MASTER THESIS

PIPMIL: Multiple Instance Learning using Patch-based Intuitive Prototypes

submitted by

ABHAY AUGUSTINE JOSEPH

Submitted to the

Chair for Data Science in the Economic and Social Sciences

within the

Faculty of Business Administration

at University of Mannheim

November 14, 2024

Supervisor:

Prof. Dr. Jörg Schlötterer

First Examiner

Prof. Dr. Jörg Schlötterer

Second Examiner

Prof. Dr. Margret Keuper

Contents

Abstract *The task of Whole Slide Image (WSI) classification is critical in digital pathology, especially for diagnosing diseases such as cancer. Traditional approaches to WSI analysis, however, face significant challenges due to the immense size of these images, the need for interpretability in high-stakes medical contexts, and the ambiguity introduced by labels available only at the slide level. This thesis addresses these challenges by developing PIPMIL (Patch-Based Intuitive Prototypes for Multiple Instance Learning), an interpretable MIL-based framework inspired by PIP-Net, adapted specifically for WSI classification tasks. PIPMIL introduces a prototype-based approach where each WSI is divided into instances , and intuitive, human-recognizable prototypes are learned to represent semantically meaningful parts of the tissue. These prototypes enable PIPMIL to produce both accurate and interpretable predictions, allowing pathologists to understand the rationale behind each classification. The training strategy incorporates self-supervised pre-training of prototypes at an instance-level to learn prototypical parts that align with human visual perception. For each bag, instances are compared with the learned prototypes to generate prototype presence scores. These instance-level scores are then aggregated into a single bag-level prototype presence score vector. To address the high computational costs associated with processing WSIs, several strategies were incorporated, including selective instance fine-tuning, patch encoding with a pre-trained CNN, and patch sampling techniques. While the model demonstrated improved interpretability over baseline MIL approaches, the overall accuracy fell short, in comparison, indicating a trade-off between interpretability and predictive performance. This limitation suggests the need for further optimization in balancing these two objectives. Nonetheless, PIPMIL's prototype-based approach offers a promising pathway towards interpretable, clinically applicable machine learning in pathology, setting a foundation for future enhancements that could reconcile interpretability with improved accuracy.*

1

Introduction

In the realm of medical imaging, the analysis of Whole Slide Images (WSIs) has become essential in the diagnosis and treatment of various diseases, especially in pathology. WSIs are high-resolution digital images of tissue samples, capturing comprehensive visual information that pathologists rely on for accurate diagnostics. However, the sheer size and complexity of WSIs introduce significant challenges to effective analysis. Traditionally, pathologists manually inspect these images, a time-intensive and subjective process prone to human error and variability in diagnosis. As the volume of imaging data continues to grow, there is a pressing need for automated methods that assist pathologists in processing WSIs efficiently and reliably.

To address these challenges, computer-aided diagnosis (CAD) systems have emerged, leveraging advances in computational pathology to support pathologists in WSI analysis. Recent developments in deep learning, particularly in supervised learning models, have shown promise in disease diagnosis and pathological research. However, these conventional approaches encounter limitations with WSIs: their assumption of high-quality, instance-level labels does not align with real-world medical datasets, where individual patches within a WSI are rarely labeled, and labeling is impractical. WSIs are typically gigapixel-scale, where only a single label is assigned to an entire image, without any indication of which patches are critical for diagnosis.

To navigate this limitation, Multiple Instance Learning (MIL) has been introduced as a suitable approach for WSI analysis. In MIL, data is organized into "bags" of instances, each bag representing a WSI, with only the label for the entire bag provided. This approach circumvents the need for instance-level annotations, making it ideal for large-scale WSI datasets where fine-grained labels are unavailable. While MIL models combined with neural networks have achieved significant success, they often fall short in terms of interpretability, failing to provide insights into why specific predictions are made, which is essential for clinical adoption.

In recent years, interpretability in MIL models has been enhanced by incorporating case-based reasoning, where models learn prototypes—semantically meaningful parts of images that help explain predictions. ProtoMIL is one such approach that aims to align prototypes with human-understandable concepts. However, ProtoMIL and similar methods face a semantic gap between

similarity in the model's latent space and human-interpretable concepts. Additionally, these models often fix the number of prototypes per class, resulting in large, redundant prototype sets that limit interpretability.

To address these challenges, this thesis introduces Patch-based Intuitive Prototype MIL (PIPMIL), an adaptation of the PIP-Net model to the MIL framework. Originally developed for interpretable image classification, PIP-Net learns meaningful prototypes through self-supervised learning. By extending PIP-Net to the MIL setting, PIPMIL aims to capture the key components of WSIs in a way that is both accurate and interpretable. This adaptation involves segmenting each WSI into smaller patches and determining which parts of these patches bear similarity to a set of learned prototypes. These prototypes are trained in a self-supervised manner, and the instance-level scores are aggregated to form a bag-level representation for classification. Through extensive experimentation, this thesis demonstrates PIPMIL's capacity to deliver reliable and interpretable predictions in WSI classification tasks.

1.1 Problem Statement

The analysis of WSIs is critical to medical diagnoses, particularly in pathology, where WSIs provide invaluable insights into tissue characteristics and disease states. However, WSIs are characterized by high resolution and complexity, posing challenges for manual analysis and traditional computational methods. The primary issues are:

1. **Computational complexity:** The high-resolution nature of WSIs demands significant computational resources for processing.
2. **Lack of instance-level labels:** In clinical practice, obtaining detailed, instance-level labels for WSI patches is impractical, meaning models must operate with only coarse slide-level labels.
3. **Need for interpretability:** Clinical settings demand that AI models not only be accurate but also interpretable, providing human-understandable insights into their predictions.

While MIL has emerged as a viable approach to handling WSIs without instance-level labels, existing MIL models often lack interpretability. This limits their potential clinical utility, as medical professionals need to understand the rationale behind each prediction to trust and act upon model outputs.

1.2 Contributions and Applications

This thesis addresses the need for an interpretable, efficient, and accurate approach to WSI classification by proposing PIPMIL, an adaptation of PIP-Net to the MIL setting. The key contributions of this thesis include:

1. **Adaptation of PIP-Net for MIL** - This thesis extends PIP-Net to the MIL framework. This adaptation allows the model to handle large, high-resolution WSIs segmented into patches and classifies entire slides based on the aggregate patch information.

2. **Efficiency in Handling Large-Scale WSIs** - The model incorporates methods such as patch sampling, patch encoding, and selective instance learning to address computational limitations. These methods allow PIPMIL to process WSIs efficiently without exhaustive computational resources, broadening its applicability to large-scale datasets.

The applications of this work are extensive within computational pathology and medical diagnostics. By providing interpretable predictions, PIPMIL offers a powerful tool for aiding pathologists in diagnostic decision-making, particularly in high-stakes settings such as cancer diagnosis.

1.3 Report Organization

The report content is arranged in the following manner:

– **Chapter 2: Background**- This chapter provides an overview and background on Whole Slide Imaging.

– **Chapter 3: Related Work**- This chapter delves into the original PIP-Net architecture, its design, self-supervised prototype learning, and how it achieves interpretability. It also provides an overview of Multiple Instance Learning, and explainable AI models.

– **Chapter 4: Approach**- Here, we describe the adaptation of PIP-Net to the MIL setting, detailing model modifications and methods to handle computational restrictions specific to WSI classification.

– **Chapter 5: Experimental Evaluation and Results**- This chapter presents the datasets, evaluation metrics, and experimental setup and results, highlighting PIPMIL's performance in terms of accuracy, and interpretability.

– **Chapter 6: Discussion**– This chapter provides an in-depth discussion of the results, examining PIPMIL's strengths, limitations, and the implications of its findings for computational pathology.

– **Chapter 7: Conclusion**- The final chapter summarizes the contributions of the thesis, discusses the potential impact of PIPMIL in clinical settings, and suggests directions for future research.

2

Background

2.1 Whole Slide Imaging

Whole Slide Imaging (WSI) is a digital pathology technique that enables the scanning of entire tissue slides at high resolution, creating digitized images that can be viewed, analyzed, and stored electronically. Whole Slide Imaging (WSI) has revolutionized digital pathology by allowing the capture of high-resolution, large-scale digital scans of entire histological slides, often at multiple magnifications. These digitized slides, typically in the form of multi-gigapixel images, provide a unique opportunity for computational analysis and machine learning applications in pathology, particularly in the fields of disease diagnosis, prognosis, and research [8, 18]. The vast amount of data generated by WSIs, however, brings significant challenges in terms of data handling, storage, computational power, and algorithm development.

One of the primary challenges in WSI analysis is related to the heterogeneity of histopathological features, which vary widely across different regions of a single slide. This complicates conventional machine learning approaches, as identifying relevant features for diagnostic tasks often requires considering subtle, context-dependent patterns that only appear in certain regions of a slide. As a result, more recent machine learning methods, particularly those in the realm of Multiple Instance Learning (MIL), are being increasingly applied to WSIs to address these unique challenges.

2.1.1 Data Characteristics and Challenges

Whole Slide Images are complex in structure, with significant differences in tissue morphology across slide regions. This variability requires models that can capture both local and global patterns within the image. For instance, a tumor's appearance might change across different regions of a slide, requiring models to distinguish between normal and pathological tissues at a granular level. Additionally, the scale and resolution of WSIs present storage and processing challenges that are non-trivial for computational workflows. Conventional approaches to image analysis, which often

assume uniformity across an image, struggle to generalize across the diverse and often sparse histopathological features within WSIs [14].

Furthermore, WSIs often contain large areas of irrelevant information, such as background regions, folds, or other artifacts that may arise during slide preparation. These irrelevant areas can introduce noise and complicate the learning process for machine learning models. Effective WSI analysis therefore requires preprocessing steps that include tissue segmentation, stain normalization, and artifact removal to ensure that models focus on diagnostically relevant areas of the slide [20].

2.1.2 Patch-based Analysis in Whole Slide Imaging

Given the size and complexity of WSIs, analyzing the entire slide at once is impractical. Instead, WSIs are typically divided into smaller, manageable patches, which are then processed independently. Patch-based analysis enables more feasible computation, as it reduces the memory and processing power requirements. However, patch-based methods present their own set of challenges, particularly in ensuring that the information extracted from individual patches can be combined effectively to capture the overall characteristics of the slide.

Multiple Instance Learning proves to be effective in such a case, as they are well-suited to handling scenarios where individual patches (instances) are not labeled directly but contribute to a coarse slide-level label (e.g., cancer presence or absence). MIL approaches are designed to aggregate information across multiple patches, allowing the model to learn from weakly labeled data [12]. Prototypical models in MIL can further enhance this process by identifying representative patches, or prototypes, that embody the most informative features of the slide [3].

In these models, each patch is processed independently through a feature extractor, such as a deep convolutional neural network (CNN), and the features are then aggregated to produce a final slide-level prediction. This approach is particularly advantageous for WSI data, where only a subset of patches may contain relevant diagnostic information.

2.1.3 Feature Extraction and Representation Learning in WSI

Feature extraction from WSI patches often involves the use of pre-trained deep learning models, which are fine-tuned or adapted to histopathological data. Convolutional neural networks (CNNs) have been extensively used for feature extraction in pathology, particularly for extracting spatial and morphological information from tissue images [4]. CNN-based feature extractors can capture local patterns in histological structures, such as cellular morphology and tissue organization, which are crucial for identifying pathologies like cancer.

However, recent advances in feature learning for WSI analysis go beyond simple CNNs, incorporating attention mechanisms and self-supervised learning approaches to capture more complex dependencies and context-aware features [**lu2021data**]. For example, attention-based mechanisms allow models to focus on the most relevant patches within a slide, prioritizing diagnostically

important regions while ignoring irrelevant ones. These mechanisms have proven effective for capturing nuanced patterns that may be sparsely distributed across WSIs, thus enhancing model interpretability and performance in downstream tasks.

Representation learning approaches, such as contrastive learning, have also shown promise in pathology. By training models to differentiate between similar and dissimilar patches, contrastive learning can help models learn robust, discriminative features that capture subtle differences in tissue morphology. This approach is particularly useful in the context of prototypical models for MIL, as it allows the model to create representations of "prototype" patches that best represent key histopathological features within a slide [**shao2021transmil**].

2.1.4 Aggregation Strategies for WSI Analysis

A critical component of WSI analysis using MIL-based approaches is the aggregation of information from multiple patches to make a slide-level prediction. Simple aggregation techniques, such as max-pooling or mean-pooling, have been used traditionally, but these methods may fail to capture complex relationships among patches. More sophisticated aggregation methods, such as attention-based pooling or learned pooling mechanisms, have been developed to better capture the contextual relevance of each patch within the broader slide [**lu2021data**].

For example, attention-based models like CLAM (CLustering-constrained Attention Multiple instance learning) have been shown to improve diagnostic accuracy by dynamically weighting patches according to their relevance [**lu2021data**]. In CLAM, patches that are more similar to known pathology patterns receive higher weights, guiding the model to focus on these regions. This type of adaptive aggregation is particularly advantageous in WSIs, where relevant patterns may only appear in a few patches, but these patches are crucial for diagnosis.

2.1.5 Prototypical Models in WSI Analysis

Prototypical models for MIL leverage the concept of prototypes to represent characteristic regions within a slide. By identifying a set of representative patches, these models can reduce the complexity of WSI data while preserving the essential information required for diagnostic tasks [**lu2021data**]. In practice, prototypical models select or learn a set of patches that best capture the diversity of histopathological features within a slide, facilitating robust predictions even when there is significant intra-slide variability.

In the context of WSI analysis, prototypical models can serve as an effective mechanism for simplifying the interpretation of complex tissue structures, enabling pathologists to understand which regions of the slide contributed most significantly to a model's prediction. This interpretability is particularly valuable in clinical settings, where understanding the reasoning behind a diagnosis is often as important as the diagnosis itself.

Whole Slide Imaging has introduced new opportunities and challenges for digital pathology, espe-

cially as the scale and complexity of WSIs require novel computational approaches for effective analysis. Patch-based analysis combined with MIL frameworks, particularly those that incorporate prototypical models, represents a promising direction for WSI analysis, enabling models to leverage the weakly-labeled and heterogeneous nature of WSI data. Advanced feature extraction techniques and aggregation strategies further support the adaptation of machine learning to this domain, making it possible to capture the complex, multi-scale patterns inherent in pathology.

While this chapter focused on the background and motivation for using WSI in conjunction with MIL-based approaches, a deeper exploration of Multiple Instance Learning and Prototypical Models will follow in the subsequent chapters. These approaches provide the technical foundation for achieving robust, interpretable, and scalable analysis of whole slide images in computational pathology.

3

Related Work

3.1 Patch-based Intuitive Prototype Network (PIP-Net)

PIP-Net [17] (Patch-based Intuitive Prototypes Network) represents a significant advancement in the field of interpretable image classification. Developed to bridge the semantic gap between machine learning models and human understanding, PIP-Net is designed to learn prototypical parts of images in a way that aligns with human visual perception. By identifying and utilizing intuitive prototypes in a self-supervised manner, PIP-Net provides both global and local explanations of its predictions, making the model's decision-making process transparent. This chapter explores the architecture of PIP-Net, its self-supervised prototype pre-training, and the full training process that allows the model to provide interpretable and accurate image classification.

3.1.1 Model Architecture and Reasoning

PIP-Net's architecture is designed to be both simple and intuitive, relying on the extraction of prototypical image patches that directly contribute to the classification decision. Figure 3.1 illustrates an overview of the model's architecture and its components. The model follows the "scoring-sheet reasoning" framework, where the presence of certain prototypes in an image adds evidence for a particular class. The core components of PIP-Net's architecture are:

CNN Backbone and Prototype Presence Scores

At the foundation of PIP-Net is a convolutional neural network (CNN) such as ResNet [11] or ConvNeXt [15], which extracts feature maps corresponding to unnormalized prototype presence scores from input images.Each output map in the resulting set of D two-dimensional maps represents the degree of similarity between a particular prototype and each spatial region in the image. Formally, let $z = f(x, w_f)$ denote the set of D unnormalized prototype presence scores for an image

x, where f is the CNN and w_f represents the learned weights of the network. These maps retain spatial arrangement, preserving localized information across the patch grid of the image. By later applying a softmax operation over D (across prototypes), PIP-Net normalizes these scores, ensuring that each spatial region is associated with a specific prototype. This design allows the model to identify and match specific image regions to semantically meaningful prototypes, enhancing interpretability.

Softmax and Max-Pooling

The generated output maps (unnormalized prototype presence scores) from the CNN are passed through a layer consisting of a set of learned prototypes, each representing a specific concept in the dataset that aligns with human visual perception. These prototypes are visualized as patches from the training data, making them easily interpretable by humans. The output maps are then matched against the prototypes using a softmax operation. The softmax is applied over D such that $\sum_d^D z_{h,w,d} = 1$, to ensure that a patch $z_{h,w,:}$ belongs to exactly one prototype, resulting in a prototype activation map across D. The Softmax layer, thus, normalizes the prototype presence scores. To aggregate the information from the resulting normalized scores, PIP-Net applies max-pooling across each prototype's activation map. This operation identifies the highest activation for each prototype across the D dimensions, resulting in a compact vector representation p that captures how strongly each prototype is present in the image.

Sparse Linear Layer and Scoring-Sheet Reasoning

The image encoding p is used as the input to a sparse linear layer, which is responsible for connecting the prototype activations to the final class labels, with weights $w_c \in \mathbb{R}_{\geq 0}^{D \times K}$. The learned weights $w_c^{d,k}$ indicate the relevance of prototype d to class k. To enhance interpretability, this linear layer is designed to be sparse, meaning that only a few prototypes are connected to each class. PIP-Net enforces a sparse linear layer by incorporating gradient clipping; weights in the classification layer that fall below a certain threshold (default, 0.001) are set to zero. This ensures that each prediction relies on a limited set of prototypes, making the model's reasoning more interpretable. By minimizing the number of prototypes actively contributing to each class, the model provides focused and concise explanations, helping users trace each decision to a small, meaningful subset of patterns. Importantly, the weights in this layer are constrained to be non-negative, meaning that the presence of a prototype can only add evidence for a class. This constraint ensures that the reasoning process remains interpretable, as the model cannot "negatively" assign a prototype to a class. The final class score is the sum of the prototype activation scores multiplied by the incoming class weights of the linear layer, creating a clear and intuitive explanation of the model's decision.

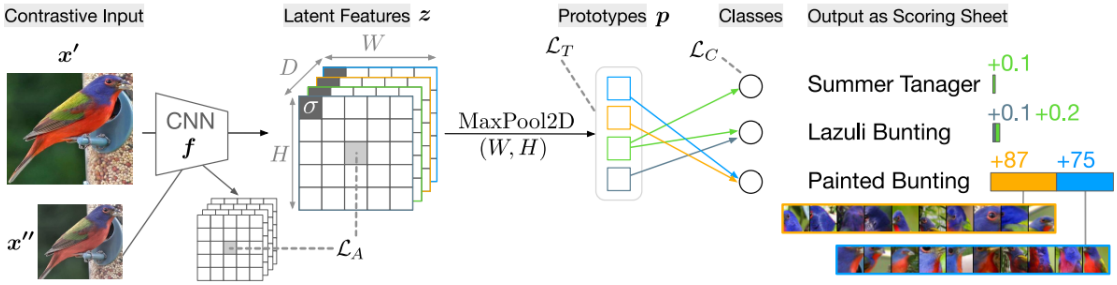


Figure 3.1: *Architecture overview of PIP-Net. (adapted from [17]). A CNN backbone is used to learn prototypical representations z, which are normalized using a Softmax layer, and pooled to a vector of prototype presence scores p. Two views of an image patch (the original and an augmented version) are processed through the CNN backbone. Contrastive learning forces representations of the image pair to be assigned the same prototype in the latent feature space, using the alignment loss $\mathcal{L}_A$. Tanh loss $\mathcal{L}_T$ regularizes the model to make use of all available prototypes. The learned part-prototypes are connected to the classes via a sparse linear layer. Standard negative log-likelihood loss $\mathcal{L}_C$, is the loss function for the classifier. The output is visualized as a scoring sheet.*

3.1.2 Self-Supervised Pre-Training of Prototypes

One of the unique features of PIP-Net is its approach to learning prototypes in a self-supervised contrastive manner, ensuring that they are semantically meaningful and aligned with human perception. This pre-training phase focuses solely on learning the prototypes without relying on image labels or the final classification task. The self-supervised pre-training process begins by generating positive pairs of images (say x' and x''), as used in other self-supervising methods[13]. These pairs are created by applying various data augmentations to the same image (say x), such as random cropping, rotation, or color jittering. The goal is to create two distinct views of the same image that a human would still recognize as representing the same object or concept. By using such augmentations, PIP-Net incorporates human visual perception into the learning process.

The essence of the self-supervised pretraining phase is to optimize for alignment and uniformity of the feature representations. Alignment enforces two similar images to be mapped to nearby latent feature vectors, and uniformity induces a uniform distribution of the feature vectors on a unit hypersphere [21]. PIP-Net, however, optimizes for *patch* alignment as opposed to alignment on an *image* level, by employing a combination of two loss terms. Specifically,

1. **Alignment Loss ($\mathcal{L}_A$):** This loss term encourages the model to activate the same prototype for similar patches. In the context of training, the model would learn to activate the same prototype for augmented versions of the same image patch. This is achieved by optimizing two views of the same image patch to belong to the same prototype:

$$\mathcal{L}_A = -\frac{1}{HW} \sum_{(h,w)\in H\times W} log(z'_{h,w,:} \cdot z''_{h,w,:}) \tag{3.1}$$

where $z'_{h,w}$ and $z''_{h,w}$ are the prototype activations for the two views at spatial location (h,w). By minimizing this loss, the model is optimized to assign consistent prototypes to similar patches across different views of the same object. Ideally, both views of the image patch would activate the same single prototype. This would occur if the normalized prototype presence scores for both views were one-hot encoded, with a value of 1 in the same dimension and 0 in all other dimensions.

The objective of solely minimizing this loss term can lead to a naive solution by trying to get $\mathcal{L}_A = 0$, wherein one prototype node is activated on every image patch in every image in the dataset. In order to avoid falling into this naive solution, PIP-Net uses the Alignment loss in combination with a second loss term, namely the tanh loss.

2. **Tanh Loss ($\mathcal{L}_T$):** The tanh loss is employed by PIP-Net to ensure that the prototypes are diverse and cover a wide range of visual concepts. This regularization prevents the model from falling into trivial solutions where only a few prototypes are used for all patches. The tanh loss encourages the model to make use of all available prototypes by penalizing underutilization:

$$\mathcal{L}_T(p) = -\frac{1}{D}\sum_{d}^{D} log(tanh(\sum_{b}^{B} p_b) + \varepsilon) \tag{3.2}$$

where p_b is the activation scores of prototypes in a mini-batch sample b and ε is a small number for numerical stability. This loss ensures that each prototype is present at least once within a mini-batch and regularizes the model to make use of all available prototypes, promoting diversity in the learned prototypes. It also does not account for how *often* a prototype is present in the mini-batch, as some prototypes may occur more frequently than others.

The overall objective during the self-supervised pre-training phase is to optimize a combination of the alignment and tanh losses:

$$\mathcal{L}_{pretrain} = \lambda_A \mathcal{L}_A + \lambda_T \mathcal{L}_T \tag{3.3}$$

where λ_A and λ_T are hyperparameters that control the relative importance of each loss term. By minimizing this combined loss, PIP-Net learns prototypes that are both consistent and diverse, forming a robust basis for the subsequent classification task.

3.1.3 Training PIP-Net

Once the prototypes have been learned in the self-supervised pre-training phase, PIP-Net proceeds to the second phase, where the entire model is trained for classification. In this phase, the prototypes are fine-tuned to become even more relevant to the specific classes in the dataset.

During this phase, the goal is to align the prototypes more closely with the class labels, ensuring that the prototypes correspond to discriminative features for each class. The linear layer connect-

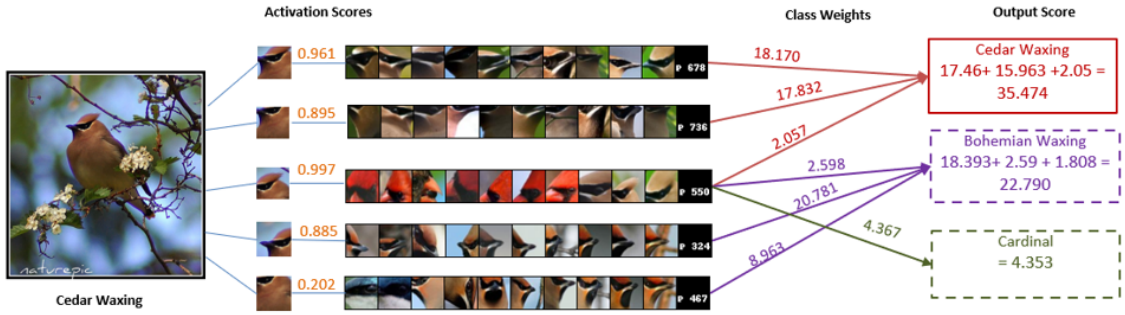


Figure 3.2: *Inference of PIP-Net with score-sheet reasoning. The figure illustrates the model's prediction of the class 'Cedar Waxing'. The most activated prototypes in the image are identified with their activation scores and their respective weights to the class labels. The predicted class is determined by the class label with the highest output score.*

ing the prototypes to the class labels is unfrozen and also trained during this phase, allowing the model to learn the relevance of each prototype for each class. The primary loss function during the full pipeline training is the classification loss $\mathcal{L}_C$, which is the standard non-negative likelihood loss between the predicted class labels and the ground truth labels. This loss adjusts both the prototype activations and the linear layer weights to improve classification accuracy.

To enhance the interpretability of PIP-Net, sparsity is enforced in the linear layer. This regularization ensures that only a small subset of prototypes contributes to the classification of any given class. By limiting the number of non-zero weights in the linear layer, PIP-Net produces compact and easily interpretable explanations. The final model outputs a sparse set of prototype activations for each image, making it clear which prototypes contributed to the final decision. Figure 3.2 illustrates the inference of PIP-Net, reflecting the model's reasoning process.

3.2 Multiple Instance Learning

Multiple Instance Learning (MIL) is a type of weakly-supervised learning method that addresses problems where only coarse-level annotations are available, making it particularly suited for tasks such as medical image analysis, drug activity prediction, and image retrieval. Unlike traditional supervised learning, which assumes that each instance in a dataset is independently labeled, MIL operates on "bags" of instances, where only the label for the entire bag is provided. This paradigm was first introduced by Dietterich et al. [6] in the context of drug activity prediction, where a molecule's activity is determined based on multiple conformations (instances) of the molecule, but only the overall activity label of the molecule (bag) is known.

3.2.1 Fundamental Concepts and Assumptions

The fundamental assumption of MIL is that each instance within a bag has a hidden class label, either positive or negative, and a bag is classified as positive if and only if at least one of its instances is positive [22]. Conversely, a bag is classified as negative if all its instances are negative. This standard MIL assumption has been extensively explored and formalized in various studies.

Early approaches to MIL include the Axis-Parallel Rectangles (APR) algorithm by Dietterich et al.[6], which aimed to identify regions in the instance space that correspond to positive bags. Auer (1997) presented a more practical algorithm, *MULTINST* to efficiently learn the APR principles, based on simple statistics of the bag [1]. Another notable method is the Diverse Density (DD) algorithm introduced by Maron and Lozano-Pérez[16], which searches for a point in the instance space that is close to instances from positive bags and far from instances in negative bags. Zhang and Goldman (2001) extended this algorithm to formulate the expectation maximization-diverse density (EMDD) algorithm [24] which has been used on several MIL problems including drug activity prediction, image retrieval and stock selection. These foundational methods laid the groundwork for more advanced MIL techniques.

3.2.2 Obstacles in Multiple Instance Learning

One of the primary challenges in MIL is the ambiguity in instance labeling within bags. Since only bag-level labels are available, there is significant uncertainty regarding which instances contribute to the bag's label. This ambiguity can lead to difficulties in training models and may result in poor generalization performance. Moreover, MIL datasets often suffer from class imbalance, where positive instances are rare compared to negative ones, further complicating the learning process .

Another obstacle is the computational complexity associated with processing large bags of instances, especially in domains such as medical imaging, where each bag can contain thousands of high-dimensional instances. Efficiently handling and learning from such large-scale data requires significant computational resources and sophisticated algorithms .

3.2.3 Advancements in Multiple Instance Learning

Recent advancements in MIL have focused on addressing these obstacles through the development of more sophisticated algorithms and the integration of deep learning techniques. For instance, deep MIL models leverage neural networks to learn powerful representations of instances and bags. The mi-Net model by Wang et al. [7] uses neural networks to embed instances and then aggregate these embeddings to form a bag-level representation, demonstrating improved performance on various MIL tasks .

Attention-based mechanisms have also been incorporated into MIL to enhance model interpretability and performance. The Attention-based Deep MIL (ABMIL) model by Ilse et al.[12] introduces an attention mechanism that learns to weigh instances within a bag according to their relevance, allowing the model to focus on the most informative instances . This approach not only

improves classification accuracy but also provides insights into which instances are critical for the bag-level prediction.

ProtoMIL [19] is another notable advancement that combines case-based reasoning with prototype learning. It uses a convolutional network to learn prototypes that represent typical parts of positive and negative classes. These prototypes are then used to compare instances within bags, and an attention pooling mechanism aggregates these comparisons to form a bag-level prediction. ProtoMIL has shown promise in tasks such as WSI classification by providing interpretable predictions that align with human visual perception. Another recent prototype-bsaed model Prototypical Multiple Instance Learning (PMIL) [23] utilizes Bag-of-visual-words (BoVW) for image representation- which is characterized by using the occurrence counts of a vocabulary of local image features to encode an image. The prototype discovery module of PMIL comprises of two clustering procedures- an intra-slide clustering (ISC) applied to the patches within each WSI separately to yield the first-level centroids, and inter-slide clustering (XSC) performed over the first-level centroids to obtain the second-level centroids. They adopt Affinity Propagation Clustering (APC) as the basic clustering algorithm.

3.2.4 Applications of Multiple Instance Learning

MIL has been successfully applied in various domains, showcasing its versatility and effectiveness. In the medical field, Campanella et al. [3] demonstrated the use of MIL for the classification of WSIs in pathology introduceing an innovative RNN-based aggregation method, paired with instance-space MIL featuring max-pooling. Their model, which combines instance-level embeddings with recurrent neural networks for aggregation, achieved state-of-the-art performance in detecting metastatic breast cancer .

In another example, the DeepMIML[9] network extends MIL to a multi-instance, multi-label setting, enabling the classification of images with multiple labels. This model leverages deep learning to extract features from instances and then uses a multi-layer neural network to aggregate these features, achieving high accuracy on image classification benchmarks.

The versatility of MIL is further exemplified by its application in remote sensing. In this context, MIL models have been used to classify land cover types from satellite imagery, where each image (bag) consists of multiple regions (instances) with varying land cover types. The MIL framework effectively handles the variability and complexity of such data, leading to improved classification performance.

In summary, Multiple Instance Learning offers a powerful framework for addressing tasks where only coarse-level annotations are available. Despite its challenges, the continued development of advanced algorithms and deep learning techniques holds promise for enhancing the performance and interpretability of MIL models across a wide range of applications.

3.3 Explainable AI and Interpretable Models

Explainable Artificial Intelligence (XAI) and interpretable models have gained significant attention in recent years, driven by the need for transparency, accountability, and trust in AI systems. The black-box nature of many state-of-the-art machine learning models, particularly deep learning networks, poses challenges in understanding and interpreting their decisions. This lack of interpretability is particularly problematic in high-stakes domains such as healthcare, finance, and criminal justice, where the implications of model decisions can be profound. This section provides a detailed overview of the key concepts, obstacles, advancements, and examples in the field of XAI and interpretable models.

3.3.1 Obstacles in Explainable AI and Interpretable Models

One of the primary obstacles in XAI is the trade-off between model accuracy and interpretability. Highly accurate models, such as deep neural networks, are often complex and difficult to interpret, whereas simpler models, such as linear regression or decision trees, are more interpretable but may lack the predictive power of their more complex counterparts. This trade-off poses a challenge in achieving both high performance and high interpretability.

Another significant challenge is the lack of standardization in evaluating and comparing the interpretability of different models. Interpretability is inherently subjective and context-dependent, especially in domain of medical imaging, making it difficult to develop universal measures and benchmarks. Furthermore, the effectiveness of interpretability methods can vary across different domains and applications, complicating the assessment of their generalizability .

The black-box nature of many machine learning models also presents challenges in debugging and improving models. Without understanding why a model makes certain decisions, it becomes difficult to identify and correct errors, biases, or unintended behaviors. This lack of transparency can erode trust in AI systems, particularly in critical applications where explainability is essential for accountability and ethical considerations.

3.3.2 Advancements in Explainable AI and Interpretable Models

Despite these challenges, significant advancements have been made in the field of XAI. Post hoc explanation methods, such as saliency maps, LIME (Local Interpretable Model-agnostic Explanations), and SHAP (SHapley Additive exPlanations), provide insights into model decisions by highlighting the most influential features for a given prediction. Saliency maps, for example, visualize the regions of an input image that contribute most to the model's output, offering a way to interpret the decisions of convolutional neural networks (CNNs) .

Model-agnostic explanation methods like LIME and SHAP generate local explanations for individual predictions by approximating the complex model with an interpretable surrogate model. These methods have gained popularity due to their flexibility and applicability to any black-box model, making them valuable tools for improving model transparency .

In addition to post hoc methods, there have been advancements in designing intrinsically interpretable models. The Prototypical Part Network (ProtoPNet) is one such model that learns prototypical parts of images during training, which can then be visualized to explain the model's predictions. ProtoPNet aligns its learned prototypes with human-understandable concepts, thereby enhancing the interpretability of its decisions. This model has been particularly useful in medical image analysis, where understanding the basis of a model's decision is crucial.

Another advancement is the use of decision trees and rule-based systems for interpretable machine learning. Neural Prototype Trees, for example, combine the strengths of neural networks and decision trees by learning prototypes that are used as decision nodes in a tree structure. This approach provides both high accuracy and interpretability, making it suitable for fine-grained image recognition tasks.

The development of self-explaining models, which incorporate interpretability directly into their architecture, represents another significant advancement. These models are designed to provide explanations as part of their output, reducing the need for separate explanation methods. An example is the Transparent Embedding Space (TesNet), which learns interpretable latent representations of data, facilitating understanding of model decisions at a deeper level.

3.3.3 Applications of Explainable AI and Interpretable Models

Explainable AI and interpretable models have been applied successfully across various domains. In healthcare, XAI techniques have been used to interpret deep learning models for diagnosing diseases from medical images. For instance, ProtoPNet has been employed to classify skin lesions by learning and visualizing prototypes that correspond to different types of lesions, thereby providing interpretable and clinically relevant explanations .

In the financial sector, interpretable models are crucial for risk assessment and fraud detection. Rule-based systems and decision trees are commonly used to create transparent models that can be audited and understood by human experts. These models help ensure that decisions are made based on clear and justifiable criteria, which is essential for regulatory compliance and trust .

XAI has also found applications in criminal justice, where it is used to interpret risk assessment models that predict recidivism. Transparent models help ensure that decisions are fair and unbiased, addressing concerns about the potential for AI to perpetuate existing biases and inequalities. By providing clear explanations for their predictions, these models can facilitate more informed and equitable decision-making .

3.3.4 Future Scope

As the field of XAI continues to evolve, future research is expected to focus on developing more sophisticated interpretability methods and integrating them into real-world applications. One promising direction is the combination of XAI with human-in-the-loop approaches, where human experts interact with AI systems to refine and validate explanations. This collaboration can enhance the reliability and trustworthiness of AI systems.

Another important direction is the development of standardized measures and benchmarks for evaluating interpretability. Creating a common framework for assessing the quality and effectiveness of interpretability methods will facilitate more rigorous and comparable research in this area. Additionally, advances in computational efficiency will be crucial for scaling XAI methods to handle large and complex datasets, enabling their widespread adoption in industry and academia.

In summary, Explainable AI and interpretable models are essential for ensuring transparency, accountability, and trust in AI systems. While significant challenges remain, ongoing advancements in this field hold promise for creating more understandable and trustworthy AI models that can be effectively applied across a wide range of domains.

4

Approach

Patch-based Intuitive Prototype Network (PIP-Net) was originally developed as a prototype-driven, interpretable classification model that makes decisions based on prototypical parts of an image. In its original form, PIP-Net operates on single images, learning prototypes that represent intuitive and human-recognizable parts of objects, or concepts. However, for Whole Slide Image (WSI) classification tasks, the problem scenario requires a Multiple Instance Learning (MIL) approach, where each image (or slide) is divided into a number of smaller patches (or instances) that are associated with a single slide-level label, but the labels of individual patches are unknown. Instead, each instance is associated with the single slide-level label.

To address this, PIP-Net needed to be adapted for the MIL framework, which is inherently different from conventional supervised learning paradigms. This adaptation required significant changes to the model's architecture, the way prototypes are learned, and how the loss functions are defined and applied across a "bag" of instances (i.e., all the patches in a WSI). In this section, we will discuss how PIP-Net was adapted to operate in a MIL setting and the modifications made to the model's loss functions to accommodate this change.

4.1 Adaptation of PIP-Net for Multiple Instance Learning (MIL)

In a traditional MIL setup, each training example is a "bag" that consists of a collection of instances, and the task is to predict a label for the entire bag, rather than individual instances. In the case of WSIs, each bag corresponds to an entire slide, and the individual patches of the slide act as instances. The key assumption in MIL is that for a positive bag (e.g., a malignant slide), at least one of the instances is positive (contains cancerous tissue). Conversely, in a negative bag (benign slide), none of the instances should be positive.

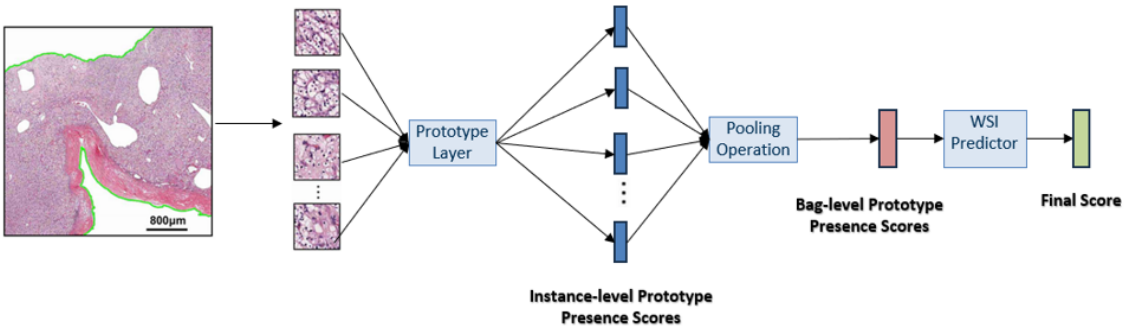


Figure 4.1: *Overview of the architectural adaptation of PIP-Net for an MIL setting. A complete WSI, which is divided into a bag of instances, is passed as input to the model. Each instance in the bag is processed to obtain normalized instance-level representations. These representations are aggregated into a single bag-level representation using a MaxPool function, over all instances in the bag. The bag-level vector serves as input to the sparse linear layer.*

4.1.1 Architectural Changes

To adapt PIP-Net to this setting, the core architecture remains intact, with a Convolutional Neural Network (CNN) backbone that processes each instance (patch) and the Softmax layer that matches patch features to learned prototypes. However, the following key changes were made to handle the MIL setting:

- **Instance-level Processing:** Each patch (instance) from a slide is processed independently through the CNN backbone to extract feature maps. These feature maps are then matched to the learned prototypes by applying a softmax over the D feature maps. We then apply a max-pooling operation per resulting feature map $z_{:,:,d}$ to obtain instance-level representations or prototype scores.

- **Max-Pooling for Bag-Level Representation:** Since the goal is to classify the entire bag, not individual instances, a max-pooling operation is applied across all instance-level prototype scores. This operation selects the maximum prototype score from all instances in the bag, aggregating the most relevant information. The idea is that the instance(s) that most closely match important prototypes should drive the bag-level decision, in line with the MIL assumption that only a few key instances are responsible for the overall bag label.

- **Sparse Linear Classifier:** he aggregated prototype scores are then passed through a sparse linear classifier, which assigns a label to the bag. As in the original PIP-Net, this layer uses sparse, non-negative weights to ensure interpretability, with only a few prototypes contributing to each decision.

Figure 4.3 depicts an overview of the architectural adaptation of PIP-Net for a Multiple Instance Learning scenario. By adapting PIP-Net to process instances independently, aggregate their prototype scores using max-pooling, and classify at the bag level, we enable the model to handle MIL tasks where instance labels are unavailable, but bag labels must still be predicted.

4.1.2 Adaptation of Loss Terms

To ensure that the model adapts properly to the MIL paradigm, it was also necessary to modify the loss functions that guide the learning process. The original PIP-Net uses a combination of alignment and tanh losses during self-supervised pre-training and a cross-entropy loss during classification training. In PIPMIL, these losses needed to be adapted to operate over bags of instances, not individual images, while maintaining the consistency and diversity of learned prototypes.

Alignment Loss:

In the self-supervised pre-training phase, the goal remains to learn prototypes that represent semantically meaningful parts of the images. The alignment loss in PIPMIL encourages patches from the same bag (e.g., patches of a malignant slide) to activate similar prototypes. However, unlike the single-image scenario, this alignment now needs to be enforced over multiple instances in a bag. The alignment loss is computed as the negative log of the dot product between the normalized prototype presence scores of two augmented views of patches, but aggregated over all instances in a bag. This ensures that similar parts across different instances contribute to the same prototype, promoting consistency.

Tanh Loss:

To avoid the model from assigning all patches in a bag to a small subset of prototypes (leading to trivial solutions), the tanh loss was adapted to ensure diversity in prototype usage across bags. This loss penalizes the model if certain prototypes are underutilized across a mini-batch of bags, ensuring that all prototypes are activated at least once in each mini-batch. The tanh loss, therefore, helps the model learn a diverse set of prototypes that capture a wide range of features from the WSIs.

4.2 Computational Restrictions

One of the key challenges encountered during the development of PIPMIL was the significant computational demands posed by WSIs. WSIs are extremely large images, often containing billions of pixels, and must be processed at high resolution to preserve diagnostic details. This results in bags containing thousands of patches, each of which needs to be processed through the CNN backbone and compared to learned prototypes. The sheer size and complexity of these bags lead to severe memory and computational bottlenecks, making it infeasible to train the model using traditional methods without adaptation. In this section, we will explore the computational restrictions faced and the methods developed to address these challenges.

Large Bag Size:

Each WSI, when divided into smaller patches, produces a large number of instances. For example, in larger datasets, a typical slide may contain around 10,000 patches. Processing all of these patches simultaneously through the CNN backbone requires a large amount of memory, as each patch must be stored, passed through multiple layers of the network, and compared to prototypes.

Feature Map Storage:

After passing patches through the CNN, the resulting feature maps are high-dimensional, and storing these feature maps for thousands of patches leads to significant memory overhead. Additionally, the model must store intermediate activations for backpropagation during training, further exacerbating memory consumption.

Computational Time:

Even if memory issues are addressed, the time required to process thousands of patches is still prohibitive. Each patch must be processed individually through multiple layers of the CNN, and the prototype matching and max-pooling operations must be repeated for each instance in the bag. This results in long training times, especially when backpropagating gradients through such a large number of patches.

4.3 Methods to Handle Computational Restrictions

To address the above computational challenges, several strategies were employed to reduce memory usage, accelerate training, and ensure the model could handle large-scale WSI datasets.

4.3.1 Patch Sampling

One of the fundamental strategies implemented to manage the computational load and facilitate the training of the PIPMIL model is the Patch Sampling method. This approach was devised to address the constraints associated with processing entire Whole Slide Images (WSIs) by selecting a manageable subset of patches from each bag and passing them through the network end-to-end. The Patch Sampling method ensures that the model can still capture essential features from WSIs without overwhelming computational resources.

Processing WSIs, with their gigapixel resolutions, in their entirety presents challenges, including excessive memory usage and prolonged processing times. Patch Sampling method focused on reducing the number of patches processed per bag while maintaining the integrity of the essential features required for accurate classification.

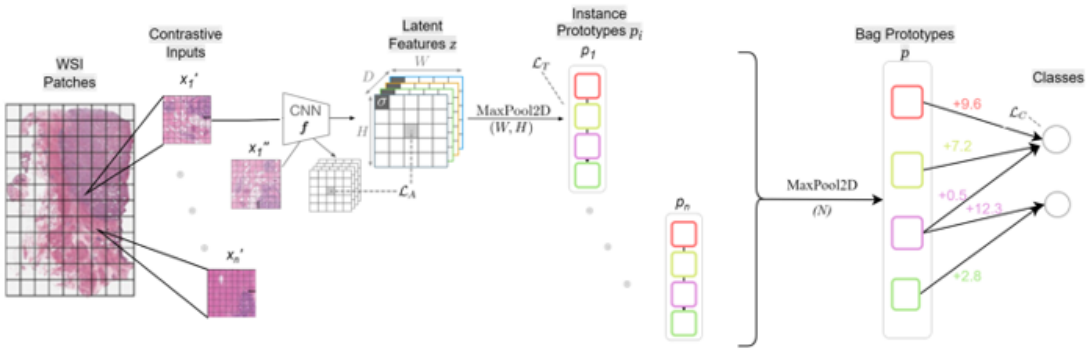


Figure 4.2: *Architectural adaptation for the Patch Sampling method. A subset of instances* $(x_1, x_2, ..., x_n)$ *are sampled randomly per bag. Each instance is passed through the network to obtain instance-level representations, which are aggregated into a single bag-level representation.*

Implementation Details

The implementation of the Patch Sampling method involving the following steps, attempts that the selected patches provide a representative subset of the WSI.

1. **Patch Extraction:** Each WSI is divided into non-overlapping patches of size 224x224 pixels. This size was chosen based on standard practices in histopathological image analysis, balancing resolution with computational feasibility. For the Camelyon16 dataset, this results in an average of 8,871 instances per slide.

2. **Random Sampling:** During each training epoch, a subset of patches is randomly selected from the total available patches in each WSI. The number of patches selected (N) is a hyperparameter that can be tuned based on the available computational resources and desired balance between efficiency and information retention. For this study, initial experiments were conducted with N set to 60, 80, and 100 patches per bag, to evaluate the impact of different sampling sizes on model performance.

3. **End-to-end Training:** The prototypes are pre-trained over all instances in a self-supervised manner utilizing PIP-Net's contrastive learning, optimizing them to learn semantic similarities that align with human visual perception, independent of the classification problem. During the training phase, each randomly sampled patch is passed through the ResNet18 backbone to extract feature maps. Subsequent Softmax and MaxPool layers yield the prototype presence scores for each patch. The prototype presence scores of all sampled patches are aggregated using a MaxPool layer to obtain a bag-level representation. The aggregated representation is passed through a sparse linear layer to produce the final classification output for the WSI. The network is trained using the selected patches, with the training process involving backpropagation and optimization of the model parameters.

To evaluate the effectiveness of the Patch Sampling method, experiments were conducted on the Bisque Breast Cancer and Camelyon16 datasets. The results demonstrated that the method significantly reduced memory usage and computational time but may compromise on classification performance.

Advantages and Limitations

The Patch Sampling method offers several advantages including:

- **Reduced Computational Load:** By processing only a subset of patches, the method significantly reduces memory usage and computational time, enabling the training of large models on standard hardware.
- **Flexibility:** The number of patches sampled per bag can be adjusted based on the available computational resources and specific requirements of the study.

However, this method also poses a number of limitations including:

- **Information Loss:** Sampling only a subset of patches may result in the loss of critical information, potentially impacting the model's performance, especially if the random samples are exclusive of features that correspond to the positive class.
- **Hyperparameter Tuning:** The number of patches to sample (N) is a hyperparameter that requires careful tuning to balance efficiency and information retention.

4.3.2 Pretrained Network for Patch Encoding

Although achieving end-to-end training over bags of instances, the random sampling approach falls short in ensuring complete representation of the entire WSI. In order to address this issue, we introduce an effective strategy employed to manage the computational load. This involves the use of a pretrained network for patch encoding. By leveraging a ResNet18 network pretrained on ImageNet, patches are efficiently encoded into lower-dimensional feature representations before being passed to the PIPMIL model. This approach significantly reduces the dimensionality of the data and the computational burden of processing raw patches directly.

Fine-tuning the Pretrained Network

The first step in this approach involves fine-tuning the pretrained ResNet18 network on the specific dataset used in this study. The ResNet18 model, originally trained on the ImageNet dataset, provides a robust starting point due to its ability to extract rich features from a wide variety of images. However, the characteristics of histopathological images differ from those in ImageNet, necessitating fine-tuning to adapt the network to the specific domain.

To achieve this, the ResNet18 network is subjected to self-supervised learning using the patches extracted from the WSIs. This involves training the network to recognize patterns and features

relevant to histopathological classification without explicit labels. The self-supervised learning framework helps the network learn semantic similarities that align with human visual perception, which is crucial for subsequent stages of the PIPMIL model.

During fine-tuning, several data augmentation techniques are employed to enhance the robustness of the network. These include random rotations, horizontal and vertical flipping, and color normalization. The use of these augmentations ensures that the network can generalize well to different variations in the patches, thereby improving its feature extraction capabilities.

Encoding Patches

Once the ResNet18 network is fine-tuned, it is used to encode all patches in the dataset into lower-dimensional feature representations. This step involves passing each patch through the pretrained network, which transforms the high-dimensional pixel data into a compact feature vector. The final layer of the ResNet18 network, which originally produces class probabilities, is removed, and the output of the second to last layer is used as the encoded representation.

Training the PIPMIL Classifier

The encoded patches are then used as inputs to the PIPMIL model, essentially replacing the backbone of the network. The number of encoded patches selected (N) is a hyperparameter tuned based on the available computational resources as well as balance desired between efficiency and information retention. In this approach, experiments were conducted with N set to 2000, 4000, 6000 and 10000 patches per bag, to evaluate the impact of different sampling sizes on model performance. The PIPMIL classifier operates on these lower-dimensional feature representations, leveraging the prototypes learned during the self-supervised pretraining phase. The classifier is trained to map the encoded patch features to bag-level predictions, determining whether the entire WSI corresponds to a positive or negative label.

During the training phase, the encoded feature vectors are kept fixed and are passed through the PIPMIL model to compute prototype presence scores for each instance. The instance-level scores are aggregated using a MaxPool layer to obtain a bag-level representation. This step ensures that the most relevant prototypes are highlighted in the final representation. The aggregated bag-level representation is passed through a sparse linear layer, which maps the prototypes to the final classification output. The training process utilizes standard backpropagation techniques with a learning rate schedule tailored for efficient convergence. The use of Adam optimizer with cosine annealing ensures that the model adapts dynamically to different stages of training, improving overall performance. As the encoding vectors are kept fixed, during the training phase, only the classifier is fine-tuned.

Advantages and Limitations

The use of a pretrained network for patch encoding offers several benefits:

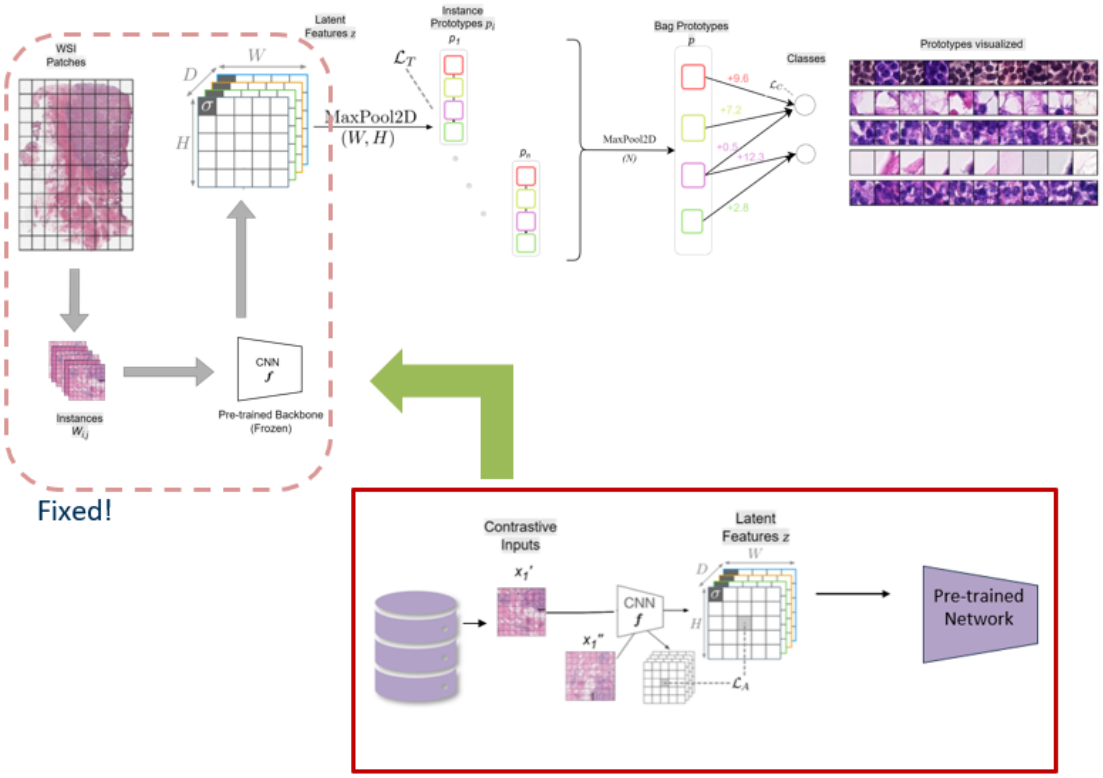


Figure 4.3: *Architectural adaptation for the Patch Encoding approach. The pre-trained CNN backbone extracts latent features from WSI patches, which are then used to compute instance-level prototype activations. The fixed backbone ensures consistency in feature extraction, while contrastive learning (bottom right) helps refine the latent space, encouraging similar patches to activate the same prototypes. Bag-level representations are formed through max pooling over instance prototypes, supporting interpretable classification.*

- **Reduced Memory Footprint:** By encoding patches into lower-dimensional feature vectors, the approach significantly reduces the memory requirements for processing large WSIs, allowing for an increased number of instances per bag.
- **Improved Efficiency:** The computational load is reduced as the model operates on compact feature representations rather than raw pixel data, and the increased number of instances increases chance of complete representation of the WSI.

However, this approach also presents certain challenges:

- **Potential Information Loss:** Reducing the dimensionality of patches might lead to the loss of fine-grained details that could be critical for accurate classification. Balancing the dimensionality reduction with the retention of essential features is crucial.
- **Lack of Fine-tuning Prototypes:** Keeping the encoded vectors fixed during the training of the classifier prevents prototypes from being fine-tuned. This can compromise on the predictive performance of the model.

4.3.3 Selective Instance Fine-tuning

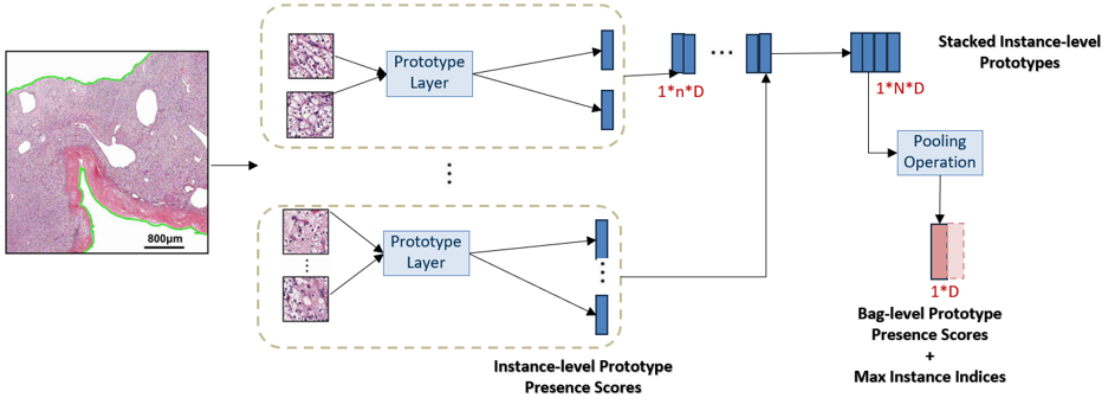
To effectively address the computational constraints associated with processing entire bags of instances, we implement a novel approach called Selective Instance Fine-tuning. This method aims to optimize the training process by focusing computational resources on the most impactful instances within each bag, thereby reducing the overall memory and processing requirements without sacrificing model performance.

The underlying premise of this approach is that not all instances within a bag contribute equally to the final prediction. In the context of Multiple Instance Learning (MIL), particularly for Whole Slide Image (WSI) classification, certain patches (instances) within a slide (bag) play a more critical role in determining the slide's label. The goal of this approach is to identify these key instances and prioritize them during the backpropagation phase of training. This is achieved through a two-step process: an initial forward pass to evaluate all instances, followed by a selective backward pass focusing only on the most relevant instances.

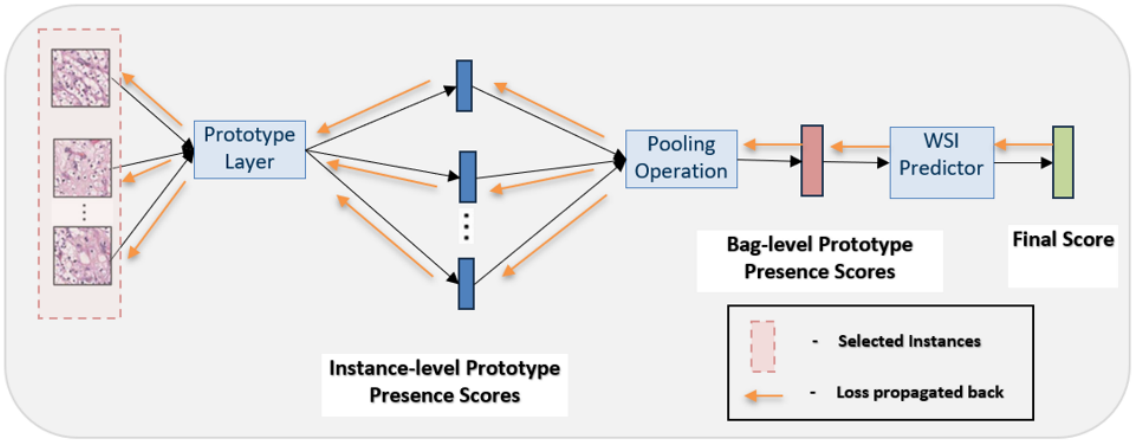
Implementation Details

The implementation of Selective Instance Fine-tuning involves several key steps:

1. **Forward Pass Over All Instances:** During the forward pass, the model processes each instance within a bag to compute instance-level prototype scores. This can be computationally intensive due to the large number of patches per WSI. To manage memory and processing constraints, the forward pass is divided into smaller chunks. Each chunk contains a subset of instances, which are processed sequentially. For each chunk, the model calculates the instance-level scores and temporarily stores these scores.



((a)) *Forward pass over all instances.*



((b)) *Selective backward pass over instances with gradients.*

Figure 4.4: *Architectural adaptation for the Selective Instance Fine-Tuning approach. (a) Forward pass over all instances to compute instance-level prototype presence scores, followed by pooling to obtain bag-level prototype scores. (b) Selective backward pass where only key instances, identified through max pooling, are used for gradient propagation, optimizing computational efficiency by focusing on the most influential patches.*

2. **Max-Pooling for Bag-level Scores:** After computing the instance-level scores for all chunks, a max-pooling operation is performed to aggregate these scores into a single bag-level score. This score represents the most significant instance-level contributions to the final prediction. The max-pooled score identifies the instances that have the highest impact on the bag-level prediction. These instances are considered the most relevant for the subsequent backpropagation step.

3. **Selective Backward Pass:** Rather than performing backpropagation on all instances within the bag, the selective backward pass focuses only on the instances that contributed to the max-pooled score. By limiting the backward pass to these key instances, the approach significantly reduces the computational burden. The gradients are computed and propagated only for the key instances, optimizing the use of computational resources.

Efficient chunk management is crucial for the success of this method. By dividing the forward pass into smaller, manageable chunks, the model can handle large batches without exceeding memory limits. Each chunk is processed independently, ensuring that the computational load is distributed evenly. Intermediate results from each chunk are combined to form the final max-pooled score.

5

Experimental Evaluation and Results

This chapter presents the experimental evaluation including datasets used in the evaluation, data preparation and preprocessing, the measures for assessing model performance, and the results of experiments conducted to evaluate PIPMIL's performance and interpretability.

5.1 Datasets

In this section, we describe the datasets used for training and testing PIPMIL's performance. We include details on each dataset's structure, class distribution, patch extraction, and preprocessing steps to facilitate multiple instance learning (MIL) settings. PIPMIL is evaluated on two datasets for whole-slide image classification. These are Camelyon16 dataset and Bisque Breast Cancer dataset.

5.1.1 Camelyon16 Dataset

The Camelyon16 dataset [2] is benchmark dataset used in computational pathology and contains Whole Slide Images (WSIs) from lymph node sections stained with Hematoxylin and Eosin (H&E) dye. The primary goal of this dataset is to detect metastatic breast cancer in the lymph nodes. Camelyon16 has a binary classification structure, with each slide labeled as either "benign" or "malignant". The dataset consists of 399 whole-slide images of $20x$ resolution, of which 239 are labeled normal (benign) and 160 are labeled tumor (malignant).

Data Preprocessing- Standard data preparation steps are followed from [19]. Each of the $20x$ resolution WSIs (bags) is divided into smaller patches (instances) of 224×224 pixels. Instances containing more than 70% of white space (background) were discarded. This results in 399 bags of variable number of instances per bag, with a mean of 8,871 instances and a standard deviation of 6,175 instances per bag. In addition, the bags with more than 20,000 instances were truncated to a maximum of 20,000 instances per bag, at random, in order to fit into the GPU memory, while

retaining maximum information. The positive instances (containing malignant tissue), however, are highly imbalanced as only 10% of the instances contain the malignant tissue. The dataset contains only slide-level labels, but patch-level labels are unavailable, making it well-suited for MIL. In experiments on the Camelyon16 dataset, image pairs are created by applying extensive data augmentations including random cropping, random rotations, horizontal and vertical flipping, and instance normalization.

5.1.2 Bisque Breast Cancer Dataset

Bisque dataset [10] contains 58 Hematoxylin and Eosin (H&E) dyed histology images from breast cancer biopsies, out of which 32 are benign, and 26 are malignant (containing at least one cancer cell), It provides a balanced number of WSIs for both classes. Each image is of size 896×768.

Data Preprocessing- For the Bisque dataset as well, standard data preparation steps are followed from [19]. Each of the slide images is divided into 32×32 patches (instances). Patches with at least 75% of the white pixels are discarded, resulting in 58 bags of various sizes, with a mean of 389 instances and a standard deviation of 117 instances per bag. For the subsequent experiments using the Bisque dataset, the image pairs are created by applying extensive data augmentations including random rotations, horizontal and vertical flipping, random staining augmentation, staining normalization, and instance normalization.

Table 5.1 provides a detailed comparison between Camelyon16 and Bisque Breast Cancer Datasets. Both datasets are widely used benchmarks in histopathology image analysis and are particularly suited to testing models that need to operate in a MIL framework, where labels are only available at the slide level rather than for individual image patches. The Camelyon16 dataset is a suitable dataset to assess the effectiveness of PIPMIL in handling large, high-resolution images typical of Whole Slide Images (WSIs) in the medical domain. In contrast, the Bisque dataset was chosen to test PIPMIL's adaptability and performance on smaller datasets.

Table 5.1: *Overview and comparison between Camelyon16 and Bisque Breast Cancer datasets used in PIPMIL experiments.*

	Camelyon (Camelyon16 Dataset)	**Bisque (Bisque Breast Cancer Dataset)**
Dataset Size	399 WSIs	58 WSIs
Class Split	239 benign, 160 malignant	32 benign, 26 malignant
Instance Size	224×224	32×32
Instances per bag	8871±6175	389±117

5.2 Experimental Setup

This chapter details the experimental setup used to evaluate the PIPMIL model on two benchmark datasets: the Bisque Breast Cancer Dataset and the Camelyon16 Dataset. Each dataset presented unique challenges and required specific model configurations, particularly in terms of computational handling. In this section, we focus on the model setup for each dataset, detailing the different strategies employed to handle computational demands and optimize performance.

5.2.1 Model Setting: Bisque

The Bisque Breast Cancer dataset was well-suited for an end-to-end training approach due to its smaller size and lesser number of instances per bag. In this configuration, each WSI, which was divided into instances, was processed through the model in a single, unified training pipeline without requiring additional sampling or selective processing methods.

For feature extraction, we utilize a ResNet-18 backbone with a modified first layer—a 3×3 convolution with stride 1—to match the smaller input patch size of the instances in this dataset. This backbone was pre-trained on the ImageNet dataset [5], providing robust initial weights. However, due to the distinct nature of histopathological data compared to natural images, the entire model, including the ResNet-18 backbone, is fine-tuned for this task. To ensure reliable and generalized results, we repeated each experiment five times, using a randomized 90-10 train-test split for each run, with the random seed varied across initializations. This setup allowed us to observe model stability and consistency across different data splits.

The ResNet-18 backbone was fine-tuned using the Adam optimizer with a learning rate of 0.0001, following a cosine annealing learning rate schedule. This gradually decreases the learning rate over time, helping the model to converge smoothly and avoid overfitting. The sparse linear layer connecting prototypes to classes was trained with a learning rate of 0.05. The prototypes were pre-trained for 10 epochs with a batch size of 10 bags, allowing the model to learn consistent and meaningful prototype representations before fine-tuning the classifier. The network was then fine-tuned for 60 epochs using a batch size of 5 bags to balance memory constraints while ensuring adequate data exposure.

5.2.2 Model Setting: Camelyon16

Due to the significantly larger size and higher computational demands of the Camelyon16 dataset, an end-to-end training setup was not feasible. Instead, three approaches were implemented to reduce memory and processing time requirements while preserving the accuracy and interpretability of the model. For all subsequent experimentation on the Camelyon16 Dataset, we retain the original train-test split across initializations [2], with the train set comprising 159 benign and 111 malignant WSIs and the test set comprising 80 benign and 49 malignant WSIs.

Patch Sampling

In this approach we use the convolutional backbones ResNet18 and ConvNeXt-tiny [15] indicated with R and C respectively. For ResNet18, we adopt the same configuration as used in [19] to maintain consistency and enable fair comparisons with ProtoMIL. In this setup, the width (W) and height (H) of the output feature maps are kept at 7×7. This resolution provides a balanced feature map size that retains spatial information while remaining computationally feasible. However, For ConvNeXt-tiny, we adjust the strides in the final layers, reducing them from 2 to 1. This modification increases the spatial resolution of the output feature maps from 7×7 to 26×26, providing a more fine-grained patch grid (z). The finer grid is intended to capture more localized features within each patch, allowing for more detailed prototype matching and patch similarity calculations, as recommended by the standard PIP-Net configuration [17].

In the self-supervised pre-training phase, we fine-tune the backbone across all instances in the dataset to establish meaningful and transferable representations. The pre-training process was for 10 epochs, with a batch size of 128 instances and a learning rate of 0.0005 in a cosine annealing schedule. For the patch sampling approach, we perform experiments varying the number of instances sampled per bag. This parameter allows us to test the effect of different instance counts on the model's efficiency and accuracy. Given the need to accommodate as many instances as possible in each bag, we set the batch size to 1 bag. After self-supervised pre-training, the sparse linear layer (classification layer) is trained with a learning rate of 0.05. The network undergoes fine-tuning for 60 epochs with a batch size of 1 bag per iteration, with the intent to learn from diverse patch arrangements within each slide while maintaining computational efficiency.

Patch Encoding

The patch encoding approach seeks to leverage all instances within a bag during training to minimize information loss, ensuring that the model can comprehensively learn from each patch in a Whole Slide Image (WSI). To accomplish this, the self-supervised pre-training phase is performed separate from the classifier's fine-tuning phase, allowing the model to generate high-quality embeddings for each patch independently of the final classification task. Similar to the setting in 5.2.2, in the self-supervised pre-training phase, a ResNet18 backbone is utilized to learn feature representations across all instances. The model is fine-tuned for prototype learning on the entire dataset, processing every patch (or instance) within each bag. This phase is run for 10 epochs with a batch size of 128 instances and a learning rate of 0.0005, scheduled with cosine annealing. After pre-training, all patches across the dataset are preprocessed through the fine-tuned backbone to obtain offline embeddings for each instance in every bag. This preprocessing generates fixed image embeddings that effectively capture high-level semantic information from each patch. These embeddings are stored offline and serve as the primary input to the classification model, avoiding the need to re-process patches through the backbone in subsequent training, thus significantly reducing computational overhead.

During the classification phase, these pre-computed embeddings are fed into PIPMIL's softmax layer, replacing the unnormalized prototype presence scores, to calculate normalized prototype presence scores for each patch. Importantly, these embeddings remain fixed throughout this phase and do not undergo updates during the backward pass, isolating prototype learning from the classification fine-tuning. For the patch encoding approach, we also perform experiments varying the number of instances sampled per bag, allowing us to test the impact of increasing instance counts on the model's predictive performance. We also experiment with the gradient clipping threshold, investigating the effect of explanation sizes on predictive performance. Additionally, to maximize the number of instances per bag, we set the batch size to 1 bag, allowing for the inclusion of all available patches within each WSI during classification fine-tuning. For the classification layer (the sparse linear layer), training is performed over 60 epochs with a batch size of 1 bag per iteration, using a learning rate of 0.05.

Selective Instance Fine-tuning

Selective Instance Fine-tuning is designed to optimize the fine-tuning of prototypes during the classification phase while incorporating information from all instances in each bag. This method combines the computational efficiency of selective training with the interpretability of PIPMIL by focusing on the most influential instances within each bag. Here, "influential" instances refer to those that contribute most significantly to the final bag-level representation. This helps facilitate effective prototype learning without exhaustive resource consumption.

As in the approaches discussed in sections 5.2.2 and 5.2.2, we employ a ResNet18 backbone that is pre-trained across all instances in a self-supervised pre-training phase. This backbone is initialized with weights from a pre-trained ResNet18 model and then fine-tuned on all instances in the dataset for 10 epochs, with a learning rate of 0.0001. During this phase, the model learns generalizable feature representations, aligning similar instances with consistent prototype activations.

After the pre-training phase, the model is further fine-tuned on the dataset for only 15 epochs, due to a limitation on computation runtime. With a batch size of 1 bag, and a learning rate of 0.05, we fine-tune the classification layer (the sparse linear layer). As discussed in section 4.3.3, to balance efficiency with thorough prototype representation, each fine-tuning epoch is split into two sequential phases: a forward pass over all instances in the bag computing the prototype presence scores for each instance and obtaining the indices of those instances that contribute most significantly to the final bag-level representation; and a second phase that performs a forward and backward pass over only the subset of instances identified in the first phase, updating the weights of those identified instances.

This selective training process reduces the computational load significantly by avoiding backpropagation through instances with minimal, or more specifically, no influence on the classification result, per epoch. This focuses fine-tuning efforts on instances critical for accurate predictions.

5.3 Evaluation Measures

In this section we will discuss the measures incorporated in the evaluation of PIPMIL to assess the model's strengths, weaknesses and capabilities. The evaluation of the PIPMIL model is based on a series of measures designed to assess both its predictive performance and interpretability. Predictive performance measures provide a quantitative assessment of the model's classification accuracy and reliability, while interpretability measures evaluate how effectively the model's outputs can be understood and trusted by domain experts.

5.3.1 Predictive Performance

In order to evaluate PIPMIL's predictive performance, standard classification measures are employed to assess how well the model distinguishes between different classes (benign and malignant) in WSI classification tasks. Specifically-

Accuracy:

Accuracy measures the proportion of correct predictions over all instances in the dataset. It provides a straightforward indicator of overall model performance. However, accuracy alone can be insufficient in the presence of class imbalance, as it does not distinguish between correct predictions in each class. For PIPMIL, accuracy provides an initial indication of the model's effectiveness in classifying WSIs but is supplemented by additional measures to address specific clinical concerns.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{5.1}$$

where TP refers to the number of true positives, TN is the number of true negatives, FP is the number of false positives and FN is the number of false negatives.

F1-Score:

The F1-score offers a measure to balance both Precision (ratio of true positives to all positive predictions) and the Recall (ratio of true positives to the total actual positives). In medical diagnosis especially, recall is crucial as it reflects the model's ability to identify all positive cases, minimizing the risk of missed diagnoses. The F1-score is the harmonic mean of precision and recall, balancing both measures to provide a single measure that accounts for false positives and false negatives. F1-score is particularly useful in situations with class imbalance. A high F1-score in PIPMIL indicates that the model achieves a good balance between avoiding false positives and minimizing false negatives.

$$F1score = 2 \times \frac{Precision \cdot Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN} \tag{5.2}$$

These predictive performance measures collectively assess PIPMIL's ability to accurately classify WSIs, with a focus on minimizing both false negatives (important for diagnostic sensitivity) and false positives (critical for specificity in clinical settings).

5.3.2 Interpretability Assessment

Interpretability is a core component of PIPMIL, as the model aims to provide transparent and understandable predictions. To evaluate interpretability, we examine quantitative measures as well as a qualitative assessment by a domain expert. These measures focus on assessing whether the model's learned prototypes align with clinically relevant features and if they provide meaningful insights for end-users, such as pathologists.

Quantitative Measures - Compactness

Quantitative measures assess the interpretability of PIPMIL's predictions by analyzing objective measures related to prototype utilization and compactness. Specifically, we look at prototype compactness, or explanation size. Compactness measures the model's efficiency in utilizing a small, focused set of prototypes for making predictions. In PIPMIL, the sparse linear classifier restricts the number of prototypes contributing to each class decision, resulting in compact and concise explanations. In general, a smaller explanation size indicates that fewer prtotypes were used to make the prediction. In particular, we look at both global and local explanation sizes.

1. **Global Explanation Size-** This refers to the total number of prototypes used in the classification across the entire dataset, or more specifically the number of prototypes in the model with at least one non-zero weight to the classifier.

2. **Local Explanation Size-** This is a measure of the number of prototypes that are relevant or activated in the prediction of a single instance or training example. We consider all relevant prototypes with a similarity >0.1.

Qualitative Assessment

Qualitative assessment involves manually examining the learned prototypes and their relevance to clinical diagnoses. This is achieved by visualizing prototypes, evaluating their alignment with clinically meaningful concepts, and ensuring that prototype contributions align with correct class predictions. For this assessment, we refer to the professional opinion of a domain expert (pathologist), with a focus on three key criteria-

1. **Relevance of Learned Prototypes-** One of the main qualitative interpretability checks is to assess whether the prototypes learned by PIPMIL correspond to meaningful and clinically relevant concepts in WSIs. Visual inspections are conducted by the pathologist to confirm if the prototypes match known diagnostic features. In particular, prototypes for malignant

cases should reflect histopathological artifacts (concepts) associated with malignancy, and similarly, prototypes for benign cases should reflect artifacts (concepts) associated with benign tissue.

2. **Prototype-Class Alignment-** This assessment criteria evaluates whether the model uses the appropriate prototypes to support each class decision. For instance, malignant prototypes should contribute positively to malignant predictions, and benign prototypes should be associated with benign predictions. Misalignment, where malignant prototypes contribute to benign classifications or vice versa, could indicate interpretability issues and misguidance in the model's reasoning. This alignment assessment is conducted by inspecting prototype activations for predictions across test cases.

3. **Error Analysis for Misclassified Slides-** Misclassifications are examined to determine if certain prototypes contributed incorrectly to the decision-making process. This involves identifying whether irrelevant prototypes were activated in misclassified instances and understanding the conditions under which these errors occur. Error analysis provides insights into potential refinements for prototype learning, ensuring that PIPMIL avoids activating irrelevant features in future cases.

5.4 Results

This section presents the outcomes of the experiments conducted to evaluate the effectiveness of PIPMIL across two datasets: the Bisque Breast Cancer Dataset and the Camelyon16 Dataset. The results are analyzed in terms of predictive performance, prototype compactness, and interpretability. Due to the characteristics of each dataset and the computational strategies applied, each subsection will address specific evaluation metrics relevant to that dataset.

5.4.1 Results on the Bisque Breast Cancer Dataset

The Bisque Breast Cancer Dataset allowed for an end-to-end training approach due to its relatively smaller size and instance count per slide. Here, we primarily focus on the predictive performance and prototype compactness of the model to assess PIPMIL's adaptability and performance on a smaller dataset.

In our experiments with the Bisque Breast Cancer dataset, we pre-processed the data as detailed in section 5.1.2. To evaluate the robustness and generalizability of our model, we performed a series of five independent training runs, each using a different train-test split. Each iteration of the experiment involved random sampling for train-test splits, maintaining a consistent ratio to prevent class imbalance. For each split, the model was trained on approximately 90% of the data, with the remaining 10% used exclusively for testing. This approach allowed for a comprehensive assessment of the model's stability across different subsets of the dataset.

Table 5.2 presents the averaged results across all five runs reporting the performance of the model in terms of predictive performance (Accuracy and F1-score) and prototype compactness. The performance of ProtoMIL on Bisque Breast Cancer dataset is presented as well.

Table 5.2: *Performance of PIPMIL and ProtoMIL on Bisque Dataset.*

Model	**Accuracy% (Mean ± Std.)**	**F1-Score% (Mean ± Std.)**	**Explanation Size**	
			Global	**Local**
PIPMIL	76.67%±2.9%	57.2%±34.7%	265	87±3
ProtoMIL	76.67%±2.2%	50.8%±28.5%	20	10

The results demonstrate that the PIPMIL model achieves high accuracy and F1-scores across multiple train-test splits, performing comparably to ProtoMIL on the Bisque dataset. These findings indicate that PIPMIL generalizes well to the dataset, suggesting that its prototype-driven approach is effective in distinguishing between classes, even under variations in data splits.

However, the compactness score reveals that the model relies on a relatively large number of prototypes to make predictions for each class. This high dependency on multiple prototypes per class may suggest limitations in the interpretability aspect of the model, as more prototypes can make it harder to directly trace individual predictions to specific image regions or concepts. A potential reason for this is that PIPMIL, built on the PIP-Net architecture, was originally optimized for patches of size 224224 pixels, which allow for richer feature representation and better-defined prototypes.

The interpretability of PIPMIL on the Bisque dataset is also challenging to evaluate due to the very small size of the instances, which are only 3232 pixels. At this resolution, the patches may lack sufficient detail for the model to learn meaningful prototypes that align with human-understandable features. In comparison, pretrained backbones like ResNet18 are typically trained on larger, higher-resolution images (e.g., 224224 pixels), with convolutional kernels optimized to capture distinct features at that scale. When applied to these smaller patches, the standard kernel sizes may struggle to capture the finer-grained features necessary for meaningful prototype development, potentially leading to the need for more prototypes to compensate for lost detail.

5.4.2 Results on the Camelyon16 Dataset

Given the large size and high computational demands of the Camelyon16 dataset, it was not feasible to run end-to-end training over the entire dataset. Therefore, three different methods were applied to handle this challenge- Patch Sampling, Patch Encoding, and Selective Instance Fine-tuning. Each method allowed for efficient processing of WSIs, and results are reported separately to highlight the performance differences across methods.

Quantitative Results: Patch Sampling

In our experiments with the Camelyon16 dataset using the Patch Sampling approach, we pre-processed the data following the steps outlined in Section 5.1.1. To evaluate the effects of different backbone architectures and bag sizes on model performance and computational feasibility, we experimented with two convolutional backbones—ResNet18 (R) and ConvNeXt-tiny (C)—while varying the number of instances per bag.

To establish a baseline, we first conducted experiments using 100 instances per bag, performing five independent training runs for each backbone. These experiments were executed on the original train-test split. The results of these baseline experiments demonstrated that both backbones (ResNet18 and ConvNeXt-tiny) were capable of processing bags with 100 instances efficiently within the available GPU memory, allowing successful completion of both pre-training and fine-tuning phases.

Following the baseline experiments, we increased the number of instances per bag to 500 to assess the performance and memory utilization of th backbones, at a higher bag size. This increase aimed to test the model's ability to handle a larger number of patches per slide, which could improve its capacity to capture more detailed spatial information across the whole slide. During these larger-scale experiments, we encountered differing memory limitations for the two backbones:

- **ConvNeXt-tiny (C):** When fine-tuning the ConvNeXt-tiny backbone with 500 instances per bag, we experienced a CUDA Out of Memory (OOM) error. This error suggests that ConvNeXt-tiny, due to its deeper and wider architecture, has a higher memory footprint per instance compared to ResNet18. The model's increased number of parameters and additional layers likely contributed to the OOM error when processing large numbers of instances, as each additional instance amplifies the cumulative memory load across the network.

- **ResNet18 (R):** Conversely, we were able to successfully fine-tune the ResNet18 backbone with 500 instances per bag. ResNet18's relatively smaller and more memory-efficient architecture allowed it to handle the increased instance count without exceeding GPU memory capacity. However, further increases in the number of instances per bag (beyond 500) eventually led to memory constraints and GPU OOM errors as well, underscoring the upper limit of ResNet18's handling capability under our experimental setup.

Table 5.3: *Performance of PIPMIL on Camelyon Dataset using the Patch Sampling approach.*

Model	**Accuracy% (Mean ± Std.)**	**F1-Score% (Mean ± Std.)**	**Explanation Size**	
			Global	**Local**
PIPMIL_C 100	62.46%±7.8%	48.68%± 5.8%	30±2	5±1
PIPMIL_R 100	59.69%± 2.3%	46.31%± 2.7%	40±2	9±1
PIPMIL_R 500	57.57%± 8.3%	35.47%± 17.88%	14±2	8±2

Table 5.3 presents the evaluation results of PIPMIL on the Camelyon16 dataset using the Patch Sampling approach. The analysis compares two model backbones—ConvNeXt and ResNet—across different instance sample sizes, measuring their performance in terms of accuracy, F1-score, and explanation size (number of prototypes activated per prediction). The results indicate that the ConvNeXt model consistently outperforms the ResNet model in both accuracy and F1-score, particularly when evaluated with a smaller subset of instances. This suggests that ConvNeXt may have a greater capacity to capture fine-grained details within patches, yielding more discriminative representations that enhance the model's predictive performance.

Interestingly, there is a counterintuitive observation regarding the impact of increasing the number of instances per bag on predictive performance. As the number of instances sampled increases for the ResNet model, a decline in accuracy and F1-score is observed. One possible explanation for this unexpected result could be that adding more instances may introduce redundant or non-informative patches into the model, which could dilute the significance of the most relevant prototypes. Excessive instance input can lead to prototype overlap or confusion, particularly if benign patches are introduced into a malignant bag (or vice versa), thereby reducing the model's ability to concentrate on clinically relevant patches.

In terms of explanation size, the ConvNeXt model with 100 instances yields a smaller explanation size than the ResNet model with the same number of instances. This compactness may be due to ConvNeXt's architecture, which inherently captures high-level features in a refined manner. For th ResNet models, when the number of instances is increased from 100 to 500, the explanation size decreases. This reduction in explanation size suggests that with more instances, the ResNet model can more effectively filter out redundant prototypes, activating only the most relevant ones for each prediction.

Future work could explore methods for adaptive instance sampling, where the model dynamically selects the most informative patches within each bag, potentially reducing explanation size while maximizing classification accuracy and F1-score.

Quantitative Results: Patch Encoding

In our experiments with the Camelyon16 dataset, the Patch Encoding approach was employed to reduce computational demands by leveraging pre-trained feature extractors, enabling the model to process large numbers of instances per bag without exceeding memory limitations. As described in Section 5.2.2, we used a ResNet18 backbone pre-trained on ImageNet to encode each patch into a lower-dimensional feature vector prior to prototype matching and classification. The objective of the Patch Encoding approach was twofold: (1) to maintain or improve predictive performance while reducing the computational burden, and (2) to test the model's scalability by progressively increasing the number of instances per bag. This encoding approach significantly reduced the dimensionality of each patch's representation, allowing the model to handle larger bags during training and inference without excessive GPU memory usage.

In this setup, each instance (patch) was passed through the pre-trained ResNet18 encoder to obtain a 512-dimensional feature vector, which is considerably smaller than the raw patch data. These encoded features were then used as inputs for the prototype layer, where each feature vector was compared to the learned prototypes to compute prototype activation scores. To aggregate information at the bag level, a max-pooling operation was applied across all instance-level prototype scores in each bag, yielding a compact bag-level representation that was then passed to the sparse linear classifier for slide-level prediction.To evaluate the effects of patch encoding on computational efficiency and predictive performance, we conducted experiments with different numbers of instances per bag: 2,000, 4,000, and 6,000 patches, respectively. Although the encoding allowed for an efficient representation of each patch, increasing the number of instances to 6,000 led to main memory limitations during training, highlighting an area for further optimization in memory management.

The model was fine-tuned for 60 epochs using a learning rate of 0.05 with momentum-based stochastic gradient descent. Additionally, we experimented with adjusting the gradient clipping threshold, increasing it from 0.001 to 0.005 to improve training stability. This change was introduced to help limit the size of gradient updates, which could reduce explanation complexity and potentially improve predictive performance. By constraining large gradients, gradient clipping helps mitigate the risk of exploding gradients, which can be particularly beneficial in models dealing with large-scale datasets like Camelyon16.

Table 5.4: *Performance of PIPMIL on Camelyon Dataset using the Patch Encoding approach.*

Model	**Accuracy% (Mean ± Std.)**	**F1-Score% (Mean ± Std.)**	**Explanation Size**	
			Global	**Local**
Encoding_2000	53.48%	31.81%	35	6±1
Encoding_2000 (cl.)	54.26%	41.58%	35	6±1
Encoding_4000	63.56%	42.10%	14	5±1
Encoding_4000 (cl.)	53.48%	59.67%	14	5±1
Encoding_6000	60.46%	30.13%	13	5±1
Encoding_6000 (cl.)	62.01%	26.86%	12	5±1

Table 5.4 gives us a overview of the impact of the Patch Encoding approach on the predictive performance and prototype compactness. From the table, it is evident that as the number of instances per bag increases, both accuracy and F1-score improve. This can be attributed to the model's enhanced exposure to a diverse range of patch features within each WSI, allowing it to better capture the subtle distinctions between classes, particularly in complex, high-variability datasets. The increased number of instances contributes to a more comprehensive representation of the entire slide, thereby improving the reliability of the model's predictions.

Additionally, the table shows that the compactness of explanations improves as the number of instances increases, meaning that fewer prototypes are needed to make accurate predictions. This is likely because a greater number of instances per bag allows the model to rely on more focused

prototypes that capture essential diagnostic features, reducing the need for redundant or overlapping prototypes. This improved compactness enhances interpretability by allowing the model to make predictions based on a smaller, more representative set of prototypes. Furthermore, we observe that increasing the gradient clipping threshold slightly improves predictive performance. However, this change does not significantly affect prototype compactness as initially expected.

Despite these improvements, the overall predictive performance of the model with Patch Encoding still falls slightly short. This shortfall may stem from the loss of finer details in the encoding process, as compressing patch information could result in less specific features that, while computationally efficient, may not fully capture all relevant diagnostic nuances needed for high accuracy. Further optimization of the encoding process or inclusion of complementary strategies, such as selective instance training, may be necessary to close this performance gap.

Quantitative Results: Selective Instance Fine-Tuning

The goal of Selective Instance Fine-Tuning was to maximize the use of available prototypes while maintaining computational efficiency and achieving strong predictive performance. This approach was particularly critical for handling large bags of instances in the Camelyon16 dataset without encountering memory issues. The data pre-processing steps are detailed in Section 5.1.1.

In this approach, model training proceeded in a structured manner, split into distinct phases. The model was first pre-trained in a self-supervised manner, allowing the model to learn meaningful and semantically consistent prototypes without requiring bag-level labels, improving generalizability and interpretability. This was followed by the fine-tuning process in two phases- a forward pass was performed across all instances in each bag. During this pass, the model computed prototype presence scores for each instance, followed by a max-pooling operation to create a single bag-level representation vector, and instances that contributed most significantly to the max-pooled representation were identified. This was followed by a forward and backward pass over these instances, updating the network's weights based solely on these influential patches.

Due to computational limitations, the model was trained for a total of 15 epochs. With each epoch processing approximately 20,000 instances per bag, training required around 12 hours per epoch, while processing 1,000 instances per bag required around 2 hours per epoch. This reduction allowed the model to handle large bags of instances without encountering CUDA out-of-memory (OOM) errors, improving both efficiency and scalability.

Table 5.5 presents the results of training PIPMIL with two different instance configurations per bag: 1,000 instances and 20,000 instances. While increasing the number of instances per bag yields a slight decrease in accuracy, it leads to an improvement in the F1-score. This improvement is particularly significant, as a higher F1-score aligns well with our objective of maximizing the model's ability to accurately detect malignant cases while minimizing false positives and false negatives.

In addition to evaluating PIPMIL, we compared its performance on the Camelyon16 dataset to other part-prototype models, ProtoMIL and PMIL, which serve as baselines for interpretability

Table 5.5: *Performance of PIPMIL on Camelyon Dataset using the Selective Instance Training approach.*

Model	**Accuracy% (Mean ± Std.)**	**F1-Score% (Mean ± Std.)**	**Explanation Size**	
			Global	**Local**
SIL_1000	63.56%	7.84%	26	17±2
SIL_20000	52.71%	64.32%	27	24±4
ProtoMIL	87.29%	2.7%	20	10
PMIL	87.5%	-	128	-
PMIL (-Backbone)	92.5%	-	128	-

in MIL settings. The results indicate that PIPMIL's predictive performance trails behind these models, particularly in terms of overall accuracy. However, it is important to highlight that unlike ProtoMIL and PMIL, PIPMIL does not regularize interpretability on a class-specific level, potentially accounting for some of the observed differences in performance. Both ProtoMIL and PMIL use explicit interpretability regularization strategies that assign class-specific prototypes, which may enhance their class-discriminative power but could potentially constrain flexibility in prototype learning.

Furthermore, PMIL reports results both with and without fine-tuning the CNN backbone. Interestingly, the results indicate that without fine-tuning, PMIL achieves improved predictive performance. This observation suggests that for part-prototype models in MIL settings, backbone fine-tuning might introduce noise or lead to overfitting on smaller regions, possibly detracting from generalizable feature learning. This insight underscores the importance of carefully balancing backbone tuning and prototype alignment in PIPMIL for optimal performance in WSI classification tasks.

5.4.3 Qualitative Results

The interpretability of PIPMIL's predictions is one of the model's primary advantages, allowing it to provide explanations for its classification decisions through visual prototypes. To evaluate this interpretability, we conducted a qualitative assessment of the learned prototypes in collaboration with a domain expert in pathology. This assessment aimed to determine whether the prototypes align with clinically relevant tissue patterns and if they provide meaningful insights that could aid in diagnostic decision-making.

Preliminary Assessment of Learned Prototypes

The preliminary interpretability assessment was structured to examine whether the prototypes corresponded to diagnostically significant features within Whole Slide Images (WSIs). Each prototype represents a distinct pattern learned by the model, and these patterns are expected to map onto identifiable histopathological features that are clinically relevant. In this assessment, we

conducted an in-depth review of PIP-Net's prototype learning capabilities, specifically focusing on the model's ability to capture and represent meaningful, semantically distinct features from histopathological images. To evaluate the interpretability and clinical relevance of the learned prototypes, a pathologist examined the entire set of 512 prototypes generated during the pre-training phase.

The pathologist analyzed each prototype individually, identifying common patterns and grouping them into five primary lexical categories based on visual and morphological similarities. These categories are as follows:

1. **Carcinoma:** Prototypes representing malignant characteristics, such as irregular cell shapes, large nucleus, high nuclear density, irregular nuclear margin, multiple nucleoli and atypical mitotic figures. These features are hallmarks of cancerous tissue and align well with expected patterns in metastatic regions.

2. **Lymphocytes:** Prototypes representing lymphocyte clusters. Lymphocytes are generally associated with benign tissue, as they are part of the immune response; however, the pathologist noted that lymphocytes can also be dispersed within metastatic regions, particularly in areas where immune responses are active. This double association requires contextual analysis within surrounding cell structures to interpret their relevance accurately.

3. **Stroma:** Prototypes representing stromal tissue, characterized by connective tissue and extracellular matrix regions. These areas are essential for understanding the tissue's structural context and provide insight into regions of non-cancerous tissue.

4. **Red Blood Cells (RBCs):** Prototypes representing RBC clusters. RBCs often appear within blood vessels or regions affected by hemorrhage. Although not directly indicative of malignancy, RBC presence can add context to tissue vascularity and inflammation.

5. **Blank:** Prototypes representing background or empty spaces. These "blank" prototypes primarily capture edges, spaces left by fat vacuoles, or slide artifacts, which do not contain diagnostically relevant tissue features.

Figure 5.2 illustrates a distribution of the prototypes across the lexical categories. It is observed that a large proportion of the prototypes represent concepts that align with a pathologist's visual perception of lymphocytes. Approximately 22% of the prototypes did not appear to align with a single, human-perceivable concept. The pathologist identified these as either unclear or "mixed" prototypes, representing overlapping features from multiple lexical categories (e.g., lymphocytes interspersed within carcinoma regions or regions with mixed stromal and carcinoma features). These ambiguous prototypes may represent areas where the model has generalized features that are not distinct enough for classification, possibly due to limitations in the dataset or the prototype learning process.

The presence of such mixed prototypes highlights the complexity of histopathological interpretation, where certain features are not exclusively characteristic of one category. For instance,

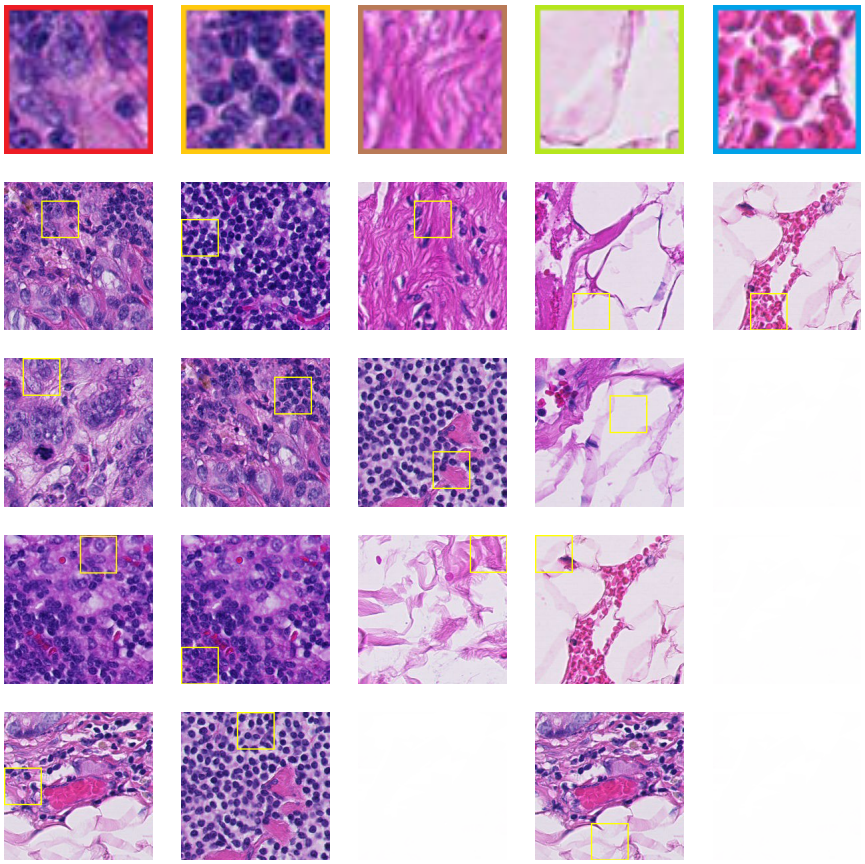


Figure 5.1: *Examples of prototypes discovered in this dataset. The first row displays examples of prototypes bearing semantic similarity to different clinically relevant concepts namely- Carcinoma, Lymphocyte, Stroma, Blank, and RBCs, respectively. The image patches below each prototype illustrate exemplary image patches where these prototypes are most activated.*

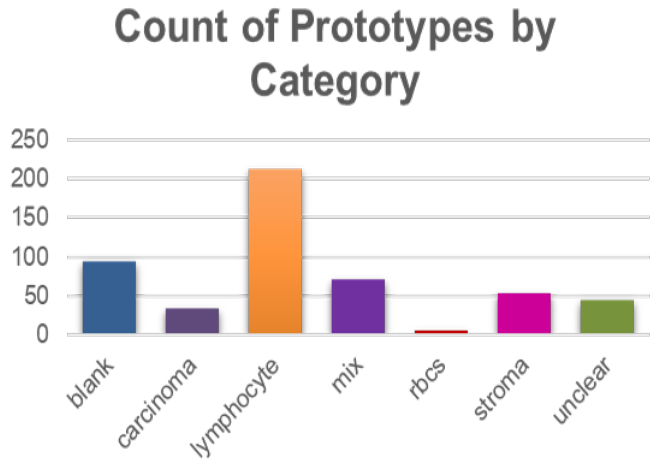


Figure 5.2: *Distribution of learned prototype across the lexical categories.*

while lymphocytes are often associated with benign tissue, they are also found in areas of metastasis where immune response activity is present. Similarly, regions classified as "mix" prototypes cannot be dismissed without considering surrounding cellular context, as adjacent structures may influence their interpretation. The pathologist emphasized that these prototypes, though not clearly defined, might still hold diagnostic relevance when viewed in combination with other prototype activations within a slide.

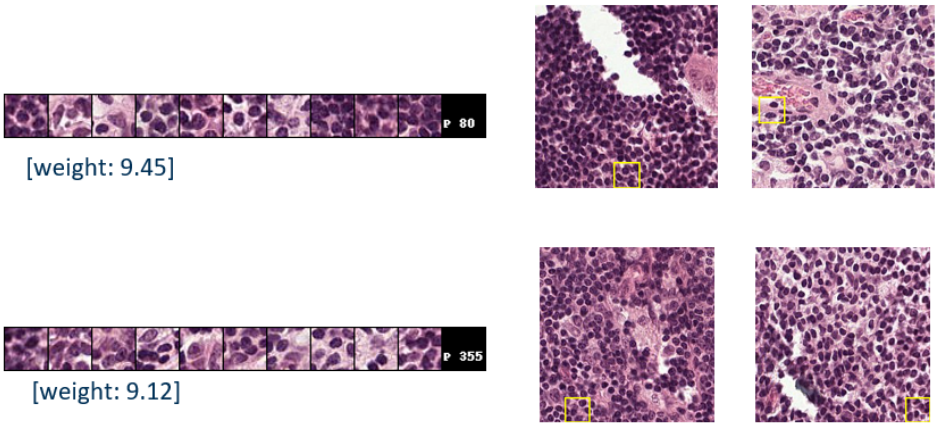
The pathologist's feedback suggests that PIP-Net has successfully learned distinct and clinically relevant prototypes in the majority of cases, particularly in categories like carcinoma and stroma. The model's ability to group similar histopathological patterns into meaningful prototypes validates its interpretability and suggests that the prototypes align closely with visual cues used by pathologists in manual diagnosis. However, the mixed prototypes reveal areas where the model could benefit from enhanced refinement or additional context to differentiate overlapping features.

Qualitative Interpretability Assessment

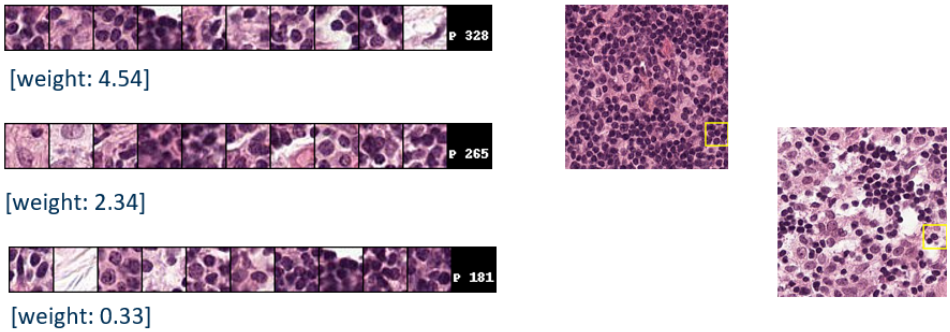
A subsequent qualitative interpretability assessment was conducted to validate the relevance and clinical significance of PIPMIL's learned prototypes. This process involved close collaboration with a domain expert to evaluate the model's prototype-based explanations and assess whether the identified regions corresponded to meaningful visual patterns in pathology. The assessment was carried out in two phases:

1. **Prototype Relevance Evaluation:** For each class (e.g., benign and malignant), the five most relevant prototypes were selected based on their weight in the sparse linear classifier. These prototypes represent the most influential features contributing to each class prediction. For each of these key prototypes, the top 10 image patches were identified based on activation scores, representing instances where each prototype was most prominently activated. These patches were reviewed by the domain expert to assess whether the visual characteristics matched expected features of benign or malignant tissue. The expert assessed the interpretability of these prototypes by examining whether they highlighted diagnostically relevant regions, such as nuclear morphology, cellular density, or atypical structures. This step ensured that PIPMIL's prototypes corresponded to visually perceptible, clinically relevant features, thereby enhancing the model's interpretability and trustworthiness.

2. **Instance-Level Analysis of Crucial Regions:** For more in-depth interpretability, crucial instances were selected from each WSI based on their overall contribution to the bag-level prediction. These instances were chosen by identifying patches with high prototype activation scores and weighting them by the prototype's importance in the final classification. Within these crucial instances, the most activated prototypes were determined by calculating the product of the activation score and the prototype's classifier weight. This approach allowed us to focus on instances where prototype activations were both high and influential in the model's decision-making process. The domain expert then reviewed these crucial



(a) Visually similar prototypes



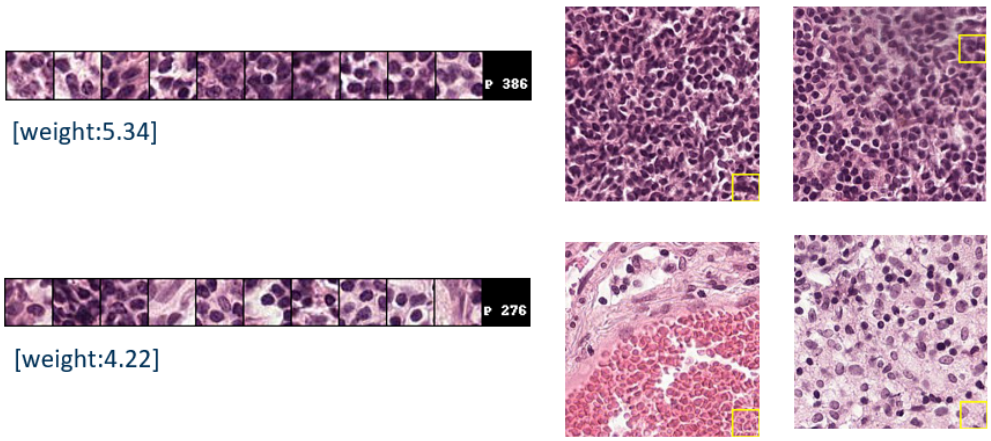
(b) Visually dissimilar prototypes

Figure 5.3: *The 5 most relevant prototypes for the benign class. The two prototypes on the top are deemed to be good prototypes as the top 10 image patches where prototype was most activated all align with the human visual perception of lymphocytes. The prototypes at the bottom, although visaully dissimilar are still deemed good prototypes as the top image patches where prototype most activated correspond to the benign class.*

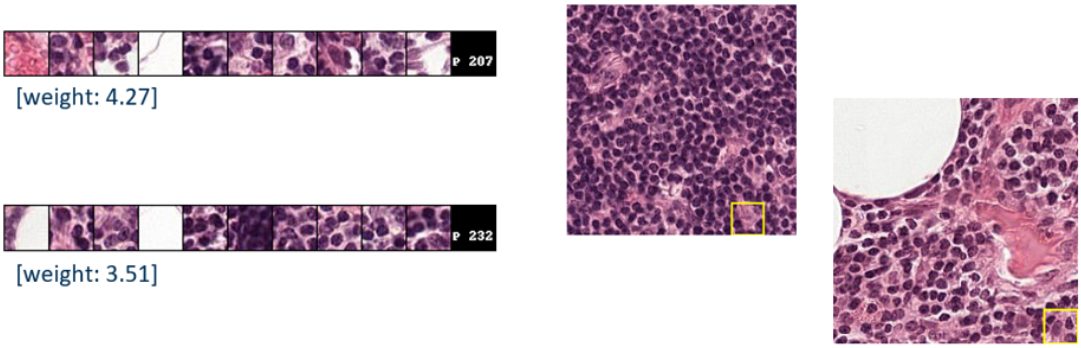
patches, assessing the correctness of prototype activations by examining if the highlighted regions displayed visually significant patterns. For example, in malignant cases, the expert looked for characteristics like nuclear pleomorphism, high nuclear-to-cytoplasmic ratio, or abnormal cell clustering. This evaluation verified whether the activated prototypes aligned with known diagnostic features, further validating the model's interpretability.

We examine some crucial cases as seen in figure 5.5, which illustrates a rightly predicted instance. While the image is primarily a benign region, we have some malignant prototypes also activating in the same region. However, upon inspection, it is clear that the activated patches contain artifacts that can be mistaken for a concept that aligns with the malignant class. Moreover, the prototypes have a low score and hence do not contribute much to the prediction.

In another crucial example that we deem a model failure, as seen in figure 5.6, the image patch is a benign instance. It is correctly predicted by the model as a benign instance. However, we



(a) Visually similar prototypes



(b) Visually dissimilar prototypes

Figure 5.4: *The 5 most relevant prototypes for the malignant class. a) Although the top 10 image patches where prototypes was most activated all appear semantically similar, the prototypes are deemed bad as they activate regions that correspond to the benign class. The prototypes at the bottom, although visaully dissimilar are still deemed good prototypes as the top image patches where prototype most activated correspond to the benign class.*

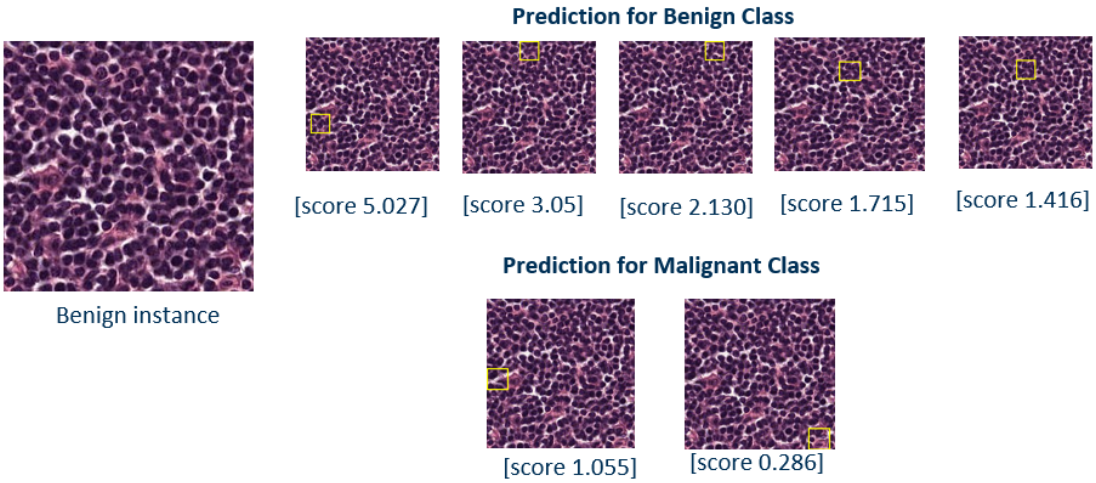


Figure 5.5: *Prototype activations for a benign instance. The top row depicts the regions where prototypes relevant for the benign class are most activated in the image. The bottom image patches indicate regions where prototypes relevant for the malignant class are most activated in the image.*

deem this example a model failure as the most activated malignant prototypes are also activated in a benign region. Moreover, the malignant prototypes have a high score as well, indicating that they are relevant to the prediction of the instance. Such an example however is not so frequently observed in the analysis.

To conclude, the findings presented demonstrate the effectiveness and limitations of the PIPMIL model across various configurations and datasets. Through rigorous testing on the Bisque and Camelyon16 datasets, PIPMIL showcased its capacity to perform interpretable WSI classification by leveraging prototype-based representations within a Multiple Instance Learning framework. The end-to-end training on the Bisque dataset highlighted PIPMIL's strong performance in settings with manageable computational requirements, while the implementation of computational strategies—such as patch sampling, patch encoding, and selective instance learning—enabled the model to handle the more demanding Camelyon16 dataset. These approaches balanced computational efficiency with interpretability, with an observed trade-off in accuracy but an improvement in the F1-score, aligning well with the high-stakes demands of medical imaging where both precision and recall are paramount. Comparisons with baseline models like ProtoMIL and PMIL further emphasized areas where PIPMIL excels and where it falls short, particularly regarding class-level interpretability constraints. These results not only validate the PIPMIL model as a viable and interpretable solution for WSI classification but also point to potential enhancements that could bridge existing performance gaps, ultimately supporting PIPMIL's applicability in real-world diagnostic environments.

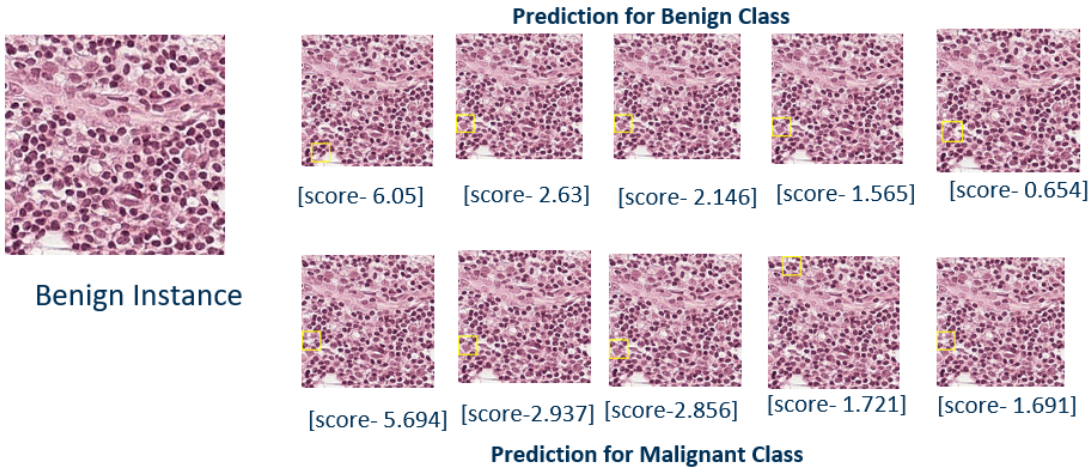


Figure 5.6: *Prototype activations for a benign instance. The top row depicts the regions where prototypes relevant for the benign class are most activated in the image. The bottom image patches indicate regions where prototypes relevant for the malignant class are most activated in the image.*

6

Discussions

In this chapter, we discuss the key findings of the experimental evaluation and analyze the strengths and limitations of PIPMIL. We explore the impact of adapting PIP-Net for Multiple Instance Learning (MIL), highlight the limitations identified during the study, and discuss the practical implications of our results. Finally, we outline possible directions for future work that can further enhance PIPMIL's applicability and performance in Whole Slide Image (WSI) classification tasks.

6.1 Adaptation of PIP-Net for Multiple Instance Learning

Adapting PIP-Net to an MIL setting presented unique challenges and required substantial architectural modifications to make the model suitable for WSI classification. PIP-Net's original architecture was designed for single-image classification, leveraging intuitive prototypes for interpretable predictions. However, the MIL paradigm required extending this approach to handle bags of instances (patches) rather than single images. This shift in structure allowed PIPMIL to operate on WSIs effectively by aggregating information from individual patches to form a single prediction for the entire slide.

Key changes included the integration of a max-pooling operation to aggregate instance-level prototype scores and compute a bag-level representation. This adjustment allowed the model to capture the most relevant features across instances, aligning with MIL's assumption that only certain patches in a slide drive the overall diagnosis. Additionally, the alignment and tanh loss terms, initially developed for single-image prototype consistency, were modified to ensure that meaningful prototypes emerged across the heterogeneous instances within each bag.

6.2 Limitations of PIPMIL

While PIPMIL offers a promising framework for interpretable WSI classification, several limitations emerged during the study, reflecting areas where the model can be further refined.

1. **Predictive Performance in Large-Scale Datasets:** Although PIPMIL demonstrated reasonable accuracy on smaller datasets, it struggled to match the performance of models like ProtoMIL and PMIL on larger datasets such as Camelyon16. The absence of class-specific interpretability regularization may have contributed to this performance gap. Unlike ProtoMIL and PMIL, which enforce prototype relevance to specific classes, PIPMIL assigns prototypes in a more generalized manner. As a result, PIPMIL may lack the same level of class-discriminative power, affecting its accuracy in complex datasets with high inter-class similarity.

2. **Computational Constraints:** Training PIPMIL on large WSI datasets was computationally intensive. Processing thousands of patches per slide required significant memory resources, which limited our ability to train the model in an end-to-end fashion on larger datasets. Although techniques like patch sampling, patch encoding, and selective instance learning (SIL) were implemented to manage these constraints, they introduced trade-offs, such as reduced spatial coverage or loss of fine-grained details. Consequently, PIPMIL's efficiency and scalability remain limited, posing challenges for deploying it in high-throughput clinical settings where large datasets and high resolution are standard.

3. **Adoption of Suitable Loss Terms:** In PIPMIL, we observed a significant drop in explanation size due to the model's inability to activate prototypes consistently across all instances within large bags. In a standard PIP-Net setting, the tanh loss function ensures that, within a mini-batch of 64 images, each prototype is activated at least once, encouraging a balanced and diverse use of prototypes across different image regions. This mechanism works well for traditional image classification tasks, where each instance has a reasonable chance of activating prototypes, thanks to the relatively smaller batch sizes and lower variability among instances. However, in PIPMIL, the large bag size—with an average of 8,000 instances per bag—leads to a substantial proportion of instances not activating any prototype. Specifically, approximately 7,500 out of 8,000 instances in each bag may not activate a single prototype. This issue arises because the tanh loss function, originally optimized for mini-batches of 64 images, fails to effectively scale to large bags, where only a small subset of instances contain features relevant enough to activate prototypes. The disparity between the output dimensions of the CNN backbones (768 for ConvNeXt and 512 for ResNet) and the smaller batch size requirement in the original PIP-Net setup exacerbates this issue. Since the tanh loss encourages each prototype to activate at least once per batch, it performs adequately when the number of features (e.g., 512 or 768 channels) is greater than the batch size (e.g., 64). However, when extended to bags of thousands of instances, this constraint

leads to excessive sparsity in prototype activations, as the vast majority of instances do not meet the threshold for activation under the tanh regularization scheme.

6.3 Practical Implications and Future Scope

The findings from this study have significant implications for the practical deployment of interpretable MIL models like PIPMIL in clinical settings, as well as for further research aimed at refining such models.

6.3.1 Implications for Clinical Deployment:

PIPMIL's interpretability, enabled by its prototype-based design, is advantageous for clinical applications where transparent and interpretable AI is critical. By identifying human-understandable features and localizing them within WSIs, PIPMIL allows pathologists to trace back its predictions to distinct patterns, aiding in trust and validation. However, given PIPMIL's computational requirements, practical deployment in high-throughput clinical workflows may require additional optimizations, such as real-time patch selection or more efficient encoding mechanisms, to ensure feasible processing times. Further, adapting PIPMIL to enforce class-specific interpretability constraints could make it more useful in tasks requiring fine-grained, class-dependent explanations.

6.3.2 Future Directions for PIPMIL:

- **Batch-Level Diversity Constraint:** Introduce a diversity constraint that enforces each prototype to be activated across a set proportion of instances in a bag, rather than just ensuring activation within a mini-batch of images. This modification would adapt the diversity mechanism of the tanh loss to the MIL framework, accounting for the large bag size by requiring each prototype to cover a minimum percentage of instances in each bag.
- **Bag-Wide Regularization for Prototype Coverage:** Implement a bag-wide regularization term that penalizes bags where prototypes are underutilized across instances. This regularization term would encourage each bag to achieve a balanced prototype activation by penalizing bags where a large proportion of instances do not activate any prototype.
- **Instance-Level Interpretability for Finer Diagnosis:** While the MIL framework inherently focuses on bag-level classification, future adaptations of PIPMIL could explore methods for finer-grained instance-level interpretation. Techniques such as instance selection or localized prototype matching could help PIPMIL provide more detailed insights into individual patches within a WSI, supporting clinicians in pinpointing specific regions of concern.

In conclusion, PIPMIL represents a promising step towards interpretable and efficient MIL for WSI classification, particularly in high-stakes medical imaging applications. By addressing the limitations identified and pursuing enhancements in interpretability, computational efficiency, and data integration, future iterations of PIPMIL could significantly improve its applicability and impact in clinical pathology.

7

Conclusion

In this thesis, we introduced PIPMIL: an interpretable model for Whole Slide Image (WSI) classification in a Multiple Instance Learning (MIL) setting. The primary goal of PIPMIL is to extend the prototype-based interpretability of PIP-Net to handle large-scale, high-dimensional WSI data while retaining clinical relevance and computational feasibility. Our approach leverages the core principles of PIP-Net but adapts the architecture and loss functions to the unique challenges posed by MIL, where only bag-level labels are available, and instances within each bag (patches of WSIs) contribute variably to the final prediction.

To achieve this, we implemented several key modifications:

1. **Adaptation for MIL:** PIPMIL was restructured to process instances within each bag individually, applying a max-pooling aggregation of instance-level prototype scores to produce a coherent bag-level representation. This design aligns with the MIL paradigm, where positive bag classification hinges on the presence of at least one positive instance.

2. **Self-Supervised Prototype Learning:** We extended PIP-Net's self-supervised alignment and tanh loss functions to adapt to large bags, ensuring meaningful prototype activation across diverse patches within WSIs. These loss functions helped PIPMIL learn prototypes that capture relevant features, enhancing both interpretability and diagnostic value.

3. **Computational Optimizations:** To address the high computational demands of WSIs, we incorporated strategies like Patch Sampling, Patch Encoding, and Selective Instance Learning (SIL). These techniques allowed PIPMIL to handle large-scale data efficiently while retaining high classification performance and interpretability.

Our experimental evaluation demonstrated that PIPMIL achieves competitive performance in WSI classification tasks. The model's accuracy and F1-score indicate its ability to make reliable predictions, while its interpretability metrics confirm that the learned prototypes represent meaningful features that can be used by clinicians for validation. However, the comparison with other MIL approaches highlighted areas for improvement, particularly in terms of prototype specificity for class-based interpretability.

This work presents PIPMIL as a promising solution for interpretable MIL in high-stakes applications, such as computational pathology. The thesis concludes that while PIPMIL effectively extends PIP-Net to an MIL framework, further improvements—such as class-specific prototype regularization and advanced aggregation methods like attention pooling—could enhance its performance and interpretability. This foundation paves the way for future research into hybrid IS and ES paradigms for interpretable WSI analysis, ultimately advancing the role of explainable AI in medical imaging.

Bibliography

[1] Peter Auer. "On learning from multi-instance examples: Empirical evaluation of a theoretical approach". In: *Proceedings of the fourteenth international conference on machine learning*. 1997, pp. 21–29.

[2] Babak Ehteshami Bejnordi et al. "Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer". In: *Jama* 318.22 (2017), pp. 2199–2210.

[3] Gabriele Campanella et al. "Clinical-grade computational pathology using weakly supervised deep learning on whole slide images". In: *Nature medicine* 25.8 (2019), pp. 1301–1309.

[4] Nicolas Coudray et al. "Classification and mutation prediction from non–small cell lung cancer histopathology images using deep learning". In: *Nature medicine* 24.10 (2018), pp. 1559–1567.

[5] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[6] Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. "Solving the multiple instance problem with axis-parallel rectangles". In: *Artificial intelligence* 89.1-2 (1997), pp. 31–71.

[7] Yuling Fan et al. "MI-Net: A deep network for non-linear ultrasound computed tomography reconstruction". In: *2020 IEEE International Ultrasonics Symposium (IUS)*. IEEE. 2020, pp. 1–3.

[8] Navid Farahani, Anil V Parwani, and Liron Pantanowitz. "Whole slide imaging in pathology: advantages, limitations, and emerging perspectives". In: *Pathology and Laboratory Medicine International* (2015), pp. 23–33.

[9] Ji Feng and Zhi-Hua Zhou. "Deep MIML network". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 31. 1. 2017.

[10] Elisa Drelie Gelasca et al. "Evaluation and benchmark for biological image segmentation". In: *2008 15th IEEE international conference on image processing*. IEEE. 2008, pp. 1816–1819.

[11] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

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

[13] Longlong Jing and Yingli Tian. "Self-supervised visual feature learning with deep neural networks: A survey". In: *IEEE transactions on pattern analysis and machine intelligence* 43.11 (2020), pp. 4037–4058.

[14] Daisuke Komura and Shumpei Ishikawa. "Machine learning methods for histopathological image analysis". In: *Computational and structural biotechnology journal* 16 (2018), pp. 34–42.

[15] Zhuang Liu et al. "A convnet for the 2020s". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 11976–11986.

[16] Oded Maron and Tomás Lozano-Pérez. "A framework for multiple-instance learning". In: *Advances in neural information processing systems* 10 (1997).

[17] Meike Nauta et al. "PIP-Net: Patch-Based Intuitive Prototypes for Interpretable Image Classification". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 2744–2753.

[18] Liron Pantanowitz et al. "Review of the current state of whole slide imaging in pathology". In: *Journal of pathology informatics* 2.1 (2011), p. 36.

[19] Dawid Rymarczyk et al. "Protomil: Multiple instance learning with prototypical parts for whole-slide image classification". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2022, pp. 421–436.

[20] David Tellez et al. "Whole-slide mitosis detection in H&E breast histology using PHH3 as a reference to train distilled stain-invariant convolutional networks". In: *IEEE transactions on medical imaging* 37.9 (2018), pp. 2126–2136.

[21] Tongzhou Wang and Phillip Isola. "Understanding contrastive representation learning through alignment and uniformity on the hypersphere". In: *International conference on machine learning*. PMLR. 2020, pp. 9929–9939.

[22] Nils Weidmann. "Two-level classification for generalized multi-instance data". In: *Master's Thesis. Albert-Ludwigs-Universität, Freiburg, Germany* (2003).

[23] Jin-Gang Yu et al. "Prototypical multiple instance learning for predicting lymph node metastasis of breast cancer from whole-slide pathological images". In: *Medical Image Analysis* 85 (2023), p. 102748.

[24] Qi Zhang and Sally Goldman. "EM-DD: An improved multiple-instance learning technique". In: *Advances in neural information processing systems* 14 (2001).

Declaration of Authorship

I hereby declare that the thesis presented is my own work and that I have not called upon the help of a third party. In addition, I declare that neither I nor anybody else has submitted this thesis or parts of it to obtain credits elsewhere before. I have clearly marked and acknowledged all quotations or references that have been taken from the works of others. All secondary literature and other sources are marked and listed in the bibliography. The same applies to all charts, diagrams and illustrations as well as to all internet resources. Moreover, I consent to my thesis being electronically stored and sent anonymously in order to be checked for plagiarism. I am aware that if this declaration is not made, the thesis may not be graded.

Mannheim, May 22, 2024 ……