feature vector, whereas a distance measure is used to calculate the "distance" between two vectors. The right choice of both the components are crucial for optimal functioning of a CBIR system. The different distance calculations used in the thesis were discussed in §3.3.1. The three image descriptors used for this thesis are LBP, BoW, and deep descriptors (see Table 3.1). Deep descriptors are based on CNN (the specialized deep networks for extracting descriptive image-features). The next chapter discusses various essential architectural components used for designing CNN.

Chapter 4

Convolution Neural Networks

> “*Never worry about theory as long as the machinery does what it’s supposed to do.*”
>
> — Robert A. Heinlein

4.1 Introduction

Experiments in this thesis are conducted using the deep networks specialized for classifying and extracting discriminative local features from images, known as Convolution Neural Network (CNN). The chapter discusses the background and theory of CNN.

Deep Learning (DL) overview: DL is a sub-field of Machine Learning (ML) based on the algorithms inspired by structure and functions of the biological brain, known as Artificial Neural Network (ANN). CNN is a type of ANN specialized for data with spatial information (e.g., images). Five major reasons for success of Deep Learning (DL) over the last few years are **(i)** state-of-the-art achievements in the field of computer vision, natural language processing, visual reasoning, and voice recognition, **(ii)** *transfer learning* or reusability of the learned parameters in a different domain, **(iii)** radical improvement in chip processing abilities (e.g., GPUs), **(iv)** reduction in cost of data storage and computational hardware, and **(v)** acceptance within the industry which brings generous investments and populous community of researchers.

CNN overview: A standard reference for the invention of CNN's architecture and its training with *backpropagation* is credited to the paper by LeCunn et al. titled *"Object Recognition with Gradient-Based Learning"* [63]. Before that time, an architecture very similar to CNN known as *Neocognitron* existed, introduced during the early 1980s by Fukushima in [64]. Neocognitron did not receive much attention due to lack of training algorithm (e.g., backpropagation). Neocognitron was based on the idea of *simple* and *complex cells*; where the simple cells perform convolution and the complex cells, *average pooling*, akin to the operations in contemporary CNN architectures. It is fair to say that, CNN's architecture is the result of achievements in several research areas, such as graphical models, neural networks, pattern recognition, optimization, digital signal processing, feature engineering, and bio-inspired intelligent systems.

Like many machine learning algorithms, CNN takes the inspiration from biological systems. Existing literature suggests that the architecture of CNN mimics a human visual system; both use a confined receptive field and multi-layered processing pipeline that continuously extracts features with higher abstractions by going deeper into the layers [65].

4.2 Chapter Organization

Some of the preliminary concepts related to CNN's architecture, such as layer, loss function, regularizer, and gradient descent, are discussed in the next section (§4.3). The Forward Pass (FP) and Backward Pass (BP) are two essential concepts in functioning and training of CNN (§4.4).

The architecture of CNN is composed of multiple layers; four types of layers are germane to this thesis **(i)** ReLU layer is a non-linearity layer (§4.6), **(ii)** convolution layer is the most crucial layer for CNN's functionality as the name implies (§4.7), **(iii)** pooling layer is a subsampling layer that makes training CNN computationally in-expensive (§4.8), **(iv)** *Dropout* layer is a regularization technique that penalizes connections between neurons to prevent *over-fitting* (§4.10).

4.3 Preliminary

Below is the summary of high-level architectural components of CNN.

(i) **Layer:** A differentiable operation (parameterized or non-parameterized). CNN is a multi-layered structure with each layer capturing the different abstraction of an image.

(ii) **Error or loss function:** A differentiable function. The goal of CNN during the training phase is to optimize its *loss function* by changing parameters of its layers.

(iii) **Regularizers:** A form of penalties on the layer's parameters or its activities. Regularizers are incorporated into the loss function or as a separate layer.

(iv) **Stochastic Gradient Descent (SGD):** An algorithm that changes parameters of all the layers to achieve an optimal value of the loss function.

The architecture of CNN is interpreted as multiple layers stacked in a feed-forward manner with a distinct loss function and some regularizers (combined into a single global loss function), optimized by changing the parameters of its layers using SGD.

4.4 CNN in Nutshell

CNN is a composite structure built from different types of *layers* connected to each other in a feed-forward manner. The input of CNN is usually a 3^{rd} order tensor, e.g., an image. The output of CNN is its final prediction and the size of output is a design choice (dependent on the problem handled by CNN). Two fundamental concepts in functioning and training of CNN are as follows:

- Forward Pass (FP) is an operation that computes the output of layer. An output of intermediate layer becomes the input to the next layer. The output of last layer constitutes the final prediction of CNN.

- Backward Pass (BP) updates internal parameters of a layer to optimize the loss (the discrepancy between the computed and desired output).

CNN performs FP during the prediction phase whereas conducts both during the training phase, i.e., FP to predict the output and then BP to update its parameters. Some layers (e.g., activation layers and pooling layers) do not have any internal parameters yet they have well-defined FP and BP.

4.4.1 Forward Pass (FP)

FP is an operation that computes the output for any given input. The FP of CNN is a chain of FP computed at each layer (starting from the first till the last, i.e., feed-forward manner). FP of the last layer yields the final output of CNN, as shown below:

$$\begin{gathered} x^1 \rightarrow \boxed{w^1} \rightarrow x^2 \rightarrow \cdots \rightarrow x^N \rightarrow \boxed{w^N} \rightarrow z \\ \text{where,} \\ \begin{aligned} x^i &: \text{input to } i^{th} \text{ layer,} \\ w^i &: \text{parameters of } i^{th} \text{ layer} \\ z &: \text{final output of CNN} \end{aligned} \end{gathered} \tag{4.1}$$

Transforming x^1 to x^2 is FP of layer 1, and obtaining the final output (prediction) z from the given input x^1 is FP of entire CNN.

4.4.2 Backward Pass (BP)

In supervised learning, an input x^1 has a desired output or ground truth $\hat{z}$. The goal of BP is to reduce the discrepancy between the final output (prediction) z and the desired output (ground truth) $\hat{z}$. The discrepancy is quantified using a *loss function* E, also known as the *error function*.

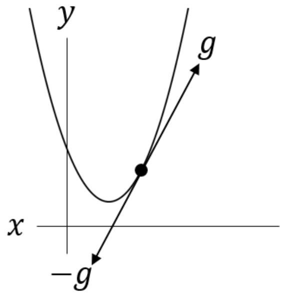


Figure 4.1: Visual interpretation of gradient descent (4.3). g represents the direction of gradient.

The final output z is based on the parameters of layers in CNN; these parameters are collectively denoted as ϕ where $\phi = \{w^1, w^2, ..., w^N\}$. Now, the training of CNN is formulated as the following minimization problem:

$$\phi = \arg\min_{\phi} E(z, \hat{z}). \tag{4.2}$$

For minimizing $E(z, \hat{z})$, parameters of each layer are updated using *gradient descent* given by

$$w^i \leftarrow w^i - \eta \frac{\partial E(z, \hat{z})}{\partial w^i} \tag{4.3}$$

The previous equation (4.3) updates the parameters w^i of i^{th} layer in the direction opposite to the gradient ($\frac{\partial E}{\partial w^i}$). The update value is scaled by a parameter know as learning rate η. The gradient descent is illustrated in Figure 4.1.

The process of updating the parameters of a layer in order to attain an optimal loss value is known as *parameter learning*. The parameter learning of i^{th} layer requires a gradient $\frac{\partial E}{\partial w^i}$ (4.3) which is calculated by BP using the following two gradients:

(i) $\frac{\partial E}{\partial w^i}$ is used to update the parameters w^i of i^{th} layer according to (4.3), and

(ii) $\frac{\partial E}{\partial x^i}$ is not used by i^{th} layer itself. However, it is passed to the previous layer $(i-1)^{th}$ as a prerequisite for the gradient calculations of that layer.

The flow of gradients during BP is illustrated below

$$\boxed{w^1} \xleftarrow{\frac{\partial E}{\partial x^2}} \boxed{w^2} \xleftarrow{\frac{\partial E}{\partial x^3}} \cdots \xleftarrow{\frac{\partial E}{\partial x^N}} \boxed{w^N} \xleftarrow{\frac{\partial E}{\partial z}} \tag{4.4}$$

The i^{th} layer simplifies the calculation of gradient $\frac{\partial E}{\partial w^i}$ by multiplying two easy-to-obtain gradients, i.e., $\frac{\partial E}{\partial x^i}$ and $\frac{\partial y^i}{\partial w^i}$. This simplification is derived by *chain rule* as follows:

$$\begin{aligned} \frac{\partial E}{\partial x^i} &= bp_{i-1} \\ &= \frac{\partial E}{\partial x^{i+1}} \cdot \frac{\partial x^{i+1}}{\partial x^i} \\ bp_{i-1} &= bp_i \cdot \frac{\partial y^i}{\partial x^i} \end{aligned} \tag{4.5}$$

$$\begin{aligned} \frac{\partial E}{\partial w^i} &= \frac{\partial E}{\partial x^{i+1}} \cdot \frac{\partial x^{i+1}}{\partial w^i} \\ &= bp_i \cdot \frac{\partial y^i}{\partial w^i} \end{aligned} \tag{4.6}$$

where,

bp_i : gradient flowing into i^{th} layer from $(i+1)^{th}$ layer,
y^i : output of i^{th} layer or input to $(i+1)^{th}$ layer x^{i+1}, and
w^i : parameters of i^{th} layer

In the above equation (4.6), the gradient $\frac{\partial E}{\partial x^{i+1}}$ is obtained from the $(i+1)^{th}$ layer whereas the gradient $\frac{\partial y^i}{\partial w^i}$ is calculated analytically (the layer's output y^i is a differentiable function of its parameters w^i). The i^{th} layer calculates another gradient $\frac{\partial E}{\partial x^i}$ (4.5) which is passed to the $(i-1)^{th}$ layer for gradient calculations of that layer. The very last layer receives the gradient $\frac{\partial E}{\partial z}$ which can be analytically calculated as well. The BP of CNN start at the last layer and continues until the very first layer, updating parameters of each layer in the process.

4.5 Notations

Inputs, outputs and parameters: At the l^{th} layer, the input is denoted as x^l, layer's parameters as w^l and output as y^l. In CNN the output of l^{th} layer becomes the input for the next $(l+1)^{th}$ layer; therefore, y^l is interchangeably written as x^{l+1}.

Feature maps: A given input x^l with shape $H \times W \times D$ is acknowledged as a collection of D feature maps, each with the shape $H \times W$.

Size of input: At the l^{th} layer, the input x^l is of size $H^l \times W^l \times D^l$, and the output y^l or x^{l+1} is of size $H^{l+1} \times W^{l+1} \times D^{l+1}$.

Index notation: A tensor $y_{i,j,d}$ represents a single scalar value at the (i,j) spatial location in the d^{th} channel of a given 3^{rd} order tensor y.

4.6 ReLU layer

Rectified **L**inear **U**nit (ReLU) is a non-linearity layer for deep networks. ReLU operation does not change the size of its input, i.e., x^l is the same size as y^l. For a given input x^l, the output y^l is obtained by FP of ReLU layer, it is computed as follows

$$y^l_{i,j,d} = max\{0, x^l_{i,j,d}\} \tag{4.7}$$

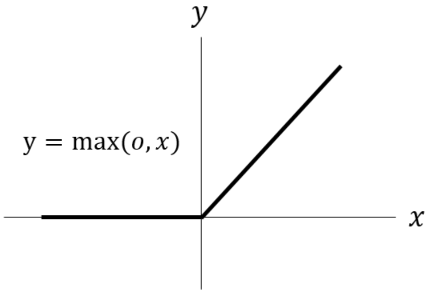
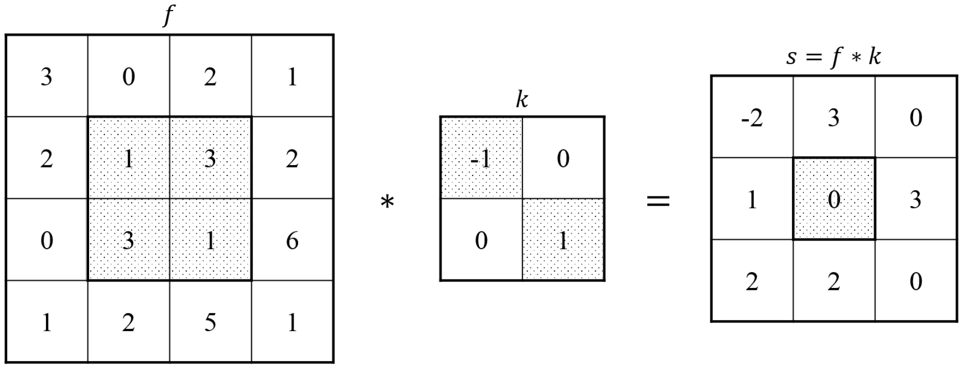


Figure 4.2: ReLU function

ReLU layer does not require any parameter learning since there are no parameters. However, it needs to calculate $\frac{\partial y^l}{\partial x^l}$ as a requirement for BP, calculated as follows

$$\left[\frac{\partial y^l}{\partial x^l}\right]_{i,j,d} = \begin{cases} 1 & \text{if } x^l_{i,j,d} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.8}$$

During FP, ReLU layer acts as a truncation operation by setting negative inputs to zero (4.7). During BP, it serves as a *gate* by allowing the backpropagation only for positive inputs and completely blocking it for negative inputs (4.8). The *gating* property of ReLU layer resolves the *vanishing* or *diminishing gradients* problem for deep CNNs to some extent.



$$\overset{f}{\begin{bmatrix}3&0&2&1\\2&1&3&2\\0&3&1&6\\1&2&5&1\end{bmatrix}} * \overset{k}{\begin{bmatrix}-1&0\\0&1\end{bmatrix}} = \overset{s=f*k}{\begin{bmatrix}-2&3&0\\1&0&3\\2&2&0\end{bmatrix}}$$

Figure 4.3: Convolution of a matrix f of size 4×4 with a kernel k of size 2×2, and the output s of size 3×3.

The semantic information of an image is a highly nonlinear function of its pixel values. The purpose of ReLU layer is to increase the non-linearity of CNN's architecture. However, ReLU layer has one caveat, i.e., it "dies" when the input x^l is negative as the backpropagation is completely blocked. It causes an issue if a large negative bias is learned in the previous layer, resulting mostly in a negative output (given as the input to ReLU layer). The "dead" ReLU layer will hence output zero for almost all the activities of CNN. In fact, ReLU layer is unable to recover from the problem as zero gradient values result in no updates and CNN goes in a state of *unrecoverable corruption*.

4.7 Convolution Layer

Convolution layer (also known as `conv layer`) is the most important layer of CNN regarding the contributions to its functionality. Convolution of a function f with a kernel k is donated by $f * k$. Figure 4.3 shows convolution of a 2^{nd} order tensor f with a kernel k resulting in another tensor s. The value at the location (x, y) of the output s (i.e., $s_{x,y}$) is calculated by the overlapping the kernel k on the top of input f at the same location (x, y) and then computing a dot product of overlapping areas. The entire output s is rendered by moving the kernel k to all the possible locations of input f. A convolution operation for a higher order tensor (> 2) is defined similarly.

Both input and output of a convolution layer are 3^{rd} order tensors, known as *feature maps*. The depth of output D^{l+1} is determined by the number of convolution kernels in the layer. The size of individual feature map in the output is determined by the method of boundary treatment during a convolution operation; various boundary treatments are as follows:

- **Zero padding:** A input is padded with zeros, such that the size of its feature map is divisible by the kernel's dimensions (it is the most widespread method also known as SAME padding).
- **Nearest:** It works the same way as the previous case. However, it pads with the value of pixel closest to the boundary instead of the zeros.
- **Valid:** The size of feature map in the output is reduced depending on the size of kernel. The size of output becomes $(H^l - H + 1) \times (W^l - W + 1) \times n$ given the *stride* of 1.

Stride is another important parameter that determines the size of output. It represents the number of steps by which a convolution kernel moves on input feature maps. In Figure 4.3, a kernel is convolved with an input at all the possible spatial locations within that input; this corresponds to the stride of 1. For the case where $s > 1$, a convolution operation skips every $s - 1$ cells while moving in the horizontal or vertical direction which reduces the size of the final output.

There are three hyperparameters in a convolution layer: **(i)** number of kernels $D^{l+1} \in \mathbb{N}_{\geqslant 1}$, **(ii)** height and width of a kernel $H, W \in \mathbb{N}_{\geqslant 1}$, commonly chosen to be equal, i.e., $H = W = k$, and **(iii)** stride $s \in \mathbb{N}_{\geqslant 1}$.

FP of a convolution layer is illustrated in Figure 4.4. In precise mathematics, the n^{th} feature map (out of the D^{l+1} number of feature maps) of the output y^l of a convolution layer (with zero padding and stride $s = 1$) after FP is given by

$$\begin{aligned} y^l_{i,j,n} = \sum_{a=0}^{H} \sum_{b=0}^{W} \sum_{c=0}^{D^l} k_{a,b,c,n} \times x^l_{i+a,j+b,c} \\ \text{where,} \\ k : \text{convolution kernels} \end{aligned} \tag{4.9}$$

The above computation is repeated for all the D^{l+1} kernels. A bias b_d is also added to all the output feature maps. However, it is excluded for simplicity (the bias b is a 1^{st} order tensor of length D^{l+1}).

BP of a convolution layer is beyond the scope of this thesis. It requires computing the partial derivative $\frac{\partial y^l}{\partial x^l}$, i.e., the output y^l (4.9) w.r.t to the input x^l. Such derivative is

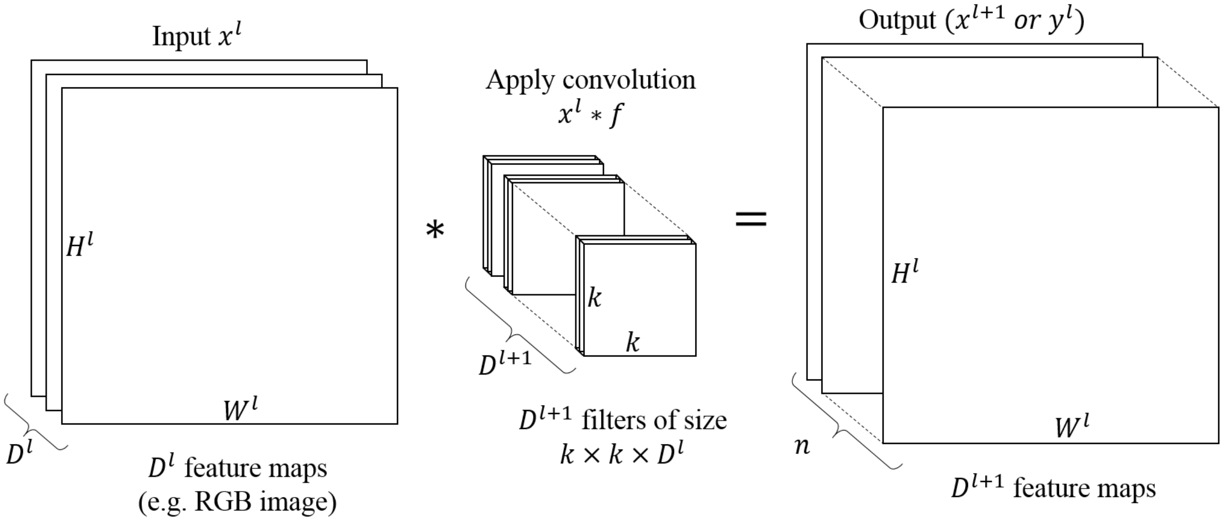


Figure 4.4: Schematic diagram for FP of a convolution layer. An input x^l to the l^{th} layer is convolved with n convolution filters f to create the output of n feature maps.

calculated by converting the output (4.9) as a matrix multiplication which is explained by Wu J. in [66] on p. 15 – p. 22. Zhang Z has written another good article explaining BP of convolution layer [67].

4.8 Pooling Layer

Pooling layer (also referred as `pool` or `sub-sampling` layer) is a down-sampling layer that summarizes $p \times p$ sub-regions of an input x^l. The hyperparameters for a pooling layer is p which defines the spatial extent of sub-sampling operation.

Pooling operation in the layer divides each of the $H^l \times W^l$ input feature map into $p \times p$ non-overlapping sub-regions and then applies either a *maximum* or *average* operation on them; depending on the choice of operation, i.e., *max* or *avg*, the pooling procedure is known as *max-pooling* or *avg-pooling* respectively. Figure 4.5 shows the max-pooling with $p = 2$ on an input tensor x^l of shape $4 \times 4 \times n$ which results in another tensor of shape $2 \times 2 \times n$. The shape of output tensor after a pooling operation is given as

$$H^{l+1} = \frac{H^l}{p}, \; W^{l+1} = \frac{W^l}{p}, \; D^{l+1} = D^l \tag{4.10}$$

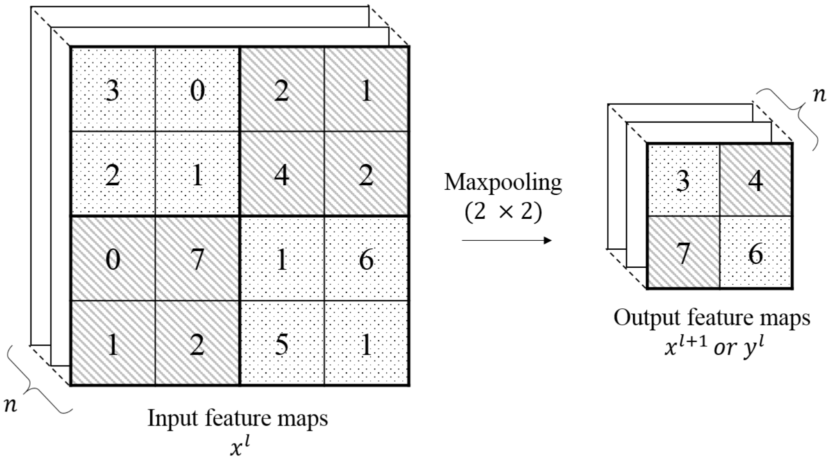


Figure 4.5: Illustration of the 2 × 2 max pooling operation.

The d^{th} feature map obtained in the output of max-pooling layer after FP is given as

$$y^l_{i,j,d} = \max_{n,m \in [0,\ p)} x^l_{i \times p+n, j \times p+m,d} \quad , \tag{4.11}$$

which is repeated D^l times to render the entire output y^l. Similarly, FP for avg-pooling is calculated by replacing *max* operation with *avg* operation. BP of a max-pooling layer computes a partial derivative of the output y^l (4.11) w.r.t the input x^l. It is explained in detail by J. Wu in [66] on pages 23 – 25.

A pooling layer is inserted in between successive convolution layers (convolution layers are followed by non-linearity and then pooling layers). Its function is to summarize the $p \times p$ areas of input. It selectively routes features from the $p \times p$ non-overlapping sub-regions of input based on either maximum or average operation. Multiple pooling layers make CNN insensitive to location-specific features. This behavior is suitable for histopathology images as a general requirement of histopathology is to identify presence of malignant pattern without knowing its exact location. The two primary purposes of a pooling layer are:

(i) **Dimensionality reduction**: A $p \times p$ pooling layer reduces the size of input data by $\frac{1}{p^2}$ thus preventing the *curse of dimensionality* and makes CNN computationally inexpensive.

(ii) **In-variance against rotation, position, and minor local changes**: Pooling operation extracts robust and invariant features from input feature maps. E.g., $p \times p$

max-pooling operation extracts the maximum value of a $p \times p$ spatial region which resists the small translation of changes of that small region.

4.9 Fully Connected Layer

A fully connected layer is added to utilize the entire representation of an input for the subsequent classification and higher level understanding of a given image. The fully connected layer have connections to all the “activations” of input as seen in a regular ANN. A fully connected layer is placed after convolutions layers.

A fully connected layer is a special case of convolution layer; a convolution layer with an input x^l (size of $H^l \times W^l \times D^l$) with D convolution kernels (each of the size $H^l \times W^l \times D^l$) is a fully connected layer with D neurons. The hyperparameters of a fully connected layer is D which represents the number of neurons in the layer.

4.10 Dropout layer

Dropout is a regularization technique, first introduced by Hinton et al. in [68] and explained further in [69]. It removes connections between neurons with a probability p as illustrated in Figure 4.6. Dropout layer is active only during the optimization phase of CNN (i.e., during the training of CNN) and it is completely deactivated during the prediction phase.

FP for Dropout layer requires a special tensor $\mathcal{D}$ of the same shape as the input x^l such that $\mathcal{D} \in \{0,\ 1\}$ and each element of $\mathcal{D}$, i.e., d_i is sampled from Bernoulli distribution $\mathcal{B}(1, p)$. FP is defined as element-wise multiplication of the input x^l and the special tensor $\mathcal{D}$; to compensate for the “dropped” connections, the output is multiplied by $\frac{1}{1-p}$, given as

$$
\begin{aligned}
& y^l = \frac{1}{1-p} * (\mathcal{D} \circ x^l) \qquad \text{with } d_i \sim \mathcal{B}(1, p) \\
\text{where,} & \\
& \circ : \ \text{Hadamard product,} \\
& d_i : \ \text{Any given element in } \mathcal{D}, \\
& p : \ \text{“dropout” probability}
\end{aligned}
\tag{4.12}
$$

Just like ReLU layer, BP of Dropout layer acts as the “gate” entirely blocking the backpropagation for certain connections between neurons. It “gates” the backpropagation for

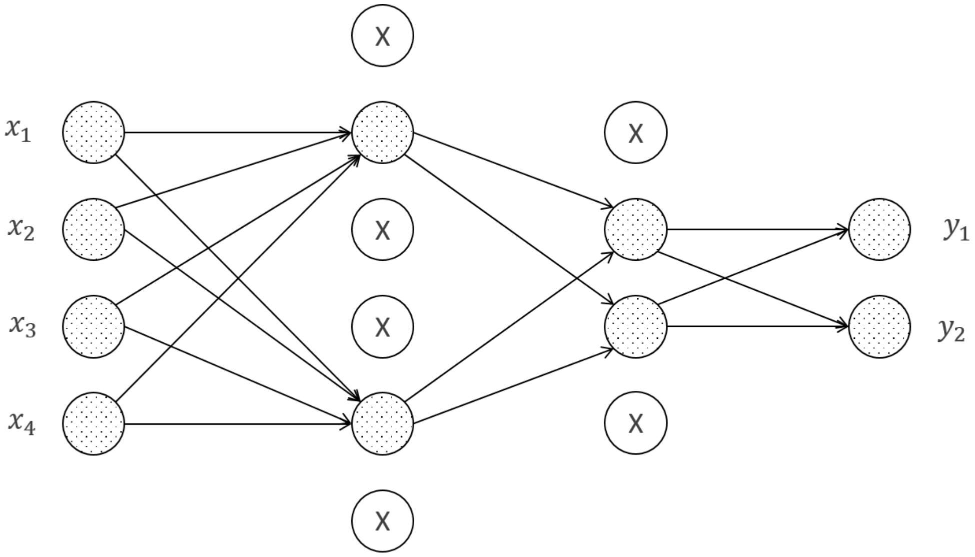


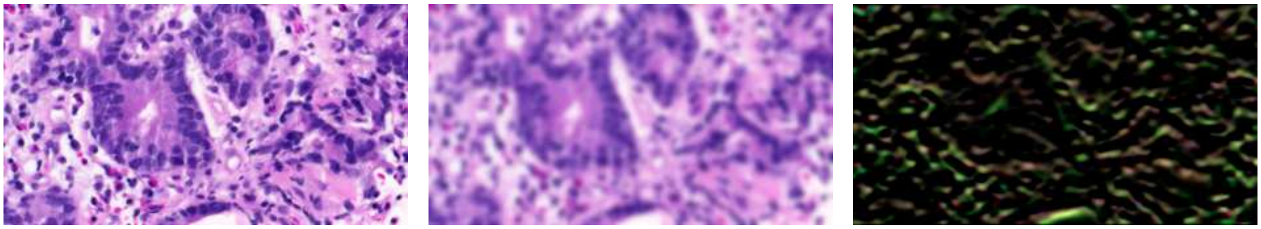
Figure 4.6: Illustration of connections as seen during the optimization step of a network with *Dropout* regularization.

the parameters that collides with the locations of zero in $\mathcal{D}$. Although $\mathcal{D}$ is a stochastic variable, it is redefined at every training step.

A model trained with Dropout is regarded as ensemble learning of the models with different number of neurons and their inter-connections [70]. Dropout is usually applied between two fully-connected layers. Summarizing, Dropout is an excellent regularization technique used to prevent overfitting and co-adaptations of neurons within CNN.

4.11 CNN as Feature Extractor

sCNN is a powerful feature extractor capable of understanding semantic concepts in images by end-to-end training. CNN encodes semantic concepts in its multi-layered architecture — every successive layer abstracts the higher level representation of an image. Image-features can be extracted from any layer of CNN. The feature extraction involves two steps. Firstly, FP of CNN is performed for a given image. Secondly, CNN is "cut" at the desired layer (from where the features needs to be extracted) and its activation maps are extracted. These activation maps constitute the feature vector for the given image. The initial layers encode low-level image features whereas the deep layers contain semantic information.



(a) Original image (b) Smoothed image (c) Horizontal edge

Figure 4.7: Application of different convolution kernels on **(a)** and input image, resulting in **(b)** obtained by applying a 2 × 2 Gaussian kernel, and **(c)** obtained by performing convolution with a horizontal edge detection kernel.

The most important layer of CNN is its convolution layer. A convolution layer performs a convolution operation which is a type of *linear image filter*. Figure 4.7 shows an example of a colored histopathology image (4.7a) and the resultant images as obtained after applying convolution operation with two different kernels (4.7b, 4.7c). One of the image (4.7c) highlights the horizontal edges in the input image because a 3 × 3 kernel used to create it, as follows

1	2	1
0	0	0
-1	-2	-1

has a high correlation with the patterns representing horizontal edges. Such high correlation results in a large and positive pixel value in presence of horizontal edges; referred as *maximal activation* for a feature (in the case, horizontal edges). Similarly, different convolution kernels *maximally activate* on different kinds of low-level patterns.

Stacking multiple layers: The architecture of CNN is highly flexible. Multiple layers can be stacked together as both the input and output are tensors of same order (3^{rd} order). Multiple convolution layers along with non-linearity and pooling layers learn to activate for the complex but specific patterns, e.g., groups of edges forming a particular shape. By adding even more layers these complex patterns assemble to activate on a semantically meaningful information within an image, e.g., a malignant pattern in a histopathology image.

Parameter sharing provides translation invariance and better feature visibility: In a convolution layer, all spatial locations share the same convolution kernels, increasing the parameter sharing while reducing the overall quantity of parameters (compared to *fully connected* layers). It also provides the translation in-variance and increases the visibility of features. For example, if a malignant pattern appears at multiple places within a pathology image, then the convolution kernel responsible for activating for such malignant pattern would fire multiple times. Even if this malignant pattern moves somewhere else within the image, the convolution kernel activates but at the different location in the output activation map[1].

Distributed representation of semantic concepts: CNN encodes semantic concepts, such as presence of cancer cells or malignant patterns within an image as M number of feature maps (also known as activation values). It is the essential characteristic of CNN, especially for Content-based Image Retrieval (CBIR) systems. E.g., in a context of histopathology images, if a specific neuron activates in presence of many closely-located cell nuclei, and another neuron activates in presence of irregular cell structures; then a disease which is identified by the two features, the irregular cells and the closely-located cell nuclei, can be recognized very effectively by CNN. Summarizing, semantic information within an image is a distribution of various abstract patterns and CNN is good at encoding distributed representations of such abstract patterns through its multi-layer architecture.

4.12 CNN Training

Training CNN for a classification problem requires the output z to be a probability distribution. The output is normalized as a probability distribution using *Softmax* function, given by

$$\tilde{z}_i = \frac{e^{z_i}}{\sum_j e^{z_j}}, \tag{4.13}$$

The above equation transforms z into a probability distribution $\tilde{z}$ such that $\arg\max \tilde{z}$ is the predicted class label.

[1] Single convolution kernel is not responsible for understanding semantic concepts in image. Instead, it is manifestation of series of activations that are carried forward through multiple stacked layers in CNN.

4.12.1 Cross-Entropy Loss Function

Cross-entropy is the most common loss function used for training CNN for a classification scenario. In information theory, cross-entropy $H(p,q)$ between two distributions p and q is given by

$$H(p,q) = -\sum_{x} p(x) \ \log \ q(x). \tag{4.14}$$

It quantifies the discrepancy between the "true" distribution p and the "estimated" distribution q. The output of CNN $\tilde{z}$ (normalized with Softmax) represents the "estimated" distribution of the predicted class. Whereas, the "true" probability distribution is obtained by setting the probability value of desired class to 100% and rest all to 0, also known as *one-hot encoding*. Therefore, cross-entropy loss in CNN is simplified to the follow equation

$$L = -\log \ \tilde{z}_j, \tag{4.15}$$

which is simply the $-log$ of element at the j^{th} index of output $\tilde{z}$ such that j is the desired class label.

4.12.2 SGD with Momentum

In the practice, training CNN uses *mini-batch strategy* version of SGD. In that case, an input to CNN x^l is a 4^{th} order tensor with the shape $H^l \times W^l \times D^l \times N$, where N is the size of "mini-batch". In other words, instead of feeding a single image to CNN, N images are fed. In the mini-batch gradient decent, a regular SGD (4.3) is changed as follows

$$\begin{aligned} &w^i \leftarrow w^i - \frac{\eta}{N}\sum_{j=1}^{N}\frac{\partial E_j}{\partial w^i} \\ &\text{where,} \\ &E_j : \ \text{loss for } j^{th} \text{ input within mini-batch} \end{aligned} \tag{4.16}$$

In this chapter, all the layers were explained assuming $N = 1$. However, most of the discussed formulas can be easily adapted to a higher value of N without changing the underlying mathematics.

For this thesis, mini-batch SGD with momentum is used to train all the CNN models. Momentum term $\Delta w_i(t-1)$ is added to a regular mini-batch SGD as given

$$\Delta w^i(t) = \frac{\eta}{N}\sum_{j=1}^{N}\frac{\partial E_j}{\partial w^i} + \alpha \Delta w_i(t-1), \tag{4.17}$$

allowing the update step to “move” in the direction of last gradient $\Delta w_i(t-1)$. The update step is scaled by a parameter known as “decay” α. To understand “momentum” consider a situation — a ball rolling down a hill and when it reaches the minimum point, it continues to move upward due to the attained “momentum”. The addition of momentum helps SGD to ovoid local minima thus increasing exploration capacity of the algorithm in the error landscape.

Chapter 5

Implementation of Image Descriptors

> “*It would be possible to describe everything scientifically, but it would make no sense; it would be without meaning, as if you described a Beethoven symphony as a variation of wave pressure.*”
>
> — Albert Einstein

5.1 Introduction

This chapter explains the implementation details of three different types of image descriptors used in this thesis. The chapter is broken down into three sections. The first section (§5.2) discusses Local Binary Pattern (LBP), the second section (§5.3) covers Bag of visual Words (BoW), and the third section (§5.4) covers deep descriptors. The third section on the deep descriptors (§5.4) discusses three different deep Convolution Neural Network (CNN) models regarding their training, architectures, and the fine-tuning protocols. Out of these three CNN models, one is implemented and trained from scratch whereas other two are popular pre-trained models, namely VGG16 and Inception v3. The material presented in the chapter is mostly self-contained; however, readers can refer to Chapter 4 to understand various architectural components of CNN.

5.2 Local Binary Pattern (LBP)

LBP is a feature extraction technique, first introduced by Ojala and Pietikainen et al. in 1994 [71]. However, its texture spectrum model was proposed much earlier in 1990 [72]. It is a simple yet competitive technique for image classification and segmentation. LBP approach does not require training before using it for extracting features from images; therefore, it comes under category of *hand-crafted image descriptors*.

Several versions of LBP algorithms exist in the literature with different in-variance capabilities (e.g., invariant to rotation and translation), making them suitable for a broad range of applications [73, 74, 75]. The variant of LBP used for the thesis work is called *uniform LBP*. The main reasons for choosing *uniform LBP* is due to its simple implementation, good performance in various fields of computer vision, and high popularity as learned from the literature [73, 74, 75].

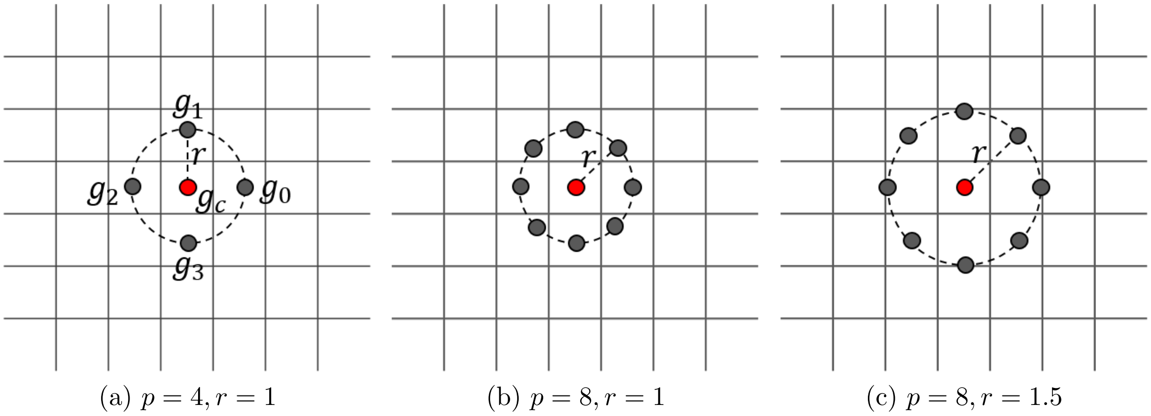
5.2.1 Implementation Details

The local binary-pattern of a pixel in a gray scale image is a pattern formed by an arrangement of pixels in the local neighborhood of the given pixel. These local binary-patterns are quantified using an LBP operator. The feature extraction pipeline for a general LBP algorithm consists of two steps **(i)** apply an LBP operator to all the pixels of an image, and **(ii)** aggregate all the quantified binary-patterns (obtained from the LBP operator) to a histogram. The obtained histogram constitutes as the final feature vector for the image. Different variants of LBP algorithms mostly differ in the second (aggregation) step.

Uniform LBP (the one used for this thesis work) uses a *circular LBP operator* (§5.2.1a.) for the first step and *uniform LBP codes* (§5.2.1b.) to construct a histogram as required by the second step.

5.2.1a. Circular LBP operator

A circular LBP operator, the one used in this thesis work, is a type of LBP operator that employs a circular region as its local neighborhood. It is denoted by $LBP_{p,r}$, where p controls discretization of the neighboring pixels and r defines the radius of the circular region. Figure 5.1 shows three circular LBP operators (parameterized differently w.r.t p and r) applied to a center pixel g_c (red dot) of an image (represented by a mesh). Each pixel of the image is represented as a square region within the mesh, and the p number of neighboring pixels (black dots) are denoted by $\{g_0, g_1, \ldots, g_{p-1}\}$.



(a) $p = 4, r = 1$ (b) $p = 8, r = 1$ (c) $p = 8, r = 1.5$

Figure 5.1: Illustration of an LBP operator with different p and r values. p controls discretization of local angular space and r sets spatial resolution of the operator.

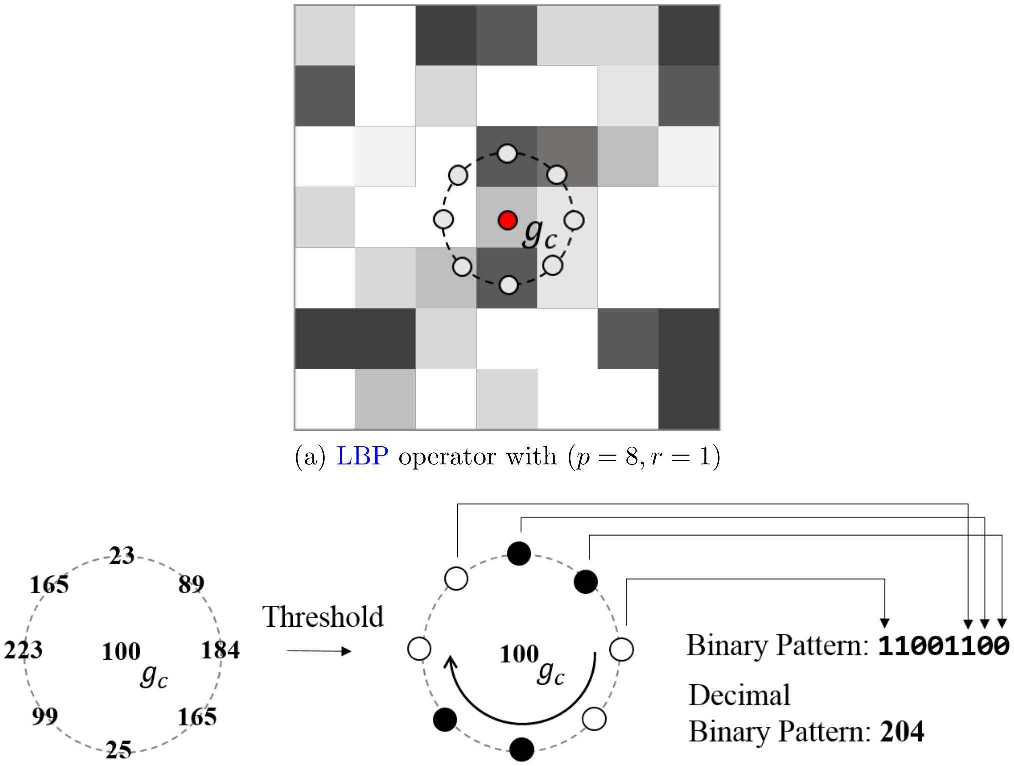
The parameter p controls the number of neighbors, resulting in 4, 8, 8 number of neighbors in figures 5.1a, 5.1b, 5.1c respectively. Every i^{th} neighbor g_i lies on the imaginary circle $\mathcal{C}$ with the center at g_c and the radius of r pixels (the second parameter of circular LBP operator). The neighbors are separated by equal distances across the perimeter of the imaginary circle. In precise mathematics, the coordinates of i^{th} neighbor g_i are given by

$$g_i = \left(- rsin\left(\frac{2\pi i}{p}\right),\ rcos\left(\frac{2\pi i}{p}\right)\right) \tag{5.1}$$

If a neighbor g_i, does not fall in the center of a square region, its value is interpolated. Otherwise, the value of the neighbor is assigned same as the intensity of the enclosed pixel, i.e., the enclosed square region. The output of circular LBP operator $LBP_{p,r}$ for a center pixel g_c is a single scalar value, given by

$$\begin{gathered} LBP_{p,r} = \sum_{i=0}^{p-1} 2^i \times s(g_p - g_c) \\ \text{where,} \\ s(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases} \end{gathered} \tag{5.2}$$

The mechanism of circular LBP operator is further illustrated in Figure 5.2. The idea is that each neighboring pixel is assigned a binary value of 1 if larger than the center



(a) LBP operator with $(p = 8, r = 1)$

(b) Quantization of "binary-pattern" formed by neighbors of g_c

Figure 5.2: Illustration summarizing the steps performed by an LBP operator g_c; **(a)** shows an LBP operator with a center pixel g_c (red) and neighboring pixels (gray), and **(b)** shows quantification of the "binary-pattern" formed by the neighboring pixels.

pixel and 0 otherwise. By doing so, a binary number of p bits is obtained such that the most significant bit comes from comparing g_{p-1} with g_c and the least significant bit from comparing g_0 with g_c. A p-bit binary number obtained from a circular LBP operator for a given pixel is the *LBP code* of that pixel.

5.2.1b. Uniform LBP codes

An *LBP code* of a pixel is a p-bit binary number which in decimal number system ranges between $[0,\ 2^p]$. An LBP code is "uniform" if its binary representation contains at most two transitions from either $0 \rightarrow 1$ or $1 \rightarrow 0$. E.g., `11100001` is a uniform LBP code whereas, `10010101` is a non-uniform LBP code. The total number of uniform LBP codes is denoted by $n_{\mu,p}$ (where μ represents the uniform coding scheme), and it is given by:

$$n_{\mu,p} = p \times (p-1) + 2 \tag{5.3}$$

In *uniform LBP* feature extraction model all the non-uniform LBP codes are grouped into a single class, whereas each of the uniform LBP code forms a class of its own (i.e., $n_{\mu,p}$ number of classes). Therefore, the total number of classes in *uniform LBP* algorithm is $n_{\mu,p} + 1$ (i.e., 1 class for all the non-uniform LBP codes and $n_{\mu,p}$ classes for uniform LBP codes).

5.2.2 Feature Extraction

The feature extraction in *uniform LBP* algorithm requires a special mapping U_p. It maps all the possible *LBP codes* to their associated classes. For a given parameter p, the output of a circular LBP operator $LBP_{p,r}$ is a single number from set A such that

$$A = \{0, 1, \ldots, 2^{p-1}\}$$

and all the classes (under the uniform classification scheme) belong to another set B such that

$$B = \{0, 1, \ldots, n_{\mu,p} + 1\}$$

then U_p maps all the LBP codes in the set A to their associated classes in the set B, given by

$$U_p : A \to B,$$

such that for a given LBP code a, its associated class is $b = U_p(a)$. Now, extracting a feature vector F_q from a given image I_q involves the following steps:

(i) Build the mapping U_p (can be cached for a given value of p).

(ii) Apply a circular LBP operator $LBP_{p,r}$ to every location (x, y) of the image I_q and store all the resultant LBP codes in an array f_q[1].

(iii) Apply mapping U_p on every element of f_q and create another array c_q.

[1] The neighbors beyond the extent of local neighborhood are set to intensity of $-\infty$ resulting in a zero bit on comparing with the center pixel.

(iv) The final feature vector F_q is calculated by aggregating all unique values in c_q as a histogram.

The length of the feature vector F_q is same as the number of classes $n_{\mu,p}$ (5.3). There are two parameters in a uniform LBP approach, i.e., radius r and the number of neighbors p. Changing either of the parameters has a significant impact on the captured semantics of an image. Moreover, the parameter p affects the length of feature vector F_q. Summarizing, LBP is a simple approach as it does not require any prior training and it is a fast feature extraction method with time complexity of $O(m)$, for a given image with m number of pixels.

5.3 Bag of visual Words (BoW)

A bag of visual words (BoW) model is a powerful computer vision tool. It has proven to be a useful method for reducing the semantic gap in features obtained from low-level image descriptors [76]. Many variations of BoW are extensively applied in the field of histopathology image classification [77, 4, 78, 79]. Unlike LBP, BoW requires training before extracting feature vectors from an image.

5.3.1 Implementation Details

A general BoW model has two phases, training and feature extraction phase. The two phases are discussed below.

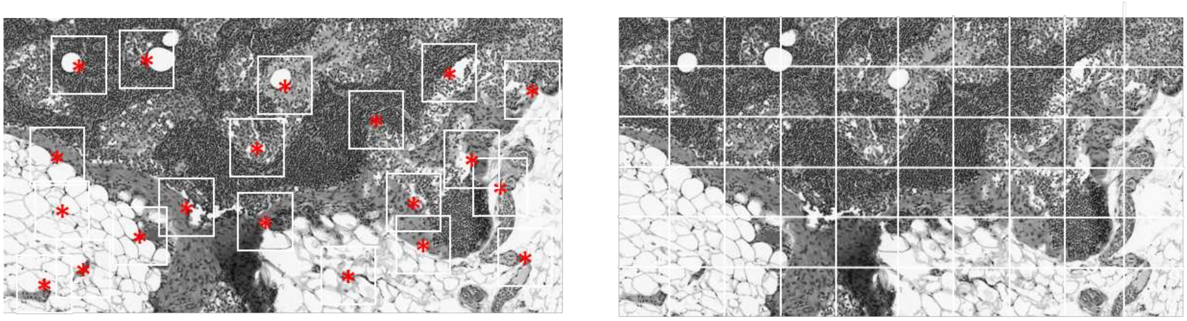
Training phase of a BoW model builds a structure called *codebook*. It is built from images in training data. The training pipeline of BoW approach consists of **(i)** sampling a large number of local and low-level features from different locations within training images (§5.3.1a. and §5.3.1b.), and **(ii)** clustering the extracted features using K-means clustering algorithm. An individual cluster center obtained from the clustering algorithm is called *visual word*, whereas all the k cluster centers are collectively known as *codebook* or visual vocabulary (§5.3.1c.).

Feature extraction phase of a BoW model uses the *codebook* constructed during the training phase. The codebook is used to generate feature vectors for unseen images. The pipeline of feature extraction phase consists of **(i)** extracting local and low-level feature vectors from different locations within a given image using the same protocol as the training phase (§5.3.1a. and §5.3.1b.), and **(ii)** building the final feature vector for the image using the learned *codebook*. It involves successive applications of *encoding* and *pooling* operations (§5.3.1d.).

5.3.1a. Location sampling

The first step of a BoW framework defines a location sampling scheme. The sampled locations are used to extract local features from them. In the literature, two different types of location sampling schemes are suggested **(i)** Interest Points (IP) and **(ii)** dense sampling [76].

IP sampling scheme is based on extracting low-level features from the "interesting" locations within an image (5.3a). These interesting locations can be either corners, blobs, or selected based on some other criteria. Dense sampling scheme, on the other hand, collects low-level features from an entire image. It divides the image into a mesh and local features are extracted from each square region formed within the mesh (5.3b). These two methods are shown below; however, for this thesis only the dense sampling scheme is used.



(a) Interest Point (IP) sampling

(b) Dense Sampling

Figure 5.3: Illustration of two location sampling techniques. Low-level features are extracted from white-outlined regions. In **(a)** only interesting location points are selected shown with red stars, whereas in **(b)** entire image in split into equal sized squares.

5.3.1b. Local features extraction

After sampling the locations, the next step is to extract local features from the sampled locations. Many choices of the local features exist in the literature, such as Scale Invariant

Feature Transform (SIFT) [80], Speed Up Robust Features (SURF) [81], and Binary Robust Independent Elementary Features (BRIEF) [82].

In this thesis, LBP (§5.2) features are used as the local features. Although it is unconventional to use higher-order features (e.g., LBP histograms) as the local features, the two main reasons to select LBP features for this thesis are — **(i)** to reduce the computational expense by processing the condense LBP histograms instead of the raw pixels, and **(ii)** using the BoW with LBP as the local feature extractor provides a comparative baseline against the LBP approach alone.

5.3.1c. Codebook construction

The most common approach for constructing a *codebook* is by clustering the local features using K-means clustering algorithm [76].

A set of n local features,

$$x_1, x_2, ..., x_n \in \mathbb{R}^D,$$

are extracted from training images. These local features are clustered using K-means algorithm. The clustering algorithm provides k number of vectors (cluster centers),

$$\mu_1, \mu_2, ..., \mu_k \in \mathbb{R}^D,$$

and data-to-cluster assignments,

$$q_1, q_2, ..., q_n \in \{1, 2, ..., k\},$$

such that the squared distance d between each data point x_i and their respective cluster μ_{q_i}, given by

$$d = \sum_{i=1}^{N} \parallel x_i - \mu_{q_i} \parallel^2,$$

is minimized. The k vectors or the cluster centers obtained from K-means algorithm $\{\mu_1, \mu_2, ..., \mu_k\}$, individually represents a "visual word", and collectively called *codebook*. Figure 5.4 shows an example of a *codebook* learned from histopathology images [4].

5.3.1d. Encoding and pooling

After the codebook construction step, the next step allows transforming a given image into the final feature vector. Representation of an image as the feature vector in BoW framework

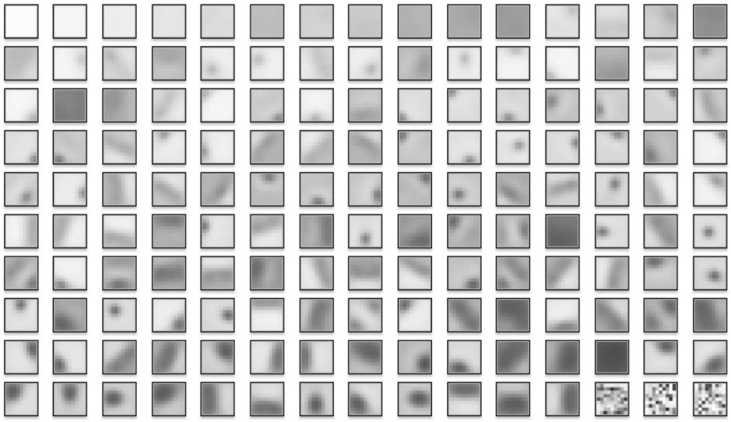


Figure 5.4: An example of a *codebook* with 150 visual words created using histopathology images. Image is taken from [4].

is composed of two successive steps of encoding and pooling. The encoding step assigns the local features onto the visual words in the codebook. Whereas, the pooling step aggregates the assigned words into a histogram. The obtained histogram is the final feature vector for the image.

In the most general form of BoW, the one implemented for this thesis, the histogram obtained from encoding-pooling step represents frequency distribution of the sampled local features w.r.t visual words in the codebook. The histogram thus created contains the information of an image in a compact form. The process of successive encoding-pooling is known as *vector quantization*. The steps involved in the vector quantization are below:

Given:

$$\mathcal{C} = \{c_1, c_2, ..., c_k\}$$
$$X = \{x_1, x_2, ..., x_n\}$$

where,

$\mathcal{C}$: codebook containing k visual words

X : set of n local features extracted from a given image

Encoding phase calculates a vector α_n for each element in X, as given

$$\alpha_{n,k} = 1 \textit{ iff } j = \underset{j \in \{1..k\}}{\arg\min} \parallel x_n - c_j \parallel^2 .$$

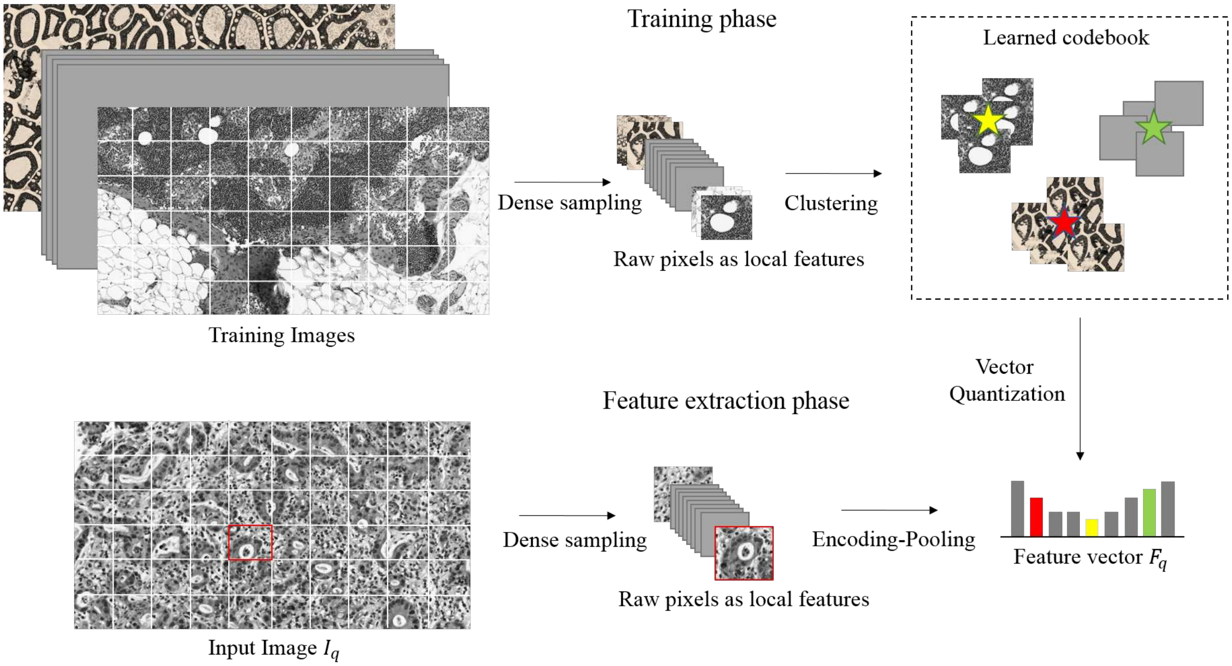


Figure 5.5: Schematic diagram of the feature extraction with Bag of visual Words (BoW) model.

The vector α_n has size k (number of visual words in codebook) and the j^{th} element of vector α_n is set to one (rest all are zeros) such that c_j (the j^{th} visual word) has the least squared distance to x_n (the n^{th} local feature).

Pooling phase calculates a vector γ of size k. The i^{th} element of γ is given as

$$\gamma_i = \sum_{j \in \{1..n\}} \alpha_{j,i}$$

Finally, the vector γ is normalized using ℓ_2 norm, i.e., $\gamma = \frac{\gamma}{\|\gamma\|_2}$.

5.3.2 Feature Extraction

The two parameters of BoW approach are: **(i)** a low-level feature extraction method and **(ii)** the size of codebook defined by the parameter k of K-means algorithm. Both the choices are very crucial for optimal working of a BoW approach. Figure 5.5 shows an overview of training and feature-extraction phases of a BoW approach as used in this thesis[1].

5.4 Deep Descriptors

#	Type	# filters @ patch size/stride	Parameters	Output size
0	Image			$1 \times 128 \times 128$
1	Convolution	64 @ $3 \times 3/1$	640	$64 \times 128 \times 128$
2	Convolution	64 @ $3 \times 3/1$	$36,928$	$64 \times 128 \times 128$
3	Max pooling	2×2	0	$64 \times 64 \times 64$
4	Convolution	128 @ $3 \times 3/1$	$73,856$	$128 \times 64 \times 64$
5	Max Pooling	2×2	0	$128 \times 32 \times 32$
6	Convolution	256 @ $3 \times 3/1$	$295,168$	$256 \times 32 \times 32$
7	Max pooling	2×2	0	$256 \times 16 \times 16$
8	FC	1024	$\mathbf{67,108,864}$	
	Dropout($p = 0.5$)		0	
9	FC	24	$24,576$	
$\sum$			$\mathbf{67,540,042}$	

Table 5.1: Architecture of the CNN model trained from scratch (CNN$_1$). The dotted line divides the model of into multiple conceptual blocks. The dropout probability is 0.5.

For a very long time, the handcrafted feature engineering had dominated as the ideal choice for the image descriptor. However, now, it is germane to only certain image analysis fields that benefit from the meticulous nature of hand-crafted features. The major weakness of handcrafted features is their brittle quality. They do not generalize well for a large and diversified image analysis problem [83]; some of the problems are best approached with flexible solutions like CNN that adapts to the problem by training and achieves better performance.

[1] For this thesis, LBP is used as the local-feature extractor in BoW model as opposed to the raw pixels (shown in the figure).

5.4.1 Implementation Details

Three different CNN models are used in this thesis, referred to as "Deep Descriptors". CNN models are used to extract representations of histopathology images. These image representations are extracted in form of the activation values from a layer of a CNN model[1].

CNN models: Out of the three models, one CNN model, namely, CNN_1 is trained from scratch and other two models, namely, VGG16 and Inception-v3 are popular pre-trained models existing in the literature. The architecture of CNN_1 model in shown in Table 5.1, and architecture of VGG16 and Inception-v3 are presented in Table B.1, Table B.2 respectively of Appendix B.

Image features: Image features can be extracted from any layer in CNN model. However, usually activations from the last pooling layer are used. In this thesis, image-features from the CNN models are extracted as follows:

- For CNN_1 (Table 5.1), activation values from the layer 8 (FC) are extracted (the output size of 1024) as image-features.
- VGG16 (Table B.1, pp. 92) is "cut" at the 18^{th} layer, its activation values (with the output size of $512 \times 7 \times 7$) are extracted, and then a 7×7 avg-pooling operation is applied, giving a image-feature vector of length 512.
- For Inception-v3 (Table B.2, pp. 94), activation values from the last pooling layer with the output size of 2048 are extracted as image-feature vectors.

Fine tuning: CNN models are fine tuned for a domain adaption. VGG16 and Inception-v3 used in this thesis are pre-trained on the natural images from ImageNet dataset [84]. Therefore, fine-tuning allows these CNN model to adapt towards a histopathology domain. When a deep network is fine-tuned, an optimal setup varies between the applications. For this thesis, the final convolutional block (layer $15 - 18$ both included) of VGG16 and the final two inception blocks within Inception-v3 are re-trained.

[1] Refer to §4.11 in Chapter 4 for the details regarding various concepts of CNN as feature extractor.

Chapter 6

New Dataset – Kimia Path24

> *If you focus your mind on the freedom and community that you can build by staying firm, you will find the strength to do it.*
>
> — Richard M. Stallman

6.1 Introduction

This chapter describes a new and open dataset, *Kimia Path24*, specially developed for conducting the experiments undertaken in this thesis[1]. The dataset is designed at Kimia Lab, University of Waterloo. It was first introduced in [85] at CVPR workshops in 2017. The dataset facilitates the research in image classification and content-based retrieval of histopathology images.

Kimia Path24 dataset is provided as a single HDF5[2] file of size 29 GB. The dataset contains 24 whole-slide images from different types of tissues and stained with different types of dyes. The total of $1,325$ test patches (each sized 1000×1000) are extracted from the 24 whole-slide images. The test patches are manually selected with the special attention to textural differences among the patches. Weighted accuracy measurements are provided to enable a unified benchmark for future works (§6.4).

[1] Available at http://kimia.uwaterloo.ca/kimia_lab_data_Path24.html

[2] https://support.hdfgroup.org/HDF5/

Each of the 24 whole-slide image in the dataset is captured by *TissueScope LE 1.0*[1]. The Whole Slide Imaging (WSI) for each slide was performed in the bright field using a 0.75 NA lens. The resolution of a given whole-slide image is determined by checking its description tag in its header, e.g., if the resolution is $0.5\mu m$ then the magnification is 20× and if the resolution is $0.25\mu m$ then the magnification is 40×.

The proposed dataset is structured to mimic the retrieval tasks in real clinical practices. The dataset allows the algorithm designers to have the flexibility to create their own training patches from the provided 24 whole-slide images. One could create $27,000$ to over $50,000$ number of training patches each of size 1000×1000.

6.2 Motivation

The dataset is created especially for the research of Content-based Image Retrieval (CBIR). *Kimia Path24* differs from other existing Digital Pathology (DP) datasets as the visual attention is employed on the diversity of patterns and not on the anatomies or malignancies. Therefore, it is preferably a computer vision dataset as in contrast to a pathological dataset. The dataset contains whole-slide images stained with different dyes, taken from different part of bodies, and also offer a significant variability regarding the grades and malignancies thereby making it very suitable for CBIR experiments. Another motivation behind creating this dataset is to provide a fixed number of testing samples. The fixed testing samples facilitate a standard benchmark but enables the design freedom of algorithm designers to generate their own training data.

6.3 Dataset Creation

For designing *Kimia Path24*, 24 whole-slide images were manually picked from a large pool of digital slides, purely based on the visual distinction (from a non-clinical perspective). In the selection process the conscious effort was made to represent the different *textural* patterns and the different types of tissues. Figure 6.1 shows thumbnails of six whole-slide images from the dataset. Figure 6.2 displays the magnified portion of each of the 24 whole-slide image. There is a substantial inter and intra-class variability among the 24 whole-slide images as shown in Figure 6.3.

[1] http://www.hurondigitalpathology.com/tissuescope-le-3/

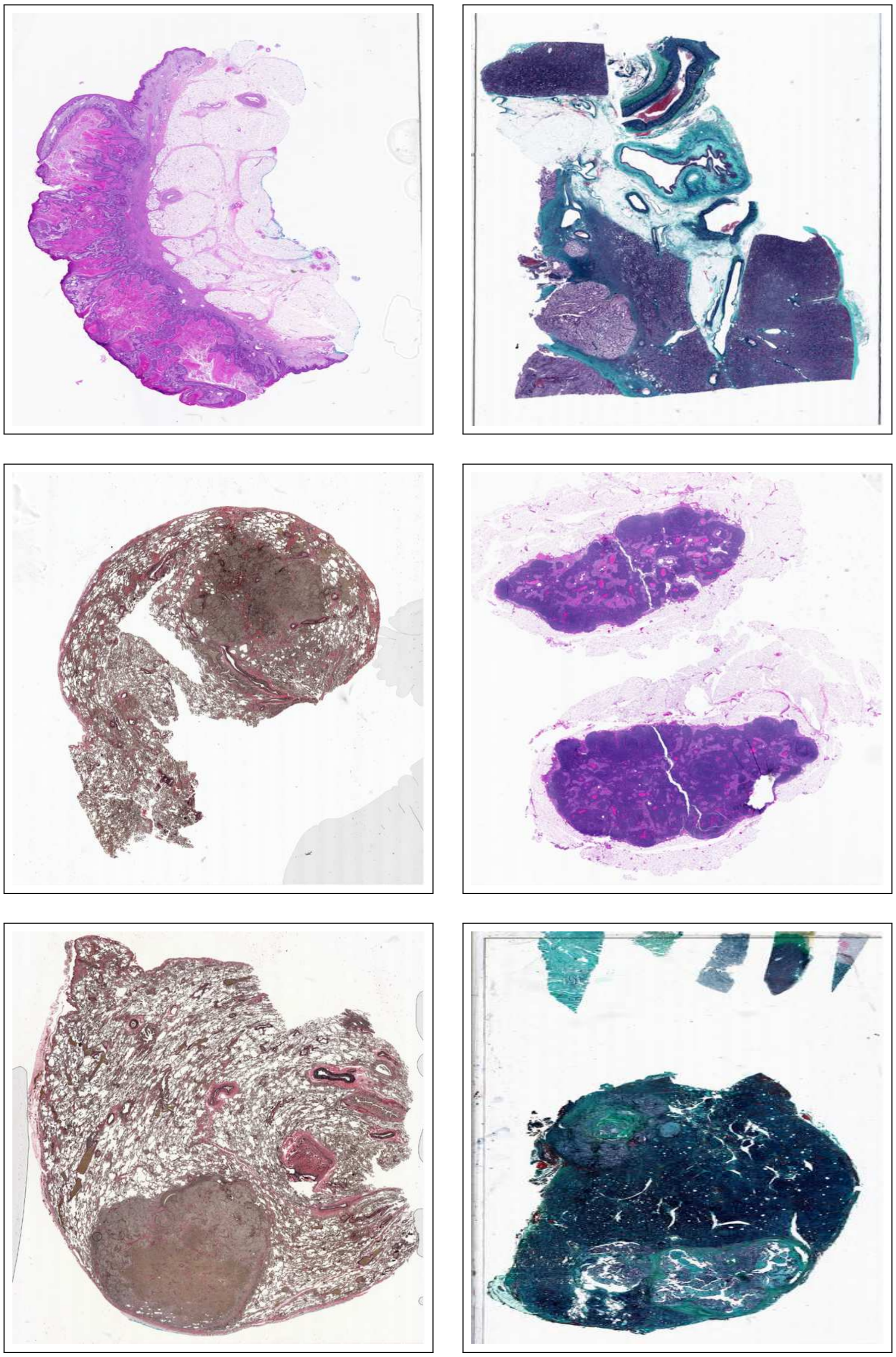


Figure 6.1: Thumbnails of six whole-slide images (aspect ratios changed for convenient display) from *Kimia Path24* dataset.

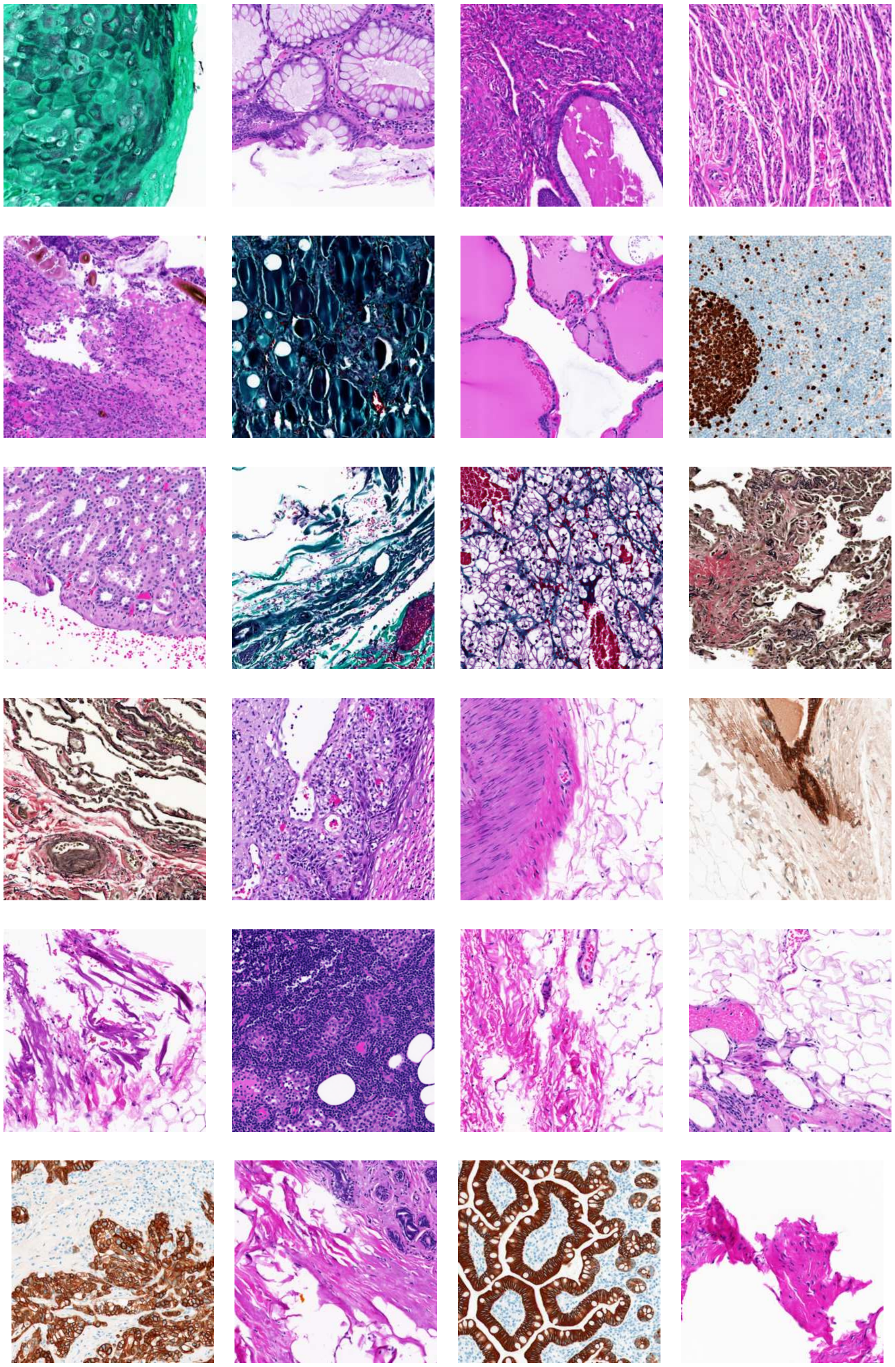


Figure 6.2: 24 sample patches for each of the 24 whole-slide images within *Kimia Path24* dataset.

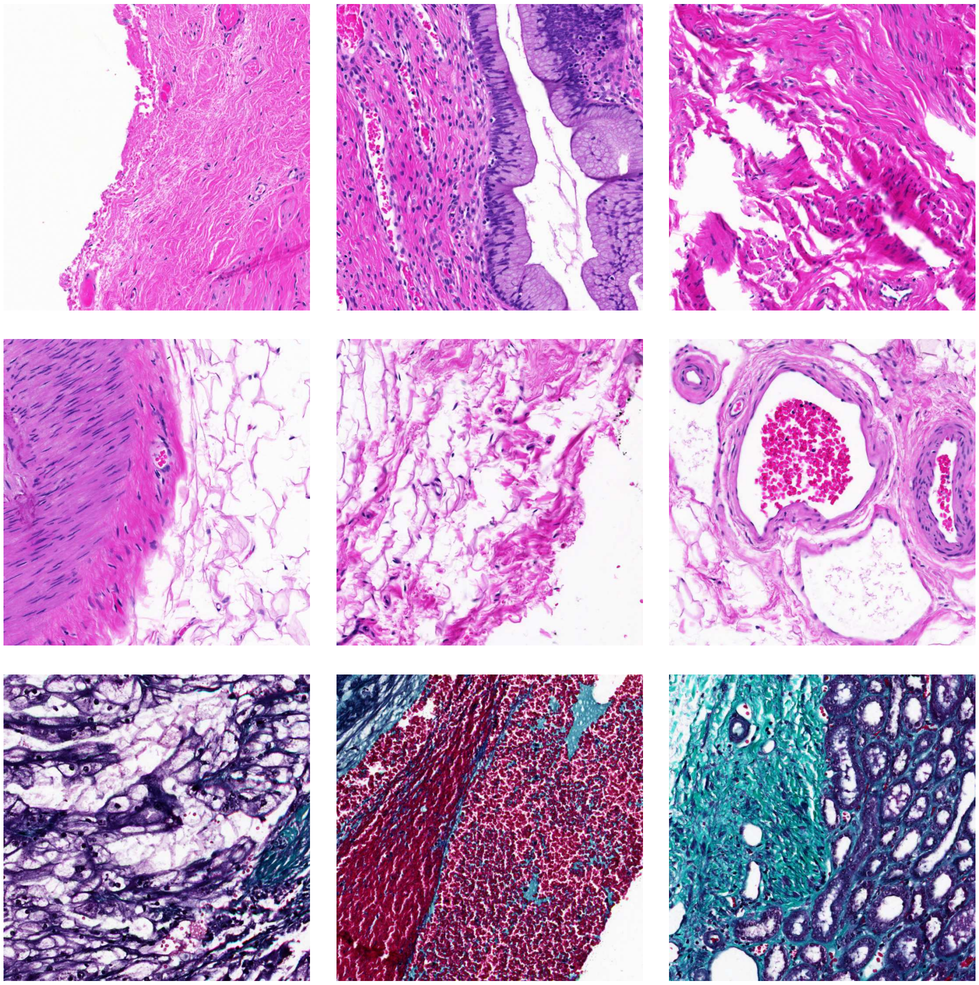


Figure 6.3: Examples of a large intra-slide textural variability in Kimia Path24. Each row shows three sample patches from a individual whole-slide image of Kimia Path24.

The following steps were performed to design and create *Kimia Path24*:

(i) The total of 1325 number of testing patches were selected (each sized 1000 × 1000 pixels that correspond to $0.5mm \times 0.5mm$ of the actual size).

(ii) The n_i patches were manually selected per whole-slide image such that $i = \{1, 2, \dots, 24\}$. The visual patch selection aimed to extract a small number of patches that represent all the dominant tissue textures in each whole-slide image.

(iii) Each of the selected patch was then removed from the whole-slide image and saved separately as a testing patch.

(iv) The remaining parts of the whole-slide image is used to construct the training dataset.

Scan #	Dimensions	# of Test Patches(n_{Γ_s})
0	40,300 × 58,300	65
1	37,800 × 50,400	65
2	44,600 × 77,800	65
3	50,100 × 77,200	75
4	26,500 × 13,600	15
5	27,800 × 32,500	40
6	38,300 × 51,800	70
7	29,600 × 34,300	50
8	40,100 × 41,500	60
9	40,000 × 50,700	60
10	47,500 × 84,700	70
11	44,100 × 52,700	70
12	45,400 × 60,100	70
13	79,900 × 56,600	60
14	42,800 × 58,200	60
15	20,200 × 57,100	30
16	35,300 × 46,300	45
17	48,700 × 61,500	45
18	26,000 × 49,600	25
19	30,700 × 70,400	25
20	48,200 × 81,400	65
21	38,500 × 40,500	65
22	40,500 × 45,700	65
23	36,900 × 49,000	65

Table 6.1: Properties of different scans in *Kimia Path24* dataset.

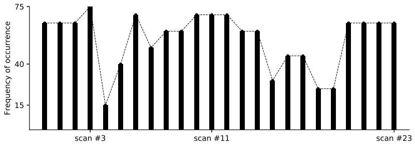


Figure 6.4: Class distribution of test data within *Kimia Path24* dataset.

6.4 Accuracy Calculation

The final accuracy calculation for the *Kimia Path24* dataset is based on two types of accuracy calculations, namely path-to-scan and whole-scan accuracies.

Notation: The total number of test patches is denoted by n_{tot} and for the dataset $n_{tot} = 1,325$. There are 24 different classes (one for each whole-slide image) denoted by set S, i.e., $S = \{0, 1, \dots 23\}$. Any given test patch from the dataset is denoted by P_s^i, where $s \in S$ represents its class and $j \in [0, n_{\Gamma_s}]$ is index to identify it among all the patches associated with class s. The Γ_s is set of patches P_s^i that belongs to class s such that $\Gamma_s = \{P_s^i | s \in S, i = 1, 2 \dots, n_{\Gamma_s}\}$ with n_{Γ_s} is number of patches in s^{th} class, as reported in last column of Table 6.1.

Accuracy measurements: Looking at a set of retrieved images R for any given CBIR experiment, the *patch-to-scan accuracy* η_p is given as

$$\eta_p = \frac{\sum\limits_{s \in S} |R \cap \Gamma_s|}{n_{tot}}, \tag{6.1}$$

which represents the standard accuracy measurement, i.e. ratio of number of correct predictions to total samples in the test data.

The *whole-scan accuracy* η_W as

$$\eta_W = \frac{1}{24} \sum_{s \in S} \frac{|R \cap \Gamma_s|}{n_{\Gamma_s}}, \tag{6.2}$$

which measures average of normalized ratio of number of correct predictions per class to total number of samples in that class (average of normalized accuracy per class). Whole-scan accuracy is not greater than patch-to-scan accuracy η_p, and it is highly dependent of distributions of class labels within testing data (see n_{Γ_s} in Table 6.1).

And finally, the *total accuracy* η_{total} as

$$\eta_{total} = \eta_p \times \eta_W \tag{6.3}$$

which takes into the account both the accuracy measurements, i.e., patch-to-scan and whole-scan accuracy.

Remark: Python code for the accuracy calculations for *Kimia Path24* dataset is provided (§C.1.1 of Appendix C).

6.5 Summary

The chapter discussed Kimia Path24, a new dataset for retrieval and classification of histopathology images. The dataset contains 24 whole-slide images from a large pool of digital slides. These 24 whole-slide images are selected mostly through visual inspection, i.e., selecting the *texturally different* images. Hence, the proposed dataset is a computer vision dataset (as in contrast to a pathological dataset) because visual attention is spent on the diversity of patterns and not on the anatomies and malignancies. The dataset also establishes the accuracy measurements for assessing the quality of retrieval experiments (§6.4). The next chapter describes various experiments performed with Kimia Path 24 using Local Binary Pattern (LBP), Bag of visual Words (BoW), and Convolution Neural Network (CNN).

Chapter 7

Comparative Study

> *The important thing is not to stop questioning. Curiosity has its own reason for existing.*
>
> — Albert Einstein

7.1 Introduction

This work intends to design the experiments capable of providing a comparative evaluation of three image analysis models, i.e., Local Binary Pattern (LBP), Bag of visual Words (BoW), and Convolution Neural Network (CNN). Kimia Path24 dataset is used as the primary dataset for performing all the experiments. The non-overlapping patches of size 1000 × 10000 are extracted from each of the 24 whole-slide images in the dataset. A simple homogeneity measurement is used to filter out the irrelevant patches (with mostly the white background); these filtered patches are converted into a training data which is stored separately. With the newly created training data, the image analysis models are trained and qualitatively assessed on their retrieval performances using the benchmark technique of Kimia Path24. Each of the image analysis algorithm contains a range of hyperparameters, tuning these hyperparameters result in feature-vectors of different sizes (encoding different semantic information within images). Therefore, changing the hyperparameters affects the benchmark results. This chapter presents the results and discussion of three series of experiments, designed to explore and investigate the effectiveness of LBP, BoW, and CNN in extracting the discriminative image-features for the classification and retrieval of histopathology images.

7.2 Materials

The Kimia Path24 dataset is used for conducting all the experiments; it can be obtained from the website of Kimia Lab[1]. Different aspects of the dataset have been thoroughly explained in Chapter 6.

7.2.1 Software Resources

`Python 3.6` is used as the primary computer language for conducting all the experiments. The development environment for the research was setup using `Anaconda Distribution`[1] which comes pre-installed with many Python-compatible libraries for the data science and machine learning fields. The following are the essential libraries used in this thesis:

(i) Numpy and SciPy [86]: These are two widely popular scientific computing libraries in Python. They provide fast array manipulations and some common algorithms in algebra. For this thesis they are used in implementing KD-Trees.

(ii) Scikit Learn [87]: It is a machine learning library in Python that provides simple and efficient implementation of many common classification, clustering, preprocessing, dimensionality reduction algorithms. For this thesis Support Vector Machines (SVM), K-Nearest Neighbors (KNN), K-Means, and some common data preprocessing routines are used directly from this library.

(iii) Scikit Image [88]: It is a Python package that provides a collection of image processing algorithms. it is used for the implementation of uniform LBP and various other image preprocessing algorithms.

(iv) TensorFlow [89]: It is a symbolic math library for implementing and training deep neural networks. For this thesis it is used for implementing and training various CNN models on GPU clusters.

[1] http://kimia.uwaterloo.ca/kimia_lab_data_Path24.html

[1] https://anaconda.org/

(v) Keras [1]: It is a high-level neural networks API, written in Python and capable of running on the top of TensorFlow. It is extensively used for conducting CNN related experiments, and it is also used for augementing images.

7.2.2 Computational Resources

The experiments conducted in the thesis are computationally expensive as they require processing gigapixel histopathology images. Furthermore, training and fine-tuning CNN models need powerful GPUs. Two primary computational platforms used for the experiments are **(i)** Sharcnet and **(ii)** Microsoft Azure Cloud. Most of the CPU intensive tasks are carried on Sharcnet whereas GPU intensive tasks are conducted on both the platforms.

Sharcnet is a cluster computing platform, accessible to all Canadian researchers[1]. For this thesis Copper cluster[2] of Sharcnet is used for conducting the CPU intensive experiments. The CPU intensive tasks are related to LBP, BoW, preprocessing images, and creating Kimia Path24 dataset whereas the GPU intensive tasks mainly involve training and tuning CNN models. For every computational job on Copper, around (but not limited to) 10 CPUs (Intel Xeon E5-2630 v3 2.4 GHz each) and around 150 GB RAM can be allocated. The Copper cluster also has GPUs and one could ask up to 4 NVIDIA Tesla K80 GPU (8 GB memory each) for every single GPU computing job.

Microsoft Azure Cloud[3] is an online platform that allows to create GPU enabled virtual machines. These virtual machines can be accessed, managed and used remotely. For this thesis, two virtual machines each containing 12 CPUs, 112 GB RAM, and 2 Nvidia Tesla K-80 (8 GB GPU memory each) are used for conducting the GPU intensive experiments.

7.3 Experimental Setup and Design

Three series of experiments are conducted for the comparative study. The experimental series are designed to analyze the image analysis algorithms for the unbiased assessment of their effectiveness in extracting the discriminative image-features from histopathology images. Three major steps in a experiment series are:

[1] https://www.sharcnet.ca/

[2] https://www.sharcnet.ca/my/systems/show/108

[3] https://azure.microsoft.com/en-ca/

(i) **Patch selection and construction of training dataset:** Kimia Path24 contains 24 full resolution whole-slide images. The large resolution of a whole-slide image makes it impossible to directly apply to directly feed it to an image analysis or machine learning algorithm. Therefore, firstly a training dataset is created by selecting only the "useful" image-patches from each of the 24 whole-slide images. The selected image-patches are of manageable sizes yet provide the useful and relevant information for training the image analysis models.

(ii) **Training image analysis models:** BoW and CNN models are trained or fine-tuned using the training data created in the previous step.

(iii) **Performing retrieval experiments:** The training data contains image patches and their corresponding class labels. This training data constitutes the repository of images which is used for performing CBIR queries. In each experimental series, the training data is indexed using one of the three image analysis algorithm, i.e. LBP, BoW or CNN. Every experiment in a series usually differs in the choice of hyperparameters. In an experiment, CBIR is performed on the indexed dataset with the test patches as a query image. The different distance measurements, i.e., ℓ_1, ℓ_2, χ^2 are used for the retrieval. All the experiments are evaluated based on the standard accuracy calculations of Kimia Path24.

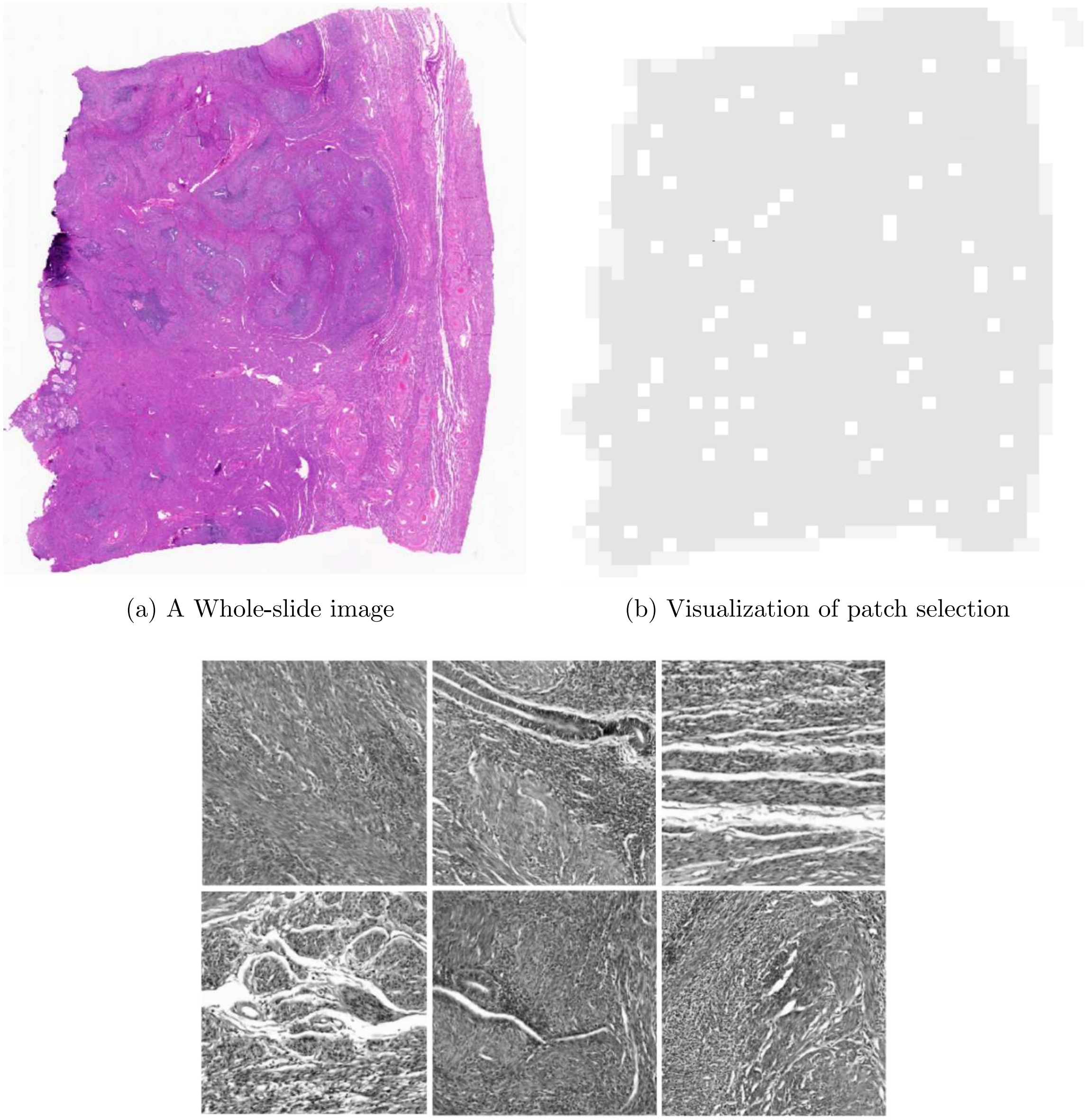
7.3.1 Training Data

For creating the training data, whole-slide images in Kimia Path24 are converted to grayscale and non-overlapping patches (each sized 1000×1000 pixels) are extracted from them. A homogeneity measurement is used to filter out the patches with mostly the background pixels (i.e. "white" patches). A homogeneity measurement of a gray-scale image I, denoted by $h(I) \in [0, 1]$ is given as

$$h(I) = 1 - \frac{1}{H \times W} \sum_{h=1,w=1}^{H,W} |I_{h,w} - \overline{vec(I)}|, \tag{7.1}$$

The above equation measures an average variability between the intensities of pixels of an image I and the median value $\overline{vec(I)}$. The patches obtained after the homogeneity-based filtering are referred as "non-white" patches. The "non-white" patches contain significant textural information allowing an optimal training of an image analysis algorithm.

The process of extracting "non-white" patches from a given whole-slide image is referred as *patch-selection* (shown in Figure 7.1). The homogeneity measurement proves to be a



(a) A Whole-slide image

(b) Visualization of patch selection

(c) Selected patches ($h < 0.99$)

Figure 7.1: Illustration of patch-selection process; **(a)** is a sample whole-slide image from Kimia Path24, and **(b)** is the visualization of its patch selection. White squares are selected for testing. The grey region can be used to generate training/indexing dataset. **(c)** Shows six samples “non-white” patches, each of size 1000 × 1000 and grey-scaled, extracted from the whole-slide image.

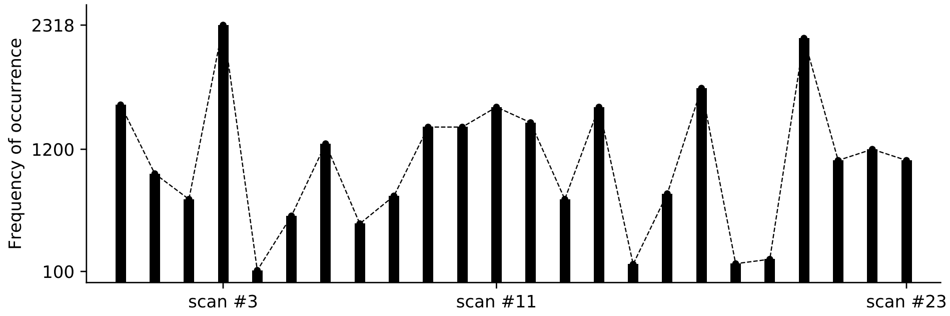


Figure 7.2: Class distribution within the training data obtained from patch-selection process.

powerful technique for patch-selection in Kimia Path24 dataset. The patches with a homogeneity measurement above 99% are excluded and the remaining patches constitute the training data. The training data thus created contains $27,055$ number of patches. There are 24 whole-slide images in Kimia Path24, therefore, training patches are distributed among the 24 different classes where each class represent an individual whole-slide image. The class distribution of training data is shown in Figure 7.2; there is a high class imbalance within the training data. However, no extra effort is invested in balancing the class distribution; since the training data is same across all the experiments, therefore, the class imbalance does not introduce any inter-experimental bias instead it makes the problem more challenging.

7.4 Experiment Series 1: Uniform LBP

For the first experiment series, uniform LBP feature extraction method is implemented using Scikit Image library(in §5.2 of Chapter 5). LBP used for the experiments has two hyperparameters, i.e, radius r and neighbors p. Both testing and training data are used in their original dimensions (i.e., 1000×1000) and no pre-processing is applied.

The total of nine experiments are performed in this experimental series. Each of the nice experiment consists of different configuration for LBP's parameters and distance measurement for the retrieval. The LBP operator with radii 1, 2 and 3 with neighbors 8, 16 and 24 respectively are applied to create histograms of length 59, 243 and 555. The descriptors are created for both testing and training data for each of the configuration. After that,

#	LBP Parameters	Descriptor Length	Distance	Accuracy η_p	 η_W	 η_{total}
1	$(p = 8,\ r = 1)$	59	χ^2	62.49	58.92	**36.82**
2			ℓ_1	61.13	57.50	35.15
3			ℓ_2	56.45	52.95	29.89
4	$(p = 16,\ r = 2)$	243	χ^2	63.62	59.51	**37.86**
5			ℓ_1	62.26	58.19	36.23
6			ℓ_2	55.77	52.12	29.07
7	$\boldsymbol{(p = 24,\ r = 3)}$	**555**	χ^2	64.67	61.08	39.50
8			$\boldsymbol{\ell_1}$	66.11	62.52	**41.33**
9			ℓ_2	59.01	55.94	33.01

Table 7.1: Performance of LBP on Kimia Path24 using different distance measures (ℓ_1, ℓ_2 and χ^2) in different configurations (p, r) for p neighbors and radius r. The length of descriptor is $p \times (p - 1) + 3$. Best results for η_{total} are highlighted in bold.

discriminative power of the descriptors are evaluated against Kimia Path24 using k-NN search (with $k = 1$) to find the similar patches. SciKit-Learn's implementation of k-NN algorithm is directly used without any modification. The test images with original sizes (1000×1000 each) are fed to an LBP model to extract their feature vectors. The $27,055$ number of training images are compared against each of the test patch for the benchmark of dataset using three different kinds of distance measurements (i.e, ℓ_1, ℓ_2 and χ^2).

The results are reported in Table 7.1, showing path-to-scan η_p, whole-scan-accuracy η_W, and total-accuracy η_{total} for all the nine experiments. Varying the radius r helps LBP to capture textural information in different scales. As shown in Table 7.1, a large radius contributes to the higher accuracy values ($2\% - 3\%$ at most). The highest total-accuracy achieved is with the longest descriptor (the length of 555). For all the experiments with descriptors of length 59, the maximum accuracy achieved is $\eta_{total} = 36.82\%$ whereas the descriptor of length 555 achieved the highest accuracy of $\eta_{total} = 41.33$; this corresponds to increase of $\sim 4.51\%$ in the accuracy scores for about $\sim 840\%$ increase in the descriptor length. Therefore, selecting the parameter p of LBP is a design choice, selecting a large value may result in slightly better accuracy however it also comes at the cost of increased storage requirements (needs more space for storing all the feature vectors). If storage cost is not a problem then the choice of the higher value of p is preferable over the smaller value of p.

7.5 Experiment Series 2: BoW

For the second experiment series, BoW is used as image descriptor (§5.3 of Chapter 5). The local features used for BoW model is extracted from uniform LBP with the parameters $r = 1$ and $p = 8$; one may refer the implemented BoW model as "Bag of LBP words" (since each word in the visual vocabulary is an LBP histogram).

Two main reasons for choosing LBP as local feature extractor are **(i)** to save the computation expense required for training a codebook; the length of local feature vectors obtained from LBP is 59 which is considerably smaller than the raw pixels, and **(ii)** to maintain the generality of experiments since LBP method has been used already, therefore, it is more compelling to compare the two approaches (the LBP alone and the BoW with an LBP as local feature).

SciKit Learn's implementation of K-Means algorithms is directly used for constructing a codebook. K-Means in SciKit Learn provides a parameter `n_jobs` which can be set to greater than one for better support on multi-CPU environment. The dense sampling strategy is used rather than the interesting point (IP) for the following reasons: **(i)** the computational complexity of IP detection, e.g., via SIFT and SURF is generally high; **(ii)** an imbalanced distribution of IP for different patches would be an issue, i.e., patches with different types of textures will probably lead to a different number of IP; **(iii)** redundancy of neighboring IP is high. As more IP are likely to be detected around the strong texture regions than the weak texture regions, reduction of the redundancy of samples from dense IP regions is a challenge.

Before extracting descriptors and training a codebook, 300 patches are randomly selected from all the available patches such that their homogeneity is larger than the average homogeneity value of all the patches for each whole-slide image. To accelerate dictionary training, all $7{,}200 = 24 \times 300$ patches are down-sampled to 500×500 and meshed into windows of sizes 16×16 and 32×32 grids without an overlap. Raw pixel descriptors and LBP features are extracted from these sub-patches. The extracted descriptors are used to train dictionaries whose size is set to 250, 500, or 1000 (§5.3.1c., pp. 49). The word-frequency histogram of each patch is then extracted using a successive application encoding-pooling (§5.3.1d., pp. 49).

The total of 18 experiments are performed with different configurations of windows szize ($w \times w$), codebook size (k), and distance measurement. The experiments are conducted using the same protocol as described in the previous section (§7.5). The different accuracy values obtained from all the 18 experiments are provided in Table 7.2. The size of image descriptors from BoW approach is same as the size of codebook, i.e. k. The best performing

Codebook		Window size		Accuracy		
size (k)	#	($w \times w$)	Distance	η_p	η_W	η_{total}
$k = 250$	1	$w = 16$	χ^2	48.53	46.61	22.61
	2		ℓ_1	50.26	48.17	24.21
	3		ℓ_2	52.45	49.92	**26.18**
	4	$w = 32$	χ^2	46.11	43.39	20.00
	5		ℓ_1	46.57	43.31	**20.16**
	6		ℓ_2	46.57	42.74	19.90
$k = 500$	7	$w = 16$	χ^2	65.36	48.53	31.71
	8		ℓ_1	66.94	50.26	33.64
	9		ℓ_2	66.04	52.45	**34.63**
	10	$w = 32$	χ^2	60.23	56.85	**34.24**
	11		ℓ_1	60.08	52.45	31.51
	12		ℓ_2	58.72	54.53	32.02
$\boldsymbol{k = 1000}$	13	$\boldsymbol{w = 16}$	$\boldsymbol{\chi^2}$	75.77	72.16	**54.67**
	14		ℓ_1	74.04	70.18	51.96
	15		ℓ_2	71.77	64.47	46.27
	16	$w = 32$	χ^2	71.55	67.43	48.24
	17		ℓ_1	71.02	66.99	**47.57**
	18		ℓ_2	72.15	63.14	45.55

Table 7.2: Performance of BoW on Kimia Path24 using different distance measures (ℓ_1, ℓ_2 and χ^2) in different configurations varying codebook size and window size for local feature extraction. The length of the descriptor is same as codebook size s (second column). Best results for η_{total} are highlighted in bold.

BoW model on Kimia Path24 dataset has the following parameters: codebook size $k = 1000$, window size $w = 16$, and distance measurement using χ^2.

7.6 Experiment Series 3: CNN

#	Model	Remarks	Descriptor Length	η_p	η_W	η_{total}
1	CNN$_1$	Trained from scratch	1024	64.98	64.75	**41.80**
2	VGG16	Pre-trained	512	65.21	64.96	**42.36**
3		Fine-tuned		63.85	66.23	42.29
4	**Inception-v3**	Pre-trained	2048	70.94	71.24	50.54
5		**Fine-tuned**		74.87	76.10	**56.98**

Table 7.3: Performance of three CNN models on Kimia Path24, **(i)** CNN$_1$, is trained from scratch, **(ii)** VGG16 with both pre-trained and fine-tuned scheme, and **(iii)** Inception-v3 again with both schemes. The best score for each model are highlighted in bold, whereas best performing model is highlighted in first column.

For the last experiment series, CNN are used for extracting image-features. Three different CNN models are used, and their implementation details are explain in §5.4.1 of Chapter 5.

7.6.1 CNN$_1$ Model

Training data is re-sized to 128 × 128 using bi-cubic interpolation (from SciPy library). The mini-batch SGD with momentum with learning rate of 0.1 and large momentum of 0.9, along with cross-entropy loss function are used to train the CNN$_1$ model. The CNN$_1$ model is presented with Kimia Path24 and its classification accuracy is calculated directly from its Softmax predictions. The final total accuracy η_{total} obtained from CNN$_1$ model is 41.80%, other accuracy values are reported in Table 7.3.

7.6.2 VGG16 Model

VGG16 is implemented using Keras library and it is initialized with pre-trained weights. The training data is first resized to (224×224) and then $(103.939, 116.779, 123.68)$ is subtracted from each of the red, green and blue channel respectively (each pixel is between 0.0 – 255.0 floating point number). The two sets of experiments are conducted using VGG16 network.

The pre-trained network is first used as a feature extractor without any fine-tuning. As pre-trained networks are trained in other domains (very different image categories) and hence cannot be used as classifier, therefore the deep features are used to train a linear Support Vector Machine (SVM) for classification. The Python package scikit-learn as well as LIBSVM are used to train SVM classifiers with a linear kernel. Both NumPy and SciPy were leveraged to manipulate and store data during these experiments.

For the second expeirment, VGG network is fine-tuned with Kimia Path24 dataset. Using the Keras library, the convolutional layers were first separated from the top fully connected layers. The training patches were fed through the model to create a set of "bottleneck" features[1] to initially pre-train the new fully-connected layers. These features were used to initialize the weights of a fully connected MLP consisting of one 256 full-connected layer followed by ReLu layer and another 24-neuron fully connected layer with Softmax normalized outputs.

The accuracy were calculated based on the classification accuracy of the fine-tuned VGG16 model. The total accuracy η_{total} from both the models, pre-trained and fine-tuned, are 42.36% and 42.29% respectively. Apparently, pre-trained model performs better, the reason could be linear SVM is a better classifier than the MLP trained during the fine-tune process. All the final accuracy for both experiments on VGG16 are reported in Table 7.3.

7.6.3 Inception-v3 Model

Inception-v3 model is implemented using Keras library and initialized with pre-trained weights. The training data is first resized (299 × 299) and every pixel value is scaled to $[-1, 1]$. Just like VGG16, two sets of experiments are performed for Inception-v3 as well, one with pre-trained networks and second with fine-tuned network.

For fine-tuning, the fully connected layers were replaced with one 1024 dense ReLU layer and a Softmax classification layer. The fully connected layers were pretrained on bottleneck features and then attached to the convolutional layers and training on the final two inception blocks was then performed. The fine-tuned Inception-v3 model performed best out of all with total accuracy $\eta_{total} = 56.98$, as reported in Table 7.3. The class

[1] Large dimensional features are projected to low dimensional space and then to solution space, the intermediate low dimensional features are known as bottleneck features.

Image Descriptor	Remarks	Descriptor Length	η_p	η_w	η_{total}
CNN	Fine tuned Inception-v3	2048	74.87	**76.10**	**56.98**
BoW	$k = 1000,\ w = 16$	1000	**75.77**	72.16	54.67
LBP	$p = 24, r = 3$	555	66.11	62.52	41.33

Table 7.4: Comparison of performance of Kimia Path24 for each of the image descriptor achieving best total accuracy η_{total}.

activation mappings (CAMs) for the fine-tuned Inception-v3 network on randomly selected test patches are illustrated in Figure 7.3, CAM images are generated using GRAD-CAM algorithm from [5].

7.7 Analysis and Discussion

The results demonstrate that Kimia Path24 is a challenging dataset since most of the Machine Learning (ML) techniques scored less than 60% accuracy. The fine-tuned Inception-v3 model achieved the best total accuracy score of 56.98% followed by the BoW model with the accuracy score of 54.67%. CNN models can improve even further if a more training images can be extracted from the whole-slide images.

It was surprising to find out that using the features from the pre-trained CNN models deliver the results comparable with the CNN network (CNN_1) trained with considerable effort and resources. Another surprising effect was that the transfer learning via fine-tuning VGG16 was not able to provide any improvement compared to the deep features extracted from the pre-trained network. However, for Inception-v3 the improvement was immediate. Perhaps, the most obvious reaction to this finding is that if there were larger quantity of samples, i.e., millions of histopathological images, and there were enough computation resources to train them, then the fine-tuned CNN would deliver much better results than the transfer learning. Although this statement is supported by the comparable empirical evidence, it remains speculation for a sensitive field like medical imaging.

LBP performed surpsingly well and reached closed to both the approaches in the accuracy scores. One should bear in mind that LBP did in fact process the images in their original dimensions whereas CNNs and BoW required substantial downsampling. Taking into account the training complexity, one may prefer LBP over CNNs since the former is intrinsically simple and fast. As well, LBP has been designed to deal with textures at the spatial level.

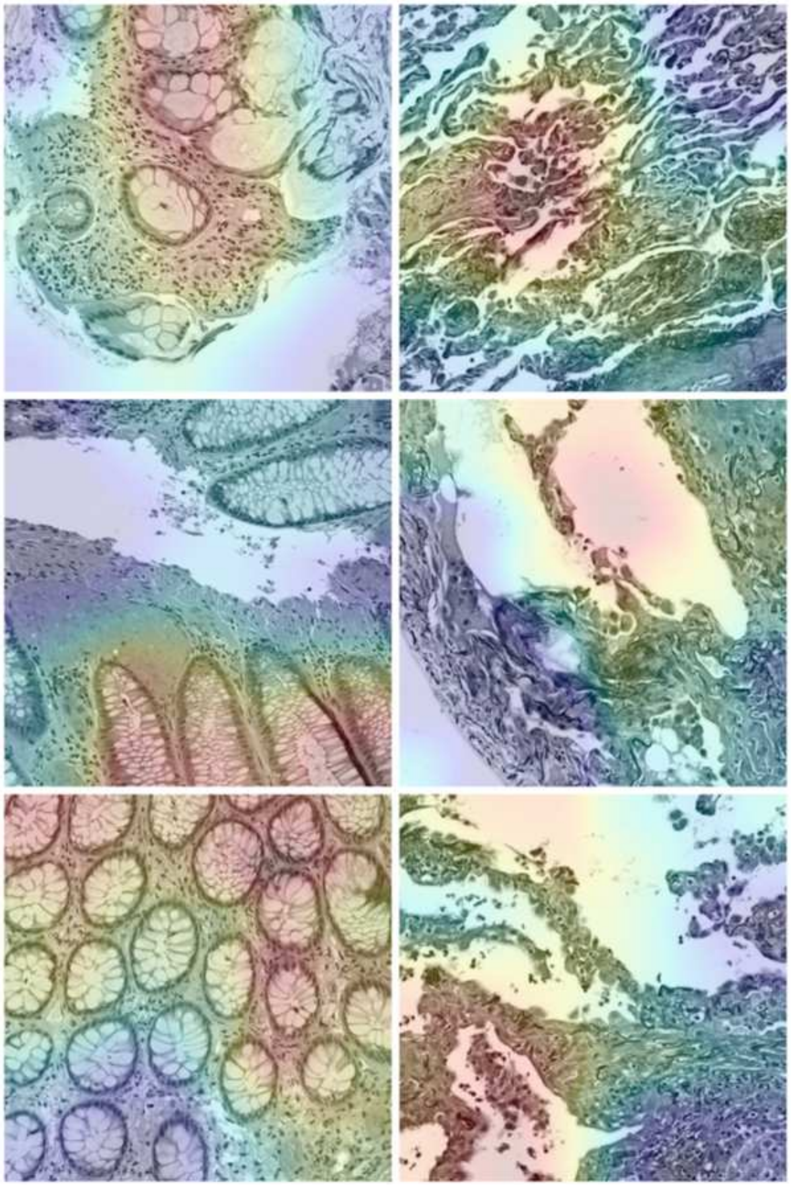


Figure 7.3: Activation maps using randomly selected patches from the Kimia Path24 testing data. The patches within each column are the same class and the labels per column are 4 and 8 respectively. The activation maps are created using the Keras Visualization Toolkit and the Grad-CAM algorithm [5]. Red areas has more influence on the label prediction (seen best in color).

Chapter 8

General Conclusion

> “*Problems worthy of attack prove their worth by fighting back.*”
>
> — Piet Hein

All three image analysis approaches, Local Binary Pattern (LBP), Bag of visual Words (BoW), and Convolution Neural Network (CNN) are able to capture discriminative visual features in pathology images as reflected in the benchmark measurements on *Kimia Path24* dataset. Testing different hyperparameters demonstrate the design flexibility of the three approaches, and also help in understanding the effects of hyperparameters on the quality of extracted image features.

The proposed dataset, *Kimia Path24* may be regarded easy because the benchmark is based on matching patches that come from the same scan/patient. However, as the results demonstrate, this is clearly not the case (total accuracy scores are under 60%). The best total accuracy η_{total} scores achieved by LBP, BoW, and CNN are 41.33%, 54.67%, and 56.98% respectively.

Fine-tuned Inception-v3 (CNN) achieved the best total-accuracy $\eta_{total} = 56.98\%$ followed by BoW (dictionary size of $k = 1000$ and window size of $w = 16$) with $\eta_{total} = 54.67\%$. Interestingly, the best CNN model scored slightly less on the patch-to-scan accuracy ($\eta_p = 74.87\%$) compared to the second best by BoW ($\eta_p = 75.77\%$). However, the whole-scan accuracy of the best CNN approach is $\eta_w = 76.10\%$ whereas for the BoW is much lesser, i.e. $\eta_w = 72.16\%$. The higher whole-scan accuracy suggests that the CNN approach has better generalization capabilities — reason could be the MLP connected at the end of the CNN model which provides an advantage in terms of classification capabilities.

The experiments on BoW reveal that a small codebook is not as capable of describing image information as a large codebook. The total-accuracy drastically improved from 26.18% to 54.67% by increasing the codebook size k from 250 to 1000. In all the experiments with BoW, window size of $w = 16$ outperformed the $w = 32$, suggesting that a smaller windows size is a preferable option for the BoW feature extraction model.

One of the surprising result is that the total-accuracy score of self trained CNN model (CNN_1) and LBP (number of neighbors $p = 24$ and radius $r = 3$) are very close to each other — 41.80% and 41.33% respectively. CNN_1 model scores better than LBP due to the higher whole-scan accuracy ($\eta_w = 64.75\%$ vs 62.52%), i.e. better classification accuracy due to the attached MLP at the end. In fact, the other pretrained CNN models are not superior than LBP approach, i.e., VGG16 ($\eta_{total} = 42.36\%$) and Inception-v3 ($\eta_{total} = 50.54\%$). These results suggest that using off the shelf CNN models for feature extraction (common approach in the community at the present time) perform equally well as the handcrafted methods, such as LBP. This serves as a reminder to the community that black box methods (i.e., CNN) should be used with caution and similar results can be achieved with a good understanding of the problem domain even with the simpler methods.

8.1 Recommendations

In the practice of medical imaging, class labels of images are generally *a priori* information as it is known that an image is depicting a breast or prostate tissue. Hence, the accurate retrieval algorithms appear to be more needed than the classification methods (even if a class contains all malignant cases, we still need to find the most similar case). The LBP approach lacks the training, therefore, it cannot take the benefit of knowledge existing in the training data. Although the BoW approach has a training phase yet it does not exploit the known class labels. Perhaps, information about class labels can be incorporated in the codebook construction step of BoW. One can even hypothesize that the extra information about class labels might confuse the Machine Learning (ML) algorithms. But there is no such evidence from the results in this thesis.

Comparing to natural images and other biomedical image categories, the histopathology images are generally acknowledged to be more complicated in structure, dimensionality, and texture. The overlapped clustered or tightly clumped nuclei in histopathology image make them more difficult to recognize or classify. As in case with high image resolution and large scale dataset, LBP was used in the BoW approach, to minimize the computational expense. The BoW model can be tested with various different local-level extractor to study its full potential for the retrieval in histopathology images.

A significant pitfall of *Kimia Path24* is the missing pathological indicators and annotations. The pathological indicators include disease type, and region of interests (ROIs), but they require expert opinions from real pathologists making them difficult to acquire.

CNNs are more scalable than the other two approaches. Hypothetically, with an availability of a large amount of annotated and labeled pathological data, CNNs can be very a powerful approach. However, such dataset at present is not available in the histopathology domain.

References

[1] François Chollet et al. Keras. https://github.com/fchollet/keras, 2015.

[2] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

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

[4] Juan C. Caicedo, Angel Cruz, and Fabio A. Gonzalez. Histopathology image classification using bag of features and kernel functions. In *International Conference on Artificial Intelligence in Medicine, Lecture Notes in Computer Science*, pages 126–135.

[5] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *See https://arxiv. org/abs/1610.02391 v3*, 7(8), 2016.

[6] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going Deeper with Convolutions. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[7] Institute of Medicine. *To Err Is Human: Building a Safer Health System.*

[8] C. W. Elston and I. O. Ellis. Pathological prognostic factors in breast cancer. I. The value of histological grade in breast cancer: Experience from a large study with long-term follow-up. *Histopathology*, 19(5):403–410, 1991.

[9] Eric Kreit, Lydia M Mäthger, Roger T Hanlon, Patrick B Dennis, Rajesh R Naik, Eric Forsythe, and Jason Heikenfeld. Biological versus electronic adaptive coloration: How can one inform the other? *Journal of The Royal Society Interface*, 10(78):20120601.

[10] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708.

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

[12] Liron Pantanowitz. Digital images and the future of digital pathology. *Journal of Pathology Informatics*, 1.

[13] H. Irshad, A. Veillard, L. Roux, and D. Racoceanu. Methods for Nuclei Detection, Segmentation, and Classification in Digital Histopathology: A Review #x2014;Current Status and Future Potential. 7:97–114.

[14] Huazhang Guo, Joe Birsa, Navid Farahani, Douglas J. Hartman, Anthony Piccoli, Matthew O'Leary, Jeffrey McHugh, Mark Nyman, Curtis Stratman, Vanja Kvarnstrom, Samuel Yousem, and Liron Pantanowitz. Digital pathology and anatomic pathology laboratory information system integration to support digital pathology sign-out. *Journal of Pathology Informatics*, 7.

[15] Jon Griffin and Darren Treanor. Digital pathology in clinical use: Where are we now and what is holding us back? 70(1):134–145.

[16] Andreas Kårsnäs. Image Analysis Methods and Tools for Digital Histopathology Applications Relevant to Breast Cancer Diagnosis, 2014. OCLC: 941290134.

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

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

[19] 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.

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

[21] David C. Wilbur, Kalil Madi, Robert B. Colvin, Lyn M. Duncan, William C. Faquin, Judith A. Ferry, Matthew P. Frosch, Stuart L. Houser, Richard L. Kradin, Gregory Y. Lauwers, David N. Louis, Eugene J. Mark, Mari Mino-Kenudson, Joseph Misdraji, Gunnlauger P. Nielsen, Martha B. Pitman, Andrew E. Rosenberg, R. Neal Smith, Aliyah R. Sohani, James R. Stone, Rosemary H. Tambouret, Chin-Lee Wu, Robert H. Young, Artur Zembowicz, and Wolfgang Klietmann. Whole-slide imaging digital pathology as a platform for teleconsultation: A pilot study using paired subspecialist correlations. *Archives of Pathology & Laboratory Medicine*, 133(12):1949–1953.

[22] 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.

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

[24] 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.

[25] 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.

[26] Thomas Kalinski, Ralf Zwönitzer, Saadettin Sel, Matthias Evert, Thomas Guenther, Harald Hofmann, Johannes Bernarding, and Albert Roessner. Virtual 3D microscopy using multiplane whole slide images in diagnostic pathology. 130(2):259–264.

[27] Liron Pantanowitz, AnilV Parwani, ClaytonA Wiley, Ishtiaque Ahmed, William Cable, Lydia Contis, Anthony Demetris, and Andrew Lesniak. Experience with multimodality telepathology at the University of Pittsburgh Medical Center. *Journal of Pathology Informatics*, 3(1):45.

[28] Filippo Fraggetta, Salvatore Garozzo, GianFranco Zannoni, Liron Pantanowitz, and EstherDiana Rossi. Routine digital pathology workflow: The Catania experience. *Journal of Pathology Informatics*, 8(1):51.

[29] Thomas W. Bauer, Renee J. Slaw, Jesse K. McKenney, and Deepa T. Patil. Validation of whole slide imaging for frozen section diagnosis in surgical pathology. *Journal of Pathology Informatics*, 6(1):49.

[30] John R Gilbertson, Jonhan Ho, Leslie Anthony, Drazen M Jukic, Yukako Yagi, and Anil V Parwani. Primary histologic diagnosis using automated whole slide imaging: a validation study. *BMC Clinical Pathology*, 6(1):4, apr 2006.

[31] Bernard Têtu, Émilie Perron, Said Louahlia, Guy Paré, Marie-Claude Trudel, and Julien Meyer. The Eastern Québec Telepathology Network: A three-year experience of clinical diagnostic services. *Diagnostic Pathology*, 9 Suppl 1:S1.

[32] Sten. Thorstenson, Jesper. Molin, and Claes. Lundstrm. Implementation of large-scale routine diagnostics using whole slide imaging in Sweden: Digital pathology experiences 2006-2013. *Journal of Pathology Informatics*, 5(1):14, 2014.

[33] Thomas W. Bauer. Commentary: Whole-slide Images–Good enough for primary diagnosis? *Journal of Pathology Informatics*, 9(1):3.

[34] C Stratman, L Drogowski, and J Ho. Digital pathology in the clinical workflow: A time and motion study. *Pathology Visions*.

[35] Ronald L. Sirota. Defining Error in Anatomic Pathology. *Archives of Pathology & Laboratory Medicine*, 130(5):604–606.

[36] Pathologic Mistake — AHRQ Patient Safety Network.

[37] Raphael Rubin, David S Strayer, Emanuel Rubin, and others. *Rubin's Pathology: Clinicopathologic Foundations of Medicine*. Lippincott Williams & Wilkins.

[38] Richard J. Zarbo, Frederick A. Meier, and Stephen S. Raab. Error detection in anatomic pathology. *Archives of Pathology & Laboratory Medicine*, 129(10):1237–1245.

[39] Daniele Giansanti, Livia Castrichella, and Maria Rosaria Giovagnoli. The design of a health technology assessment system in telepathology. *Telemedicine Journal and E-Health: The Official Journal of the American Telemedicine Association*, 14(6):570–575.

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

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

[42] 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.

[43] Juan C Caicedo, Fabio A González, and Eduardo Romero. Content-based histopathology image retrieval using a kernel-based semantic annotation framework. *Journal of biomedical informatics*, 44(4):519–528, 2011.

[44] Metin N Gurcan, Laura E Boucheron, Ali Can, Anant Madabhushi, Nasir M Rajpoot, and Bulent Yener. Histopathological image analysis: A review. *IEEE reviews in biomedical engineering*, 2:147–171, 2009.

[45] Kirsten L Weind, Cynthia F Maier, Brian K Rutt, and Madeleine Moussa. Invasive carcinomas and fibroadenomas of the breast: comparison of microvessel distributions–implications for imaging modalities. *Radiology*, 208(2):477–483, 1998.

[46] PH Bartels, D Thompson, M Bibbo, and JE Weber. Bayesian belief networks in quantitative histopathology. *Analytical and quantitative cytology and histology/the International Academy of Cytology [and] American Society of Cytology*, 14(6):459–473, 1992.

[47] PW Hamilton, N Anderson, PH Bartels, and D Thompson. Expert system support using bayesian belief networks in the diagnosis of fine needle aspiration biopsy specimens of the breast. *J. of Clinical Pathology*, 47(4):329–336, 1994.

[48] 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.

[49] 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.

[50] 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.

[51] Lei Zheng, Arthur W Wetzel, John Gilbertson, and Michael J Becich. Design and analysis of a content-based pathology image retrieval system. *IEEE Transactions on Information Technology in Biomedicine*, 7(4):249–255, 2003.

[52] 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.

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

[54] 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.

[55] 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.

[56] Scott R Granter, Andrew H Beck, and David J Papke Jr. Alphago, deep learning, and the future of the human microscopist. *Archives of pathology & laboratory medicine*, 141(5):619–621, 2017.

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

[58] Jiawei Su, Danilo Vasconcellos Vargas, and Sakurai Kouichi. One pixel attack for fooling deep neural networks. *arXiv preprint arXiv:1710.08864*, 2017.

[59] Gaurav Sharma and Alexis Carter. Artificial Intelligence and the Pathologist: Future Frenemies? 141(5):622–623.

[60] S Doyle, M Hwang, MD Feldman, JE Tomaszewski, and A Madabhushi. Using manifold learning for content-based image retrieval of prostate histopathology. In *Workshop on Content-Based Image Retrieval for Biomedical Image Archives*, 2007.

[61] J. G. Dy, C. E. Brodley, A. Kak, L. S. Broderick, and A. M. Aisen. Unsupervised feature selection applied to content-based retrieval of lung images. 25(3):373–378.

[62] Akshay Sridhar. *Content-Based Image Retrieval of Digitized Histopathology via Boosted Spectral Embedding (BoSE).* Rutgers The State University of New Jersey-New Brunswick and University of Medicine and Dentistry of New Jersey.

[63] Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. Object Recognition with Gradient-Based Learning. In *Shape, Contour and Grouping in Computer Vision*, Lecture Notes in Computer Science, pages 319–345. Springer, Berlin, Heidelberg.

[64] Kunihiko Fukushima and Sei Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and Cooperation in Neural Nets*, pages 267–285. Springer.

[65] Saeed Reza Kheradpisheh, Masoud Ghodrati, Mohammad Ganjtabesh, and Timothée Masquelier. Deep Networks Can Resemble Human Feed-forward Vision in Invariant Object Recognition. 6(1).

[66] Jianxin Wu. Introduction to convolutional neural networks.

[67] Zhifei Zhang. Derivation of Backpropagation in Convolutional Neural Network (CNN), 2016.

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

[69] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

[70] Kazuyuki Hara, Daisuke Saitoh, and Hayaru Shouno. Analysis of dropout learning regarded as ensemble learning. In *International Conference on Artificial Neural Networks*, pages 72–79. Springer, 2016.

[71] T. Ojala, M. Pietikainen, and D. Harwood. Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In *Proceedings of 12th International Conference on Pattern Recognition*, volume 1, pages 582–585.

[72] Dong-chen He and Li Wang. Texture Unit, Texture Spectrum, And Texture Analysis. 28(4):509–512.

[73] Rakesh Mehta and Karen Egiazarian. Dominant Rotated Local Binary Patterns (DRLBP) for texture classification. 71:16–22.

[74] Timo Ahonen, Jiří Matas, Chu He, and Matti Pietikäinen. Rotation invariant image description with local binary pattern histogram fourier features. In *Scandinavian Conference on Image Analysis*, pages 61–70. Springer.

[75] Lin Zhang, Lei Zhang, Zhenhua Guo, and David Zhang. Monogenic-LBP: A new approach for rotation invariant texture classification. In *Image Processing (ICIP), 2010 17th IEEE International Conference On*, pages 2677–2680. IEEE.

[76] M. Dammak, M. Mejdoub, and C. B. Amar. A survey of extended methods to the bag of visual words for image categorization and retrieval. In *2014 International Conference on Computer Vision Theory and Applications (VISAPP)*, volume 2, pages 676–683.

[77] A. Pastor López-monroy, Hugo Jair Escalante, Angel Cruz-roa, and Fabio A. González. Bag-of-Visual-Ngrams for Histopathology Image Classification. In *International Seminar on Medical Information Processing and Analysis*, 2013.

[78] U. Avni, H. Greenspan, E. Konen, M. Sharon, and J. Goldberger. X-ray Categorization and Retrieval on the Organ and Pathology Level, Using Patch-Based Visual Words. 30(3):733–746.

[79] Angel Cruz-Roa, Juan C. Caicedo, and Fabio A. González. Visual pattern mining in histology image collections using bag of features. 52(2):91–106.

[80] David G. Lowe. Distinctive image features from scale-invariant keypoints. 60(2):91–110.

[81] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded Up Robust Features. In *Computer Vision – ECCV 2006*, Lecture Notes in Computer Science, pages 404–417. Springer, Berlin, Heidelberg, 2006.

[82] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. BRIEF: Binary Robust Independent Elementary Features. In *Computer Vision – ECCV 2010*, Lecture Notes in Computer Science, pages 778–792. Springer, Berlin, Heidelberg.

[83] Randy Allan and W Kinsner. A study of microscopic images of human breast disease using competitive neural networks. In *Electrical and Computer Engineering, 2001. Canadian Conference On*, volume 1, pages 289–293. IEEE.

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

[85] 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.

[86] Travis E Oliphant. Python for scientific computing. *Computing in Science & Engineering*, 9(3), 2007.

[87] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[88] Stefan Van der Walt, Johannes L Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D Warner, Neil Yager, Emmanuelle Gouillart, and Tony Yu. scikit-image: image processing in python. *PeerJ*, 2:e453, 2014.

[89] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[90] Shaimaa Al-Janabi, André Huisman, Aryan Vink, Roos J. Leguit, G. Johan A. Offerhaus, Fiebo J. W. ten Kate, and Paul J. van Diest. Whole slide images for primary diagnostics of gastrointestinal tract pathology: A feasibility study. 43(5):702–707.

[91] Liron Pantanowitz, AndrewJ Evans, JohnD Pfeifer, LauraC Collins, PaulN Valenstein, KeithJ Kaplan, DavidC Wilbur, and TerenceJ Colgan. Review of the current state of whole slide imaging in pathology. *Journal of Pathology Informatics*, 2(1):36.

[92] Mike Isaacs, Jochen K. Lennerz, Stacey Yates, Walter Clermont, Joan Rossi, and John D. Pfeifer. Implementation of whole slide imaging in surgical pathology: A value added approach. *Journal of Pathology Informatics*, 2(1):39.

[93] Rajendra Singh, Lauren Chubb, Liron Pantanowitz, and Anil Parwani. Standardization in digital pathology: Supplement 145 of the DICOM standards. *Journal of Pathology Informatics*, 2.

APPENDICES

Appendix A

A.1 Background Information

A.1.1 Hematoxylin-Eosin (H&E) staining

Hematoxylin-Eosin (H&E) is most common staining method used by pathologists at present [28, 22]. Hematoxylin stains cell nuclei blue, while Eosin stains cytoplasm and connective tissue in pink, as shown in Figure A.1. Due to the long history of H&E, and many supporting data and research for its efficacy, there is a strong belief among many pathologists that H&E will continue to be the common practice over the next 50 years [3].

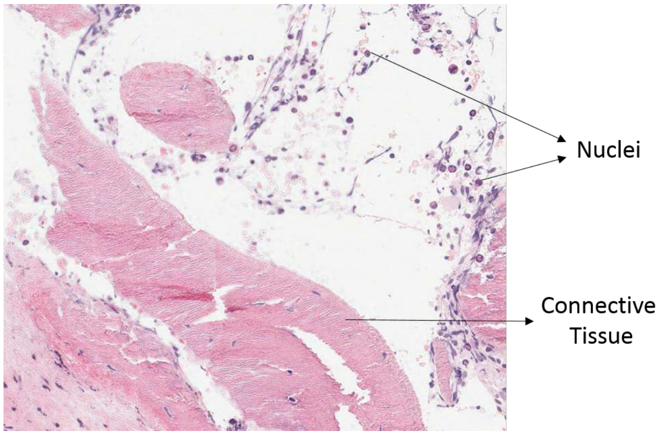


Figure A.1: A sample digital slide stained with Hematoxylin-Eosin (H&E). Hematoxylin stains cell nuclei in blue whereas Eosin stains connective tissue in pink.

A.1.2 Pro and cons of WSI technology

	Advantages	
1.	Primary diagnosis	[22, 28, 90, 24]
2.	Education, training and mentoring	
	• Easy access to students and pathologists in training	[17, 91, 18]
	• Sharing of instructive and unusual cases digitally	
	• Promotes interactive teaching environment	
3.	Telepathology	
	• Receive a second opinion remotely	[21, 18, 32]
	• No need to transport glass slides physically	
	• As a part of quality assurance protocol	
	• More educated and assertive decisions for difficult cases	
4.	Archiving interesting and legal cases	
	• Digital slides are not subjected to degradation	[22, 18]
	• Easy access to patient's previous histology	
	• Improves traceability for misdiagnosis or legal matters	
5.	Image analysis and computational pathology	Section TODO
6.	Improved patient safety	
	• Integration with Laboratory Information System (LIS)	[18, 20, 21]
	• Prevent wrongful labeling and transportation errors	
	• Prevent damage to valuable and irreplaceable glass slides	

(a)

	Disadvantages	
1.	Costly for initial setup	
	• A single WSI scanner can cost around $135,000 [92]	[27, 20, 18]
	• Requires upgrade to IT infrastructure	
	• High initial costs discourages small pathology labs	
2.	Limited focus control	[19]
3.	Massive file size	
	• A single whole-slide image can be up to 1 gigabyte or more	[91, 19]
	• Imposes storage, archiving and transmission challenge	
4.	Different vendors and their standards	
	• Limited interoperability between different vendors	[91, 19]
	• However, DICOM standard for digital slides exists ([93])	
5.	Pathologists reluctance to adapt	[27, 20]

(b)

Table A.1: Pros and cons of WSI technology are respectively discussed in **(a)** and **(b)**.

Appendix B

B.1 Pretrained CNNs used for the experiments

B.1.1 VGG 16

The VGG-16 is very popular CNN model developed by Visual Geometry Group at the University of Oxford [84]. It has 16 layers which can learn total of around 138 million parameters. Detailed textual description of the VGG-16 architecture is given in Table B.1.

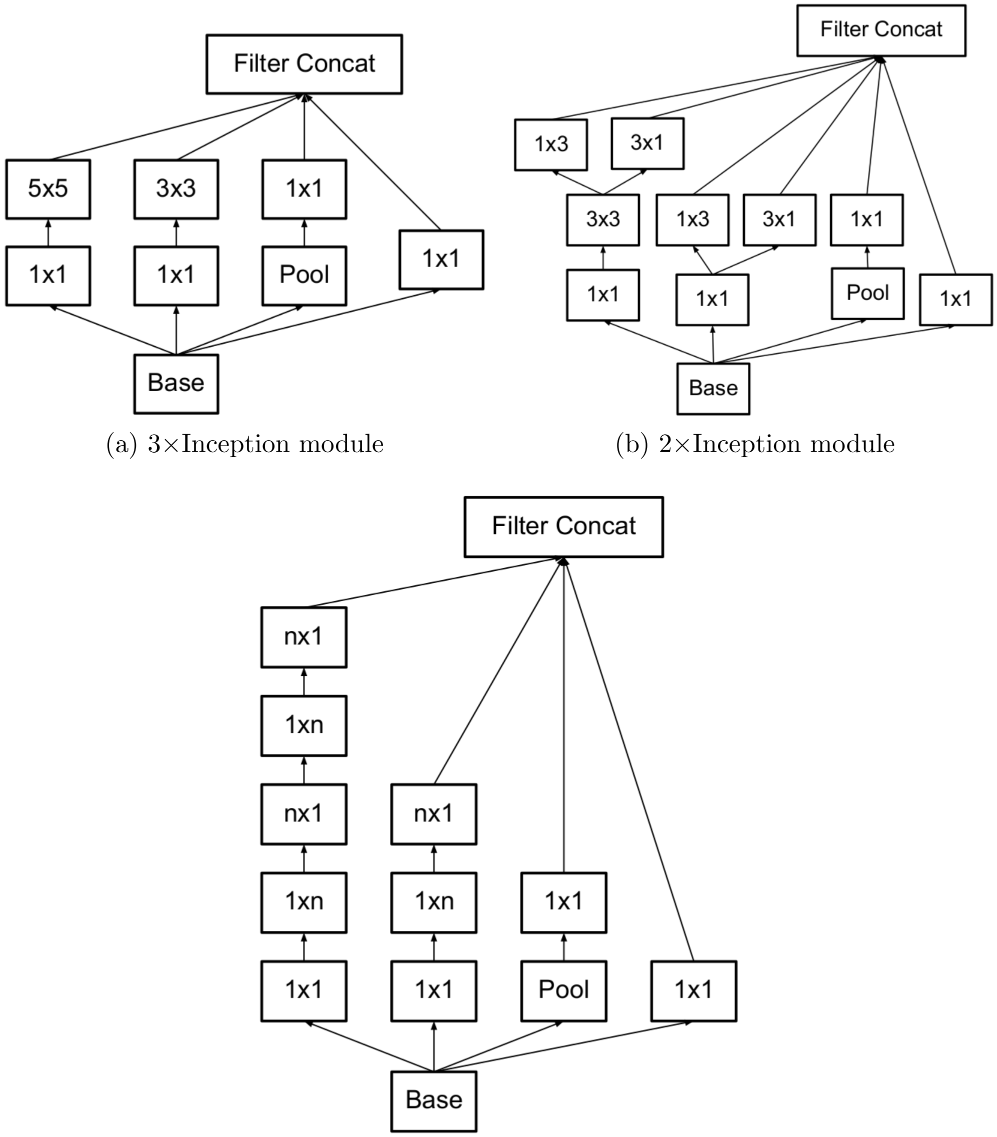
B.1.2 Inception-v3

Inception-v3 is third version of the *Inception* architectures created at Google's DeepMind. First Inception module was introduced by M. Szegedy et al (DeepMind) around 2014 in [6]. The key insight of the Inception module is the realization that conventional convolutional filters can only learn linear functions of their inputs, therefore, need is to increase their learning abilities and abstraction power by having more complex filters.

The earliest Inception module computes 1 × 1 filters, 3 × 3 filters and 5 × 5 filters in the parallel, then applies the bottleneck 1 × 1 filters [6]. However, architecture of Inception-v3, introduced within a year of time from the first version of the Inception, removed the 5 × 5 filters and replaced them by two successive layers of 3 × 3 filters along with batch normalization layers [2]. Full textual description of the Inception-v3 is given in Table B.2 and different Inception modules that creates the Inception-v3 have been presented in Figure B.1.

#	Type	# filters @ patch size/stride	Parameters	Output size
0	Image			$3 \times 224 \times 224$
1	Convolution	$64 @ 3 \times 3/1$	1729	$64 \times 224 \times 224$
2	Convolution	$64 @ 3 \times 3/1$	$36,928$	$64 \times 224 \times 224$
3	Max pooling	2×2	0	$64 \times 224 \times 224$
4	Convolution	$128 @ 3 \times 3/1$	$73,856$	$128 \times 112 \times 112$
5	Convolution	$128 @ 3 \times 3/1$	$147,584$	$128 \times 112 \times 112$
6	Max Pooling	2×2	0	$128 \times 56 \times 56$
7	Convolution	$256 @ 3 \times 3/1$	$295,168$	$256 \times 56 \times 56$
8	Convolution	$256 @ 3 \times 3/1$	$590,080$	$256 \times 56 \times 56$
9	Convolution	$256 @ 3 \times 3/1$	$590,080$	$256 \times 56 \times 56$
10	Max pooling	2×2	0	$256 \times 28 \times 28$
11	Convolution	$512 @ 3 \times 3/1$	$1,180,160$	$512 \times 28 \times 28$
12	Convolution	$512 @ 3 \times 3/1$	$2,359,808$	$512 \times 28 \times 28$
13	Convolution	$512 @ 3 \times 3/1$	$2,359,808$	$512 \times 28 \times 28$
14	Max pooling	2×2	0	$512 \times 14 \times 14$
15	Convolution	$512 @ 3 \times 3/1$	$2,359,808$	$512 \times 14 \times 14$
16	Convolution	$512 @ 3 \times 3/1$	$2,359,808$	$512 \times 14 \times 14$
17	Convolution	$512 @ 3 \times 3/1$	$2,359,808$	$512 \times 14 \times 14$
18	Max pooling	2×2	0	$512 \times 7 \times 7$
19	FC	4096	$\mathbf{102,764,544}$	
	Dropout		0	
20	FC	4096	$16,781,312$	
	Dropout		0	
21	FC	1000	$4,097,000$	
$\sum$			$\mathbf{138,357,544}$	

Table B.1: VGG-16 architecture: All the convolutions are zero padded to prevent changes in the sizes. ReLU activation functions is used applied after Max pooling. The channels mean is subtracted from each pixel as a preprocessing step $(103.939, 116.779, 123.68)$ [1]. The dropout probability is 0.5.



(a) 3×Inception module

(b) 2×Inception module

(c) 5×Inception module

Figure B.1: Different Inception modules used in architecture of Inception-v3. Image is taken from [6]

Type	patch size/stride	Output size
Input		$3 \times 299 \times 299$
Convolution	$3 \times 3/2$	$32 \times 149 \times 149$
Convolution	$3 \times 3/1$	$32 \times 147 \times 147$
Convolution Padded	$3 \times 3/1$	$64 \times 147 \times 147$
Max Pooling	$3 \times 3/2$	$64 \times 73 \times 73$
Convolution	$3 \times 3/1$	$80 \times 71 \times 71$
Convolution	$3 \times 3/2$	$192 \times 35 \times 35$
Convolution	$3 \times 3/1$	$288 \times 35 \times 35$
3×Inception (Fig. B.1a)		$768 \times 17 \times 17$
5×Inception (Fig. B.1c)		$1280 \times 8 \times 8$
2×Inception (Fig. B.1b)		$2048 \times 8 \times 8$
Max Pooling		$2048 \times 1 \times 1$
Global Average Pooling		$1000 \times 1 \times 1$
Softmax		1000

Table B.2: Inception-v3 network. Taken from [2]

Appendix C

C.1 Kimia Path24 Dataset

C.1.1 Accuracy calculation code

```
import numpy as np
import h5py as h5
import sys
from sklearn.metrics import accuracy_score, confusion_matrix

data_file_path = sys.argv[1]
csv_file_path = sys.argv[2]

data = h5.File(data_file_path, 'r')
correct_labels = data['/test_data/targets']
predicted_labels = np.genfromtxt(csv_file_path)

np = accuracy_score(correct_labels, predicted_labels)*100.
cnf_mat = confusion_matrix(correct_labels, predicted_labels)
e = cnf_mat.diagonal()
nw = (e/cnf_mat.sum(axis=0)).mean()*100.
n_total = (nw*np)/100.

print n_total
```

Glossary

Blame game A situation in which one party blames others for something bad or unfortunate rather than attempting to seek a solution 1

Biopsy An examination of tissue removed from a living body to discover the presence, cause, or extent of a disease 2, 10

Light microscopy A type of microscopy that uses visible light and a system of lenses to magnify images of small subjects 3, 11, 13, 14

Histodiagnosis A diagnosis made from examination of the tissues, esp. by use of microscopy 4, 6, 19

Hyperparameters In machine learning, a hyperparameter is a parameter whose value is set before the learning process begins 7, 33, 34, 36, 62, 65, 67, 75

Radiology Radiology is the science that uses medical imaging to diagnose and sometimes also treat diseases within the body. For e.g. X-ray, ultrasound 10, 16