A Self-Supervised Contrastive Learning Approach for Whole Slide Image Representation in Digital Pathology

by

Parsa Ashrafi Fashi

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, 2022

Author's Declaration

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

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

Abstract

Digital pathology has recently expanded the field of medical image processing for diagnostic reasons. Whole slide images (WSIs) of histopathology are often accompanied by information on the location and type of diseases and cancers displayed. Digital scanning has made it possible to create high-quality WSIs from tissue slides quickly. As a result, hospitals and clinics now have more WSI archives. As a result, rapid WSI analysis is necessary to meet the demands of modern pathology workflow. The advantages of pathology have increased the popularity of computerized image analysis and diagnosis.

The recent development of artificial neural networks in AI has changed the field of digital pathology. Deep learning can help pathologists segment and categorize regions and nuclei and search among WSIs for comparable morphology. However, because of the large data size of WSIs, representing digitized pathology slides has proven difficult. Furthermore, the morphological differences between diagnoses may be slim, making WSI representation problematic. Convolutional neural networks are currently being used to generate a single vector representation from a WSI (CNN). Multiple instance learning is a solution to tackle the problem of giga-pixel image representation. In multiple instance learning, all patches in a slide are combined to create a single vector representation.

Self-supervised learning has also shown impressive generalization outcomes in recent years. In self-supervised learning, a model is trained using pseudo-labels on a pretext task to improve accuracy on the main goal task. Contrastive learning is also a new scheme for self-supervision that aids the model produce more robust presentations. In this thesis, we describe a self-supervised approach that utilizes the anatomic site information provided by each WSI during tissue preparation and digitization. We exploit an Attention-based Multiple instance learning setup along with supervised contrastive learning. Furthermore, we show that using supervised contrastive learning approaches in the pretext stage improves model embedding quality in both classification and search tasks. We test our model on an image search on the TCGA depository dataset, a Lung cancer classification task and a Lung-Kidney-Stomach immunofluorescence WSI dataset.

Acknowledgements

I would like to thank both Professor Tizhoosh and Babaie whose continuous support made this thesis possible. I would also want to thank Sobhan, Amir, Daniel and Milad that helped me on the progress of thesis. Finally, I would like to thank Ghazal that was beside me at all stages of writing this thesis, both emotionally and mentally, and encouraged me to the end.

Dedication

To the passengers of flight PS752.

Table of Contents

List of Figures viii

List of Tables xi

1 Introduction 1

1.1 Introduction . . . 1

1.2 Motivation . . . 2

2 Digital Pathology, Deep Learning and Image Search 4

2.1 Digital Pathology . . . 4

2.2 Deep Learning . . . 6

2.2.1 Basic Artificial Neural Networks . . . 7

2.3 Related topics in Deep Learning . . . 13

2.3.1 Content-Based Image Retrieval . . . 14

2.3.2 WSI Representation Learning . . . 16

2.3.3 Multiple Instance Learning . . . 17

2.3.4 Self-Supervised Learning . . . 18

2.3.5 Contrastive learning . . . 19

3 Methodology 22

3.1 Proposed Methodology and Architecture . . . 22

3.2 Patch Selection 24
3.3 Feature Extraction 24
3.4 Attention-Based Pooling 25
3.5 Self-Supervision and Contrastive Learning-Based on Primary Site Information 27

4 Experiments and Results **30**
4.1 WSI Search Results 30
4.2 Lung Cancer: LUAD/LUSC Classification 35
4.3 Attention Pooling Effectiveness 37

5 Summary and Conclusions **39**
5.1 Summary 39
5.2 Conclusions 40
5.3 Potential Directions 40

References **42**

Appendix **51**

List of Figures

2.1 A digital pathology scanner. WSI scanners are capable of producing high quality images of multiple slides at the same time. Image taken from Leica Biosystems official website 5

2.2 Multi-magnification representation of a WSI. Image taken from [6] 6

2.3 A basic artificial neural network. First, inputs (blue block) are fed to a number of hidden layers (red blocks). The output (yellow block) of these hidden layers are then used as ground-truth for comparison to learn predictions. . 8

2.4 Detail of a ANN layer. The input (blue block) are multiplied by the corresponding weights and then aggregated. A bias term (orange block) is added to the summation and the final result is then passed to an activation function (green block), which is a rectified linear unit (ReLU) in this case. The output (yellow block) is then utilized for further computations during the forward path. 9

2.5 Comparison of different values of learning rate. Taken from this website . 10

2.6 Loss surface of VGG [77]. Taken from original repository of [57] 11

2.7 Exponential decay, Cosine decay and Step decay schedulers [23] [59]. 12

2.8 An illustration of convolutional block computations. The computational block (marked green) is multiplied elementwise with each same size subsquares of the input (marked blue), and the pooled summations of this multiplications make the output (red) of the convolutional operation. . . . 13

2.9 an Overall illustration of a CNN model. This shape shows the convolutional computation and the forward path. The final layer of the CNN is then pooled and fed to a number of fully connected layers for prediction and error measurement. 14

2.10 A complete workflow of a CBIR system. The image (marked in grey) is passed to some computational blocks for feature extraction (marked in blue). The features are then indexed for archiving (green block). A query image (marked in cyan) goes through the same feature extraction modules as the indexed images in archive. Then similarity of the query and the indexed images are computed (orange block). Subsequently, the results are ranked based on the similarity score from highest to lowest (marked in dark grey). Based on the user (expert) feedback, the feature extractor, similarity measurements and the indexing can be optimized (red block). 15

2.11 The overall illustration of a multiple instance learning setup in the case of digital pathology. The cancerous region is shown with a yellow contour. The WSI is broken down into multiple smaller patches. In the case of MIL, the patches do not have individual labels but all have a single label, namely the WSI label. After CNN feature extraction (blue blocks), the features (marked in red) are aggregated via a MIL technique (yellow block) to make a single representation for all instances. This representation is then used for comparison with the label in a classification setup (green block). 17

2.12 A comparison of basic supervised learning and contrastive learning. In contrastive learning, instead of comparing the instances only with their own labels, the distances of different instances with similar labels are minimized, and the negative samples are pushed away in the feature space. 20

3.1 The proposed SS-CAMIL concept. The blocks that the transferred knowledge of pretext task (e.g., for label "kidney" as the primary site) are used for the downstream task (e.g., for label "KIRP", *kidney renal papillary cell carcinoma*, as the primary diagnosis), outlined with a grey line. For the LUAD/LUSC classification task, only the blocks on the right side of the dashed red line are used when only pre-trained features will be used. . . . 23

3.2 The complete model architecture 23

3.3 The pipeline for Yottixel patch extraction. Taken from [44]. 24

3.4 Comparison of average pooling and attention-based pooling. In average pooling, each input index (blue blocks) are averaged among different instances. In attention-based pooling, each feature is multiplied by trainable weights first (yellow block). 26

4.1 Exponential learning rate decay with different exponential bases. 32

4.2 t-SNE of CNN-DS [32] (Taken from the paper) (top left) and CAMIL (top right) and SS-CAMIL (bottom). 36

List of Tables

3.1 EfficientNet Comparison. 25
3.2 Tumor types, subtypes and primary sites. 28

4.1 Horizontal Search Results. F1-scores of Majority-3 (in %) are reported. . . 33
4.2 Vertical Search Results. F1-scores of Majority-3 (in %) are reported. . . . 34
4.3 LUAD/LUSC classification. 37
4.4 Attention pooling scores of 9 different WSIs. 38

1 Cancer subtype abbreviations. 51

Chapter 1

Introduction

1.1 Introduction

In recent years, the development of digital pathology has opened up new possibilities in the field of medical image analysis for diagnostic purposes. Images of histopathology, also known as whole slide images (WSIs), are typically accompanied by information on the location and type of illnesses and malignancies being depicted. Recent advancements in digital technology have made it possible to make high-quality WSIs from tissue slides in a short period of time using digital scanning. A direct outcome of this has been a significant increase in the number of WSI archives in hospitals and clinics. It has therefore become evident that quick analysis of WSIs is required in order to fulfil pressing requirements in the everyday workflow of modern pathology. As a result, computerised techniques for image analysis and diagnosis have become increasingly popular as a result of the digital scanning of slides, in addition to the other advantages of pathology.

The field of digital pathology has been drastically changing due to the recent success of artificial neural networks in AI domain. Various pathological tasks, including segmentation and categorization of areas and nuclei, as well as searching among WSIs to locate similar morphology, can be made easier with deep learning. However, because of the huge data size of WSIs (which is typically greater than 50,000×50,000 pixels), the depiction of digitized pathology slides has proven to be a difficult task. Furthermore, the morphological traits that distinguish between different diagnoses may be microscopically small, posing a significant difficulty for WSI representation. The process of directly generating a single vector representation from a WSI is currently being investigated using convolutional neural networks (CNN).

Additionally, in recent years, self-supervised learning has demonstrated remarkable results in terms of generalization. A model is trained using pseudo-labels on a pretext task in self-supervised learning, which allows the model to produce more accurate results on the main target task. In this thesis, we present a self-supervised technique that takes advantage of the primary site information provided by each WSI, which is always available during the tissue preparation and subsequent digitization process of the tissue. We also demonstrate that including supervised contrastive learning techniques into the pretext stage can increase the quality of model embeddings in both the WSI classification and search tasks.

1.2 Motivation

Digital pathology slides are acquired and scanned from various anatomic sites. Each anatomic site has different cancer types. For example, two of the major cancers in the Lung are Lung Adenocarcinoma and Lung Squamous Cell Carcinoma. Also, two significant categories of primary brain neoplasms with different malignant potential and behavior are low grade gliomas and high grade gliomas, including glioblastoma multiforme. Each cancer type and malignancy in general is unique in each anatomic site and has different characteristics based on what anatomical site the tissue is extracted from. Therefore, identifying the primary site of the tissue can help the model identify the characteristics of cancer in each slide in a more efficient way. Since the site of each tissue is always available with the tissue, we can use it as prior information to help our deep learning model understand cancers better.

Considering state-of-the-art self-supervision approaches, contrastive learning is a practical approach to train models for a pretext task. Contrastive learning helps the model learn rich representation from the provided information. In this thesis, we will use the anatomical site as a pseudo-label for training, and we will use Supervised Contrastive Loss for training the model with the pseudo-label.

Another aspect of the pathology slides is their enormous size. The classification of slides needs to be broken down into smaller images (patches). To consider the patches as representatives of a slide, Multiple instance learning (MIL) approaches are exploited. In the MIL setup, all patches from each slide are considered instances of a bigger bag. The extracted features of each patch are aggregated to represent a whole slide to classify each slide. Therefore, we conduct our experiments in a MIL setup. Different approaches to MIL exist, such as Deep sets, Graph representations and attention blocks. In this paper, we use an attention-based MIL setup for aggregation.

These have motivated us to develop a pathology-related self-supervised learning approach and a model to train and classify digital pathology slides.

Chapter 2

Digital Pathology, Deep Learning and Image Search

With the rise in importance and exploitation of digitized histopathology slides, computer-aided diagnosis (CAD) has become a popular approach and area of research. In recent years, learning-based approaches have proven to be dominant among CAD systems. Artificial neural networks are being exploited in various tasks, such as instance classification, image segmentation, and information retrieval. Digital Pathology has also been widely explored as an application field of artificial intelligence.

2.1 Digital Pathology

Pathology is an area of medicine that entails the investigation and diagnosis of disease using surgically removed organs, tissues (biopsy samples), physiological fluids, and, in certain situations, the entire body (autopsy) [48]. Pathology examines the causes, mechanisms of disease formation, structural changes in cells, and the effects of these changes [48]. Pathologists are specialists in various diseases, including cancer, and are responsible for the great majority of cancer diagnosis [48]. A light microscope is used to examine the cellular pattern of tissue samples to identify whether they are malignant or not (benign) [48]. Pathologists also use genetic research and gene markers to diagnose and classify various diseases [65].

Digital pathology is concerned with the acquisition and analysis of scanned and digitized pathology glass slides [2]. In digitizing, the slides are scanned with specific microscopic scanners, (pictured in Figure 2.1) and through this digitization, whole slide images

(WSIs) can be viewed and analyzed via computer-assisted programs and software. Digital pathology has become significant in recent years due to the better accessibility to slides for pathologists and accurate diagnosis and prediction with the help of current artificial intelligence advances [83].

Figure 2.1: A digital pathology scanner. WSI scanners are capable of producing high quality images of multiple slides at the same time. Image taken from Leica Biosystems official website

Digital pathology slides are categorized as gigapixel images. They can naturally be larger than 100,000 × 100,000 pixels in size. Due to their massive size, they are represented in a pyramid structure [6]. Each slide is represented in different magnification levels based on the desired zoom level. The lowest zoom level is typically called the thumbnail of a slide, and the highest level is the highest resolution of the slide image with the most resolved details. Figure 2.2 shows the pyramid representation of a WSI. Each magnification level contains specific information. In the highest magnification, the detail of orientation, placement, mitosis and containment of each nucleus can be seen. In contrast, in lower magnification, the gland information and the growth of cancerous clusters is observed [68].

Due to their large size, WSIs are often broken into smaller images (patches) for computer-assisted analysis and diagnosis tasks. The process of selecting the area and the magnification of each patch are mainly divided into two main categories of "supervised" and

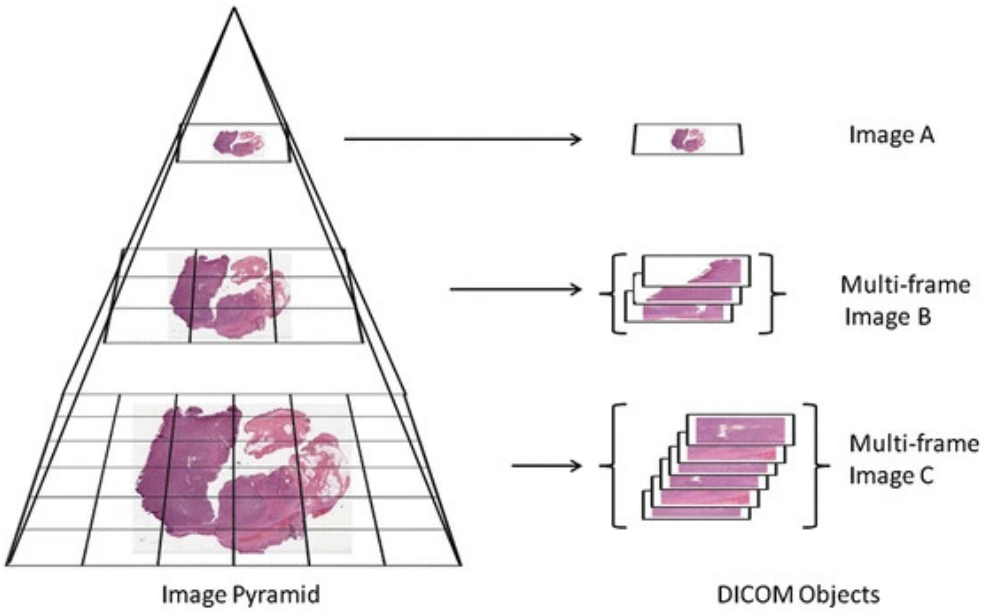

Figure 2.2: Multi-magnification representation of a WSI. Image taken from [6]

"unsupervised". In the supervised setup, firstly, one or more pathology experts specify the locations associated with the cancerous regions in a WSI. Then, patches are selected from these regions specifically [73]. On the other hand, unsupervised patch selection is done without the collaboration of a pathologist. This patch selection method is mainly based on low-level features like colour and location[45].

2.2 Deep Learning

Deep learning (DL) is an area of artificial intelligence concerned with the design and training of artificial neural networks with many layers, inspired by the construction and process of the human brain [53]. Deep learning models are constructed by multiple neural layers. Each layer consists of different parameters called "weights", which are tuned to predict specific pieces of information from various inputs [41]. Deep learning models, which are generally artificial neural networks (ANNs), are predictive models employed in different tasks, such as classification, object detection, segmentation and sequence prediction [53].

Deep Learning can be divided into three major subsections: Supervised, Unsupervised and reinforcement learning [53]. In supervised learning, the model learns its parameters

based on labels assigned to each input. After the input has been fed to an ANN, the output is compared to the given label, and with the help of various objective functions, the difference of the output and the desired label is computed, and the parameters are then "trained" with different optimization algorithms to decrease the distance between the label and the prediction [52].

In an *unsupervised* setup, the labels are absent, perhaps because it is too expensive to label the data, and the parameters are tuned with the help of objective functions that do not require any guidance from labels. Also, in reinforcement learning problems, an agent interacts with an environment and tries to solve specific problems based on rewards and punishment that it receives from interacting with an environment [79]. ANNs can be applied to various data types, including but not limited to images , text, time series and digital signals [86] [15] [40] [67]. With the help of DL, the field of artificial intelligence is growing rapidly and is currently one of the most active fields in the field of computer science and engineering.

2.2.1 Basic Artificial Neural Networks

Figure 2.3 shows a simple ANN. As it can be seen, an ANN consists of two or more layers. Each layer also consists of multiple neurons. The input features are multiplied by particular values, which is called weights, and then summed to create a new value. The weights of the next layer also apply the new value to produce new values. In the final layer, the aggregation of all the final summations creates the model's output. The weights of each layer are then subject to gradient-based adjustments to predict the output more efficiently. Figure 2.4 shows the working process of each ANN layer. If the inputs to a layer are denoted with x_i and the corresponding weights with w_i, the corresponding neuron in the next layer will be computed as

$$y_j = f(\Sigma w_i x_i + b), \tag{2.1}$$

where b is the bias term of the neuron and $f(\cdot)$ is called an "activation function".

Activation functions are non-linear functions that mimic the spike characteristics of the brain neurons and control the neuron's output value. Activation functions limit the value of the neuron's output to a specific range. They also add non-linearity to the model, which helps the model predict inputs with non-linear decision boundaries [28].

If the output of the ANN is denoted as as y and the corresponding label with $\hat{y}$, the distance between the prediction and the label (the loss value) can be computed with the help of an objective function given as

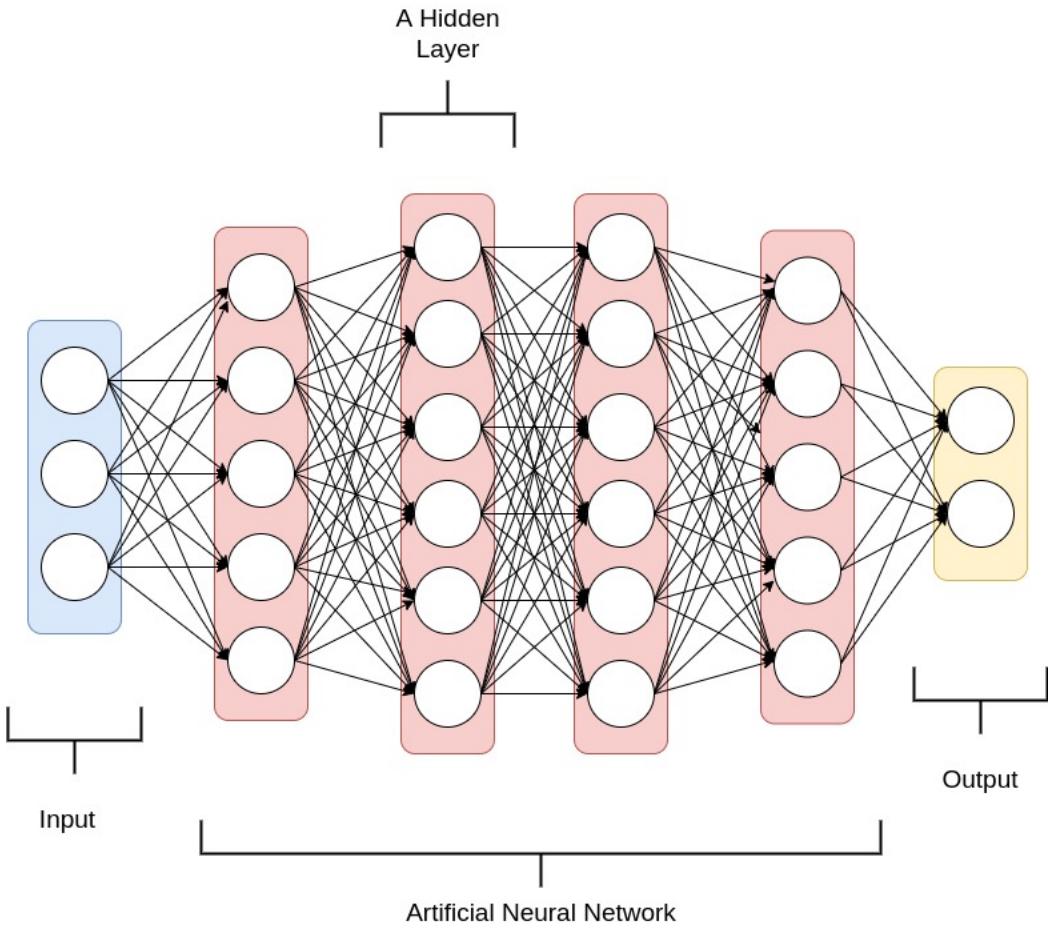

Figure 2.3: A basic artificial neural network. First, inputs (blue block) are fed to a number of hidden layers (red blocks). The output (yellow block) of these hidden layers are then used as ground-truth for comparison to learn predictions.

$$\text{loss} = L(y, \hat{y}), \tag{2.2}$$

where L is the objective (loss) function. Based on the task and the desired output of the ANN, different loss functions can be used, such as mean squared loss, cross-entropy loss and contrastive loss [63] [43] [46] [10].

The process mentioned above is also described as a "forward pass". After an iteration of the forward pass, The back-propagation (BP) process is initiated to optimize the model's parameters [54]. In BP, an optimization method is used to find the optimal value of

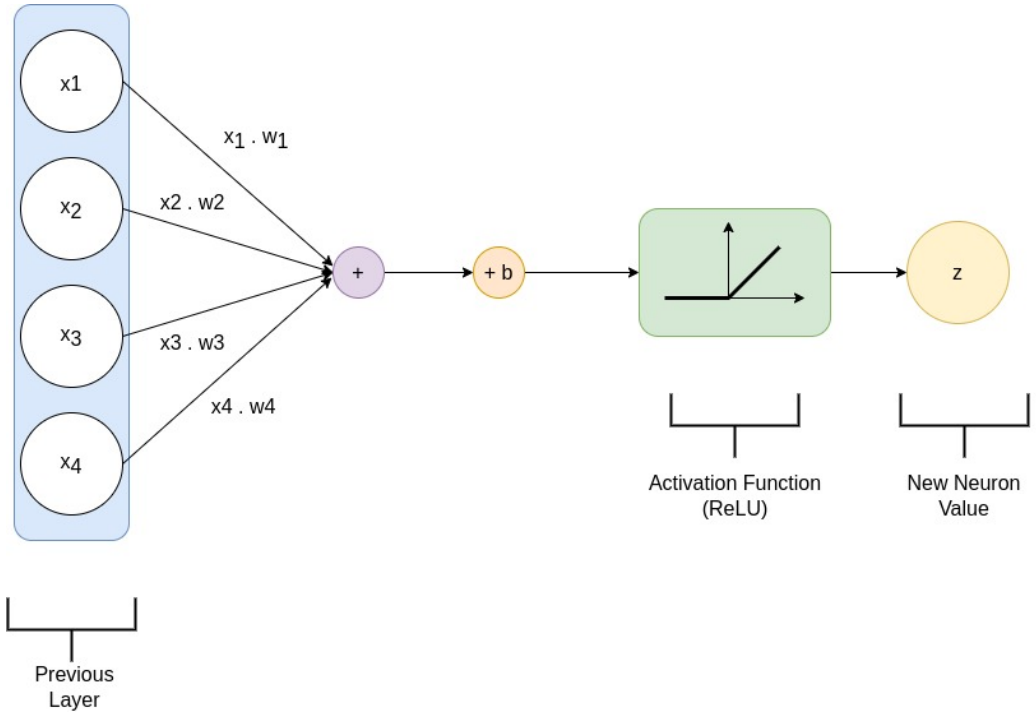

Figure 2.4: Detail of a ANN layer. The input (blue block) are multiplied by the corresponding weights and then aggregated. A bias term (orange block) is added to the summation and the final result is then passed to an activation function (green block), which is a rectified linear unit (ReLU) in this case. The output (yellow block) is then utilized for further computations during the forward path.

weights. A standard optimization algorithm used in the literature is called stochastic gradient descent (SGD) [52]. In SGD, the gradient of the loss function is computed with respect to each weight value. The new weight value is then computed as

$$w' = w - \alpha \frac{\partial L}{\partial w}, \tag{2.3}$$

where α is called the *learning rate*[63]. The intuition behind the SGD is that if parameters move against the gradient of the optimization function, they will eventually reach the optimal point [52]. Learning rate is also a very important hyperparameter in a learning setup, and it indicates the size of the steps with which the model moves toward the optimal points [63]. Large learning rates will help the model reach its optimal value faster, but it may be too large to find the actual optimal point. Smaller step sizes have better accuracy

but tend to be stuck in the local minima if not used carefully [22].

As can be seen in Figure 2.5, small learning rates slow the training setup and can sometimes mislead the model into getting stuck at a local minimum [22]. Large learning rates can also miss the global optimum if the steps are larger than the global optimums domain [22]. The loss surface of a VGG model (a very deep convolutional neural network that is utilized widely in the literature) can be observed in Figure 2.6, which is a classification model [77]. It can be observed how challenging it could be to converge to a solution through a non-convex, non-smooth loss function [42].

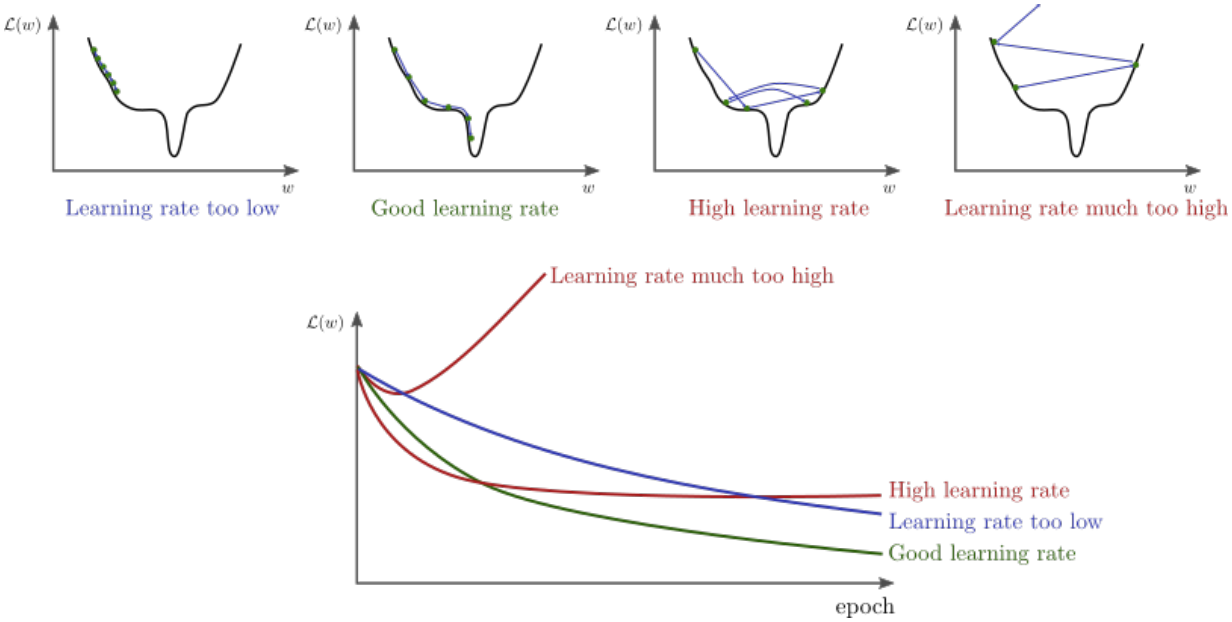

Figure 2.5: Comparison of different values of learning rate. Taken from this website

It is common in the literature to reduce the learning rate gradually to improve convergence [23]. Figure 2.7 shows three different learning rate schedulers, namely exponential decay , cosine decay and step decay [59] [23].

The above process is then repeated for different inputs. To help the model learn more from the inputs, the passing of the entire training data is repeated in multiple epochs. The input is also commonly fed to large batches to help the optimization algorithm produce better gradients and efficiently utilize the computational power [29].

Before initiating the training process, the dataset is divided into three sections: train, validation, and test set [27]. The train set is the portion of the data used to train the

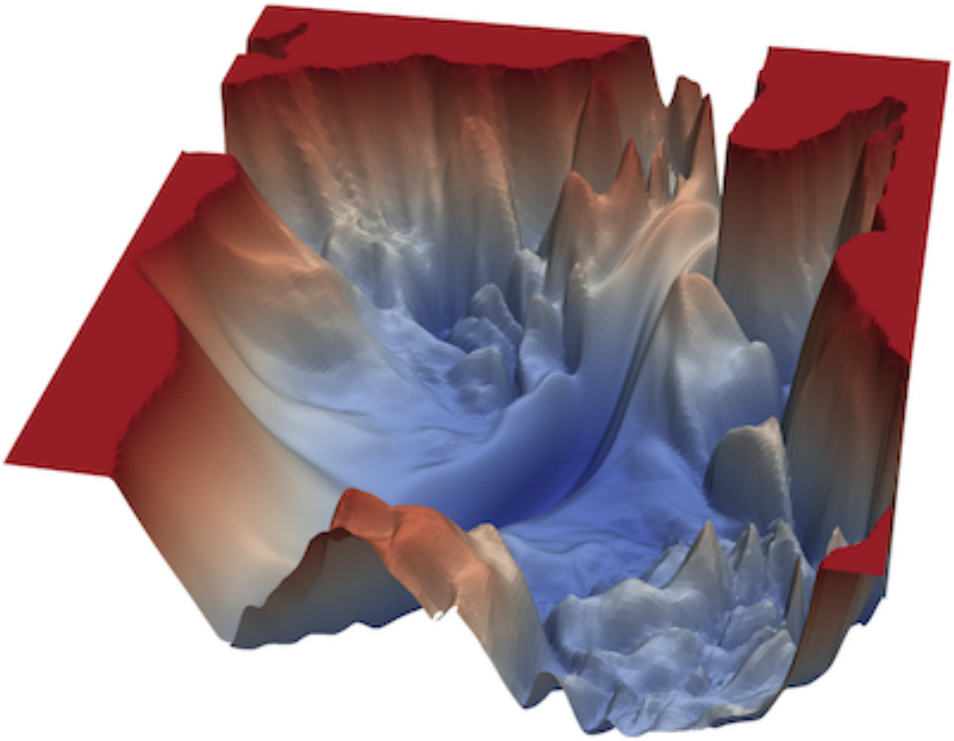

Figure 2.6: Loss surface of VGG [77]. Taken from original repository of [57]

model parameters. After each training epoch, the model predicts the data available in the validation set to check whether the training hyperparameters, such as learning rate and batch size, are appropriate for the training phase [29]. Since the model has not observed the validation set data in the training phase, it should verify whether the model being trained can generalize to unseen data instances. It can reveal whether a model is overfitting the training data. "Overfitting" occurs when the model has excellent performance on the training set but performs poorly on the validation and test dataset [7]. After the training process is completed, the model is evaluated on the test dataset to measure how the model performs on an unseen dataset.

Previously described ANN learning uses basic one-dimensional layers. However, the recent deep learning models use more complex learning layer architecture called *convolutional neural networks (CNN)* [58]. CNNs are widely used deep learning modules to learn image data and, as the name suggests, perform 2D convolution operation on the input. Figure 2.8 shows a basic convolution layer. As it can be observed, the convolution of the

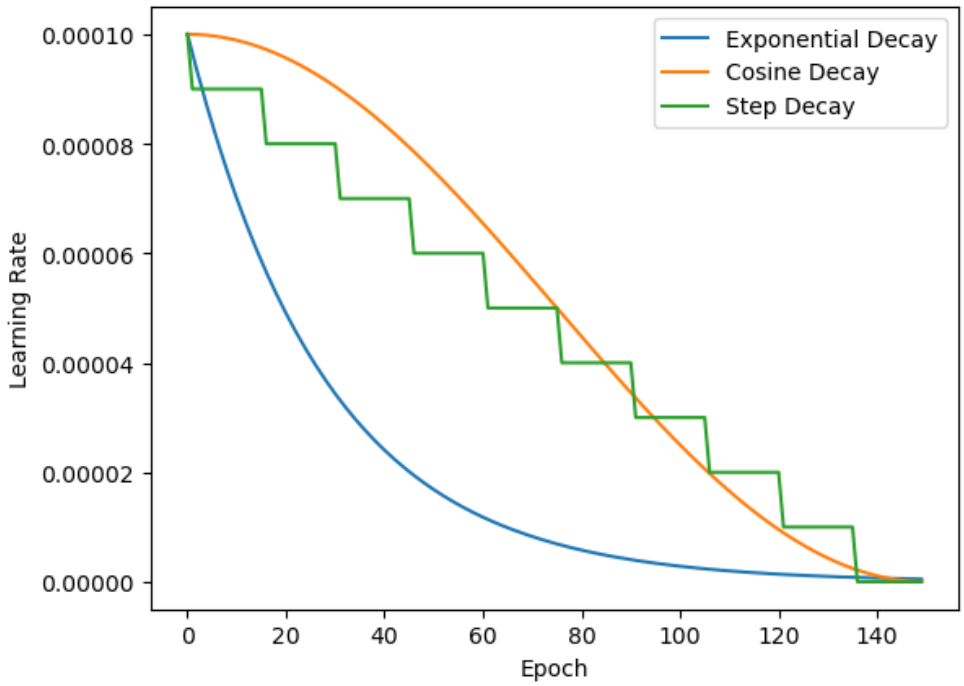

Figure 2.7: Exponential decay, Cosine decay and Step decay schedulers [23] [59].

input image and the convolutional filter are computed with a moving window technique. Here, the convolutional operation is identical to a matrix dot product. Commonly, each convolutional layer consists of multiple convolutional filters [3]. Like basic ANN layers, the trainable parameters in a convolutional layer are the filter coefficients. CNNs, like any other ANN, are trained with the help of an objective function and an optimization method.

The main advantage of choosing CNN models over conventional ANNs is that they consider neighbourhood information of input and learn spatial features of images, texts and speech data [3]. Due to the spatial overlap of convolutional outputs, a CNN conserves this information and carries it through the training session. Another main advantage of CNNs is the exploitation of fewer parameters than a fully connected layer since the convolutional layer outputs multiple inputs to a fewer output [21]. Furthermore, The total size of a convolutional layer output is usually smaller than the input. CNNs have become a favourite choice of model blocks in the literature for as state-of-the-art classification, segmentation, recognition and detection models.

Figure 2.9 shows an overall topology of a CNN. As can be seen, the general architecture of a CNN model is similar to a feedforward ANN. Instead of regular layers, CNNs exploit

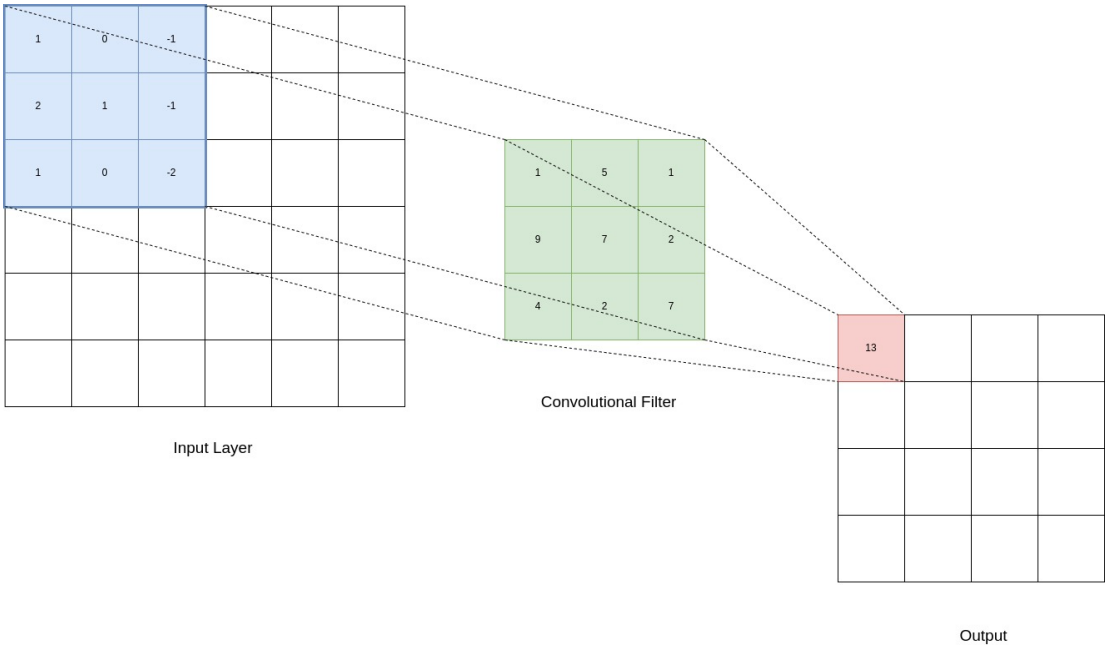

Figure 2.8: An illustration of convolutional block computations. The computational block (marked green) is multiplied elementwise with each same size sub-squares of the input (marked blue), and the pooled summations of this multiplications make the output (red) of the convolutional operation.

consecutive convolutional layers. Also, like ANNs, the output of each convolutional block is passed to an activation function. Typically in the case of classification, after the final convolutional block, the output is flattened and passed to a few fully connected layers to compare the final output to the labels [38] [77] [30]. The section of a classification CNN before the fully connected layer is commonly called "feature extractor", and the remaining fully connected layers are named classification block.

2.3 Related topics in Deep Learning

In the previous section, the basics of ANNs and CNNs were discussed. In this section, deep learning applications related to the thesis will be discussed. First, the definition of content-based image retrieval and its application in digital pathology will be elaborated. Then multiple-instance Learning (MIL), self-supervised learning and contrastive learning will be explained.

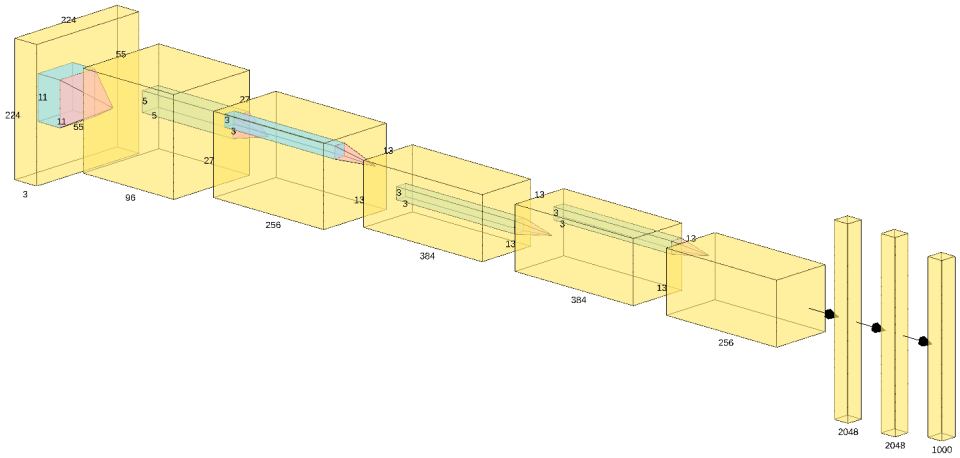

Figure 2.9: an Overall illustration of a CNN model. This shape shows the convolutional computation and the forward path. The final layer of the CNN is then pooled and fed to a number of fully connected layers for prediction and error measurement.

2.3.1 Content-Based Image Retrieval

Content-Based Image Retrieval (CBIR) is the process of searching through databases containing various images that have been indexed before [25] [25]. CBIR systems perform image retrieval depending on the image's content. Extracting semantic information is required for image identification through meaningful indexing. In the context of text-retrieval systems, documents can be broken down into words and compared using word-based characteristics [72]. Digital images are composed of pixels; decomposing them and comparing them on the basis of the pixel-based features may not possible, as two similar images captured from the same scene or object would have different pixel distributions due to natural image modifications. As a result, developing a suitable representation for a digital image is a significant challenge in CBIR. Numerous classical and modern learning algorithms have been developed for this purpose [25].

There are various processes involved in implementing an accurate and efficient CBIR system. To begin, distinct representations of an image should be derived from pixel infor-

mation. These representations are mainly extracted from deep feature extractors [94]. For a retrieval request, the query or queries must first be transformed in some way before they can be compared to images stored in indexed archives. Additionally, relevant similarity metrics should be employed to rank the results before they are displayed to the CBIR user. Additionally, these models must be assessed for improvement. All of these points have been discussed in detail in the following paper. The configuration of a CBIR system is depicted in Figure 2.10.

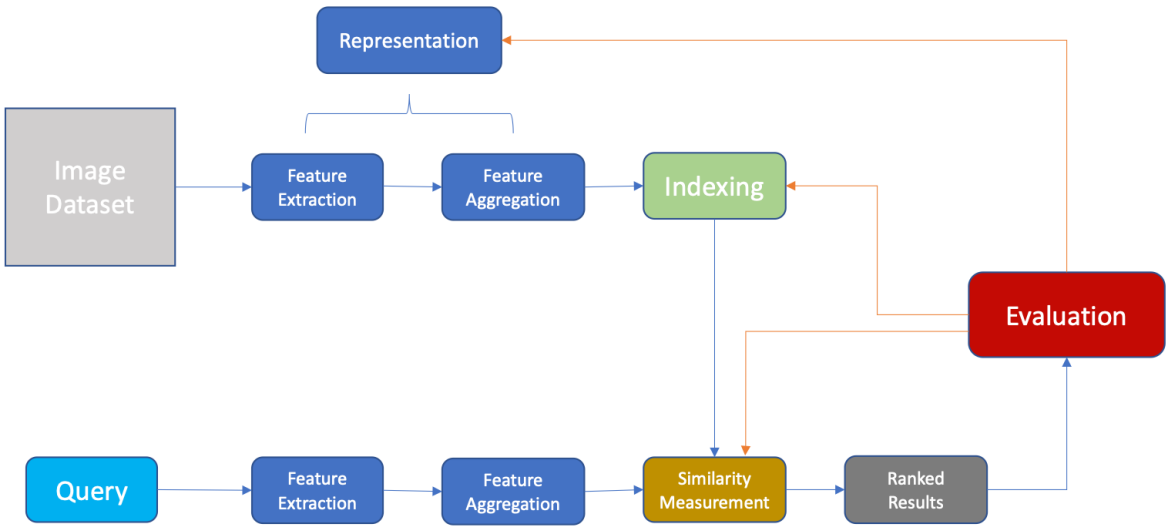

Figure 2.10: A complete workflow of a CBIR system. The image (marked in grey) is passed to some computational blocks for feature extraction (marked in blue). The features are then indexed for archiving (green block). A query image (marked in cyan) goes through the same feature extraction modules as the indexed images in archive. Then similarity of the query and the indexed images are computed (orange block). Subsequently, the results are ranked based on the similarity score from highest to lowest (marked in dark grey). Based on the user (expert) feedback, the feature extractor, similarity measurements and the indexing can be optimized (red block).

To extract reliable features from images, it is common to use previously trained CNNs, and use these features as data points to compare the images. This is one aspect of what

is called "transfer learning" [95]. Zeiler et al. have shown that the features produced from the final convolutional layers (high-level features) contain more semantic information than the features from the starting CNN blocks (low-level features) [92]. As a result, during CBIR feature extraction, high-level image features are extracted to represent the image. To extract more salient features, CNNs should be trained on large image datasets to ensure that the model has had the chance to see many samples containing semantic structures relevant for the application at hand. Then, the classification module of the model is discarded, and the CNN blocks are used as feature extractors [87].

CBIR can become a critical tool in medical imaging, especially in the field of digital pathology [82] [31] [45]. To increase the pathologist's confidence in identifying the tissue characteristics and in cancer diagnosis, it is helpful to compare images of the new patients with the images of previously diagnosed cased in the archive [82].

2.3.2 WSI Representation Learning

As mentioned in the previous passage, for a CBIR model to perform reliable search tasks, it needs to be trained on a set of representative cases. Whole slide images (WSIs) are gigapixel images, meaning they generally have been made of very large dimensions with billions of pixels [66] [49]. It is practically impossible to input gigapixel images directly to CNNs due to their computational cost and hardware bottlenecks. It is quite common to divide a WSI into smaller patches and select a subset of patches to perform classification [37] [45] [12].

Early WSI representation approaches primarily investigated patch-level classification. Hou et al. reported an early classification of WSI slides in 2016 [37]. In this paper, the authors extracted and classified patch-level features with a CNN iterative fashion. The authors first train a CNN with WSI patches. Then they compare the patch prediction with the WSI label and create an intensity map of correct predictions to aid their patch extraction algorithm. They create a histogram of predicted patches in their second stage and compare it with the actual WSI label. Coudray et al. extracted multi-magnification features from 20x and 5x magnifications and aggregated the features with an average of the probabilities of the corresponding patches [13]. Their work mainly focuses on Lung Adenocarcinoma (LUAD) and Lung Squamous Cell Carcinoma (LUSC) slides. Kalra et al. first cluster the entire tissue with colour clustering, then select a small number of patches based from each cluster [45]. They employed patch-level embeddings for WSI search.

2.3.3 Multiple Instance Learning

Multiple instance learning (MIL) is a specific learning scheme where a label is assigned to a bag of instances [16] [91] [39]. In many applications, like digital pathology, there may only be a few available labels for all instances. Hence, it may be more convenient to label multiple objects with a single label. A most common example is an image consisting of multiple objects and task includes detecting and classifying a single object inside the image.However, it is expensive to annotate the whole image and classify it based on those annotations. Therefore, algorithms need to be defined to detect the object among multiple other instances in an image.

MIL is a common approach for the classification and retrieval of WSI images [32] [39] [55]. As mentioned in previous sections, for the classification of WSI images, each slide is first broken down into multiple patches due to computational complexities. This training setup is often translated into a MIL setup, where each patch is considered an instance and the WSI is therefore denoted as a "bag of instances" [16]. In most cases, the features extracted from patches are aggregated into a single representation, and exploited for different representation learning tasks. Figure 2.11 illustrates an example of a multiple instance learning scheme.

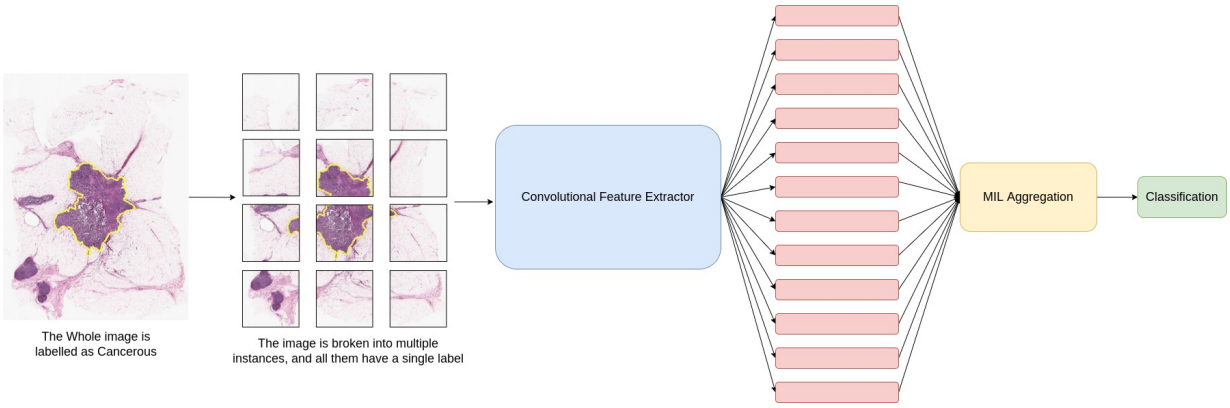

Figure 2.11: The overall illustration of a multiple instance learning setup in the case of digital pathology. The cancerous region is shown with a yellow contour. The WSI is broken down into multiple smaller patches. In the case of MIL, the patches do not have individual labels but all have a single label, namely the WSI label. After CNN feature extraction (blue blocks), the features (marked in red) are aggregated via a MIL technique (yellow block) to make a single representation for all instances. This representation is then used for comparison with the label in a classification setup (green block).

Recently, Zaheer et al. proposed MIL with deep-sets, where they demonstrated that different pooling layers could obtain permutation invariant representations [91]. Permutation invariance is a crucial characteristic in a MIL setup [39]. It suggests that the ordering of the instances should not affect the resulting representation vector. This attribute guarantees that the resulting vector is entirely dependent on the semantics of the instances and not the ordering and positions of instances with relation to each other. In the mentioned paper, the authors use the sum of each instance to produce the final representation and compare the results of the max-pooling of instances, which selects the maximum value of each feature among all the instances [91] .

Following the above paper, many MIL-based WSI representation schemes have been proposed. Ilse et al. proposed attention-based multiple instances learning to perform weighted pooling over each instance feature [39]. Attention models are recently proposed algorithms in deep learning [5]. Purpose of "attention" is to put emphasis on the feature vectors that the model thinks are more critical for the task at hand. In other words, the attention block highlights patches that can contribute more to the task at hand. Compared with a conventional deep-set with average pooling, attention block acts as a weighted average pooling.

Another example of attention-based pooling in MIL is proposed by Kalra et al. where the authors introduced memory networks (MEM) for learning permutation-invariant representations [44]. In another paper, Adnan et al. used graph CNNs to consider each instance as a node in a graph and then learned an adjacency matrix to build a graph representation of WSIs [1]. Just recently, Hemati et al. have exploited deep sets for MIL training in histopathology. They employed a conditional prediction layer where predictions of primary site labels guide the primary diagnosis predictions [32] [91].

2.3.4 Self-Supervised Learning

Self-supervised learning (SSL) refers to deep learning consisting of two stages: Pretext training and downstream (target) training [33]. In the pretext stage, a model is trained on available information that can be extracted from the data itself without any costly human supervision. The trained weights from the pretext training are then utilized for the target task. The intuition behind this approach is to teach the model basic understanding of the input that the model may not achieve in a conventional training stage. For examples, the pretext task propesed by Gidaris et al. is to train a model on different rotations (0, 90, 180 and 270 degrees) of the same image [20]. Each input instance is rotated, therefore the corresponding label is the degree of rotation. The authors then show that various computer

vision tasks such as classification, detection, or segmentation generalize better with self-supervision. Another example of early self-supervision is reported by Doersch et al. [17]. The pretext task in the paper is, given a sample image, to find relative positions between two random patches in the image. It helps the model understand the spatial connection between different image parts and objects, and therefore model will better understand the semantic information of the image.

There have also been some approaches to self-supervision in histopathology literature. In a recent publication, Koohbanani et al. propose two sets of pretext tasks: domain-agnostic and domain-specific tasks [51]. Domain-agnostic pretext tasks refer to a set of general pretext tasks such as rotation, flipping, real/fake prediction and domain prediction. These tasks are not focused on pathology-related characteristics of patches. On the other hand, domain-specific pretext tasks focus on pathology features of images and consist of magnification prediction, JigMag prediction (predicting the correct magnification order of randomly shuffled patches, like a jigsaw puzzle), and hematoxylin channel prediction. Hematoxylin and eosin are two popular colour channels for histopathology images whereas the Hematoxylin channel has a strong correlation with nuclei location and cancer characteristics [51].

To further introduce self-supervised algorithms, first contrastive learning and related optimization methods should be introduced.

2.3.5 Contrastive learning

Contrastive learning (CL) is another active field of research in deep learning where the goal is to pull similar instances together and push the non-related samples away [11] [46]. Training a model with a contrastive loss can help produce a more distinct feature vector for an input. Figure 2.12 illustrates the difference between contrastive learning and conventional supervised learning.

The first usage of a contrastive loss appeared in 2005 [11]. The authors proposed a similarity loss function that maps training data into a target space such that the L_1 norm of the target space imitates the semantic distance of the input space. They considered pairwise input and chose to either push away or pull the samples based on similarity. Thus, the embedding distance between two inputs is minimized when they belong to the same class, but it is increased when they do not. The mathematical formulation of contrastive loss is written as

$$L(x_i, x_j) = \mathbb{I}[y_i = y_j] \, ||f(x_i) - f(x_j)||_2^2 + \mathbb{I}[y_i \neq y_j] \, max(0, \epsilon - ||f(x_i) - f(x_j)||_2)^2, \quad (2.4)$$

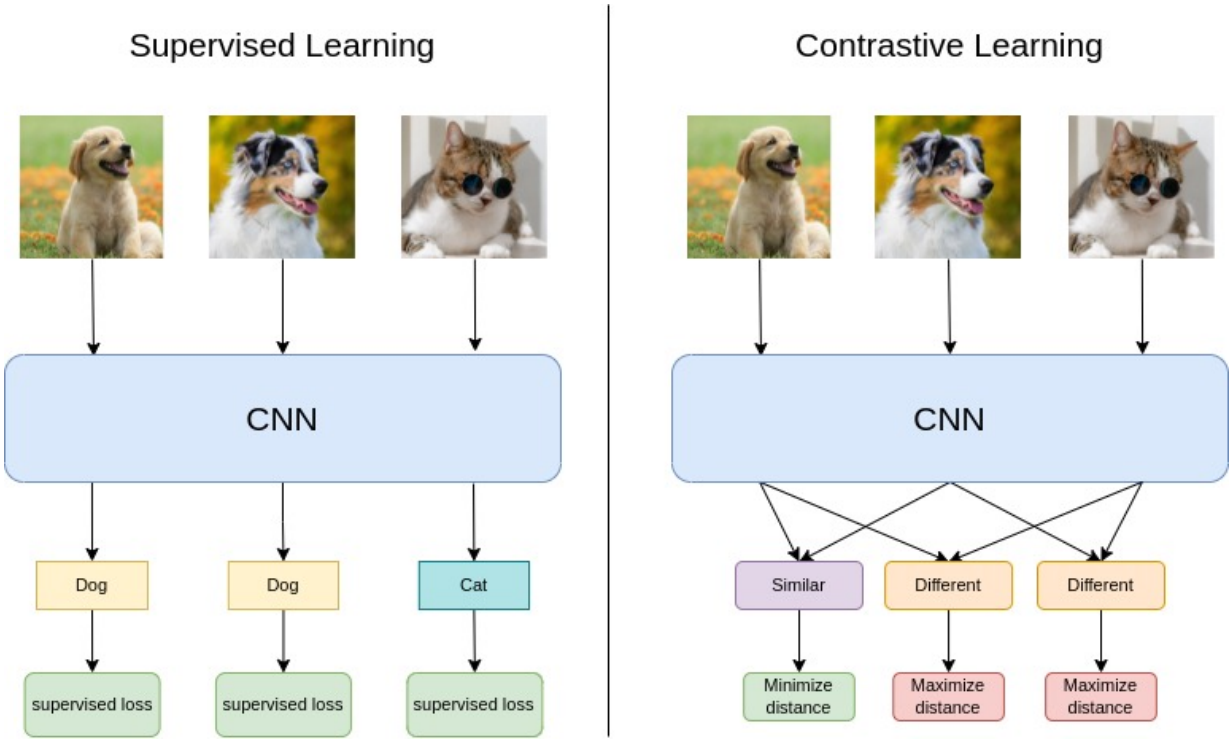

Figure 2.12: A comparison of basic supervised learning and contrastive learning. In contrastive learning, instead of comparing the instances only with their own labels, the distances of different instances with similar labels are minimized, and the negative samples are pushed away in the feature space.

where ϵ is a hyperparameter that controls the distance between negative samples.

In a paper proposed by Hoffer et al., instead of two samples for comparison, authors used one instance as an anchor, one negative and one positive sample for metric learning, simultaneously [36]. In this regard, the loss is written as

$$L(x, x^+, x^-) = \sum_x max(0, ||f(x) - f(x^+)||_2^2 - ||f(x) - f(x^-)||_2^2 + \epsilon). \tag{2.5}$$

For the sake of comparing with multiple negative samples, N-pair loss generalizes the triplet loss hypothesis [78]. They write the contrastive loss function with an anchor, a positive sample, and N-1 negative samples as

$$L(x, x^+, \{x^-\}_1^{N-1}) = \log(1 + \sum_{i=1}^{N} \exp f^\top(x) f(x_i^-) - f^\top(x) f(x^+)). \tag{2.6}$$

Soft-nearest neighbors loss considers multiple positive samples [71] [19]. For a batch of N samples, the loss function is written as

$$L = -\frac{1}{N}\sum_{i}^{N}\log\frac{\sum_{j\neq i, y_i=y_j}\exp(-f(x_i,x_j)/\tau)}{\sum_{j\neq i}\exp(-f(x_i,x_j)/\tau)}, \tag{2.7}$$

where f is a function that measures similarity, and τ is a hyperparameter called temperature that defines the amount of concentration of positive samples in the latent space (feature space).

Finally, a loss function that utilizes multiple positive and negative samples in a batch used in this thesis is supervised contrastive learning [46]. The authors suggested a fully supervised contrastive loss that draws all clusters of points belonging to the same class together while pushing clusters of samples from other classes apart. Given I as a set of indices of a batch, supervised contrastive loss is written as

$$L = \sum_{i\in I} -\frac{1}{|P(i)|}\sum_{p\in P(i)}\log\frac{\exp(z_i\cdot z_p/\tau)}{\sum_{a\in A(i)}\exp(z_i\cdot z_a/\tau)}, \tag{2.8}$$

where $z_i \equiv Proj(E(i))$, $E(i)$ is the output of a encoder block, $Proj(\cdot)$ is a projection function (a fully connected layer in implementation), $A(i) \equiv I \setminus i$ and $P(i) \equiv \{p \in A(i)|y_p = y_i\}$. As t can be seen, supervised contrastive loss is a generalization of soft-nearest neighbors loss.

In most recent papers, CL is implemented in a self-supervised fashion. Chen et al. propose SimCLR (a simple framework for contrastive learning of visual representations) that uses different augmentations as positive samples and any other samples in the batch as a negative [10].

Another approach close to SimCLR is BYOL (bootstrap your own latent) [24]. In BYOL, two different images that can be different augmentations are fed to two models to maximize the agreement between the two outputs. The exciting fact about BYOL is that it only considers positive samples.

As a pathology example, Ciga et al. employed SimCLR and achieved promising results, compared to baseline training methods, for multiple histopathology downstream tasks, including classification, regression, and segmentation [12]. Another recent pathology example is introduced in [55]. The authors perform contrastive learning on different magnification levels separately. Then they create hierarchical representation based on combined magnifications in the downstream tasks.

Chapter 3

Methodology

This thesis proposes a novel end-to-end WSI level self-supervised approach that exploits anatomic site (organ) classification as the pretext task.

The anatomic site (primary site) information corresponds to the organ type of each tissue sample and its corresponding digital slide which is always available for each WSI, i.e., it is always known if a digital slide is extracted from sites such as the brain, lung or breast. Therefore, the model is first trained on an anatomic site classification task. One can show that using the primary site information for the pretext task helps the model generalize better on the primary diagnosis classification.

Another contribution of this thesis is the exploitation of supervised contrastive learning in a MIL setup to generate more robust and distinguishable representations for classification and, specifically, for image search.

The following section provides a step-by-step explanation of the proposed method. The model architecture is broken down to illustrate the function of each module. Then, the experimental results will be reported and discussed. The complete methodology is depicted in Figure 3.1.

3.1 Proposed Methodology and Architecture

This section goes through a step-by-step explanation of the methodology and the model architecture. Figure 3.2 shows the overall architecture of the deep learning model trained and used for image search.The proposed concept is named “SS-CAMIL” which stands for

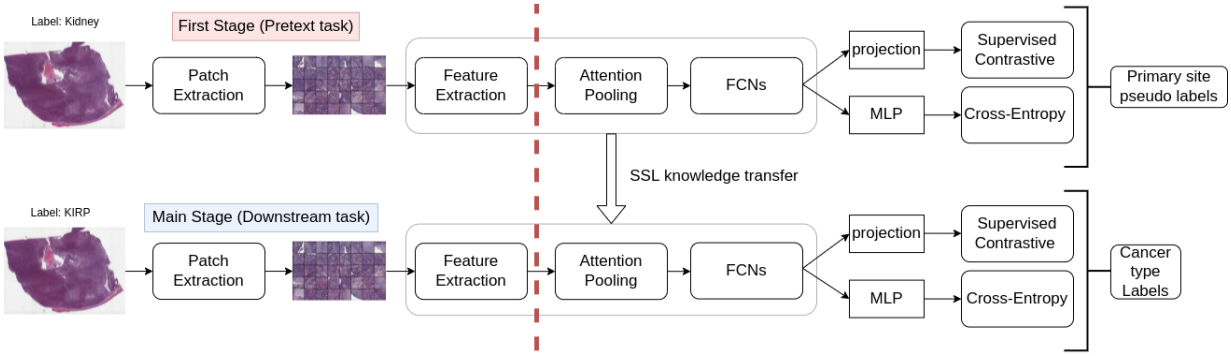

Figure 3.1: The proposed SS-CAMIL concept. The blocks that the transferred knowledge of pretext task (e.g., for label "kidney" as the primary site) are used for the downstream task (e.g., for label "KIRP", *kidney renal papillary cell carcinoma*, as the primary diagnosis), outlined with a grey line. For the LUAD/LUSC classification task, only the blocks on the right side of the dashed red line are used when only pre-trained features will be used.

self-supervised contrastive learning with attention-based multiple instance learning. Furthermore, "CAMIL" is an abbreviation for *contrastive learning with attention-based multiple instance learning*.

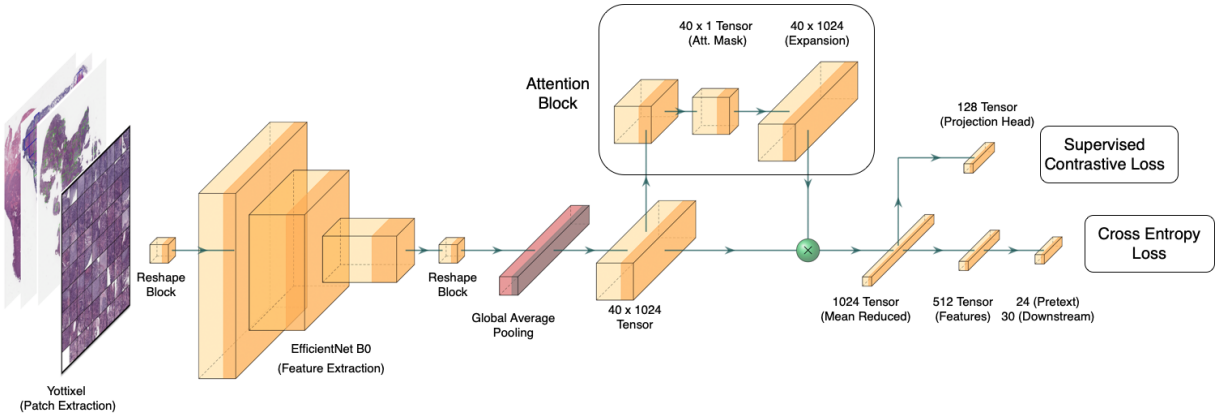

Figure 3.2: The complete model architecture

3.2 Patch Selection

As mentioned in the related work section, in order to process a histopathology WSI for a deep learning task, it is conventional to break it down into smaller patches [37] [45] [12] [55]. There exist different methodologies for extracting valuable patches from a WSI. The common exhaustive method is to select all the patches from a WSI, i.e., include all patches for processing [31].

In this thesis, for extraction of the histopathology patches, the patch selection method in Yottixel was selected [45]. The patch extraction approach is depicted in Figure 3.3. Yottixel utilized a two-step k-mean clustering. The tissue is grouped in the first step based on its colour histogram. The patch groups extracted from the first step are then subjected to a second k-means clustering based on patch location to select spatially varied patches from each colour segment. After that, random patches are selected from each cluster. Therefore, each patch represents a different WSI location and colour. As a result, more regions of a WSI are likely to be considered during training. It should be mentioned that the patches used in this paper are x40 level patches in the size of $3 \times 224 \times 224$.

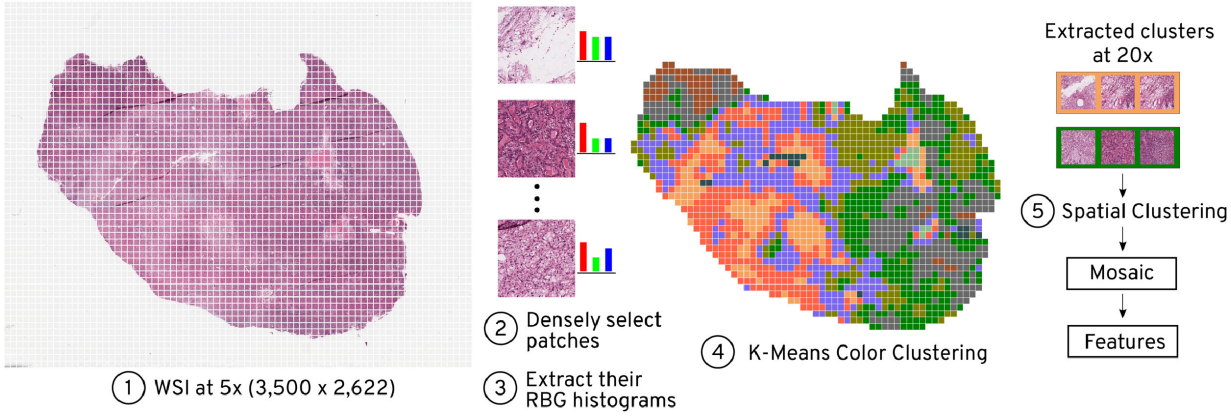

Figure 3.3: The pipeline for Yottixel patch extraction. Taken from [44].

3.3 Feature Extraction

After the patches are extracted, they must pass through a Convolutional Neural Network to extract distinctive feature vectors. If the batch size is denoted as b, the number of patches as n, and the width and height of input with w and h, respectively, the input

Table 3.1: EfficientNet Comparison.

Model	Top-1 Accuracy	Top-5 Accuracy	#Params	Ratio to EffNet
EfficientNet B0 [80]	76.3%	93.2%	5.3M	1×
ResNet-50 [30]	76.0%	93.0%	26M	4.9×
DenseNet-169 [38]	76.2 %	93.2 %	14M	2.6×

batch dimensionality to the model would be $b \times n \times 3 \times w \times h$. The number 3 indicates that coloured images have been used with three colour channels (Red, Green and Blue). All common CNNs get inputs with a dimension of 4, i.e., $b\times$ number of channels $\times w \times h$. Hence, one first needs to modify the patch order in this phase to feed the input image into the feature extractor block. The reshape layer indicated in Figure 3.2 is implemented in this regard. It changes each input from the shape $(b, n, 3, w, h)$ to $(b \times n, 3, w, h)$.

The patches are now inputted into a model for feature extraction. The model selected in this thesis is **EfficientNet B0** [80]. The reason for this particular feature extractor is that it uses fewer model parameters compared to other state-of-the-art feature extractors like ResNet and DenseNet [30] [38]. Table 3.1 illustrates the number of parameters in EfficientNet B0 compared to a ResNet-50 and DenseNet-169, and their performance based on results reported by Tan et al. [80]. It can be seen that with almost one-tenth of the baseline parameters, EfficientNet-B0 shows better or on par performance on ImageNet dataset [14]. The complete comparison of different EfficientNet variations are reported in EfficientNet paper [80].

The features from the final convolutional block have the size of $b \times n \times 1280 \times 8 \times 8$, if inputs are of size $3 \times 256 \times 256$. So each feature tensor has the size of $1280 \times 8 \times 8$. To change the features to a single 1-D feature vector, the basic features are fed to a global max-pooling layer and a fully connected layer to extract vectors of size 1,024 for each patch. The reason to do so is that $1,280 \times 8 \times 8 = 81,920$ is still too long to be practical and would drastically increase the time and computational resource requirements. Another reshape layer is then utilized to convert the output shape to $(b, n, 1024)$ to be able to be fed to a MIL aggregation module.

3.4 Attention-Based Pooling

As displayed in Figure 3.2, the feature vectors serve as the input to an attention block. There exist various ways for MIL aggregation, and as mentioned in the related works, two

of such aggregation methods are deep sets and attention-based pooling [91] [39]. Figure 3.4 illustrates the comparison between the deep sets' average pooling and attention pooling. As it can be observed, the main difference between the two is that the instances are first multiplied by a trained attention mask before averaging in attention pooling. In other words, attention pooling is a form of weighted averaging with trained weights. Therefore, the model can decide what instances have more important values to emphasize the value in the average.

Two fully connected layers plus an extension layer make up the attention block. The two dense layers produce a mask of size (b, n), which is then duplicated to get a size of $(b, n, 1024)$. Duplication is done, so the mask is in the size of the input instances. This mask is then element-wise multiplied with the attention block input and averaged to generate a 1,024-length vector representation of each WSI.

Instead of a basic average pooling layer, the mask learns the weight of each patch (importance factor) and lets the model pick which patch is more representative of the WSI. Ilse et al. showed that the representation of attention-based pooling is permutation-invariant, meaning that the output does not change when the input patches are reordered, hence establishing a significant degree of freedom for patch selection [39].

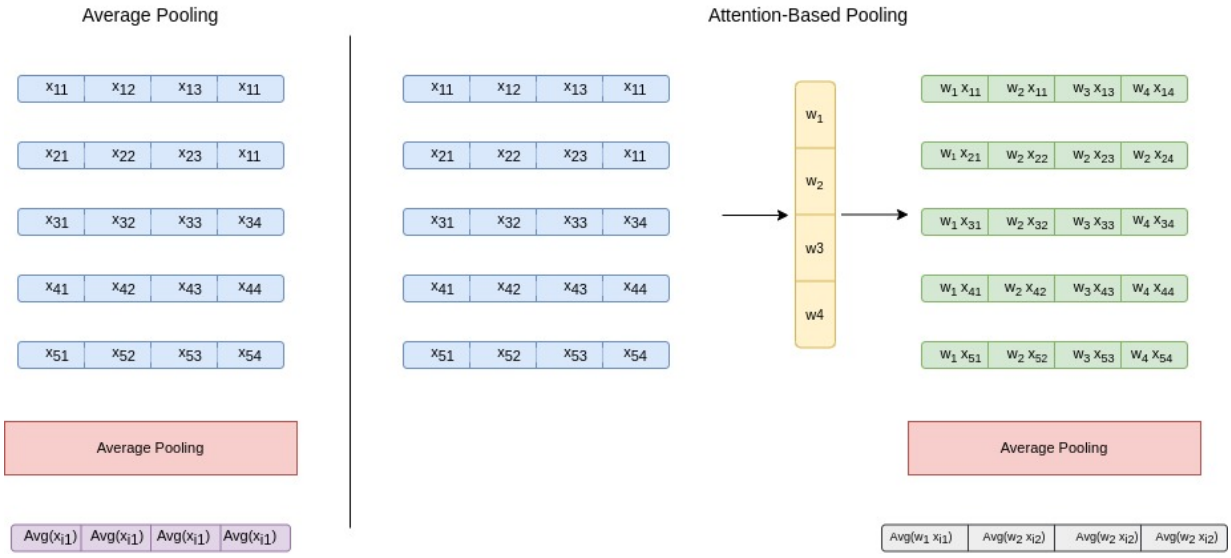

Figure 3.4: Comparison of average pooling and attention-based pooling. In average pooling, each input index (blue blocks) are averaged among different instances. In attention-based pooling, each feature is multiplied by trainable weights first (yellow block).

3.5 Self-Supervision and Contrastive Learning-Based on Primary Site Information

The main contribution of this thesis is to introduce the exploitation of **primary site information as pseudo-labels** in a self-supervised learning setup (the first training stage). Previous SSL methods in pathology used data augmentation-based self-supervision as pretext tasks. Primary site information of a WSI is a piece of information that is always available and can be used as a pseudo-label. This information basically indicates the original anatomical organ that the tissue had been extracted from. Since it is always apparent where the tissue has come from, it is considered it a readily available piece of information in this thesis.

Table 3.2 shows the tumor subtypes considered in this thesis. As it can be observed, 6,746 WSIs used in the study are from 30 types and 22 anatomic sites. There has been reported usage of the primary site as a known label in previous publications. Hemati et al. utilized the primary site directly in the training setup and only classifies the cancer subtypes for similar anatomic sites [32]. However, to the best of the author's knowledge, using this available information for self-supervision has not been explored in the literature. The second contribution is the utilization of supervised contrastive learning for both pretext and downstream tasks [46].

After extracting WSI feature vectors, the features are passed to a projection head and a contrastive loss based on the primary site labels.

The experiments will show that transferring the primary site information as a self-supervised task improves the performance of the proposed model. To evaluate the impact of self-supervision, experiments were conducted in two phases. First, the results of basic attention-based MIL without self-supervision are reported. This model is called CAMIL, which stands for contrastive learning with attention-based multiple instance structure. Then, the results of primary site self-supervision on CAMIL are reported. The second experiment uses SS-CAMIL, which stands for self-supervised contrastive learning with attention-based multiple instance structure.

It should also be mentioned that compared to all previous patch-based SSL methods, the proposed self-supervision approach is performed **on WSI-level**, which means that instead of using the contrastive loss for every single patch, the loss function for the aggregated slide representation has been used. This way of implementation helps reduce the training and prediction computational cost. It also helps the model to understand that all patches are part of a larger instance, namely the WSI, and these instances represent a semantic whole when put together.

Table 3.2: Tumor types, subtypes and primary sites.

Tumor Type	Subtype	primary site
Gastrointestinal tract	Colon Adenocarcinoma	Colon
	Stomach Adenocarcinoma	Stomach
	Esophageal Carcinoma	Esophagus
	Rectum Adenocarcinoma	Rectum
Pulmonary	Lung Adenocarcinoma	Bronchus and lung
	Lung Squamous Cell Carcinoma	Bronchus and lung
	Mesothelioma	Heart, mediastinum, and pleura
Liver, pancreaticobiliary	Liver Hepatocellular Carcinoma	Liver and intrahepatic bile ducts
	Cholangiocarcinoma	Liver and intrahepatic bile ducts
	Pancreatic Adenocarcinoma	Pancreas
Endocrine	Thyroid Carcinoma	Thyroid gland
	Pheochromocytoma and Paraganglioma	Adrenal gland
	Adrenocortical Carcinoma	Adrenal gland
Urinary tract	Kidney Renal Papillary Cell Carcinoma	Kidney
	Kidney Renal Papillary Cell Carcinoma	Kidney
	Bladder Urothelial Carcinoma	Bladder
	Kidney Chromophobe	Kidney
Brain	Brain Lower Grade Glioma	Brain
	Glioblastoma Multiforme	Brain
Prostate/testis	Prostate Adenocarcinoma	Prostate gland
	Testicular Germ Cell Tumors	Testis
Gynaecological	Ovarian Serous Cystadenocarcinoma	Ovary
	Cervical Squamous Cell Carcinoma	Cervix uteri
	Uterine Carcinosarcoma	Uterus
Breast	Breast Invasive Carcinoma	Breast
Haematopoietic	Thymoma	Thymus
Laryngeal	Head and Neck Squamous Cell Carcinoma	Larynx
Mesenchymal	Sarcoma	Retroperitoneum and peritoneum
Melanocytic malignancies	Skin Cutaneous Melanoma	Skin
	Uveal Melanoma	Eye and adnexa

In practice, one finds out that using CL for a MIL setup has a bottleneck. As mentioned before, contrastive loss tries to increase the similarity of presentation of the same instances and decrease the similarity for negative pairs. One of the necessities of CL is large batch sizes [62]. The reason is that contrastive learning requires multiple positive samples to derive an acceptable representation for the sample. Since smaller batch sizes have lower chances of having multiple positive samples, contrastive learning tends to have a poor performance on small batch sizes. On the other hand, each bag representation in a batch in MIL has multiple instances involved. Suppose the batch size of b and a fixed bag size of n. Therefore, the number of patches that needs to be processed before the MIL aggregator

is $b \times n$. This issue led to the bad performance of contrastive learning in this study. With four NVIDIA Tesla V100 PCIe GPUs with 32 gigabytes of memory and a bag size of 40, the batch size could not be enlarged over 24. However, in the literature [26], batch sizes of more than 256 are recommended for contrastive learning. To overcome this challenge, a cross-entropy term was added to the contrastive loss function. The intuition behind this idea was that since cross-entropy tries to learn a single presentation for each instance, adding a cross-entropy helps the positive instances be close to a specific point in the embedding space. On the other hand, contrastive terms help move the instances close or far from each other. The loss function has the form

$$L = \sum_{i \in I} -\frac{1}{|P(i)|} \Big(\sum_{p \in P(i)} \log \frac{\exp(z_i \cdot z_p / \tau)}{\sum_{a \in A(i)} \exp(z_i \cdot z_a / \tau)} \Big) - \; y_i \, log \, \hat{y}_i \tag{3.1}$$

where $z_i \equiv Proj(E(i))$, $E(i)$ is the output of a encoder block, $Proj(\cdot)$ is a projection function (a fully connected layer in implementation), $y_i \equiv FC(E(i))$, $FC(\cdot)$ is the output of fully connected layers after the attention block, $\hat{y}_i$ is the instance ground-truth label, $A(i) \equiv I \setminus i$ and $P(i) \equiv \{p \in A(i) | y_p = y_i\}$. The experimental results have shown that this loss functions generates robust representations in the latent space.

After the training with the above setting, the model is trained on the downstream task with diagnostic labels (i.e., primary diagnosis). After the training session, the features extracted from the last fully connected layer before the projection head are utilized for WSI search and classification.

Chapter 4

Experiments and Results

This section will discuss the details of the thesis experimentation. Three sets of deep learning experiments have been conducted. The first experiment set focuses on the performance of the model on WSI search. The extracted features will be used as representations to define two sets of image search experiments. This experimental setup will show how self-supervision will improve the latent WSI representation. The Cancer Genome Atlas (TCGA) Program dataset has been utilized as the source of data. TCGA is the largest open-source histopathology dataset [84]. The second experiment set will be conducted on a Lung cancer classification task. In the thesis, the pre-trained weights of the previous step is utilized to leverage the self-supervision information for this classification task. Finally, the impact of attention-based MIL on experimental results will be reported.

In the first three sets, first, the training setup is explained. Then the datasets and numeric details of hyperparameters used in the training setup is discussed. Finally, with the help of tables and figures, the performance of the proposed model compared to baseline methods .

4.1 WSI Search Results

WSIs from The Cancer Genome Atlas Program (TCGA) were used. TCGA is a joint project between NCI and the National Human Genome Research Institute. In this project, TCGA generated over 2.5 petabytes of genomic, epigenomic, transcriptomic, and proteomic data [84]. Over 20,000 original cancer and matched normal samples from 33 different cancer types have been molecularly characterized by TCGA. TCGA repository now holds

70 Projects, 67 Primary sites and 85,415 different cases, as reported in TCGA official repository. Since the experimental setup for image search is close to CNN-DS paper, the same subset of TCGA as in the mentioned paper was used [32]. Here, 6,746 WSIs from TCGA is utilized an 85, 5, and 10 percent of the dataset is used for training, validation, and testing, respectively. The dataset consisted of WSIs of 24 primary sites with 30 distinct primary diagnoses. In the training stage, the batch size is set to 16, and the WSI set size to 40. Patches of sizes 1000×1000 are extracted using the patching algorithm proposed in Yottixel paper and resized them to 224×224 [45]. The reason for resizing the patches is mainly due to memory limits (downsampling patches is quite common in literature [83, 61]).

It is common to use data augmentation to help the deep learning models generalize better on the test dataset (i.e., have better understanding and accuracy on the dataset). Data augmentation is the practice of changing and extending the data in each epoch iteration [76]. Hence, the model grasps the valuable information from each instance instead of shortcuts and unrelated info in the image. For data augmentation, horizontal and vertical flip, 90-degree rotation, shifting, and scaling is applied to the data from the Albumentations library [8]. Multiple positional augmentations is used for the dataset. All this augmentation is random and has a 50 % chance of occurring. These positional augmentations help the model not get distracted by positional information such as rotations and flips and focus more on the semantic image information.

We have also used a learning rate scheduler for the training setup. As mentioned in the related work section, learning rate is a hyperparameter that modifies the learning step sizes. Adjusting the learning rate correctly can help the model converge better to its global optimum. In this thesis, an exponential decay learning rate scheduler is utilized. The exponential learning rate is computed as

$$ilr \times b^e, \tag{4.1}$$

where ilr is the initial learning rate, b is the exponential base, and e is the epoch number. The illustration of a learning rate scheduler with exponential decay can be seen in the Figure 4.1. As the exponential base is increased, it can be observed that learning rate drops faster. After trying multiple bases for this experience, the exponential decay with a base of 0.96 and a coefficient of 0.0001 is used.

Each of the presented results is trained with 150 epochs utilizing three Tesla V 100 GPUs in parallel mode. In the related works section, it is alos explained that temperature in contrastive learning defines the punishment of negative samples being near the positive anchor. Based on experiments by Wang et al., the temperature is set to 0.1 for contrastive learning in both pretext and downstream tasks [88].

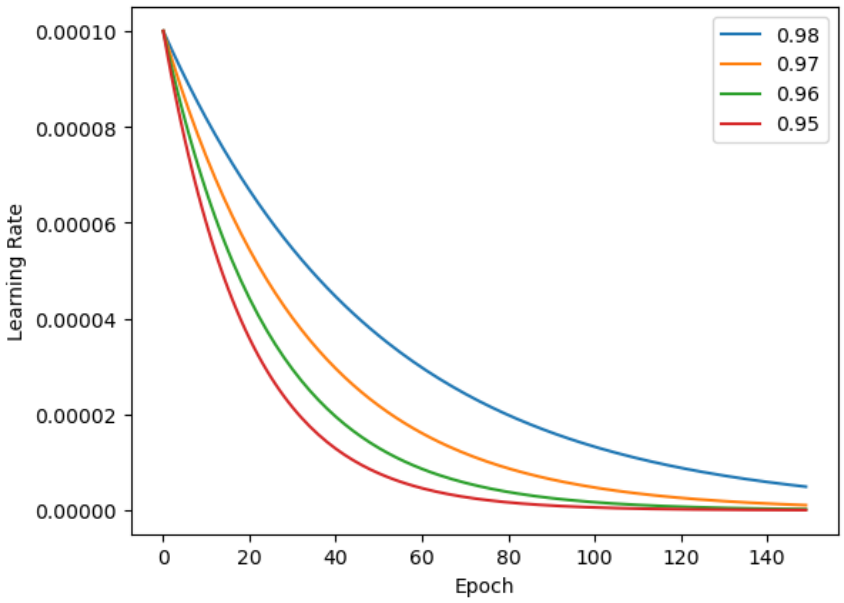

Figure 4.1: Exponential learning rate decay with different exponential bases.

For testing, horizontal (site identification) and vertical (subtype identification) WSI search tasks are established. The precision with which a tumour type can be located across the entire test archive is referred to as "horizontal search". The tumour type labels are not available through the training process, so it measures the model's ability to search and find unknown tumours. On the other hand, "vertical search" measures how well the model can identify the proper cancer subtype of a tumour type from a set of slides from a single primary site, which may have a variety of initial diagnoses. The subtypes that is used in the vertical search for evaluation are the labels that are fed to the model in the downstream task. Also, in vertical search, the subtypes are compared and evaluated in their tumour type group. Hence the tumour types with only one subtype are omitted in the vertical search task. For both search tasks, k-NN algorithm with $k = 3$ is employed to find the three instances closest to each test sample. For the results, the **leave-one-out technique** is employed, (leave-one-WSI-out, and compared it with the other slides and provide the average scores across the slides).

Tables 4.1 and 4.2 show the horizontal and vertical search results, respectively. The performance of the model is compared with Yottixel and CNN-DS [45] [32]. In both tables, CAMIL is the baseline attention-based MIL with CL and without self-supervision, and SS-CAMIL is the same as CAMIL setup but uses the weights of self-supervision of primary

sites.

Table 4.1: Horizontal Search Results. F1-scores of Majority-3 (in %) are reported.

Tumor type	n_{slides}	Yottixel	CNN-DS	**CAMIL**	**SS-CAMIL**
Brain	46	73	91	100	100
Breast	77	45	77	91	91
Endocrine	71	61	66	86	89
Gastro.	69	50	75	84	86
Gynaec.	18	16	33	56	62
Head/neck	23	17	69	74	92
Liver	44	43	56	77	84
Melanocytic	18	16	50	61	78
Mesenchymal	12	8	100	92	92
Prostate/testis	44	47	81	91	89
Pulmonary	68	58	91	81	87
Urinary tract	112	67	76	92	95
Haematopoietic	42	0	24	50	50

We can observe that the SS-CAMIL model has the best results among the four setups in 10 out of 13 cases for horizontal search. CAMIL is the dominant model in one of the remaining three cases (Prostate/Testis). In the rest of the tumour types, SS-CAMIL has shown competitive results. One of the interesting observations is that although CNN-DS utilizes primary site information as prior information for the classification of tumour subtypes, the results of CAMIL are better in most cases. This observation demonstrates the effect of attention-based pooling compared to simple average pooling. Another observation is the improvement in performance with self-supervision on the primary sites.

It can be observed that in most tumour types, SS-CAMIL has performed better than CAMIL. This observation indicates that the primary site can help the model generalize better when deciding on the subtypes within the self-supervision framework. Having better performance on horizontal search means how well the model can identify the unknown parent tumour type better. It justifies that the features extracted with the SS-CAMIL method have an overall better representation of the slides than other methods.

For the case of vertical search, in 14 subtypes of a total of 24 distinct subtypes, SS-CAMIL achieved the best F1-score. As mentioned before, some of the subtypes such as

Table 4.2: Vertical Search Results. F1-scores of Majority-3 (in %) are reported.

Tumor Type	Subtype	n_{slides}	Yottixel	CNN-DS	**CAMIL**	**SS-CAMIL**
Gastrointestinal tract	COAD	22	62	69	72	73
	STAD	27	61	64	79	92
	ESCA	10	12	44	55	89
	READ	10	30	55	26	0
Pulmonary	LUAD	30	62	61	71	76
	LUSC	35	69	60	76	75
	MESO	3	0	50	50	33
Liver, pancreaticobiliary	LIHC	32	82	95	95	95
	PAAD	8	94	94	94	94
	CHOL	4	26	0	0	0
Endocrine	THCA	50	92	98	99	100
	PCPG	15	61	81	86	90
	ACC	6	25	28	50	77
Urinary tract	KIRP	25	75	84	84	88
	KIRC	47	91	87	92	92
	BLCA	31	89	95	94	98
	KICH	9	70	53	88	80
Brain	LGG	23	78	89	91	89
	GBM	23	82	89	91	90
Prostate/testis	PRAD	31	98	97	94	100
	TGCT	13	96	93	96	100
Gynaecological	OV	9	80	82	76	80
	CESC	6	92	66	44	44
	UCS	3	75	80	100	50

Breast Invasive Carcinoma, Thymoma, Head and Neck Squamous Cell Carcinoma, Sarcoma and Skin are the only subtypes in their tumour types. Therefore, these subtypes are not included in the vertical search task.

For five subtypes, CAMIL has performed better. In the cases when both CAMIL and SS-CAMIL have poorer performance than CNN-DS and Yottixel, small sample sizes seem to be a recurrent pattern, meaning that the model did not have the chance to learn distinct features from these subtypes. It demonstrates that the size of the train data has a significant effect on the training setup and that when more data is available, the proposed feature extractor model in the thesis performs better. Again, here it can seen that in 11 subtypes, self-supervision has helped the model perform better than CAMIL. This result suggests that teaching the model the primary site information before a downstream task can significantly help the model generalize better.

The Effect of Contrastive Learning on representation To show the effect of contrastive learning, the 2-dimensional t-SNE plot of CNN-DS, CAMIL and SS-CAMIL is shown in 4.2 [34] [32]. The word t-SNE stands for t-distributed stochastic neighbour embedding, and is a method that maps high dimensional data points to a 2-dimensional or 3-dimensional space. This method was introduced in 2002 by Hinton et al. and is currently a popular approach for visualizing high dimensional spaces such as convolutional feature spaces [35]. It can be observed that CAMIL and SS-CAMIL clusters are tighter and more separable than CNN-DS. This improvement is one of the main reasons CAMIL and SS-CAMIL act better in the image search tasks.

4.2 Lung Cancer: LUAD/LUSC Classification

In another experiment, the prospoed model is employed on Lung Adenocarcinoma (LUAD) and Lung Squamous Cell Carcinoma (LUSC) classification task [85]. Lung carcinomas are among the most aggressive cancers, with the most significant fatality rate worldwide. Among non-small cell carcinomas, lung squamous cell carcinoma (LUSC) and adenocarcinoma (LUAD) account for most lung cancers. Non-small cell lung cancers (NSCLCs) are often treated with surgery initially, and chemotherapy and radiation stay the alternative choice for these cancer types. Patients diagnosed with NSCLCs frequently experience relapse, metastasis, and death [4].

LUAD and LUSC appear to be quite diverse in terms of prognosis [81]. Notably, they are regarded as different clinical cancer subtypes. LUAD is more dominant in non-smokers

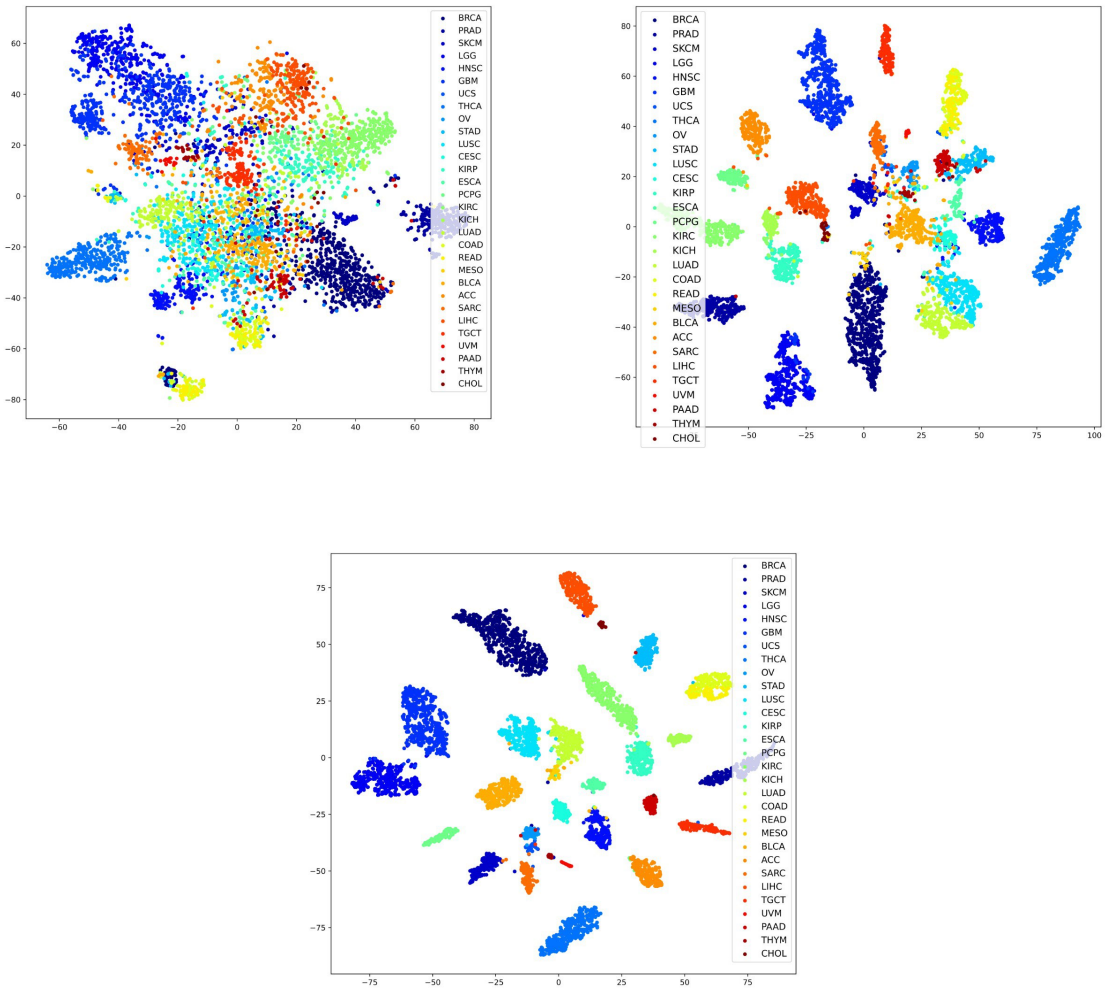

Figure 4.2: t-SNE of CNN-DS [32] (Taken from the paper) (top left) and CAMIL (top right) and SS-CAMIL (bottom).

sc

Table 4.3: LUAD/LUSC classification.

Method	Accuracy
Yu et al. [90]	75%
Khosravi et al. [47]	83%
MEM [44]	84%
Coudray et al. [13]	85%
CNN-DS [32]	86%
CAMIL	88%
SS-CAMIL	**89**%

than smokers. However, it is also reported in smokers. Typically, the tumour is more peripherally placed and grows more slowly than the other forms, albeit it is more prone to metastasis in the early stages of the disease. LUSC is the second most frequent type of lung cancer in cigarette smokers. It is highly related to smoking-induced airway lesions [9]. Therefore, LUAD and LUSC must be investigated to develop effective diagnosis and therapeutic intervention.

The dataset had 2,574 lung tissues taken from the TCGA repository. LUAD/LUSC classification is a challenging classification task that requires visual inspection of the tissue by expert pathologist [13]. In this setup 1,800 slides is used for training, and 774 slides is utilized for test [44]. For training, the convolutional feature extraction block is frozen (i.e. is not trained) to demonstrate the learned features from the previous setup. The batch size and the set size are the same as in the above setup.

The results of LUAD/LUSC classification are shown in 4.3. The suggested strategy outperformed earlier LUAD/LUSC classification approaches by 2% (delivering 88 %), which underlines the performance of attention-pooling and contrastive learning. The SS-CAMIL blocks are also employed in this task, improving the performance to 89%. This also suggests that knowing the primary site information before classifying can help the model identify distinguishing cancer type features.

4.3 Attention Pooling Effectiveness

In the final set of experiments, The effectiveness of the attention-pooling layer is investigated. Results are compared with the conventional average pooling, and to do so, nine random WSIs from Lung, Kidney, and Brain organs from the TCGA repository are chosen.

Table 4.4: Attention pooling scores of 9 different WSIs.

Weighting	Lung			Kidney			Brain			Avg
	1	2	3	1	2	3	1	2	3	
Uniform	0.97	0.89	0.80	0.89	0.70	0.89	0.94	0.87	0.88	0.87
SS-CAMIL	0.98	0.90	0.83	0.91	0.79	0.91	0.96	0.86	0.90	0.89

A pathology expert scored the effectiveness of all 40 patches from each WSI with labels 1,2, and 3, meaning "not useful", "somewhat useful", and "very useful", respectively. The normalized scores and the output of the attention block are multiplied for each WSI and the results are compared with uniform importance (with all patches having the same weight). The scores are then divided by the optimal importance (weights of patches are proportional to effectiveness label) scores to get normalized numbers. The final formulation of the score for slide j can be given as

$$score_j = \frac{\sum_{i=1}^{40} p_i \times e_i}{\sum_{i=1}^{40} \hat{e}_i \times e_i}, \tag{4.2}$$

where p_i is the pooling layer output, e_i is the evaluation number that has a value of 1, 2 and 3, and and $\hat{e}_i$ is the normalized value of evaluations with respect to the whole evaluation vector for slide j. In writing the described vector, it is considered that the inner product of two vectors have a direct correlation to the similarity of two vectors, ans this theorem is the basis for a similarity measure called "Cosine Similarity" measure.

The results are shown in 4.4. This indicates that the proposed model has better overall scores, suggesting that CAMIL has learned the relative importance of patches in the attention block and emphasizes the patches that are more related to the cancerous region. A generalization of this attention block can be utilized in future works to extract patches that correspond more to cancerous spots.

Chapter 5

Summary and Conclusions

5.1 Summary

In this thesis, the effectiveness of a self-supervised learning method in digital pathology based on anatomic site (organ) labels of WSIs was investigated. Anatomic site labels are readily available for each glass slide, and hence for each WSI, since the originating organ of a pathology slide is always known in laboratory settings. The primary site labels were used as *pretext pseudo-labels* for training to exploit the learned weights as a starting point for classifying various cancer subtypes.

Because pathology slides are considered big image data and pixel-level or regional annotations are costly to generate, a multiple instance learning framework for training was selected as a better choice to avoid these challenges. Comparing the most successful multiple instance learning models in the literature, this thesis put forward an attention-based pooling block for feature aggregation. Considering the most common self-supervised learning schemes in deep learning literature, a fully supervised contrastive learning loss function was employed as well. Since multiple instance learning methods consume considerable memory, and contrastive learning is batch-size dependent, this thesis introduced a loss function combining cross-entropy and supervised contrastive loss.

Four sets of the experiments were conducted in this thesis. A model was trained on 6000+ WSIs from the TCGA public repository in the first set of experiments. Using the trained model, two retrieval tasks, namely horizontal and vertical search, showed the performance of learned features as image representation. When enough image data is available, using the proposed CAMIL and SS-CAMIL has a superior performance for image

search tasks. Also, with the help of t-SNE plots, it was illustrated how contrastive learning contributes to WSI representations.

In the second set of experiments, the trained weights were transferred from the previous self-supervision tasks to aid in lung cancer LUSC/LUAD classification tasks. It was demonstrated that using the pretext weights can elevate the performance of a single site (organ) classification task such as lung cancer.

In the final set of experiments, the effectiveness of the attention-based aggregation block was verified and showed to be contributing to the image understanding in terms of relative patch importance.

5.2 Conclusions

Computational pathology is a fast-growing subfield of medical imaging that is currently attracting scientists from computer and medical sciences. Many areas related to this topic have the potential for exploration. The exploitation of primary pathology information and structure for auxiliary training has been seldom considered. Only a few papers have exploited pathology-specific information for deep learning tasks.

Self-supervised learning has proven to be an effective way to transfer histopathology information to deep learning models. Self-supervision can also be adequate for tasks other than classification, such as segmentation, object detection and text understanding. There is more information such as different magnifications of a WSI and various staining methods that can be formulated for pathology self-supervised learning tasks.

5.3 Potential Directions

There are other ways to share different training stages and information of multiple distinct tasks. Recently, a method has been introduced called *Deep Mutual Learning* that shares the trained information of multiple training stages in a single training setup [93]. Therefore, another possible approach would be testing the performance of deep mutual learning on the training of primary site information and a target task.

Multi-task learning is the practice of training a single network with two or more deep learning tasks, such as segmentation, depth detection, classification and detection, simultaneously [70]. Multi-task learning is another option for transferring pathological information in a training setup.

Contrastive learning is becoming a popular field in computational pathology. Although contrastive learning helps the convolutional networks produce excellent results, it tends to harm the semantic structure of the model's latent space if not used properly [88]. In other words, in a sample classification datasets, LUSC and LUAD should be split, but the distance of LUAD and LUSC samples should be less than their distance from a cancer subtype outside of the Bronchus and Lung region.

Exploiting the hierarchical information of slides (anatomic site, tumour types, subtypes, etc.) can help contrastive learning simultaneously create more useful structural features. An alternative would be rewriting contrastive learning formulation to consider these hierarchical pieces of information from a WSI.

Although multiple instance learning has been introduced and utilized very early since the introduction of deep learning, new modifications can be done on the aggregation scheme of the MIL models. Feature aggregation modules can be modelled such that the spatial correlation and magnification information of patches are preserved. Also, different methods than averaging can be explored for aggregation.

References

[1] Mohammed Adnan, Shivam Kalra, and Hamid R Tizhoosh. Representation learning of histopathology images using graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 988–989, 2020.

[2] Shaimaa Al-Janabi, André Huisman, and Paul J Van Diest. Digital pathology: current status and future perspectives. *Histopathology*, 61(1):1–9, 2012.

[3] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)*, pages 1–6. Ieee, 2017.

[4] Dorota Anusewicz, Magdalena Orzechowska, and Andrzej K Bednarek. Lung squamous cell carcinoma and lung adenocarcinoma differential gene expression regulation through pathways of notch, hedgehog, wnt, and erbb signalling. *Scientific reports*, 10(1):1–15, 2020.

[5] Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

[6] Bruce A. Beckwith. *Standards for Digital Pathology and Whole Slide Imaging*, pages 87–97. Springer International Publishing, Cham, 2016.

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

[8] Alexander Buslaev, Vladimir I Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A Kalinin. Albumentations: fast and flexible image augmentations. *Information*, 11(2):125, 2020.

[9] Joe W Chen and Joseph Dhahbi. Lung adenocarcinoma and lung squamous cell carcinoma cancer classification, biomarker identification, and gene expression analysis using overlapping feature selection methods. *Scientific Reports*, 11(1):1–15, 2021.

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[11] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005.

[12] Ozan Ciga, Tony Xu, and Anne Louise Martel. Self supervised contrastive learning for digital histopathology. *Machine Learning with Applications*, page 100198, 2021.

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

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

[15] Li Deng and Yang Liu. *Deep learning in natural language processing*. Springer, 2018.

[16] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1-2):31–71, 1997.

[17] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. *CoRR*, abs/1505.05192, 2015.

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

[19] Nicholas Frosst, Nicolas Papernot, and Geoffrey Hinton. Analyzing and improving representations with the soft nearest neighbor loss, 2019.

[20] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018.

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

[22] Marco Gori and Alberto Tesi. On the problem of local minima in backpropagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(1):76–86, 1992.

[23] Akhilesh Gotmare, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation. *arXiv preprint arXiv:1810.13243*, 2018.

[24] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020.

[25] Venkat N Gudivada and Vijay V Raghavan. Content based image retrieval systems. *Computer*, 28(9):18–22, 1995.

[26] Michael U Gutmann and Aapo Hyvärinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of machine learning research*, 13(2), 2012.

[27] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[28] Soufiane Hayou, Arnaud Doucet, and Judith Rousseau. On the impact of the activation function on deep neural networks training. In *International conference on machine learning*, pages 2672–2680. PMLR, 2019.

[29] Fengxiang He, Tongliang Liu, and Dacheng Tao. Control batch size and learning rate to generalize well: Theoretical and empirical evidence. *Advances in Neural Information Processing Systems*, 32, 2019.

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

[31] Narayan Hegde, Jason D Hipp, Yun Liu, Michael Emmert-Buck, Emily Reif, Daniel Smilkov, Michael Terry, Carrie J Cai, Mahul B Amin, Craig H Mermel, et al. Similar image search for histopathology: Smily. *NPJ digital medicine*, 2(1):1–9, 2019.

[32] Sobhan Hemati, Shivam Kalra, Cameron Meaney, Morteza Babaie, Ali Ghodsi, and Hamid Tizhoosh. Cnn and deep sets for end-to-end whole slide image representation learning. In *Medical Imaging with Deep Learning*, 2021.

[33] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. *Advances in Neural Information Processing Systems*, 32, 2019.

[34] G Hinton and LJP van der Maaten. Visualizing data using t-sne journal of machine learning research. 2008.

[35] Geoffrey E Hinton and Sam Roweis. Stochastic neighbor embedding. *Advances in neural information processing systems*, 15, 2002.

[36] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International workshop on similarity-based pattern recognition*, pages 84–92. Springer, 2015.

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

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

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

[40] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data mining and knowledge discovery*, 33(4):917–963, 2019.

[41] Anil K Jain, Jianchang Mao, and K Moidin Mohiuddin. Artificial neural networks: A tutorial. *Computer*, 29(3):31–44, 1996.

[42] Prateek Jain and Purushottam Kar. Non-convex optimization for machine learning. *arXiv preprint arXiv:1712.07897*, 2017.

[43] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

[44] Shivam Kalra, Mohammed Adnan, Graham Taylor, and Hamid R Tizhoosh. Learning permutation invariant representations using memory networks. In *European Conference on Computer Vision*, pages 677–693. Springer, 2020.

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

[46] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33, 2020.

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

[48] Edward C Klatt and Vinay Kumar. *Robbins and Cotran review of pathology*. Elsevier Health Sciences, 2014.

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

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

[51] Navid Alemi Koohbanani, Balagopal Unnikrishnan, Syed Ali Khurram, Pavitra Krishnaswamy, and Nasir Rajpoot. Self-path: Self-supervision for classification of pathology images with limited annotations. *IEEE Transactions on Medical Imaging*, 2021.

[52] Quoc V Le, Jiquan Ngiam, Adam Coates, Ahbik Lahiri, Bobby Prochnow, and Andrew Y Ng. On optimization methods for deep learning. In *ICML*, 2011.

[53] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[54] Yann LeCun, D Touresky, G Hinton, and T Sejnowski. A theoretical framework for back-propagation. In *Proceedings of the 1988 connectionist models summer school*, volume 1, pages 21–28, 1988.

[55] Bin Li, Yin Li, and Kevin W Eliceiri. Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14318–14328, 2021.

[56] Bin Li, Yin Li, and Kevin W. Eliceiri. Dual-stream multiple instance learning network for whole slide image classification with self-supervised contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14318–14328, June 2021.

[57] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018.

[58] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[59] Zhiyuan Li and Sanjeev Arora. An exponential learning rate schedule for deep learning. *arXiv preprint arXiv:1910.07454*, 2019.

[60] Sam Maksoud, Kun Zhao, Peter Hobson, Anthony Jennings, and Brian C Lovell. Sos: Selective objective switch for rapid immunofluorescence whole slide image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3862–3871, 2020.

[61] Niccolò Marini, Sebastian Otálora, Henning Müller, and Manfredo Atzori. Semi-supervised training of deep convolutional neural networks with heterogeneous data and few local annotations: An experiment on prostate histopathology image classification. *Medical image analysis*, 73:102165, 2021.

[62] Jovana Mitrovic, Brian McWilliams, and Melanie Rey. Less can be more in contrastive learning. 2020.

[63] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[64] Soojeong Nam, Yosep Chong, Chan Kwon Jung, Tae-Yeong Kwak, Ji Youl Lee, Jihwan Park, Mi Jung Rho, and Heounjeong Go. Introduction to digital pathology and computer-aided pathology. *Journal of pathology and translational medicine*, 54(2):125, 2020.

[65] Michael J Ombrello, Keith A Sikora, and Daniel L Kastner. Genetics, genomics, and their relevance to pathology and therapy. *Best practice & research Clinical rheumatology*, 28(2):175–189, 2014.

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

[67] Hendrik Purwins, Bo Li, Tuomas Virtanen, Jan Schlüter, Shuo-Yiin Chang, and Tara Sainath. Deep learning for audio signal processing. *IEEE Journal of Selected Topics in Signal Processing*, 13(2):206–219, 2019.

[68] Maral Rasoolijaberi. Multi-magnification search in digital pathology. Master's thesis, University of Waterloo, 2021.

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

[70] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[71] Ruslan Salakhutdinov and Geoff Hinton. Learning a nonlinear embedding by preserving class neighbourhood structure. In Marina Meila and Xiaotong Shen, editors, *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, volume 2 of *Proceedings of Machine Learning Research*, pages 412–419, San Juan, Puerto Rico, 21–24 Mar 2007. PMLR.

[72] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[73] Massimo Salvi, Filippo Molinari, U Rajendra Acharya, Luca Molinaro, and Kristen M Meiburger. Impact of stain normalization and patch selection on the performance of convolutional neural networks in histological breast and prostate cancer classification. *Computer Methods and Programs in Biomedicine Update*, 1:100004, 2021.

[74] Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[75] Yash Sharma, Aman Shrivastava, Lubaina Ehsan, Christopher A Moskaluk, Sana Syed, and Donald Brown. Cluster-to-conquer: A framework for end-to-end multi-instance learning for whole slide image classification. In *Medical Imaging with Deep Learning*, 2021.

[76] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

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

[78] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29, 2016.

[79] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

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

[81] Suyan Tian. Classification and survival prediction for early-stage lung adenocarcinoma and squamous cell carcinoma patients. *Oncology letters*, 14(5):5464–5470, 2017.

[82] Hamid R Tizhoosh, Phedias Diamandis, Clinton JV Campbell, Amir Safarpoor, Shivam Kalra, Danial Maleki, Abtin Riasatian, and Morteza Babaie. Searching images for consensus: can ai remove observer variability in pathology? *The American journal of pathology*, 191(10):1702–1708, 2021.

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

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

[85] Lindsey A Torre, Rebecca L Siegel, and Ahmedin Jemal. Lung cancer statistics. *Lung cancer and personalized medicine*, pages 1–19, 2016.

[86] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018, 2018.

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

[88] Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2495–2504, June 2021.

[89] Pengshuai Yang, Zhiwei Hong, Xiaoxu Yin, Chengzhan Zhu, and Rui Jiang. Self-supervised visual representation learning for histopathological images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 47–57. Springer, 2021.

[90] Kun-Hsing Yu, Ce Zhang, Gerald J Berry, Russ B Altman, Christopher Ré, Daniel L Rubin, and Michael Snyder. Predicting non-small cell lung cancer prognosis by fully automated microscopic pathology image features. *Nature communications*, 7(1):1–10, 2016.

[91] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in Neural Information Processing Systems*, 30, 2017.

[92] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks, 2013.

[93] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4320–4328, 2018.

[94] Wengang Zhou, Houqiang Li, and Qi Tian. Recent advance in content-based image retrieval: A literature survey. *arXiv preprint arXiv:1706.06064*, 2017.

[95] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.

APPENDIX

TCGA Cancer Subtype Acronyms

Table 1 explains the cancer subtype abbreviations exploited in the paper.

Table 1: Cancer subtype abbreviations.

Abbreviation	Primary Diagnosis
ACC	Adrenocortical Carcinoma
BLCA	Bladder Urothelial Carcinoma
BRCA	Breast Invasive Carcinoma
CESC	Cervical Squamous Cell Carcinoma and Endocervical Adenoc.
CHOL	Cholangiocarcinoma
COAD	Colon Adenocarcinoma
ESCA	Esophageal Carcinoma
GBM	Glioblastoma Multiforme
HNSC	Head and Neck Squamous Cell Carcinoma
KICH	Kidney Chromophobe
KIRC	Kidney Renal Clear Cell Carcinoma
KIRP	Kidney Renal Papillary Cell Carcinoma
LGG	Brain Lower Grade Glioma
LIHC	Liver Hepatocellular Carcinoma
LUAD	Lung Adenocarcinoma
LUSC	Lung Squamous Cell Carcinoma
MESO	Mesothelioma
OV	Ovarian Serous Cystadenocarcinoma
PAAD	Pancreatic Adenocarcinoma
PCPG	Pheochromocytoma and Paraganglioma
PRAD	Prostate Adenocarcinoma
READ	Rectum Adenocarcinoma
SARC	Sarcoma
SKCM	Skin Cutaneous Melanoma
STAD	Stomach Adenocarcinoma
TGCT	Testicular Germ Cell Tumors
THCA	Thyroid Carcinoma
THYM	Thymoma
UCS	Uterine Carcinosarcoma
UVM	Uveal Melanoma