AN ABSTRACT OF THE PROJECT REPORT OF

Thi Kim Phung Lai for the degree of Master of Science in Computer Science presented on April 23, 2018.

Title: Multiple Instance Learning for Histopathological Image Classification

Abstract approved: ____________________________
Raviv Raich

In histopathological image analysis, image classification as well as pattern detection play a crucial role in the diagnosis and treatment process since the goal is to not only differentiate cancer types but also identify cancerous manifestations. Fully supervised learning strategies tend to address these problems using manually annotated cancerous regions and labeled cancer-type images. The success of these approaches heavily depends on manual segmentation from pathologists/experts. However, the manual process is challenging due to two major issues of histopathological images: (i) manual segmentation process over the entire image is time-consuming and labor-intensive and (ii) boundaries of different cancerous regions in the image are naturally ambiguous, which may create inter- and intra-observation variations among experts. Therefore, weakly supervised learning approaches solely based on the label of images are well-suited for the data. Multiple instance learning (MIL), one of the weakly supervised learning methods, is recently considered as a machine learning paradigm to analyze histopathological images. Based on image labels/cancer types, MIL approaches learn to predict a cancer type as well as detect and localize cancerous regions in the image. In this report, existing strategies for modeling histopathological image analysis as MIL problems are reviewed. Recent trends and future directions are also discussed.
Multiple Instance Learning for Histopathological Image Classification

by

Thi Kim Phung Lai

A PROJECT REPORT

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented April 23, 2018
Commencement June 2018
Master of Science project report of Thi Kim Phung Lai presented on April 23, 2018.

APPROVED:

__________________________________________
Major Professor, representing Computer Science

__________________________________________
Director of the School of Electrical Engineering and Computer Science

__________________________________________
Dean of the Graduate School

I understand that my project report will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my project report to any reader upon request.

__________________________________________
Thi Kim Phung Lai, Author
ACKNOWLEDGEMENTS

I would like to acknowledge my major advisor Prof. Raviv Raich for his kindness guidance and great support during the time I have worked with him. His motivation and enthusiasm helped me to work through this project.

Additionally, I would like to thank the other committee members: Dr. Kim Jinsub, Dr. Xiao Fu, and Dr. Molly Megraw for their attendance and advice in my defense. I would also like to thank Dr. Xiaoli Fern and Dr. Sinisa Todorovic for their advice.

Moreover, I would also like to thank Dr. Arvind Rao, The University of Texas MD Anderson Cancer Center for providing the dataset used in this project.

Last but not least, I would like to have a special thank to my family for their wholehearted support to help me go through all these difficult periods.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1</strong></td>
<td></td>
</tr>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Medical imaging modalities</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Importance and challenges of histopathological image analysis</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Multiple instance learning suitability</td>
<td>4</td>
</tr>
<tr>
<td>1.4 Report structure</td>
<td>6</td>
</tr>
<tr>
<td><strong>2</strong></td>
<td></td>
</tr>
<tr>
<td>Classification of histopathological images</td>
<td>7</td>
</tr>
<tr>
<td>2.1 A general classification framework</td>
<td>7</td>
</tr>
<tr>
<td>2.2 Preprocessing</td>
<td>8</td>
</tr>
<tr>
<td>2.2.1 Color Normalization</td>
<td>9</td>
</tr>
<tr>
<td>2.2.2 Patch division</td>
<td>13</td>
</tr>
<tr>
<td>2.2.3 Irrelevance reduction</td>
<td>13</td>
</tr>
<tr>
<td>2.3 Feature engineering</td>
<td>13</td>
</tr>
<tr>
<td>2.3.1 Morphological features</td>
<td>14</td>
</tr>
<tr>
<td>2.3.2 Textural features</td>
<td>14</td>
</tr>
<tr>
<td>2.3.3 Interest point descriptor</td>
<td>16</td>
</tr>
<tr>
<td>2.3.4 Intensity-based features</td>
<td>18</td>
</tr>
<tr>
<td>2.4 Classification</td>
<td>18</td>
</tr>
<tr>
<td>2.4.1 Fully supervised learning algorithms</td>
<td>18</td>
</tr>
<tr>
<td>2.4.2 Weakly supervised learning algorithms</td>
<td>19</td>
</tr>
<tr>
<td>2.5 Model evaluation</td>
<td>20</td>
</tr>
<tr>
<td>2.5.1 Validation techniques</td>
<td>21</td>
</tr>
<tr>
<td>2.5.2 Evaluation methods</td>
<td>23</td>
</tr>
<tr>
<td><strong>3</strong></td>
<td></td>
</tr>
<tr>
<td>Histopathological image classification</td>
<td>25</td>
</tr>
<tr>
<td>3.1 Fully supervised learning approaches</td>
<td>25</td>
</tr>
<tr>
<td>3.1.1 Logistic regression single-instance single-label (LR-SISL)</td>
<td>25</td>
</tr>
<tr>
<td>3.1.2 Discriminative feature-oriented dictionary learning (DFDL)</td>
<td>26</td>
</tr>
<tr>
<td>3.1.3 Bag-level label prediction for fully supervised learning</td>
<td>29</td>
</tr>
<tr>
<td>3.2 Weakly supervised learning - Multiple instance learning</td>
<td>30</td>
</tr>
<tr>
<td>3.2.1 Multiple instance boosting (MIL-Boost)</td>
<td>31</td>
</tr>
<tr>
<td>3.2.2 Multiple clustered instance learning (MCIL)</td>
<td>32</td>
</tr>
<tr>
<td>3.2.3 Multi-instance multi-label learning in the presence of novel class instances (MIML-NC)</td>
<td>35</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
</tr>
<tr>
<td>4</td>
<td>37</td>
</tr>
<tr>
<td>4.1</td>
<td></td>
</tr>
<tr>
<td>4.1.1</td>
<td>38</td>
</tr>
<tr>
<td>4.1.2</td>
<td>38</td>
</tr>
<tr>
<td>4.2</td>
<td>38</td>
</tr>
<tr>
<td>4.2.1</td>
<td>38</td>
</tr>
<tr>
<td>4.2.2</td>
<td>38</td>
</tr>
<tr>
<td>4.3</td>
<td>39</td>
</tr>
<tr>
<td>4.3.1</td>
<td>40</td>
</tr>
<tr>
<td>4.4</td>
<td>42</td>
</tr>
<tr>
<td>5</td>
<td>45</td>
</tr>
<tr>
<td>Bibliography</td>
<td>45</td>
</tr>
<tr>
<td>Appendices</td>
<td>50</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Examples for different imaging modalities [2].</td>
<td>3</td>
</tr>
<tr>
<td>1.2</td>
<td>Example of data requirements for fully supervised learning (left) and weakly supervised learning (right) approaches.</td>
<td>5</td>
</tr>
<tr>
<td>2.1</td>
<td>A general classification framework.</td>
<td>7</td>
</tr>
<tr>
<td>2.2</td>
<td>Visual flowchart for a histopathological image classification system.</td>
<td>8</td>
</tr>
<tr>
<td>2.3</td>
<td>Color Normalization Comparison [1].</td>
<td>10</td>
</tr>
<tr>
<td>2.4</td>
<td>Two main stages of the SFTA algorithm. Left image is for decomposition process and right image is for SFTA feature extraction [13].</td>
<td>15</td>
</tr>
<tr>
<td>2.5</td>
<td>HOG feature extraction of image of the digit 2 with different CellSize [4].</td>
<td>17</td>
</tr>
<tr>
<td>2.6</td>
<td>Family of cross validation methods: a) Random sub-sampling, b) K-fold cross validation, and c) Leave-one-out cross validation.</td>
<td>22</td>
</tr>
<tr>
<td>3.1</td>
<td>DFDL detection procedure (taken from [42]).</td>
<td>29</td>
</tr>
<tr>
<td>4.1</td>
<td>Visualization example 1. Red areas indicate cancerous cells. Green ones are normal cells.</td>
<td>43</td>
</tr>
<tr>
<td>4.2</td>
<td>Visualization example 2. Red areas indicate cancerous cells. Green ones are normal cells.</td>
<td>44</td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Bag-level performances on breast, kidney, lung, and spleen cancer datasets with different frameworks on SISL and MIL approaches.</td>
<td>40</td>
</tr>
<tr>
<td>4.2 Complexity analysis for different methods.</td>
<td>42</td>
</tr>
</tbody>
</table>
## LIST OF ALGORITHMS

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1  Discriminative Feature-oriented Dictionary Learning.</td>
<td>28</td>
</tr>
<tr>
<td>2  Multiple clustered instance learning.</td>
<td>34</td>
</tr>
</tbody>
</table>
Chapter 1: Introduction

Medical imaging is commonly used in medical diagnosis and treatment. Different imaging modalities can be used to evaluate parts of human anatomy. Examining medical image data is a two-fold challenge. It is difficult to diagnose a specific disease. It is also difficult to analyze the images. Image classification plays a key role in computer-aid diagnosis. This is a complex task due to the use of different methods and strategies in investigating image modality results, image preprocessing results, feature extraction/engineering/selection results, and classification method results. There are two main goals of medical image classification approaches: (i) image-level diagnosis determining the presence/absence of medical conditions/diseases (cancer type, disease type, or survival-rate type) and (ii) localization and recognition of relevant regions in the image associated with cancer/disease manifestations. The histopathology-based approach is one of the most widely used medical imaging today. The importance and challenges of histopathological image analysis in medicine will be discussed in this introduction.

1.1 Medical imaging modalities

Medical imaging is growing at a rapid pace. Many imaging modalities, namely X-ray based imaging, ultrasound imaging, magnetic resonance imaging, cytology, and histopathological imaging have already matured.

There are several X-ray based methods including conventional X-ray (radiography) and computed tomography (CT) [7, 8]. In X-ray, particles pass through the body and parts of the energy of the X-ray beam are observed. Then, a detector or a film captures the attenuated X-rays creating a clinical image. Different organs have a different sensitivity and absorption to radiation. In radiography, while bones and tissues dense in calcium absorb X-rays well, soft tissue like muscle fiber absorbs fewer X-rays. Therefore, radiography is suitable for scans of bone fracture images and dental images. In the CT process, the scanner rotates, enabling the creation of views from different angles [5]. Therefore, CT scans provide far more detailed images than radiography, especially in the case of blood vessels and soft tissue (i.e., organs and muscles). However, these ionizing radiation methods may cause a harmful effect on humans such as a high incidence of
developing cancer [10]. Even though today doctors and manufactures are well aware of the risks and control the radiation dose to minimize the risks, X-ray based methods are not recommended for pregnant women, young children, and unborn babies (unless the examination is absolutely necessary) [5].

Ultrasound imaging techniques use high-frequency sound waves to produce images of the interior of the body [5]. The sound waves are transmitted to the examined area and the returning sound waves or echoes are processed via a computer and then converted to ultrasound images. Ultrasound is particularly suited for studying the functions of moving structures in real-time such as abdomen, heart and blood vessels, and breasts. Especially, due to its non-ionizing nature, it is a good choice in the case of concerning radiation sensitivity, i.e., pediatrics and prenatal diagnostics [5].

Magnetic resonance imaging (MRI) uses a powerful magnetic field and radiofrequency pulses to produce detailed images of the internal structures of the body [3]. MRI is particularly well-suited for a wide range of non-bone parts or soft tissue of the body such as knee and shoulder injuries, brain, nerves, and muscles. Although MRI does not emit damaging ionizing radiation, it employs a strong magnetic field that may affect humans, especially for people with implants and women who are pregnant. In an MRI scan, loud noise and expensive costs are also taken into consideration [3].

Different from these aforementioned methods, cytology and histopathology imaging are the microscopic examination of cell and tissue samples. In both modalities, cell or tissue samples are removed from the body in order to study the manifestations of disease/cancer. Cytology is applicable for rapid tests to establish a working diagnosis. Cytology images are taken at a cell level and cannot show whether disease invades surrounding tissues. On the other hand, histopathological images are typically taken at a much lower-magnification level than that of cytology. This low-magnification allows for analysis at the tissue level that is well-suited for classification of cancer images or identification of the boundary between tissue types. Cytology and histopathology are invasive methods that require an exploratory surgery. However, in comparing to X-ray based or MRI methods, images in cytology and histopathology are generally taken at a much higher-resolution. Such images provide an access to cell and tissue structures to improve cancer diagnosis accuracy. An example of the aforementioned imaging modalities (radiography, CT, MRI, ultrasound, cytology, and histopathology) is shown in Figure 1.1.
1.2 Importance and challenges of histopathological image analysis

Histopathological imaging is a major discovery in modern medicine [12, 18, 20]. In histopathology, a number of cells or tissues is taken from a body. The obtained samples are typically dyed with hematoxylin and eosin stain (H&E) that highlights the nuclei, cytoplasm, mitochondria, and other organelles. After the staining process, tissue slides are viewed under a high-magnification microscope and analyzed/studied for microscopic examination by pathologists [18, 41]. In comparing histopathology to the other medical imaging modalities discussed above, histopathological images are considered as a “gold standard” in cancer recognition and diagnosis. These images are widely used in many types of cancers, i.e., brain, breast, kidney, lung, and spleen [17, 42, 43].
To analyze histopathological images, there are numerous challenges in dealing with the data. The first challenge is variations in color among different tissue images. The color variation can arise from the staining and dying process. Different staining materials, staining time, staining light or folding tissues can lead to varied histopathological images. The second difficulty of working with histopathological images is that they are typically large (30000 × 30000 × 3 pixels and occupying 1GB of memory) [20]. This may be time-consuming as well as memory-consuming when working with these types of images. In contrast to the large size of images, cancer manifestations may appear in a minute portion of the image at a small nucleus level. Therefore, a standard supervised learning classification approach which learns and classifies cancerous manifestations at the large image level, may remove the effects of these minute regions on training and testing the classifier. Additionally, another challenge is the complexity of histopathological images. The complexity is attributed to several factors, e.g., cell boundaries are overlapped, parts of tissues are folded, and cell boundaries are blurred. These reasons make the cell region extraction using traditional image segmentations difficult. Another important challenge comes from the complex patterns in tissue images, resulting in the difficulty of obtaining reliable computer-assisted diagnosis systems. Manual analysis of histopathological images is still the primary diagnosis method, but this method has major disadvantages. This process is time-consuming and heavily depends on the experts and experience of the pathologists. In addition, due to the complex and diverse characteristics of histopathological images, there are empirically substantial inter-observation and intra-observation variations among experts. Consequently, a computer-assisted diagnosis system for histopathological images that does not rely on manual extracted regions of interest is a necessary approach that can overcome the above-mentioned challenges.

1.3 Multiple instance learning suitability

Fully supervised learning approaches are based on manual segmentation and available labels from experts in the training process. An object can be described by a feature vector or a set of feature vectors in which all feature vectors are manually labeled. The labels are generated from experts and then used to train classifiers. However, this may be inapplicable in many problems, especially in histopathological image analysis. The success of the supervised learning algorithms heavily depends on the availability and reliability of a large amount of manual labeling data. The manual annotation process is nevertheless often time-consuming and labor-intensive. Addition-
ally, with a large image generated from an entire tissue, medical experts/ pathologists usually do not segment all target patterns or mask all cancerous manifestations in the image. They tend to provide some focal areas and force less important areas in the image into non-influence on classification. This may create information distortion and mislead the learning process. Moreover, patterns in histopathological images are often ambiguous contours while experts tend to force a crisp decision by masking target areas. Therefore, alternative approaches that reduce dependence on manual labels are necessary.

Unsupervised learning approaches where no manually annotated process is needed can solve the concerning manual issues of supervised learning frameworks. However, in medical imaging analysis where human knowledge is necessary, unsupervised learning strategies are not clinically practical directions [21]. Additionally, unsupervised learning frameworks describe hidden structures of unlabeled images based on similarity and dissimilarity among data samples. Therefore, it is well-suited for clustering, anomaly detection, and dimensionality reduction tasks rather than classification problems [24].

Figure 1.2: Example of data requirements for fully supervised learning (left) and weakly supervised learning (right) approaches.

In the midst of supervised and unsupervised learning strategies, weakly supervised learning approaches which only requires image-level labels, is considered. They offer usage for a
good balance of containing a moderate amount of labels by experts while being able to explore pixel-level/patch-level classification [44, 45]. Figure 1.2 shows the differences between data requirements for fully supervised learning and weakly supervised learning approaches. Fully supervised learning frameworks require manual annotation of cancerous areas as well as the type of cancer while weakly supervised learning approaches only require a label of cancer type. In comparing weakly supervised learning approaches to fully supervised learning or unsupervised learning methods, it is described by a set of feature vectors in which only the label of the set is provided. Multiple instance learning (MIL), one of the weakly supervised learning paradigms, is well-suited for histopathological image analysis. MIL approaches are based on a collection of training histopathological images labeled by experts/pathologists to classify types of data as well as automatically detect and localize cancerous regions in the image. There is no manual segmentation or manual instance labeling (pixel-level) from experts. Moreover, MIL methods are based merely on image-level labels that is applicable in histopathological images where experts or doctors usually provide an image with its type of cancer or cancer grading only. Additionally, medical manifestations may appear in a minute portion of the image. Due to the limited presence of the manifestations, its effects on the entire image may be ignored. As a result, dividing the whole slide histopathological image into small patches and considering the influence of these patches on the image label is suitable. Therefore, to deal with the histopathological images, we consider using MIL approaches to learn the classifier and localize crucial patterns in the image.

1.4 Report structure

The rest of the report is organized as follows. Section 2 presents a classification framework for histopathological images. It includes the literature review of preprocessing and feature extraction methods and general approaches for classification problems. Section 3 reviews specific histopathological image classification strategies, the core focus of this report. The experimental settings, results, and analysis are provided in Section 4. Future direction is also briefly discussed in Section 4. Finally, summary of the report is presented in Section 5.
Chapter 2: Classification of histopathological images

2.1 A general classification framework

Figure 2.1: A general classification framework.

Histopathological images obtained from the tissue preparation and imaging production technologies [20] are ready for analyzing using computer-assisted diagnosis systems. In general, a classification system for histopathological images consists of three main stages as shown in Figure 2.1. The first stage is pre-processing. In histopathological image analysis, pre-processing steps are needed to reduce color variations and to formulate an image as a multiple instance
learning problem. This stage consists of color normalization, patch division, and irrelevance reduction. The second stage is feature engineering. The feature engineering process can be divided into several categories based on histopathological image characteristics, namely morphological-based, textural-based, fractal-based, topological-based, and intensity-based features. The final stage is classification. There are many classification approaches for histopathological images. In this work, we summarize and split them into two classification paradigms: (i) fully supervised learning approaches where types of cancers (image-level labels) and cancerous areas (instance-level labels) are available and (ii) weakly supervised learning models and multiple instance learning approaches in particular where only bag-level labels are provided. While many of the preprocessing and feature engineering methods are fairly mature in histopathological image analysis, classification frameworks remain an open area. Therefore, in this report we will briefly introduce several preprocessing and feature engineering methods while concentrating on classification approaches. A visual flowchart of the histopathological image classification system is shown in Figure 2.2.

![Figure 2.2: Visual flowchart for a histopathological image classification system.](image)

### 2.2 Preprocessing

To overcome the aforementioned challenges of histopathological image analysis, three major pre-processing steps are used. They are: (i) color normalization to eliminate color variations
among different images generated from varying materials, time, and light during the staining process, (ii) patch division to employ the effectiveness of patch-based approaches in histopathology and to formulate a multiple instance learning problem from the image, and (iii) irrelevance reduction to remove irrelevant information and to ease the burden of computation. Different approaches for each of these steps are discussed as follows.

2.2.1 Color Normalization

In histopathological images, color variations typically arise from staining histopathological tissues. Diversity in tissue preparation, staining materials, staining periods, and scanner technologies can cause noisy color distribution in histopathological images. Therefore, to reduce color variations among different images, several common color normalization methods are considered in this report: histogram specification [6], Reinhard et al. [32], and Macenko et al. [26]. To be consistent among these methods, the input image which needs to normalize the color is called source image. The reference image which is used to compare with the color distribution of the source image is named target image. The result image after matching into the color distribution of the target image is called output image.

2.2.1.1 Histogram specification

Histogram specification is a transformation of an image color space so that its histogram matches a specified histogram. It transforms the red, green and blue histograms of the image to match the three histograms of the specific image. Let \( x \) be the source image, \( y \) be the output image, and \( z \) be the target image. The objective is to find a transformation \( y = T(x) \) in which both source and output images are matched into the histogram of the target image. The mapping function can be found in three steps:

- Find histogram of the source image \( h_x \) and the histogram equalization mapping function \( z = T_1(x) \).

- Specify the output histogram \( h_y \) and the histogram equalization mapping function \( z' = T_2(y) \).

- Solve the two equalization problems \( z = T_1(x) \) and \( z' = T_2(y) \) and find the relation of these transformations to obtain the final transformation \( y = T(x) \). As the two intermediate
images \( z \) and \( z' \) have the same equalized histogram, they are actually the same image, or \( z = z' \). Therefore the overall transformation from the source image \( x \) to the output image \( y \) is: 

\[
y = T_2^{-1}(z') = T_2^{-1}(z) = T_2^{-1}(T_1(x)).
\]

This method is simple and enhances contrasts of the input image. However, a major disadvantage of histogram-based approaches as claimed in [22] is that they follow the assumption that the
proportion of pixels of each stain type is similar to the output and source images. This is not always correct and causes color mismatch due to different proportions of stains in these images.

2.2.1.2 Reinhard at el. method

In this method, the target image and the source image are given. The goal is to correct the color of the source image by applying characteristics of the target image to the source image. The idea behind the concept of the Reinhard method [32] is in the RGB color space where values of different channels are correlated. For example, most pixels will have large values of the red and green channels when the blue channel is large. This means that if we want to change the appearance of a pixel, we have to modify all color channels. This causes complication in modifying the color of the image. Therefore, the Reinhard method decorrelates color space of the image by converting RGB color space to \( l\alpha\beta \) color space [34] and processes color normalization in that space before re-converting to the RGB color space. There is a little correlation among the axes of \( l\alpha\beta \) color space that reduces the correlation among each of the color channels in that space. To match the color distribution of source images to that of a target image, there are three main steps:

- Covert RGB images to the perception-based \( l\alpha\beta \) color space by a linear transformation for all pixels.
- Normalize color in the \( l\alpha\beta \) color space as:

\[
\begin{align*}
    l_{\text{output}} &= \frac{l_{\text{input}} - \bar{l}_{\text{input}}}{\hat{l}_{\text{input}}} \hat{l}_{\text{target}} + \bar{l}_{\text{target}} \\
    \alpha_{\text{output}} &= \frac{\alpha_{\text{input}} - \bar{\alpha}_{\text{input}}}{\hat{\alpha}_{\text{input}}} \hat{\alpha}_{\text{target}} + \bar{\alpha}_{\text{target}} \\
    \beta_{\text{output}} &= \frac{\beta_{\text{input}} - \bar{\beta}_{\text{input}}}{\hat{\beta}_{\text{input}}} \hat{\beta}_{\text{target}} + \bar{\beta}_{\text{target}}
\end{align*}
\]

where \( \bar{l}, \bar{\alpha}, \bar{\beta} \) are the channel means and \( \hat{l}, \hat{\alpha}, \hat{\beta} \) are the channel standard deviations. These values are calculated over all pixels of the image.

- Revert the result back to RGB color space.

The usage of \( l\alpha\beta \) color space reduces the correlation among values of color channels. This
method is simple and easy to embed in various commercial graphics packages [32]. However, the assumption of a unimodal distribution of pixels in each channel of \(l\alpha/\beta\) color space may be compromised when using multiple colored stains. This may result in background areas being mapped as colored regions.

2.2.1.3 Macenko at el. method

In medical imaging techniques, the most popular staining method is using hematoxylin and eosin stains (H&E). Hematoxylin stains nucleic acids in a blue-purple color while eosin stains proteins in a bright pink color [35]. The color normalization approach of Macenko et al. [26] is based on the staining appearance normalization. Specially, it determines the concentrations of the individual stains for each pixel, normalizes the staining concentrations and then mixes the stains with common characteristic absorption coefficients to obtain a standardized image.

It is assumed that there is a stain vector for each of two stains in the image, and the resulting color in the optical density space (OD space) of each pixel is a linear combination of the stain vectors. Every value must exist between the two stain vectors. To obtain the specific stain vector for each stain in the image, the source image is firstly converted from the RGB color space to the OD space by the following equation:

\[
OD = -\log_{10}(I),
\]

where \(I\) is the RGB color vector with each component is normalized to \([0, 1]\). This transformation is to produce a space where a linear combination of stains results in a linear combination of values in OD space. The pixels which have low values in OD are removed (low or zero values in OD). Then, a plane that the stain vectors form (stain matrices) is calculated based on two vectors corresponding to the two largest singular values of the SVD decomposition of the OD transformed pixels. All of the OD pixels are projected onto this plane and normalized to unit length. The angle of each point with respect to the first SVD direction is calculated. This is to map the direction of each point in the plane to a scalar. For each stain, the histogram of these angles or the intensity of histograms for all pixels are computed. Then, the \(99^{th}\) percentile of these intensity values is used as a robust approximation of the maximum to map input image values to match the target image. As argued in [22], the method can fail if the stain matrix estimation fails or the mapping function is inappropriate due to color variations.
2.2.2 Patch division

After color normalization, images are divided into small patches. Later features from each of these patches are extracted as a feature vector. In histopathology, an image is typically large, hence it is computationally inefficient and spatially inefficient if working with the entire image. Also, in comparing medical manifestations to the large image, they naturally appear at the nucleus level which can be difficult to obtain information from the entire image. Therefore, the patch division process is well-suited for histopathological images. It is also to formulate a multiple instance learning problem in which an image is divided into a set of patches or it is considered as a bag (image) of instances (patches). A size of patches is picked based on the input from pathologists or chosen by cross validation. If the size is small, the patch may not capture sufficient information related to neighborhood structure of cells or invasive areas. If the size is large, the medical manifestations may occupy at small portion of the patch.

2.2.3 Irrelevance reduction

After the staining process, histopathological slides are put onto a glass microscope slide to view under the microscope. To generate a histopathological image, beside the region of interest, there is a big area of glass around that region. The glass is obviously uninformative for the classification purpose. Therefore, removing patches of glass is necessary to eliminate clearly irrelevant information as well as partly to ease the burden of computation. To determine glass patches, patches from all images are collected and then clustered by using a K-means algorithm [27]. The clusters of glass are manually selected and patches belong to these glass clusters are removed from a patch construction process.

2.3 Feature engineering

Feature engineering is the process of using domain knowledge of the data to generate feature vectors. It is an important step for understanding data and providing information for further classification and the analysis process. Based on histopathological characteristics, there are numerous feature engineering techniques which can be categorized into: morphological features (provide information about size and shape of cells, tissues, e.g., radius, area symmetry of cells) [37,39], textural features (provide information on the variation in the intensity of a surface, e.g., smoothness, coarseness, and regularity) [19, 38], and interest point descriptor (e.g., histogram
of oriented gradients)

2.3.1 Morphological features

Morphological features provide information about the size and shape of a cell. The size is presented as area, perimeter, and radius of the cell. The shape is expressed by orientation, compactness, roundness, and smoothness. Suppose there are sets of the boundary points of segmented cells/nuclei. In image processing, these morphological features are defined for each cell as follows:

- Area: the number of pixels within the boundary of the cell.
- Perimeter: the sum of the distances between each adjoining pair of pixels around the boundary of the cell.
- Radius: the average length of the radial lines towards all points lying onto the boundary.
- Orientation: the angle between the horizontal axis and the major axis of the ellipse that fairly covers the cell.
- Compactness: the ratio between the square of the perimeter with the area.
- Roundness: compactness divides by $4\pi$.
- Smoothness: the sum of the difference between the radial line towards a boundary point and the average length of the radial lines surrounding the boundary point.

Using the domain knowledge, these quantification properties are different from the malignant cells and the normal or benign cells in histopathology. Therefore, they can be used to differentiate cancerous and normal cell appearances.

2.3.2 Textural features

Textural features are to capture the repetitive patterns of regions in an image. It contains information about spatial distribution or intensity variation within a region or related area of the image. Texture can be evaluated as being fine, coarse, and smooth. At low-magnification histopathological images, textural features are commonly used to capture tissue architecture, such as cell
organization. At medium resolution, architectural organization of nuclei can be expressed by graph-based algorithms. At high-magnification, the architecture of an individual cell can be described. It can be the periphery and center of a nuclei.

![Input Image](image)

Two main stages of the SFTA algorithm. Left image is for decomposition process and right image is for SFTA feature extraction [13].

Fractal-based features are considered as a common type of textural features, especially in extracting features for complex objects. Fractal-based features are used to estimate and quantify the complexity of the shape or texture of objects. The features are obtained through a binary decomposition method [36] and a box counting algorithm [11]. Typically an object dimension is given by an integer for example 1D (a line), 2D (a square), and 3D (a cube). However, some natural objects are fractals, e.g., tissue boundary inside a kidney or a brain. In actuality, fractal dimensions help in characterizing the boundary within the kidney/brain.

One of the most popular methods for fractal-based feature extraction is segmentation-based fractal texture analysis (SFTA) [13]. The SFTA algorithm can be divided into two main parts. First, a gray-scale input image is decomposed into a set of binary images. Two-threshold binary
decomposition (TTBD) is applied to decompose the image. Second, the resulting binary images are used to compute the fractal dimension from its boundary regions. The fractal dimension features are computed from each border image using a box counting algorithm. The mean gray level and size (pixel counting) of the regions are also computed as parts of the SFTA features. The SFTA process is shown in Figure 2.4. In TTBD, the set of binary images is obtained by selecting pairs of thresholds from a predefined set of threshold values and applying a two-threshold segmentation as follows:

\[
I_b(x, y) = \begin{cases} 
1 & \text{if } t_l < I(x, y) \leq t_u \\
0 & \text{otherwise},
\end{cases}
\]

where \(I_b(x, y)\) and \(I(x, y)\) are the input binary and gray scale images, respectively. Note that \(t_l\) and \(t_u\) are lower and upper threshold values, respectively. The fractal-based features are exploited to describe the boundary complexity of objects. The boundary regions of a binary image are computed as follows:

\[
\Delta = \begin{cases} 
1 & \text{if } \exists(x', y') \in N_8[(x, y)] : \{I_b(x', y') = 0\} \land \{I_b(x', y') = 1\} \\
0 & \text{otherwise},
\end{cases}
\]

where \(N_8[(x, y)]\) is the set of pixels that are 8-connected to the pixel at position \((x, y)\). The fractal-based features are computed from each border image using a box counting algorithm. In the box counting algorithm, the image is divided into a grid of squares of size \(\epsilon \times \epsilon\). Then we count the number of the squares \(N(\epsilon)\) that contain at least one pixel of the object. By varying the value of \(\epsilon\), we create a curve of \(\log N(\epsilon) vs \log \epsilon^{-1}\). This curve is eventually approximated by a line. The fractal dimension corresponds to the slope of the line.

### 2.3.3 Interest point descriptor

In histopathology images, cancerous manifestations may appear in a small part of the image. Features from local object appearance are extracted by the distribution of local intensity gradients or edge directions. Therefore, a histogram of oriented gradients (HOG) [29] that counts occurrences of gradient orientation in localized portions of the image is applied. There are several major steps to extract HOG features:

- Divide the image into small regions (cells) and compute a histogram of gradient directions
within each cell.

- For each cell, we count the number of direction that falls within each angular bin, e.g., 8 bins for 8 directions. Pixels in each cell contribute the weighted gradient to the angular bin of the cell.

In each cell, describe the gradient orientation into angular bins. Pixels in each cell contributes the weighted gradient to its angular bin.

- Group adjacent cells as spatial regions (blocks). The block is the basis for grouping and normalization of histograms.

- Normalized group of histograms represents the block histogram. The set of these block histograms represents the descriptor.

Figure 2.5 shows HOG feature extraction with different sizes of cells. It can be seen from the figure that based on cell size, we can achieve different feature lengths as well as different values of HOG features.

![Figure 2.5: HOG feature extraction of image of the digit 2 with different CellSize [4].](image)

Figure 2.5: HOG feature extraction of image of the digit 2 with different CellSize [4].
2.3.4 Intensity-based features

The intensity-based features are extracted from the gray-level or color histogram of the image. These features are defined from the intensity histogram of pixels in a cell. The optical density (OD) of a pixel from its gray-level values is one of the intensity-based features. For color images, the intensity-based features can be pixel values of a single color channel or the relationship between the color values of different color channels.

2.4 Classification

Based on the ability to learn classifiers from fully supervised data where all label information is provided (i.e., both instance- and bag-level labels are available) or weakly supervised data where a portion of label is provided (i.e., only bag-level labels are available), we categorize classification algorithms into two groups:

- **Fully supervised learning algorithms**: The instance classifiers are learned from fully annotated instances while its bag classifiers are assigned from the instance prediction based on some certain functions. These algorithms are ideal with instance-level prediction, but may also be used for bag-level prediction.

- **Weakly supervised learning algorithms**: In these approaches, classifiers are trained based on bag-level labels only. The instance classifiers are weakly learned during the optimization of the bag-level classifiers. These algorithms are typically optimized for bag-level prediction, but may or may not be used for instance-level prediction.

2.4.1 Fully supervised learning algorithms

The fully supervised learning algorithms can be viewed as approaches that separately learn the concept of positive and negative instances and try to detect/recognize all positive instances. In this setting, instance-level labels are available. In other words, in histopathological images, these images have to be marked where the cancerous regions are in the image. Then the total or part of the marked areas are used to train the classifiers. The manual labeling process has to be done by experienced experts or pathologists to achieve the most reliable labels. Once the instance-level classifier is trained, bags can be classified easily by applying the presence-based assumption of the MIL approach: bags are labeled positive if they contain at least one positive instance [15].
In practice, the instance-level classifier cannot be perfectly trained to truly predict all instances. As a result, if a negative instance in a negative bag is wrongly predicted as positive, the bag is wrongly predicted as positive. To reduce the risk, the presence-based assumption can be relaxed to: bags are labeled positive if they contain a sufficient percentage of positive instances in the bag.

In this report, two fully supervised learning algorithms are presented in detail in Chapter 3 and their experimental evaluations are shown in Chapter 4. The first algorithm is obtained by applying a logistic regression (LR) to a single-instance single-label (SISL) version of the data, in which each instance label is available in training. We denote this benchmark by LR-SISL. Different regularization methods and different bag-level prediction criteria are also applied to LR-SISL to create different versions of LR-SISL. The second algorithm is discriminative feature-oriented dictionary learning (DFDL) [42]. DFDL learns dictionary bases on the raw image using manually extracted regions (like annotating at instance-level) and then in the testing phase, unseen instances are classified using the trained bases. The bag-level label is predicted based on the presence or absence of positive instances.

These fully supervised learning classifiers are trained based on fully supervised learning in which all data samples are labeled. Therefore, they are expected to act as an upper bound on performance due to the advantage of the availability of instance-level labels in training.

2.4.2 Weakly supervised learning algorithms

In weakly supervised learning algorithms, the classifier is trained solely based on the bag-level label. There are no instance-level labels available in training. MIL approaches typically rely on the presence-based assumption [15] in which a bag is predicted as positive if it includes at least one positive instance and a negative bag consists of all negative instances. To follow the assumption, MIL algorithms focus on finding and evaluating the most positive or the least negative instance in a bag. The bag-level classifier is trained in the concept of: the bag-level prediction function is generally associated with an instance-level prediction function by the maximum function or the softmax function. In this learning paradigm, only bag-level labels are required. In histopathological image classification, these methods classify a data type based on the label of the image such as types of cancers, cancer and non-cancer, and survival rate (low or high rate). No cancerous manifestations or disease areas are provided in training. The labels are provided by experienced experts or pathologists. In comparing the weakly supervised learning strategies
to fully supervised learning strategies, bag-level approaches are more convenient and efficient. This is due to the fact that there are a few available histopathological image datasets which consist of manual segmentation/labeling at the instance-level. An example which demonstrates the differences from data requirements between fully supervised and weakly supervised learning algorithms is shown in Figure 1.2. In fully supervised learning approaches, manual segmentation and localizing cancerous/disease areas must be provided. The type of data is also needed. On the other hand, in bag-level learning algorithms, only type of data is required.

In this report, several MIL algorithms are considered. They include (i) DFDL, (ii) multiple instance boosting (MIL-Boost) [46], (iii) multiple clustered instance learning (MCIL) [44], and (iv) multi-instance multi-label learning in the presence of novel class instances (MIML-NC). DFDL is considered as a fully supervised learning algorithm if dictionary bases are learned from randomly picking patches in a bag instead of picking patches from manual marking areas. MIL-Boost generalizes boosting to the MIL frameworks. In MIL-Boost, a bag-level label classifier is built from a linear combination of instance-level classifiers. MCIL adapts the clustering concept to MIL-Boost. In MCIL, positive instances are clustered into different groups. This aspect enables the differentiation of each of several cancer types from non-cancer samples. MIL-Boost can be viewed as a special case of the MCIL where there is one cluster considered. MIML-NC is a multi-instance multi-label framework for learning in the presence of label ambiguity. This can be simplified as a multiple instance learning by considering a bag associated with one label.

These fully supervised learning algorithms are presented in detail in the following chapter (Chapter 3). A comparison of their performances and complexity analysis is described in Chapter 5.

2.5 Model evaluation

In a learning process, it is necessary to select/train a model based on training data samples and evaluate the learned model based on testing data samples. The questions of how to select a model, how to validate/estimate a performance, and what evaluation metrics are being used, are taken into consideration. There are different validation techniques, namely holdout and cross validation. The testing performance also considers the differences between the training and testing data samples. Accuracy, area under the curve (AUC), and visualization of the results are commonly used to evaluate a model.
2.5.1 Validation techniques

Generally, classification models have one or more tuning parameters. Therefore, it is necessary to select the model parameters optimally to obtain the best performances. Once we have chosen a model and its optimal parameters, its performance is estimated. Additionally, in medicine and histopathology in particular, data separation between training and testing samples also takes the following information.

- Model selection and performance estimation:

  If there is an unlimited number of data examples, the optimal parameters are straightforwardly chosen from the model that provides the largest performance on the entire dataset where the true performance is the largest performance. However, in real-world domains, especially in medical applications, we only have access to a limited number of data samples. If the parameters are obtained naively, the overfitting may occur as exhibited by a high performance in the training set (100% correct classification) but a low performance in the testing set.

To overcome the issue, there are two methods as follows:

- Holdout method: The idea of the holdout method is to split the dataset into a training set and a testing set. Typically, a model is better when training with more data. Hence, this suggests that we should use as much data as possible in training. However, to achieve a reliable testing performance, it is better to use as much data as possible in testing. Practically, a common separation ratio is 70% of the data for training and the remaining 30% for testing. This method is simple and fast in producing the result. However, since it is a single train-and-test experiment, the holdout method is only accurate in the case of having “perfect” separation between training and testing data. When there is a sparse dataset where it is unable to separate perfectly or an “unfortunate” split happens, the holdout method is inaccurate and unreliable.

- Cross validation: The limitations of the holdout approach can be overcome by a family of cross validation strategies.

  Firstly, random sub-sampling performs K data splits of the dataset. In this method, each portion is randomly selected from a number of samples without replacement.
For each data portion, a classifier is trained from scratch with the training samples and the performance is computed as an average of the K separate experiments.

![Diagram of cross validation methods]

**Figure 2.6: Family of cross validation methods:** a) Random sub-sampling, b) K-fold cross validation, and c) Leave-one-out cross validation.

The second method is K-fold cross validation. The data is split into K-fold partitions. For each of the K experiments, one fold is chosen for testing and the rest is for training. Ideally, all the testing examples of K experiments cover the total number of data samples and they are different from each of the K experiments. As in the random sub-sampling method, the true performance is the average performance over K experiments. Compared to the random sub-sampling approach, the advantage of K-fold cross validation is that all data samples are eventually used for both training and testing. A common choice for K-fold cross validation is $K = 10$.

The third member of the cross validation family is leave-one-out. It is similar to
K-fold cross validation, but K is chosen as the total number of examples. In practice, this method is computationally inefficient due to the large value of K. Three aforementioned cross validation methods are shown in Figure 2.6.

In practice, to select model parameters, the data samples are split into training, validation, and testing sets. Training set is a set of samples used for learning parameters. Validation set is used to validate and tune the parameters. Testing set is only used for assessing the performance of the fully-trained classifier. In the testing phase, there is no further tuning of the model.

- Data separation in the training and testing process:
  In medical datasets, histopathology in particular, to collect the data, several tissues or one tissue from a patient can be viewed in different angles under the microscope. As a result, this process can generate different images from one patient in which these images are naturally similar to each other. If the patient-based information is eliminated or the separation process is image-based, both training and testing samples can include images from the same patient. This results in improved performance compared to patient-based in train and test data separation. However, it may not follow the nature of medical diagnosis and treatment where we typically transfer the learned knowledge from certain patients to unseen patients.

2.5.2 Evaluation methods

There are several ways to examine a framework. We will focus on three common methods in particular: (i) accuracy, (ii) area under the curve, and (iii) visualization.

- Accuracy:
  It measures the proportion of the number of correctly predicted samples over the whole data set. This metric is easy to understand and simple to compute. However, this metric is sensitive to the number of data samples among different classes. In an imbalanced dataset when one class dominates, the classifier obtains a high accuracy even in the case that the classifier wrongly predicts all data samples as the dominated class.

- Area under the curve (AUC):
The probabilistic interpretation of AUC is that AUC equates the probability that a classifier ranks a randomly chosen positive sample higher than a randomly chosen negative sample (rank based on the predicted score). AUC is not sensitive to the varying number of data samples in each class. Therefore, it is typically used for imbalanced label data.

- Visualization:

While accuracy and AUC are used for quantitative analysis, visualization is used for qualitative analysis of a model assessment. It provides an intuition about the obtained result. It can vary with different applications. For example, visualization can be used to display model coefficients, positive or negative examples, and misclassified examples. In histopathological image analysis which uses a multiple instance learning approach, the visualization of positive regions of cancer in the tissue image can be valuable.
Chapter 3: Histopathological image classification

3.1 Fully supervised learning approaches

In fully supervised learning strategies, both instance-level labels and bag-level labels are available. It is equivalent to a single-instance single-label learning setting (SISL) where each instance has its label. In a SISL setting, we are given a set of $M$ instances denoted as $\{x_m, y_m\}_{m=1}^M$. Each instance $x_m$ is associated with a label/class $y_m \in \{0, 1\}$. A notation $w$ is used for the weight of classifiers in SISL algorithms. In general, the goal is to train a classifier that maps an instance to its label.

3.1.1 Logistic regression single-instance single-label (LR-SISL)

LR-SISL method is obtained by applying a logistic regression (LR) to a single-instance single-label (SISL) version of the data, in which each instance label is available in training. This means that marked-cancerous regions in a cancer image are provided in the training process in the method. The regularizer is included in the objective function of the method to control the trade-off between model complexity and data fit. The normalized objective function of the method is as follow:

$$
\min_w f(w) = \frac{1}{M} \sum_{m=1}^M \left[ \log(1 + e^{x_m^T w}) - y_m x_m^T w \right] + R(w).
$$

where

$$
R(w) = \begin{cases} 
\lambda \| w \|_1, & \text{for } L_1\text{-norm}, \\
\frac{\lambda}{2} \| w \|_2^2, & \text{for } L_2\text{-norm}.
\end{cases}
$$

and $\lambda$ is a regularization parameter. This parameter can be fixed or chosen using cross validation.

There are many optimization methods for learning a logistic regression model [30]. For example, gradient descent/ascent, Newton’s method, Quasi-Newton, and conjugate gradient ascent, and iterative scaling methods can be used in logistic regression optimization problems.
The Newton-based methods require to compute the inverse of the Hessian, which is typically not easy to compute. The iterative scaling methods generally converge slowly [25]. We continue with a simple gradient descent optimization method. In the gradient descent approach, the model parameter $w$ is iteratively learned. It is updated at $t$-th iteration as follows:

$$w^{(t+1)} = w^{(t)} - \eta \frac{\partial f(w)}{\partial w},$$

where $\eta$ is the step-size of the updating process. The step-size can be fixed or chosen via line search methods [9]. The gradient of $f(w)$ is computed as follows:

$$\frac{\partial f(w)}{\partial w} = \frac{1}{M} \sum_{m=1}^{M} \left( \frac{e^{x_m^T w}}{1 + e^{x_m^T w}} - y_m \right) x_m + R'(w),$$

where $p_m = \frac{e^{x_m^T w}}{1+e^{x_m^T w}}$ and $R'(w)$ is the gradient of the regularization $R$ w.r.t. $w$ given by:

$$R'(w) = \begin{cases} \lambda \text{sign}(w), & \text{for } L_1\text{-norm}, \\ \lambda w, & \text{for } L_2\text{-norm}. \end{cases}$$

### 3.1.2 Discriminative feature-oriented dictionary learning (DFDL)

Feature extraction is one of the challenging tasks in histopathological image classification. In discriminative feature-oriented dictionary learning (DFDL) [42], the goal is to discover features based on learning class-specific dictionaries. This approach aims to have one dictionary per class such that examples from a given class are accurately approximated over the dictionary of the class and poorly approximated using the dictionary of other classes.

The vectorization of a patch extracted from images is denoted as $x_m \in \mathbb{R}^d$ where $m = 1, 2, \ldots, M$. There are $C$ different classes of data samples and $K$ bases per each dictionary. A collection of data samples from class $c$ where $c = 1, 2, \ldots, C$ is denoted as $X_c \in \mathbb{R}^{d \times M_c}$. A collection of complementary data samples from class $j$ ($j \neq c$) is denoted as $\bar{X}_c \in \mathbb{R}^{d \times \bar{M}_c}$. The dictionary for class $c$ is denoted as $D_c \in \mathbb{R}^{d \times K_c}$. For a vector $s \in \mathbb{R}^K$, $||s||_0$ is the number of its non-zero elements. The sparsity constraint is given by $||s||_0 \leq \ell$. In a matrix form, $||S||_0 \leq \ell$.
means that each column of $S$ has no more than $\ell$ non-zero elements. According to the main idea of DFDL of learning class-specific dictionary such that it can accurately represent samples from class $c$ and poorly represents samples for the other classes, it is necessary to have:

$$\min_{\|s_m\|_0 \leq \ell_c} \|x_m - D_c s_m\|_2^2, \forall m = 1, 2, \ldots, M_c$$

and

$$\min_{\|s_m\|_0 \leq \ell_c} \|\tilde{x}_m - D_c \tilde{s}_m\|_2^2, \forall m = 1, 2, \ldots, \tilde{M}_c$$

where $\ell_c$ controls the sparsity level of the class $c$. For simplicity, the class index in each notation is dropped and the two sets of conditions are simplified in the matrix form. Therefore, the optimization problem for each dictionary in the DFDL framework is presented as follows:

$$D^* = \arg\min_D \left( \frac{1}{M_c} \min_{\|S\|_0 \leq \ell} \|X - DS\|_F^2 - \frac{\rho}{M_c} \min_{\|S\|_0 \leq \ell} \|X - DS\|_F^2 \right)$$

(3.1)

where $\rho$ is a positive regularization parameter. Intuitively, the first term encourages intra-class differences to be small and the second term encourages inter-class differences to be large. The DFDL algorithm for finding dictionary bases is summarized in Algorithm 2. The DFDL-based procedure for histopathological image classification follows the main steps below:

- **Step 1: Training DFDL bases for each class.** From training images that are manually marked the cancerous regions, training patches are selected (that can be considered as labeled instance-level labels). After extracting patches, there are a set of healthy patches and a set of diseased patches for training the dictionary $D$. The training process is shown in Algorithm 2.

- **Step 2: Learning a threshold $\theta$.** The threshold indicates the proportion of healthy patches in an image. Each training patch is classified using the learned DFDL bases as follows:

$$z = \arg\min_{c \in \{1, 2, \ldots, C\}} r_c(x_m),$$

(3.2)

where $z$ is a temporal label of $x_m$, $r_c(x_m) = \|x_m - D_c \delta_c(\hat{S})\|_2$, $\hat{S} = \arg\min_{S} \left( \|x_m - DS\|_2^2 + \gamma \|S\|_1 \right)$, and $\delta_c(\hat{S})$ is a part of $\hat{s}$ associated with class $c$. The proportion of healthy patches in an image $\theta$ is learned from a simple SVM.
• Step 3: Classifying test images. For an unseen test image, the proportion of healthy patches in the image is calculated. The image is classified as healthy (diseased) if its healthy patches proportion is greater (smaller) than the threshold $\theta$.

Algorithm 1 Discriminative Feature-oriented Dictionary Learning.

1: **Input:**
   - $X, \bar{X}$: collection of all in-class samples and complementary samples.
   - $K$: number of bases in each dictionary.
   - $\rho$: the regularization parameter.
2: Choose initial $D_0$ and $\ell$ as follows:
   
   $$(D_0, S_0) = \arg\min_{D,S} \|Y - DS\|_F^2 + \lambda \|S\|_1,$$
   
   $$\ell \approx 1 / M \sum_{i=1}^M \|s_0^i\|_0.$$
3: **while** not converged **do**
4: Fix $D = D_0$, update $S$ and $\bar{S}$ by solving the following equation:
   
   $$\hat{S}^* = \arg\min_{\|S\|_0 \leq L} \|\tilde{X} - D\hat{S}\|_F^2,$$

   where $\hat{S} = [S, \bar{S}]$ and $\tilde{X} = [X, \bar{X}]$. This can be solved by an orthogonal matching pursuit (OMP) [40].
5: Fix $S$ and $\bar{S}$, calculate:
   
   $$E = \frac{1}{M} XS^T - \frac{\rho}{M} \bar{X} \bar{S}^T$$
   
   and
   
   $$F = \frac{1}{M} SS^T - \frac{\rho}{M} \bar{S} \bar{S}^T.$$
6: Update $D$ by solving:
   
   $$D^* = \arg\min_D \left( -2\text{trace}(EF^T) + \text{trace}(D(F - \lambda_{\min}(F)I_k)D^T) \right)$$
   
   subject to: $\|D_k\|_2^2 = 1$ where $k = 1, 2, \ldots, K$.
7: **end while**
8: **Output:** Dictionary $D^*$.

The detection procedure of DFDL is shown in Figure 2.6.
3.1.3 Bag-level label prediction for fully supervised learning

In fully supervised learning approaches, instance-level classifiers are learned from fully labeled data samples. A bag-level label is produced based on the collection of instance-level labels [31]. Following a presence-based assumption in multiple instance learning approaches, a bag-level label is related to the instance-level labels via the number of positive instances. The number of positive instances in the $b$-th bag denoted by $N_b$ is as:

$$P(N_b = n|y_1, y_2, \ldots, y_{n_b}) = I\left(\sum_{i=1}^{n_b} = n\right).$$

The relation between the bag-level label $Y_b \in \{0, 1\}$ and the number of positive instances $N_b$ is as follows:

$$P(Y_b = 1|N_b = n, \nu) = \phi(n; \nu),$$
where $v$ is the bag-level classifier parameter. There are two cases of the function $\phi(n; v)$ to predict the bag-level label as follows:

- **Case 1:**

  $$\phi(n; v) = \frac{e^{v_0 + n v_1}}{1 + e^{v_0 + n v_1}},$$

  where $v = [v_0, v_1] \in \mathbb{R}^2$ is an unknown bag-level classifier parameter. The model follows a logistic regression model with an input vector of $[1, n]$.

- **Case 2:**

  $$\phi(n; v) = I(n \geq v_0),$$

  where $v_0 \in \mathbb{R}^1$ is an unknown bag-level classifier parameter. This model offers a deterministic decision of a bag-level label. A bag is labeled positive if the number of positive instances is greater or equal $v_0$. If $v_0 = 1$, the setting is similar to that of traditional multiple instance learning approaches [15].

- **Case 3:**

  $$\phi(n; v) = \begin{cases} 
v_n, & \text{if } n = 0, 1, \ldots, n_b, \\
v_{n_b}, & \text{if } n > n_b,
\end{cases}$$

  where $v = [v_0, v_1, \ldots, v_{n_b}]$ is an unknown bag-level parameter vector with $n_b$ number of instances in a bag.

### 3.2 Weakly supervised learning - Multiple instance learning

Due to the suitability of multiple instance learning (MIL) and its advantages in reducing labeling tasks and overcoming data intrinsic challenges, histopathological image analysis is formulated as a MIL problem. We are given a set of $B$ bags denoted as $\{X_B, y_B\} \triangleq \{X_b, y_b\}_{b=1}^B$. Each bag $X_b$ is associated with a label $y_b \in \{0, 1\}$ and contains $n_b$ instances $\{x_{bi}\}_{i=1}^{n_b}$, where $x_{bi}$ denotes the $d$-dimensional feature vector of the $i$th instance of bag $b$. Generally, $y_{bi}$ denotes the label of the instance $i$-th in bag $b$ ($b = 1, \ldots, B$ and $i = 1, \ldots, n_b$). Typically, $y_{bi} = 0$ is for negative instances that are irrelevant for classification and positive values of $z_{bi}$ are for positive instances.
that are essential and informative for classification ($y_{bi} = 1$ for one considered cluster of positive instances or $y_{bi} \in \{1, \ldots, K\}$ for different clusters of positive instances). In general, the goal is to train a classifier that maps an instance to its visible/hidden label (depending on instance-level or bag-level learning) and based on the presence or absence of positive instance labels to predict the bag-level label.

### 3.2.1 Multiple instance boosting (MIL-Boost)

Multiple instance boosting (MIL-Boost) [46] generalizes boosting (AnyBoost [28]) to the MIL framework. The concept of the algorithm is that a strong classifier $h$ is built by linearly combining multiple weak classifiers. In MIL-Boost, instances are not individually labeled and only bags or groups of instances are labeled. Weak classifiers are added to the strong classifier iteratively. The derivation of MIL-Boost builds on MIL cost functions, namely Noisy-OR and integrated segmentation and recognition (ISR). They are described as follows:

- **Noisy-OR boost**

  The score of each instance in a bag is $y_{bi} = F(x_{bi})$ where $F(x_{bi}) = \sum_t \lambda_t f^t(x_{bi})$ is a weighted sum of weak classifiers. The probability of each instance in the bag is given by a standard logistic function (sigmoid function) as:

  $$p_{bi} = \frac{1}{1 + \exp(-y_{bi})}.$$  

  The probability of the bag follows a Noisy-OR function:

  $$p_b = 1 - \prod_{i=1}^{n_b} (1 - p_{bi}).$$

  Under the model, the likelihood function associated with a set of training examples is:

  $$\mathcal{L}(F) = \prod_{b=1}^{B} p_b^{y_b} (1 - p_b)^{(1-y_b)},$$

  where $y_b \in \{0, 1\}$ is the label of bag $b$. According to AnyBoost approach, the weight on
Each instance $w_{bi}$ is given as the derivative of the likelihood function over its score as:

$$\frac{\partial \mathcal{L}(F)}{\partial y_{bi}} = \frac{y_b - p_b}{p_b} p_{bi} = w_{bi}$$

Intuitively, the weight $w_{bi}$ on each instance is the product of the bag weight $\frac{y_b - p_b}{p_b}$ and the instance weight $p_{bi}$. In the learning process, the bag weight $\frac{y_b - p_b}{p_b}$ is assigned to always $-1$ for negative bags. Thus, it reduces the effects of negative instances (particularly in negative bags) on the learning procedure. The algorithm selects a subset of instances to assign a high positive weight and then these instances dominate the learning process.

- **ISR boost**

It is denoted that $z_{bi} = \exp(y_{bi})$, $S_b = \sum_{i=1}^{n_b} z_{bi}$, and $P_b = \frac{S_b}{1 + S_b}$. Interpretively, $z_{bi}$ is the likelihood that the object occurs at $bi$ or the instance $y_{bi}$ is positive. The quantity $S_i$ is a likelihood ratio that at least one instance is positive and $P_b$ is the probability that some instance is positive. The log likelihood assigned to a set of training bags is:

$$\log \mathcal{L}(F) = \sum_{b=1}^{B} y_b \log p_b + (1 - y_b) \log(1 - p_b).$$

The weights for the ISR framework are:

$$\frac{\partial \mathcal{L}(F)}{\partial y_{bi}} = \left( y_b - p_b \right) \frac{z_{bi}}{1 + z_{bi}}.$$

Intuitively, all of the instances in the bag including negative instances compete for weight as the weight is normalized by sum of the $z_{bi}$.

### 3.2.2 Multiple clustered instance learning (MCIL)

The key idea of multiple clustered instance learning (MCIL) is embedding a clustering concept into the MIL setting under a boosting framework. A boosting approach allows finding positive instances in positive bags while a clustering technique enables the method to separately divide positive instances into different clusters. In histopathological image analysis, these strategies of MCIL can be used to simultaneously perform image-level classification (cancer vs. non-cancer), pixel-level segmentation (cancer vs. non-cancer cells), and patch-level clustering (cancer sub-
A training dataset contains $B$ bags denoted as $\{X_b, y_b\}_{b=1}^B$. Each bag $X_b$ is associated with a label $y_b \in \mathcal{Y} = \{0, 1\}$. Each of $n_b$ instances $x_{bi} \in \mathcal{X}$ in bag $b$ has a hidden label $y_{bi} \in \mathcal{Y}$, which is unknown during training. The concept of clustering in MCIL is integrated by assuming that there are $K$ clusters $(1, 2, \ldots, K)$ of positive instances in positive bags and there are all negative instances in negative bags (cluster 0). Assuming that there is a hidden variable $y_{ki} \in \mathcal{Y}$ to denote whether the instance $x_{bi}$ belongs to $k$th cluster. Following the similar assumption of MIL approaches, in MCIL, a bag is classified as positive if at least one instance in the bag belongs to any of $K$ clusters. Negative bags consist of all negative instances that do not belong to any of $K$ clusters. The assumption of MCIL can be formulated as follows:

$$
y_{bi} = \max(y_{0i}, y_{1i}, y_{2i}, \ldots, y_{Ki}),$$
$$y_b = \max_i (y_{bi}) \text{ or } y_b = \max_{k=0:K} (y_{ki}).$$

The goal of MCIL is to learn: (i) $K$ instance-level classifiers: $h^k : \mathcal{X} \to \mathcal{Y}$ for $K$ clusters, (ii) the corresponding bag-level classifier for $k$th cluster $H^k : \mathcal{X}^{n_b} \to \mathcal{Y}$, and (iii) the overall bag-level classifier: $H : \mathcal{X}^{n_b} \to \mathcal{Y}$. These classifiers are formulated as follows:

$$H^k(X_b) = \max_i h^k(x_{bi}),$$
$$H(X_b) = \max_{k=0:K} H^k(X_b) = \max_k \max_i h^k(x_{bi}).$$

The cluster-specific instance-level probability $p_{bi}^k$, instance-level probability $p_{bi}$, and bag-level probability $p_b$ are computed as follows:

$$p_{bi}^k = p(y_{bi} = 1 | x_{bi}) = \sigma(2h_{bi}^k), \text{ where } h_{bi}^k = h^k(x_{bi}),$$
$$p_{bi} = p(y_{bi} = 1 | x_{bi}) = g_k(p_{bi}^k),$$
$$p_b = p(y_b = 1 | X_b) = g_i(p_{bi}) = g_i(g_k(p_{bi}^k)) = g_{ik}(p_{bi}),$$

where $\sigma(.)$ is a sigmoid function that is $\sigma(v) = \frac{1}{1+\exp(-v)}$ and $g(.)$ is a softmax function. In MCIL, it can be one of four models of softmax functions: noisy-or (NOR), generalized mean (GM), log-sum-exponential (LSE), and integrated segmentation and recognition (ISR) [44]. The
loss function of a general MIL approach defined in the AnyBoost is:

$$
\mathcal{L} = - \sum_{b=1}^{B} \left( I_{y_b=1} \log p_b + I_{y_b=0} \log (1 - p_b) \right)
$$

where $I(\cdot)$ is an indicator function. The weight of the instance $i^{th}$ of $b^{th}$ training data belongs to cluster $k^{th}$ is computed using a chain rule:

$$
w_{bi}^k = - \frac{\partial \mathcal{L}}{\partial h_{bi}^k} = - \frac{\partial \mathcal{L}}{\partial p_b} \frac{p_b}{\partial p_b} \frac{\partial p_b^k}{\partial h_{bi}^k}
$$

The concept of clustering is encoded by re-weighting the instance-level weight $w_{bi}^k$. Intuitively, if the $k^{th}$ cluster can explain some instances well, the weight of instances and bags for other clusters decrease in re-weighting. A detailed procedure of MCIL is summarized in Algorithm 2. Note that MCIL is considered as a advanced variant of MIL-Boost where there can be more than one cluster of positive instances.

**Algorithm 2** Multiple clustered instance learning.

1. **Input:**
   Bags $\{X_1, X_2, \ldots, X_B\}$ and bag-level labels $\{y_1, y_2, \ldots, y_B\}$.
   K: number of clusters.
   T: number of weak classifiers.
2. **for** $t = 1 \rightarrow T$ **do**
3.   **for** $k = 1 \rightarrow K$ **do**
4.       Compute weights $w_{bi}^k = - \frac{\partial \mathcal{L}}{\partial h_{bi}^k}$
5.       Train weak classifiers $h_{k}^t$ using weights $|w_{bi}^k|$
6.       $h_{k}^t = \arg \min_h \sum_{bi} I_{h(x_{bi}^k) \neq y_b} |w_{bi}^k|$  
7.       Find $\alpha_t$ via line search to minimize $\mathcal{L}(\cdot, h_{k}^t, \cdot)$
8.       $\alpha_{k}^t = \arg \min_\alpha \mathcal{L}(\cdot, h_{k}^t + \alpha h_{k}^t, \cdot)$
9.       Update strong classifiers $h_{k} \leftarrow h_{k} + \alpha_{k}^t h_{k}^t$
10. **end for**
11. **end for**
12. **Output:** $h^1, h^2, \ldots, h^K$. 

3.2.3 Multi-instance multi-label learning in the presence of novel class instances (MIML-NC)

Similar to multiple instance learning, multi-instance multi-label learning (MIML) is a framework for learning in the presence of label ambiguity. In MIML, each bag is described by multiple instances and associated with multiple class labels [47]. A MIML approach can be relaxed into a MIL problem by considering a bag being defined by a set of instances and associated with one label. In MIML, instead of providing a label for each instance, experts provide labels for groups of instances. During the labeling process, it is acceptable that a set of target classes is focused while instances outside the set are appropriately modeled. For example, ornithologists label bird audio according to a list of species while avoid labeling a rain drop or a moving vehicle sound. Multi-instance multi-label learning in the presence of novel class instances (MIML-NC) takes into account the presence or absence of irrelevant instances, namely novel instances. The goal of the model is to determine whether an instance belongs to the novel class or known classes. The bag-level label is obtained by removing the novel class from the union of the instance-level labels.

MIML-NC addresses the MIML problem in the presence of novel class instances using two aspects: (i) a class \( c = 0 \) is associated with novel class instances and (ii) bag-level label removes the novel class from the union of instance labels. The relation between the instance label and its feature vector follows a logistic regression:

\[
P(y_{bi}|x_{bi}, w) = \prod_{c=0}^{C} e^{I_{y_{bi}=c}w_{c}^T x_{bi}} / \sum_{c=0}^{C} e^{w_{c}^T x_{bi}},
\]

where \( w = [w_1, w_2, \ldots, w_C] \) and \( w_c \in \mathbb{R}^{d \times 1} \) is the weight for class \( c \). For each bag, the union of its instance-level labels is denoted as \( Y_{nb}^m = \bigcup_{j=1}^{n_b} y_{bj} \). The relation between the observed bag label \( Y_b \) and \( Y_{nb}^m \) is modeled as:

\[
P(Y_b|Y_{nb}^m) = I_{Y_b=Y_{nb}^m} + I_{Y_b \cup \{0\}=Y_{nb}^m},
\]

where \( Y_b \subseteq \{1, 2, \ldots, C\} \) and \( Y_{nb}^m \subseteq \{0, 1, 2, \ldots, C\} \). To learn the model parameters in MIML-NC, authors consider a maximum likelihood inference and apply the expectation maximization algorithm [14]. A dynamic programming algorithm is also introduced in this paper to compute the posterior probability of each instance. In the testing process, instance annotation and bag-
level label prediction are respectively computed as follows:

\[
\hat{y}_{ti} = \arg \max_{0 \leq k \leq C} w_k^T \mathbf{x}_{ti}
\]

\[
\hat{Y}_t = \bigcup_{i=1}^{n_t} \hat{y}_{ti}.
\]

An unlabeled test instance is detected as a novel instance if

\[ P(y_{ti} = 0 | \mathbf{w}_{ti}, \mathbf{w}) \geq \theta \]

where \(0 \leq \theta \leq 1\) is a manually selected threshold.
Chapter 4: Experiments

4.1 Datasets

To examine how aforementioned classification approaches work on histopathological images, we consider the following histopathological image datasets: breast, kidney, lung, and spleen. The breast dataset is obtained from the publicly available dataset provided by UCSB Center for Bio-Image Informatics [16]. The kidney, lung, and spleen dataset are provided by the Animal Diagnosis Lab, The Pennsylvania State University [42].

The breast dataset consists of 52 images of benign and malignant tissues. Manual segmentation of malignant regions in malignant images is provided while there are no areas of malignant manifestations in benign images. We consider the manual annotation as known instance-level labels. Also, the label of each image is provided in which malignant tissues are denoted as positive images and benign tissues are denoted as negative images. Therefore, in the breast dataset, both instance-level and bag-level labels are available. We use a balanced number of positive and negative bags in training and testing phases. Due to the fact that the diseased cells may appear in a minute portion of the histopathological images, the number of positive patches (including malignant manifestations) is generally significantly smaller than the number of negative patches (including no malignant manifestations). As a result, it is unbalanced in the training and testing phases for the instance-level classifier.

In the kidney, lung, and spleen datasets, only labels of images are available. There are inflamed tissues and normal tissues which are denoted as positive and negative bags, respectively. Each of these datasets contains 300 images of inflamed and normal tissues. There are 200 samples used for training, 50 samples used for validation, and the remaining samples are used for the testing process. We use a balanced number of positive and negative bags in the training and testing phases.
4.1.1 Data preprocessing

As described in Section 2, to begin with, each image is divided into small patches to formulate a multiple instance learning problem. In the breast cancer dataset, each image is split into a set of patches whose patch size is $100 \times 100 \times 3$ pixels. There are maximum of 56 instances per bag in the breast cancer dataset. In the kidney, lung, and spleen datasets, each image is also divided into a set of patches whose patch size is $100 \times 100 \times 3$ pixels. There are 130 instances per bag in each of these datasets. The Reinhard method and K-means algorithm are applied to normalize color variations and reduce irrelevance patches, respectively.

4.1.2 Feature engineering

To extract features, we use morphological features and fractal-based features. To be specific, features are the combination of (i) HOG features (36 features) (ii) morphological features (24 features), and (iii) SFTA features (24 features). When combined, there are 84 features extracted for each instance in a bag.

4.2 Setting

In the following, we present the setting experiment including evaluation metrics and parameter selection methods.

4.2.1 Evaluation metrics

We consider accuracy as a major evaluation criterion. Due to the balanced number of positive and negative bags, accuracy is used for the bag-level evaluation. Additionally, the visualization of predicted cancerous areas is considered as an instance-level assessment. A model complexity of each method is also employed as an evaluation criterion.

4.2.2 Parameter selection

The range of parameter tuning values for each of the aforementioned methods are extracted from the corresponding publications. In logistic regression methods, we search over the range of regularization parameters $\lambda \in \{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. In MIL-Boost and
MCIL frameworks, we search over the generalized mean softmax function with the sharpness controlling parameter named \( r \in \{15, 20, 25\} \) and the number of weak classifiers named \( T \in \{150, 200, 250\} \). For DFDL, we tune the regularization parameter \( \rho \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\} \), sparsity level parameter \( \lambda \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\} \), and the number of dictionary bases \( k \in \{100, 200, 500\} \). MIML-NC is a tuning-free method. All the results as well as selected parameters are reported using 10-fold cross validation [23].

4.3 Results and analysis

As described in Section 3.1.3, to achieve bag-level labels in logistic regression methods, we apply case 1 and case 2 of relation between bag-level and instance-level labels. Based on differences concerning regularization methods and bag-level label classifier approaches, we denote the logistic regression models by LR-SISL-\( L_1-M_1 \), LR-SISL-\( L_2-M_1 \), LR-SISL-\( L_1-M_2 \), and LR-SISL-\( L_2-M_2 \) corresponding to logistic regression using \( L_1 \)-norm and case 1, \( L_2 \)-norm and case 1, \( L_1 \)-norm and case 2, \( L_2 \)-norm and case 2 in predicting bag-level labels, respectively. DFDL-SISL and DFDL are considered as two versions of DFDL in which DFDL-SISL bases are learned from fully supervised data setting (instance-level and bag-level labels are available) and DFDL bases are learned from weakly supervised data setting (only bag-level labels are available).

The performances of logistic regression methods, DFDL, MIL-Boost, MCIL, and MIMIL-NC over the breast, kidney, lung, and spleen datasets are provided in Table 4.1. These results are obtained at optimal parameters that are chosen via 10-fold cross validation. In kidney, lung, and spleen datasets, there are no instance-level labels provided. Therefore, in Table 4.1, there is no accuracy for the logistic regression methods (LR-SISL family). In the breast dataset, logistic regression methods attain good results and LR-SISL-\( L_2-M_1 \) obtains the best accuracy (78%). This is because the logistic regression methods benefit from the availability of instance-level labels while other MIL approaches learn instance-level classifiers from a weak relation among known bag-level labels and hidden instance-level labels. DFDL-SISL and DFDL obtain the comparable results. Due to the feature vectors are obtained from the feature engineering process instead of using the raw representations of patches to achieve the fair comparison to others, performance of the DFDL models are worse than other approaches in the case of using feature engineering. When using raw represent features, DFDL achieves a comparable result. MIL-Boost, MCIL, and MIMIL-NC achieve comparable accuracy to the logistic regression methods. This emphasizes the capacity of MIL frameworks in learning bag-level labels. They perform
well without instance-level information. Using two-tailed t-test at 95 percent confidence level, we obtained: (i) in the Kidney and Spleen datasets, MIL-Boost and MIML-NC are statistically indistinguishable and (ii) in the Lung dataset, DFDL using raw features, MIL-Boost, and MCIL are statistically indistinguishable.

In the breast dataset, the manually cancerous cells are provided; therefore, we provide visualization of instance-level ground-truth and predicted instance-level results of LR-SISL-$L_2$-$M_1$ and MIML-NC. The available packages of another methods do not provide access to instance-level prediction, hence their visualizations are omitted. Two examples of visualization are shown in Figure 4.1 and Figure 4.2. Even the instance-level labels are provided in training logistic regression classifiers, the predicted instance-level results are still poor. LR-SISL classifiers tend to pick the high density areas to be cancerous clues. MIML-NC tends to pick the white areas to be cancerous manifestations. As the presence-based assumption in MIL, both LR-SISL and MIML-NC correctly indicate at least one cancerous instance, which creates correct bag-level prediction results.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Breast</th>
<th>Kidney</th>
<th>Lung</th>
<th>Spleen</th>
</tr>
</thead>
<tbody>
<tr>
<td>SISL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LR-SISL-$L_1$-$M_1$</td>
<td>75.00 ± 6.54</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>LR-SISL-$L_2$-$M_1$</td>
<td>78.00 ± 5.33</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>LR-SISL-$L_1$-$M_2$</td>
<td>71.00 ± 5.86</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>LR-SISL-$L_2$-$M_2$</td>
<td>73.00 ± 4.96</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>DFDL-SISL</td>
<td>61.00 ± 5.33</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MIL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DFDL</td>
<td>60.00 ± 5.98</td>
<td>60.00 ± 2.14</td>
<td>86.80 ± 2.09</td>
<td>75.60 ± 1.22</td>
</tr>
<tr>
<td>DFDL-Raw</td>
<td>77.00 ± 4.73</td>
<td>84.40 ± 1.33</td>
<td>96.80 ± 0.74</td>
<td>91.40 ± 0.43</td>
</tr>
<tr>
<td>MIL-Boost</td>
<td>75.00 ± 5.00</td>
<td>90.40 ± 2.00</td>
<td>95.80 ± 1.09</td>
<td>93.11 ± 0.95</td>
</tr>
<tr>
<td>MCIL</td>
<td>72.00 ± 4.67</td>
<td>90.40 ± 1.60</td>
<td>94.80 ± 1.31</td>
<td>90.89 ± 1.11</td>
</tr>
<tr>
<td>MIML-NC</td>
<td>74.00 ± 3.71</td>
<td>87.60 ± 1.42</td>
<td>95.00 ± 0.86</td>
<td>95.60 ± 0.72</td>
</tr>
</tbody>
</table>

Table 4.1: Bag-level performances on breast, kidney, lung, and spleen cancer datasets with different frameworks on SISL and MIL approaches.

4.3.1 Complexity analysis

In this section, we compare the training computational complexity of the aforementioned methods. The comparison is evaluated for one iteration. In this report, we consider a binary classification problem, therefore the number of classes $C = 2$. For consistency, the list of parameters
used for different methods is denoted as follows:

- \( M = B\bar{n}_b \): the total number of instances in the training process where \( B \) is the number of bags and \( \bar{n}_b \) is the average number of instances per bag.
- \( d \): data dimension.
- \( K_1 \): the number of bases per class in DFDL.
- \( N \): the number of training patches per class in DFDL.
- \( \ell \): sparsity level in DFDL.
- \( K_2 \): the number of positive clusters in MCIL.
- \( T \): the number of weak classifiers in MCIL.

In LR-SISL methods, an iteration in training instance-level classifiers takes linear time on the size of the training set. In these methods, classifiers are trained over all \( M \) instances, which are \( d \)-dimensional feature vectors. Hence, at the instance-level, the complexity of LR-SISL methods is \( O(Md) \). Similar to those of the bag-level classifier in case 1, the complexity is \( O(2B) \sim O(B) \) (dimension equals 2). In case 2, there is only comparison of the number of positive instances with a fixed value of \( v_0 \), it is in \( O(1) \). Therefore, the complexity of logistic regression for case 1 methods is \( O(Md + B) \) and for case 2 methods is \( O(Md + 1) \). In DFDL, the complexity of OMP method in solving the optimal sparse problem when the dictionary is stored in memory in its entirety is \( O(N^2(2dK_1 + \ell^2K_1 + 3\ell K_1 + \ell^3) + dK_1^2) \) [33]. As assumed that \( \ell \ll K_1 \approx d \ll N \), the complexity above is simplified to \( O(N(2dK_1 + \ell^2K)) = O(K_1N(2d + \ell^2)) \). The sparse coding step in DFDL consists of solving \( C \) sparse coding problems \( \hat{S} = \arg \min \| \tilde{X} - D_i\hat{S}_i \|_F^2 \), each problem has complexity of \( O(K_1(CN)(2d + \ell^2)) \sim O(K_1N(2d + \ell^2)) \) (since \( C = 2 \) is significantly small compared to other parameters). The total complexity of DFDL is \( O(K_1M(2d + \ell^2)) \). In MCIL, the training process requires to go over \( T \) weak classifiers and \( K_2 \) number of positive clusters. Therefore, the complexity of MCIL is \( O(MTK_2d) \). Similarly, in MIL-Boost, which is considered as a specific case of MCIL with \( K_2 = 1 \), the complexity is \( O(MTd) \). In MIML-NC, for calculating the posterior probability in the expectation step, it is required to go over \( C + 1 \) classes (\( C \) classes and a novel class) for each instance in a bag for \( B \) bags is \( O(B(C + 1)2^{C+1}\bar{n}_bd) = O(24Md) \sim O(Md) \). The maximization step requires to obtain the step size via backtracking line search and update the parameter. It needs
\( O(Bn_bCd) = O(2Md) \sim O(Md) \). As a result, the complexity of MIML-NC is \( O(Md) \). The comparison of complexities among these algorithms are provided in Table 4.2.

Table 4.2: Complexity analysis for different methods.

<table>
<thead>
<tr>
<th>METHODS</th>
<th>COMPLEXITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>LR-SISL-( M_1 )</td>
<td>( Md + B )</td>
</tr>
<tr>
<td>LR-SISL-( M_2 )</td>
<td>( Md + 1 )</td>
</tr>
<tr>
<td>DFDL</td>
<td>( K_1 M(2d + l^2) )</td>
</tr>
<tr>
<td>MIL-BOOST</td>
<td>( MTd )</td>
</tr>
<tr>
<td>MCIL</td>
<td>( MTK_2d )</td>
</tr>
<tr>
<td>MIML-NC</td>
<td>( Md )</td>
</tr>
</tbody>
</table>

4.4 Possible future directions

Weakly-supervised approaches such as multiple instance learning are of interest in histopathological image analysis. The reason is that such frameworks are convenient: they avoid the need of manual segmentation/annotation and are solely based on image labels. Even though the aforementioned methods perform well on these histopathological images, there are several recent trends that we are considering for future research.

First, this report considers feature engineering to extract feature vectors. This is heavily based on how to choose the methods of feature extraction and feature selection. To overcome the issue, researches currently use convolutional neural networks (CNNs) to extract feature from the raw image, and there is no need of domain knowledge in extracting features.

Second, in the aforementioned histopathology datasets, patient-based information is not mentioned. It means that we may use images from the same patient in both the training and testing process. This definitely provides a better performance than the usual. However, it may not follow the nature of medical diagnosis and treatment where we generally transfer learned knowledge from certain patients to unseen patients, rather than performing the similar diagnosis on the patients who we know exactly their conditions. In the future, we consider using separately patient-based images in the training and testing process.
Figure 4.1: Visualization example 1. Red areas indicate cancerous cells. Green ones are normal cells.
Figure 4.2: Visualization example 2. Red areas indicate cancerous cells. Green ones are normal cells.
Chapter 5: Conclusion

In this report, we reviewed different frameworks of dealing with histopathological image analysis. Different methods can be applied to preprocessing, extract feature, and classify histopathological image datasets. Due to the challenges in color normalization during the staining process, it is necessary to normalize color. The Reinhard method performs well on color normalization. Feature engineering, which applies the domain knowledge, is suitable for extracting features. Depending on different types of histopathological images (i.e., different cancer types, cancer and non-cancer), different features can be selected. In the report, we used morphological, texture features and interest point detector to extract features. From our experimental results, in comparing to single-instance single-label frameworks such as LR-SISL and DFDL, multiple instance learning methods such as MIL-Boost, MCIL, and MIML-NC, which avoid the manually time-consuming and labor-intensive annotation, achieved the promising results on histopathological image analysis.
Bibliography


APPENDICES