Application of Convolutional Neural Networks and Vision Transformers in Cancer Grading in Pathology Images

Capstone Report

Dosbol Erlan

Nazarbayev University
Department of Electrical and Computer Engineering
School of Engineering and Digital Sciences

This project report was created on TexStudio editing platform using LaTeX. All the figures were drawn using draw.io online software tool.

NAZARBAYEV UNIVERSITY

Electrical and Computer Engineering
Nazarbayev University
http://www.nu.edu.kz

Title:
Application of Convolutional Neural Networks and Vision Transformers in Cancer Grading in Pathology Images

Theme:
Deep Learning for Cancer Grading

Project Period:
Fall 2024 - Spring 2025

Project Group:
Applications of Signal Processing Lab

Participant(s):
Dosbol Erlan

Supervisor(s):
Muhammad Tahir Akhtar

Copies: 1

Page Numbers: 30

Date of Completion:
April 23, 2025

Abstract:

Cancer Grading is a time-consuming and labor-intensive process. There is a need for accurate and robust Machine Learning (ML) models for automated Cancer Grading in Pathology Images. Existing methods use Convolutional Neural Networks (CNNs) for image classification and attention modules like Convolutional Block Attention Module (CBAM) for intermediate feature map refinement. However, integrating the Original Sequential CBAM between Convolutional Blocks in CNNs can disrupt the information flow in a model, increases the number of parameters, and can lead to longer and more computationally intensive training; our experiments demonstrate this can negatively impact performance.
We propose Post-Convolutional Parallel CBAM for Cancer Grading in Pathology Images. We used KBSMC colon cancer dataset for training and validation for 20 epochs on three different architectures: VGG16, GoogLeNet, and ResNet34. The results indicate that the Proposed Post-Convolutional Parallel CBAM consistently outperforms Baseline and Original Sequential CBAM methods across various evaluation metrics despite resulting in fewer parameters than models using the Original CBAM integration. For example, the proposed method resulted in F-1 score of 0.810, while the Original CBAM approach got 0.658. Therefore, the proposed approach showed its effectiveness for transfer learning scenarios, and further development may lead to accurate and robust diagnostic tools.

Contents

Preface

The main motivation of this capstone project is to investigate how the cancer grading procedure can be improved with deep learning. In this project, my goal is to improve accuracy and other evaluation metrics for a computer vision model aimed at classifying whole-slide images using attention modules designed for convolutional neural networks (CNNs).

The report is structured as follows:

- **Chapter 1: Introduction** gives an overview of cancer grading and explains the motivation behind applying deep learning to the procedure.
- **Chapter 2: Background** establishes theoretical foundations and discusses related works for this project.
- **Chapter 3: Methodology** outlines the proposed architectural changes to the existing models to boost accuracy, recall, precision and other evaluation metrics for cancer grading. Furthermore, the chapter discusses other parts of the implementation such as data collection, preprocessing, hyperparameters and training.
- **Chapter 4: Results and Discussions** establishes the evaluation metrics of the models and presents the results of experiments. Furthermore, the chapter discusses the reasons, implications, and relevance of the obtained results.
- **Chapter 5: Conclusion** summarizes results, contributions and potential areas of further research for this work.

I am grateful to **Prof. Muhammad Tahir Akhtar**, whose guidance over the past few years has been crucial in finding my approach to research.

Nazarbayev University, April 23, 2025

Dosbol Erlan
<dosbol.erlan@nu.edu.kz>

Chapter 1

Introduction

1.1 Background

Cancer is a global health problem. In 2020, about 19.3 million new cancer cases were detected and 10 million people died from it [1]. Current research highlights that economics costs associated with cancer detection and treatment are expected to rise [1]. The economic burden will be even more significant for the low-resource regions of the world [1, 2]. Therefore, we need an accurate and accessible diagnostic tool to identify cancer.

Medical imaging is integral to cancer diagnostics. Techniques such as Computed Tomography (CT), Magnetic Resonance Imaging (MRI), and digital pathology are helpful in visualizing structural and functional abnormalities in tissues [3]. The developments in digital pathology have created new methods for extracting quantitative properties of tissue images like texture and morphology [4]. These new methods improve diagnostic precision [5]. However, these methods alone cannot handle vast datasets and still require an expert interpretation.

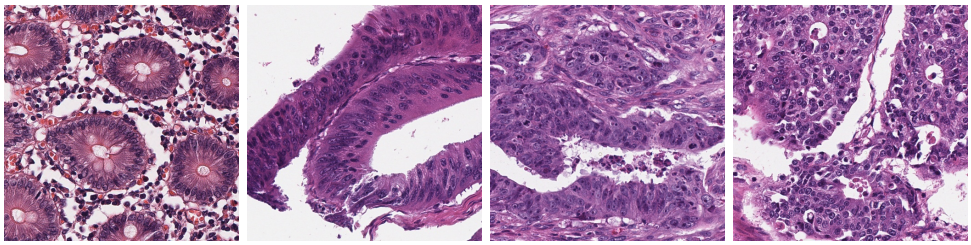
Machine Learning (ML) has risen in importance in medical diagnostics over the past decades. With more accessible computational resources, ML models that are based on Deep Learning architectures are becoming more and more instrumental in utilizing large medical datasets to predict clinical outcomes and create personalized treatment plans [6, 7]. However, cancer grading is still a challenging process for deep learning as the model needs to be highly accurate and robust to changes [8].

This project aims to address the aforementioned challenges by introducing new techniques to boost accuracy of the ML models.

1.2 Literature Review

Cancer Grading Fundamentals Cancer grading is a process of classifying cancer aggressiveness based on its morphological characteristics. Figure 1.1 illustrates representative examples of the four grades of cancer: Benign, Well-Differentiated (WD), Moderately Differentiated (MD), and Poorly Differentiated (PD), as presented in the work by Lee et al. [9]. The results of cancer grading are used for planning patient treatment and predicting clinical outcome. There are traditional histological grading systems like the Nottingham grading system designed for grading breast cancer. The Nottingham grading system evaluates the tumor aggressiveness by considering cell differentiation, mitotic rate and structure [10]. But traditional histological grading systems have common flaws such as inter-observer variability and limited reproducibility [11]. Therefore, we need a consistent and efficient method to improve grading precision.

Advancements in digital pathology like whole-slide imaging (WSI) have given the opportunity to create more detailed datasets of pathology images. Therefore, it made pathology more accessible for visual assessment. But cancer grading still requires expert labor and still is a subject to human bias [5]. Therefore, we need alternative automated approaches to cancer grading leveraging ML techniques to handle vast datasets.



(a) Benign **(b)** WD **(c)** MD **(d)** PD

Figure 1.1: Representative examples of cancer grades (Benign, WD, MD, and PD). Adapted from Lee et al., MICCAI 2023 [9].

Deep Learning & Computer Vision Foundations Deep Learning is a subset of ML that utilizes neural networks with multiple layers. The more layers a Deep Learning model has, the deeper it is. Deep Learning can be used for Image Classification tasks using Convolutional Neural Networks (CNNs). The theoretical foundations for CNNs have been laid out in the 1990s by Yann Lecun [12]. In the 2010s, with the exponential increase in computational power, CNNs were rediscovered by researchers and engineers for Image Classification. Namely, AlexNet architecture was the turning point for the shift happening in the field of computer

vision from traditional kernel based methods to machine learning methods [13]. VGG architecture showed that deeper models are more efficient and accurate than wider models [14]. GoogLeNet architecture proposed the idea of Inception blocks, which utilizes convolutional layers of different kernel sizes in parallel [15]. ResNet architecture worked on the problem of vanishing gradients in deep architectures by introducing residual connections [16].

Research Gaps and Future Directions The advancements in ML-driven cancer grading methods are rapid. But there are other problems that remain unsolved. One of them is the problem of generalizability of Deep Learning models i.e. they can be accurate in a controlled setting, but not in real-life application. Therefore, there is a need for robust and scalable algorithms for CNNs.

1.3 Related Works

The foundational CNNs used in this capstone project are VGG [14], GoogLeNet [15] and ResNet [16].

Simonyan et al. [14] introduced Visual Geometry Group (VGG) architecture in 2015 that utilized simple and homogeneous 3x3 convolutional filters. The VGG-16 model architecture is illustrated in the Fig. 1.2 .The authors experimentally proved that deep CNNs are more accurate and efficient than wide CNNs in various computer vision tasks such as image classification. Furthermore, the authors provided a simple, yet effective feature extractor that is still widely used as a baseline.

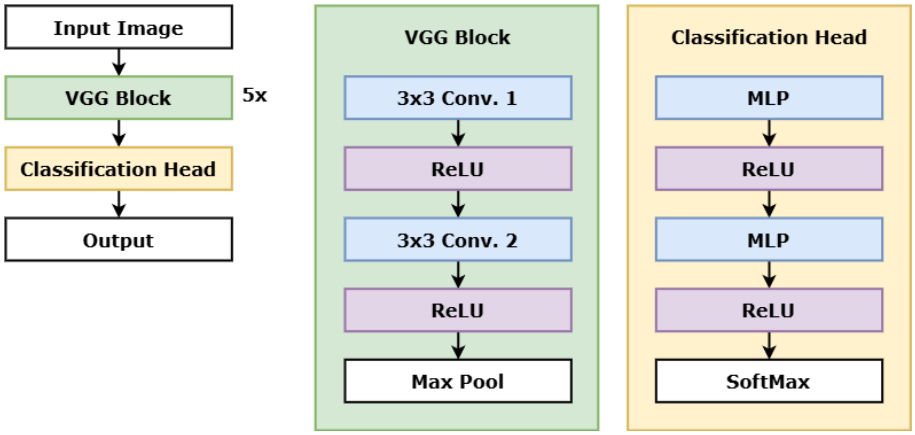


Figure 1.2: Visual Geometry Group (VGG-16) Net Architecture [14]

The input to the VGG-16 model [14] is a fixed-sized three-channel (Red, Green, Blue) image with resolution of 224x224 pixels. The core operation of the model is 3x3 convolutional filters. Furthermore, the authors used ReLU (1.1) activation

function after every convolutional layer. In addition, 2x2 max-pooling layers were utilized for spatial reduction. Crucially, the number of filters increases with depth (64,128,256,512,512). Feature extraction occurs through hierarchical representation learning. Finally, in the classification head, Multi-Layer Perceptron (MLP) (1.2) is coupled with ReLU activation function (1.1) and SoftMax function (1.3) is used to calculate the class probabilities.

$$\sigma_r(z) = \max(0, z) \tag{1.1}$$

$$\text{MLP}(\mathbf{x}) = W_1(\sigma_r(W_0(\mathbf{x}))) \tag{1.2}$$

$$\text{Softmax}_k(\mathbf{z}) = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}} \tag{1.3}$$

Szegedy et al. [15] proposed the GoogLeNet architecture that featured Inception blocks, which utilized parallel convolutional layers with different kernel sizes. The GoogLeNet architecture is illustrated in the Fig. 1.3. The core unit of the architecture was the Inception module. The purpose of it was to capture features of different scales. Furthermore, GoogLeNet introduced parallel computation of convolutions. 1x1 Convolutional layers were utilized for reducing the number of channels and feature transformations. Finally, at the end of each Inception module the parallel branches are concatanated. In addition, the proposed model achieved state-of-the-art accuracy at image classification challenges like ImageNet with significantly less parameters than that of second best.

He et al. [16] introduced ResNet architecture that used residual connections to deal with the vanishing gradient problem in very deep CNNs. The illustration of the ResNet architecture is given in the Fig. 1.4 The authors have successfully shown experimentally that very deep CNNs can exist without facing the problem of vanishing gradients by training 152 layer CNN. The core principle in the ResNet architecture is the residual connections, which adds input feature map to output of the convolutional layers after batch normalization and activation function. ResNet quickly became a standard backbone for many vision tasks because of the high accuracy of ResNet models.

Vaswani et al. [17] proposed the Transformer model, which at the time was revolutionary as the industry standard at the time was recurrent neural nets or convolutional layers. The proposed self-attention mechanism became the foundational building block for dealing with sequential data. Although the proposed idea was applied to machine translation at the time, it quickly became influential in machine vision tasks as well.

Dosovitskiy et al. [18] built upon the previous idea of self-attention and multi-head attention [17] to introduce Vision Transformers (ViT). The architecture of ViT

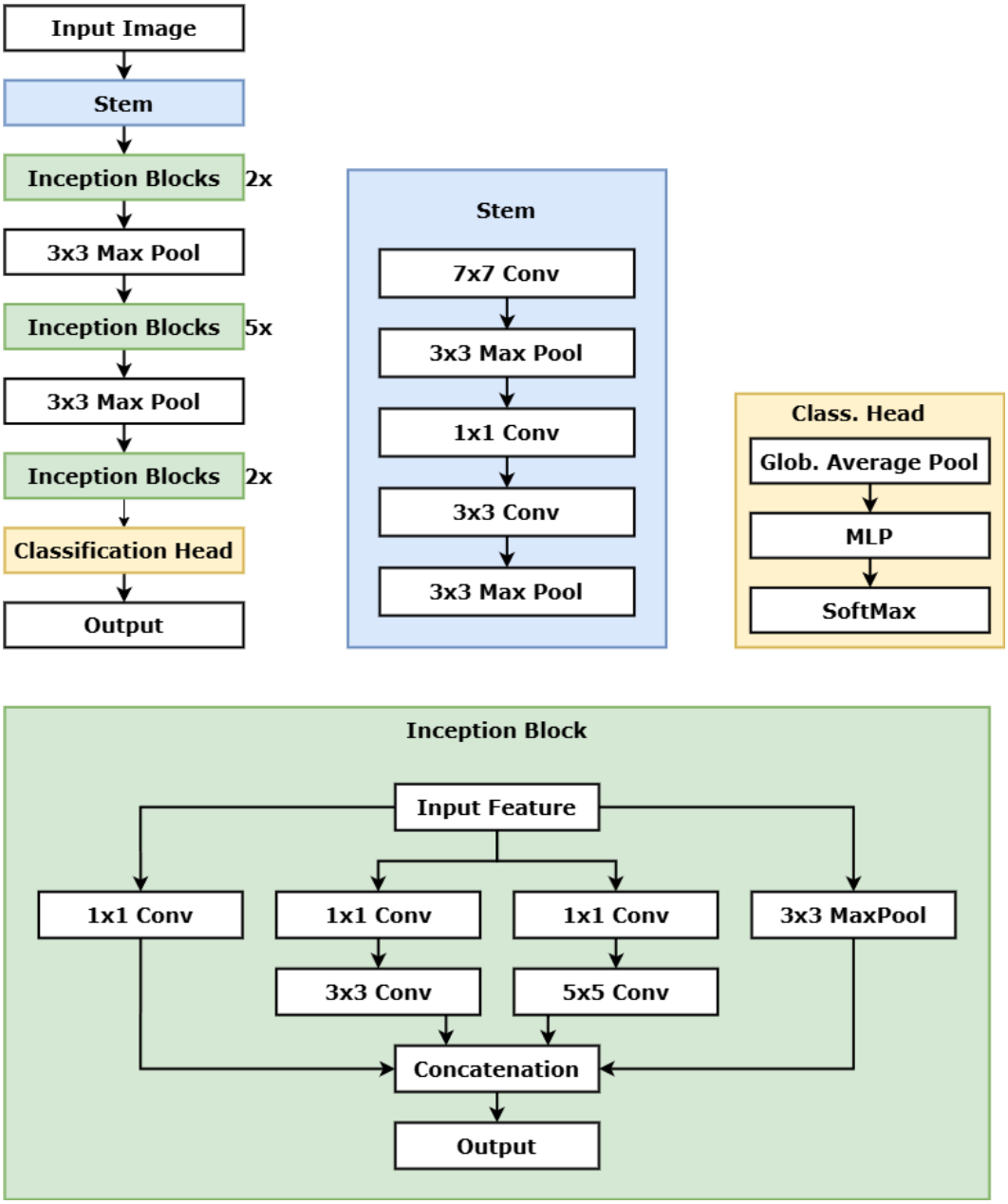


Figure 1.3: GoogLeNet Architecture
[15]

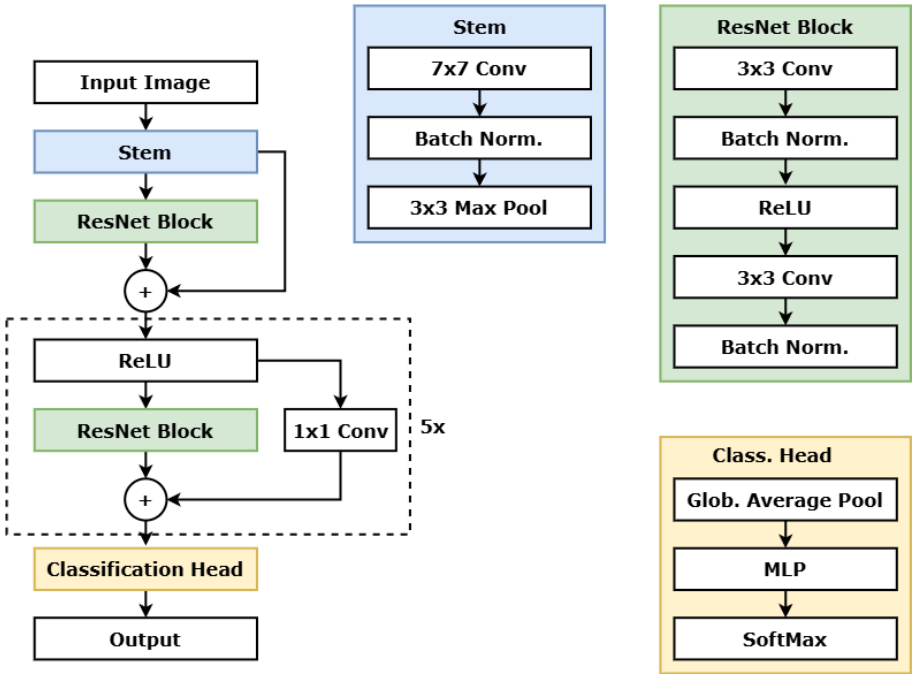


Figure 1.4: ResNet34 Architecture
[16]

is illustrated in Fig. 1.5. The input image is divided into patches before getting processed by ViT. Next, the patches are flattened to vectors, which are projected to embedding dimension. Furthermore, the patch embeddings are combined with positional embeddings. Also, special class <CLS> token is added to the combined embedding. The transformer encoder processes the input via self-attention mechanism of multi-head attention unit. Because of self-attention mechanism each patch relates to other patches. Therefore, global spatial context is captured by Transformer Encoder. MLP is used to refine intermediate features. Finally, a classification head produces class probabilities.

Liu et al. [19] built upon the idea of ViT [18] to introduce CNN-like Transformer for machine vision tasks called Swin Transformer (Fig. 1.6) that captures hierarchical patterns and global spatial context. Similar to ViT, the input to the Swin Transformer is a set of non-overlapping patches of images. However, in Swin Transformer the patches do not get flattened to vectors. Instead, linear embedding is applied to the patches, which are processed by Double Swin-Transformer blocks. The Double Swin-Transformer Blocks are similar to Transformer Encoder blocks [17] [18], but instead of Multi-Head Self-Attention, they use Windowed Self-Attention and Shifting-Window based Self-Attention.

Woo et al. [20] introduced Convolutional Block Attention Module (CBAM) that computes channel attention and spatial attention in sequence. CBAM will be

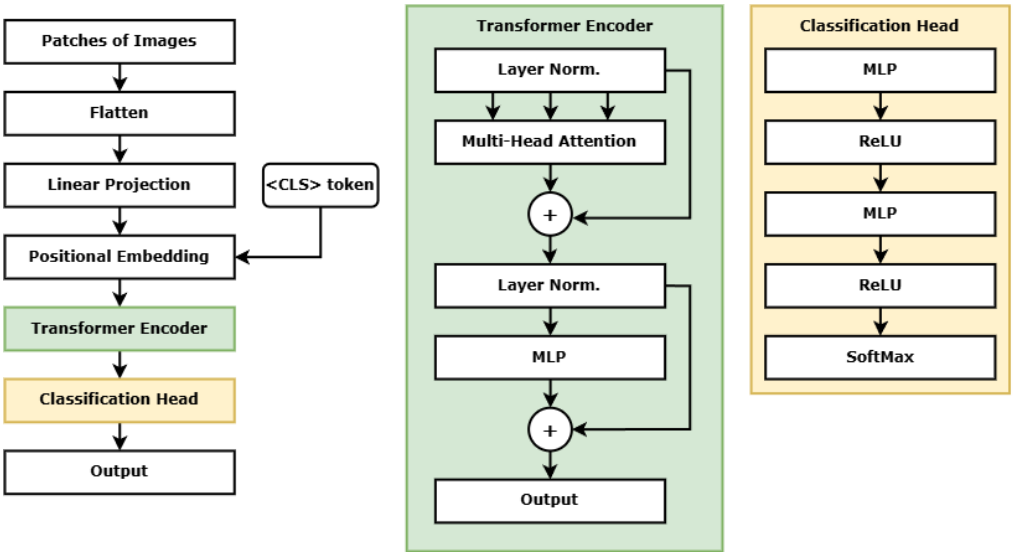


Figure 1.5: Vision Transformer (ViT) architecture
[18]

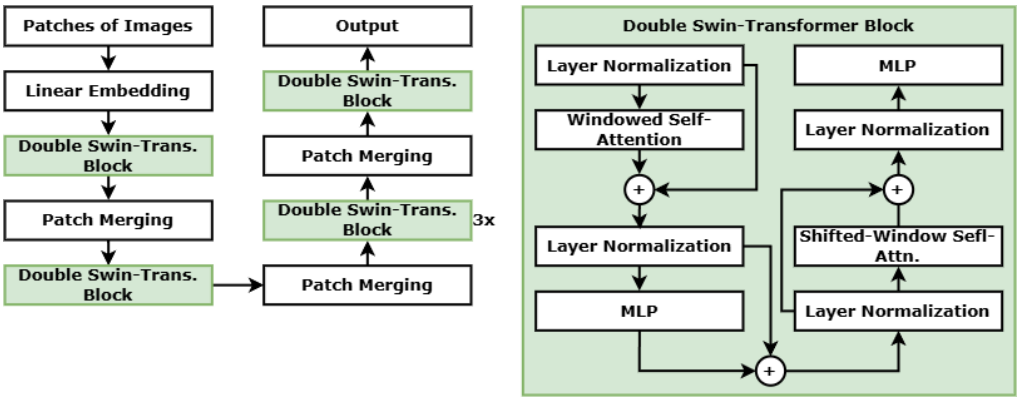


Figure 1.6: Swin Transformer architecture
[19]

discussed later in Section 2.1.

Park et al. [21] proposed Bottleneck Attention Module (BAM) that computes channel attention and spatial attention in parallel. Channel attention is computed via global pooling and multi-layer perceptron (MLP), while spatial attention is computed via 1x1 convolution, downsampling and standard convolutions. Authors proposed to integrate BAM between bottleneck blocks.

In summary, the above mentioned concepts were crucial for Experiments conducted in the Capstone Project. VGG [14], GoogLeNet [15], ResNet [16], Self-Attention [17], ViT [18] and Swin-Transformers [19] were utilized for investigation of Transfer Learning capabilities of various CNNs and Vision Transformers for Cancer Grading in Pathology Images (Experiment 1). On the hand, CBAM [20] and BAM [21] were used instead of ViT [18] and Swin Transformers [19] in investigation of Post-Convolutional Parallel CBAM for Cancer Grading in Pathology Images (Experiment 2).

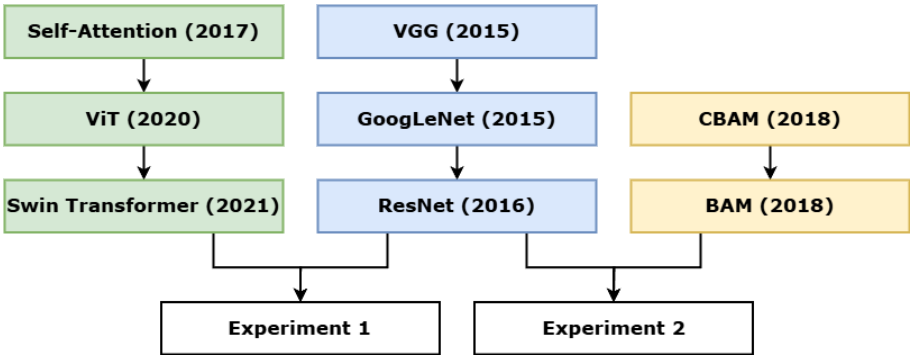


Figure 1.7: Concepts discussed in Related Works and their relation to the Experiment 2 discussed in 4.5

1.4 Motivation and Problem Statement

Motivation Accurate cancer grading is essential for an efficient personalized treatment plan. Traditionally, the treatment plan relies on the results of histopathological analysis, which is a labor-intensive method that is subject to inter-observer bias. Furthermore, other manual grading systems like Bloom-Richardson and Gleason scales can also be affected by inter-observer variability, contributing to the morbidity of the resource-poor regions of the world.

Advancements in ML algorithms over the past decades enable us to utilize accessible computational resources, vast datasets and capture complex patterns in the images. Therefore, we can use ML techniques to enhance diagnostic accuracy and precision. Moreover, integration of ML models into digital pathology can automate image analysis, decreasing the pressure to low-resource healthcare es-

tablishments. Tellez et al. [22] have shown that using image processing techniques such as color normalization and stain augmentation can improve the robustness of the ML model for different imaging conditions. On the other hand, we cannot fully rely on ML models just yet as many problems remain unsolved like technical, operational and regulatory barriers need to be addressed.

Problem statement Even though ML techniques are successful in medical image analysis in controlled settings, applying them in real-life scenarios is problematic. The problems are related to lack of imaging protocols across institutions, class imbalance in datasets, data security and privacy

Firstly, training ML models to achieve clinically viable results needs vast, annotated datasets, which itself is a labor-intensive process further straining healthcare systems. The images in datasets might have different imaging protocols that affect accuracy, robustness and generalizability of the model [23]. Patil et al. have proposed color normalization and standardization methods to achieve consistent model performance [24].

Secondly, class imbalance in datasets might distort the ML model predictions to skew from image analysis to statistical analysis. For example, in cancer grading datasets malignant samples might be underrepresented, thus incentivizing ML models to predict benign most of the time. Therefore, we need data augmentation techniques to mitigate class imbalance. Guerrero et al. [23] proposed a data augmentation method for histopathological image analysis to increase robustness of ML models.

Thirdly, the integration of ML models into existing digital pathology streamlines can be challenging due to data security, privacy and compliance with healthcare regulations. Furthermore, for real-life integration of ML models, there is a need for rigorous validation of the models by analysing evaluation metrics like sensitivity, specificity, f1 score, recall, precision and accuracy.

This capstone project addresses these problems by developing ML models that utilize image processing techniques to mitigate the varying imaging protocols problem, data augmentation techniques to mitigate the class imbalance problem and analyzing various evaluation metrics to mitigate the model validation problem.

1.5 Ethical and Professional Responsibilities

- **Ethical Responsibility:**
  During this project, some ethical concerns may arise, mostly regarding misdiagnosis, algorithmic bias in training data and data privacy & security the datasets used. We describe and address these concerns in the following ways:

First of all, risk of misdiagnosis is great and could have dire consequences for patients. It could lead to inappropriate treatment or delayed care, both of which could end up with lethal results. Therefore, the ML model should not be the sole predictor of diagnosis and/or planner of treatment. It should be only used as one more tool in healthcare experts' toolbox. Still, the problem should be addressed via training the model to be robust by rigorous validation and analysing various evaluation metrics.

Secondly, there is a risk of algorithmic bias in the training data demographics i.e. class imbalance in terms of different groups of people. Therefore, the model intended for clinical use should be trained on various datasets and validated on other real-life sets. Another way of addressing this issue is to use synthetic data using image augmentation techniques to reduce class imbalance and algorithmic bias.

Thirdly, all the datasets used for training, testing and validation should be cleaned i.e. remove all the sensitive data associated with medical records. In addition, regulations regarding data storage and security should be followed. There are many regulations across various countries like HIPAA, GDPR, CPRA, PIPEDA etc. All the datasets used for the clinical trials should meet the standards and regulations regarding data privacy and security.

- **Informed Judgments:**
  In order to make informed judgements throughout this project we conducted extensive literature review and tried to stick to the best practices noticed in existing literature. Technical decisions like model architecture, size and other decisions are based on literature review and evaluation metrics.

  Another great concern of the project is the societal aspects of this research. If the model were to be integrated into real-world clinical workflow, how would that affect the people? To answer this question, we need to first understand the complex modern clinical system where just having the classical evaluation metrics like accuracy is not enough. Due to time and resource limitations of this capstone project, it was not possible to train, test and validate on many different datasets and environments. Furthermore, the project was not tested on real-life applications. Therefore, it is not practical yet. But with more research it very well could be used in real-life clinical workflow to assist the pathologists and other experts in the field. However, for the model to be practical, there is a need for more data, more computational resources, more time, more consultation with experts and more analysis of the results. Furthermore, for real-world application, there is a need to document all design decisions and rationale transparently. In conclusion, for the capstone project to result in real-world application in a clinical environment, the technical decisions should be based on existing literature and the application

design should be based on expert consultation and clinical trials with active feedback from the clients.

- **Global Context:**
  The global context of this project is significant as it can reduce the pressure put on low-resource regions of the world when it comes to dealing with a large number of patients and a low number of experts. But the project has its own flaws like varying protocols between labs and healthcare institutions, staining methods, scanner types and genetic variations across regions. Therefore, there is a need to train a ML model on vast and various datasets that were obtained using different methods and from different parts of the world.

  Furthermore, the implementation of the project might face hurdles in low-resource regions as they might have problems with infrastructure like the internet speed, availability and computational resources. Therefore, the real-life implementation of the project in developing regions should be carried out under government programs with the support of technological companies. Still, the project has a potential to change many lives across the world as automated cancer grading is especially needed in areas with shortage of specialized pathologists. Therefore, for the project to have actual impact, it needs external support in implementing it. In addition, for successful realization of the project, there is a need for creating a local dataset that captures local genetic and demographic features, further highlighting the need for external support.

- **Economic Impact:**
  If developed properly, the project has a potential to have great economic impact on both short-term and long-term. There are short-term losses in developing the models, validating them, consulting with experts and further testing them. Also there are costs associated with integrating the ML models into existing healthcare systems that in itself could disrupt the system in the short-term. Moreover, there are administrative costs related to training personnel and change of protocols.

  However, the long-term benefits might outweigh short-term losses as it could lead to faster diagnosis throughput, which decreases the pressure put on modern healthcare systems, which are often underfunded, underpaid and understaffed. In addition, the ML models could save lives as they can detect abnormalities earlier, thus leading to better outcomes, which could save lives and treatment costs. On the other hand, the short-term losses might be a barrier for developing regions of the world as they might not be able to afford even short-term costs that could put the entire system under risk. Furthermore, some could claim that ML models could potentially put human experts

out of work further making the problem of understaffed healthcare institutions worse. These problems should be addressed by institutional changes and reforms to ensure smooth integration of ML models.

- **Environmental Impact:**
  The environmental impact of this project is mostly due to the high computational cost of training deep neural networks. To train a deep neural network, a computer should do millions of matrix manipulations for hours. Therefore, it spends some energy. We could minimize the negative environmental impact of this project by training efficient models that are both accurate and not large in terms of parameters count. We could minimize negative environmental effects of this project by using efficient hardware to train the models on the cloud platforms with green energy sources.

  However, the environmental impact of the project is potentially more positive than negative. Because the end goal is to automate the cancer grading process, which is currently done by a human who, most probably, has significantly more negative impact on the environment than a ML model. Furthermore, with more optimization on the technical side like GPU optimization and model architecture optimization, the environmental impact of ML models might become insignificant over time.

- **Societal Impact:**
  There are several benefits and potential challenges of the project in terms of its societal impact.

  Firstly, the project could lead to potentially faster and more consistent cancer grading procedures, which in itself leads to more benefits like quicker diagnosis, treatment plans, less pressure on the healthcare institutions, etc.

  Second benefit of the project is, if the implementation of the project were to be successful, it would free up time for experts like pathologists for more complex tasks. Therefore, it would lead to more productive work environments and would reduce burnout from doing repetitive manual tasks.

  On the other hand, there are some challenges related to this project. The most important problem is the trust of medical professionals and patients. It is often hard to trust a black box system that even the designers of the system often have a hard time interpreting the results. Furthermore, the system relies on existing digital infrastructure for digital pathology like internet, digital literacy, trained professionals, etc. Widespread use of the system might push disadvantaged regions of the world to increase the gap between them and developed regions, thus further exacerbating the inequality.

Chapter 2

Existing Methodology

2.1 Sequential Convolutional Block Attention Module

The seminal paper by Woo et al. [20] introduced the Convolutional Block Attention Module (CBAM) that computes channel attention and spatial attention in sequence. Figure 2.1 shows the general workflow of Sequential CBAM, where **H, W, C** are integers that represent height, width and number of channels of a feature map respectively.

The Channel Attention Module (CAM), illustrated in the Fig 2.2,flattens the spatial dimensions (height and width) to identify the "importance" of a particular channel in the final classification. Therefore, the shape of the feature map changes from **(H,W,C)** to **(1,1,C)**. In technical implementation, it should be **(C,1,1)**, but for visual understanding, it is left as **(1,1,C)**.

Similarly, the Spatial Attention Module (SAM), illustrated in the Fig. 2.3 ,flattens the channel dimension to identify the "importance" of a pixel across different channels. Therefore, the shape of the feature map changes from **(H,W,C)** to **(H,W,1)**

Both attention modules use average pooling (2.1) (2.6) and maximum pooling (2.2) (2.7) to aggregate feature maps. The outputs of both Attention Modules are multiplied element-wise by the input feature (2.5) (2.9).

2.1.1 Channel Attention Module

$$\mathbf{F}^{c}_{\text{avg}} = \text{AvgPool}(\mathbf{F}) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \mathbf{F}_{c,i,j} \tag{2.1}$$

$$\mathbf{F}^{c}_{\text{max}} = \text{MaxPool}(\mathbf{F}) = \max_{1 \le i \le H, 1 \le j \le W} \{\mathbf{F}_{c,i,j}\} \tag{2.2}$$

The channel attention module (CAM) is illustrated in 2.2. The input feature map tensor $\mathbf{F} \in \mathbb{R}^{H \times W \times C}$ is aggregated with the average pool (2.1) and the

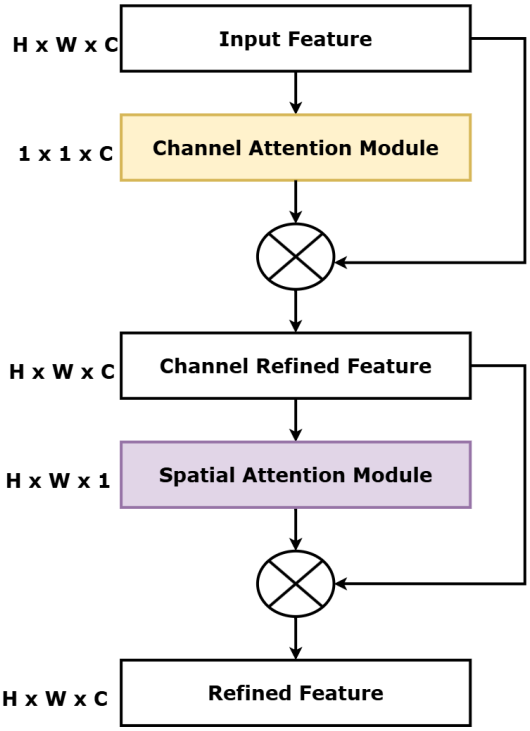


Figure 2.1: Convolutional Block Attention Module (CBAM) [20]

maximum pool (2.2) in parallel to produce channel-wise average pooled features $\mathbf{F}^c_{\text{avg}} \in \mathbb{R}^{1\times1\times C}$ and channel-wise maximum pooled features tensors $\mathbf{F}^c_{\text{max}} \in \mathbb{R}^{1\times1\times C}$.

Lin et al. [25] introduced Network-in-Network that uses convolutional layers with kernel size of 1x1. The authors have shown the equivalence between 1x1 convolutional layers and multilayer perceptrons, where weight tensors $W_0 \in \mathbb{R}^{C\times(C/r)}, W_1 \in \mathbb{R}^{(C/r)\times C}$ are used for training (1.2). Therefore, the next step in CAM is the bottleneck MLP with a reduction ratio **r** that uses the ReLU activation function (1.1). The rationale behind using reduction ratio in CAM's MLP is to reduce the parameter overhead and increase performance by exploiting both average-pooled and max-pooled features [20]

$$\mathbf{M}_c(\mathbf{F}) = \sigma_s\left(\text{MLP}(\mathbf{F}^c_{\text{avg}}) + \text{MLP}(\mathbf{F}^c_{\text{max}})\right) \tag{2.3}$$

$$\sigma_s(z) = \frac{1}{1+e^{-z}} \tag{2.4}$$

The output of CAM is Channel Attention Map $\mathbf{M}_c : \mathbb{R}^{H\times W\times C} \rightarrow \mathbb{R}^{1\times1\times C}$. The spatial dimensions collapse due to the pooling methods applied (2.1) (2.2). The

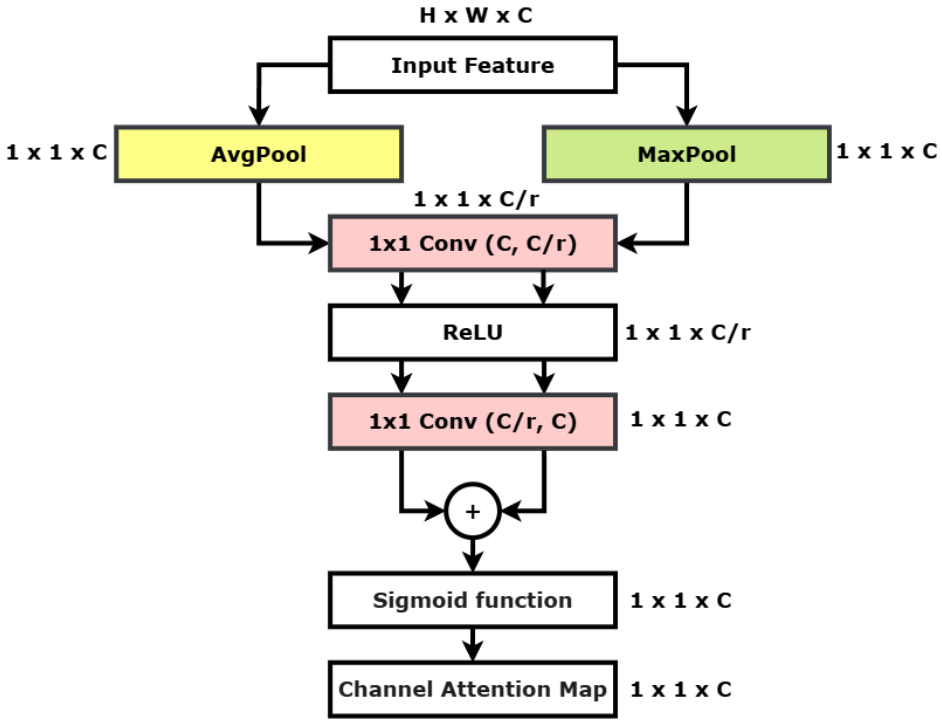


Figure 2.2: Channel Attention Module (CAM)

equation for the Channel Attention Map is given in (2.3), where the Sigmoid Activation function is defined as in (2.4).

2.1.2 Spatial Attention Module

The Spatial Attention Module (SAM) pipeline is illustrated in 2.3. The intermediate feature map tensor $\mathbf{F}' \in \mathbb{R}^{H \times W \times C}$ is defined as in (2.5). $\mathbf{F}'$ is the input to the SAM.

$$\mathbf{F}' = \mathbf{M}_c(\mathbf{F}) \otimes \mathbf{F} \tag{2.5}$$

The average-pool and maximum-pool functions used in SAM are different from that of CAM. In SAM, the channel dimension is collapsed to identify the "weight" of each pixel across channels. Spatial-wise Average Pooled Features $\mathbf{F}^s_{\text{avg}} \in \mathbb{R}^{H \times W \times 1}$ and Spatial-wise Max Pooled Features tensor $\mathbf{F}^s_{\text{max}} \in \mathbb{R}^{H \times W \times 1}$ are found using (2.6) (2.7). $\mathbf{F}^s_{\text{avg}}$ and $\mathbf{F}^s_{\text{max}}$ are concatenated into a single tensor, therefore having the shape of **(H,W,2)**.

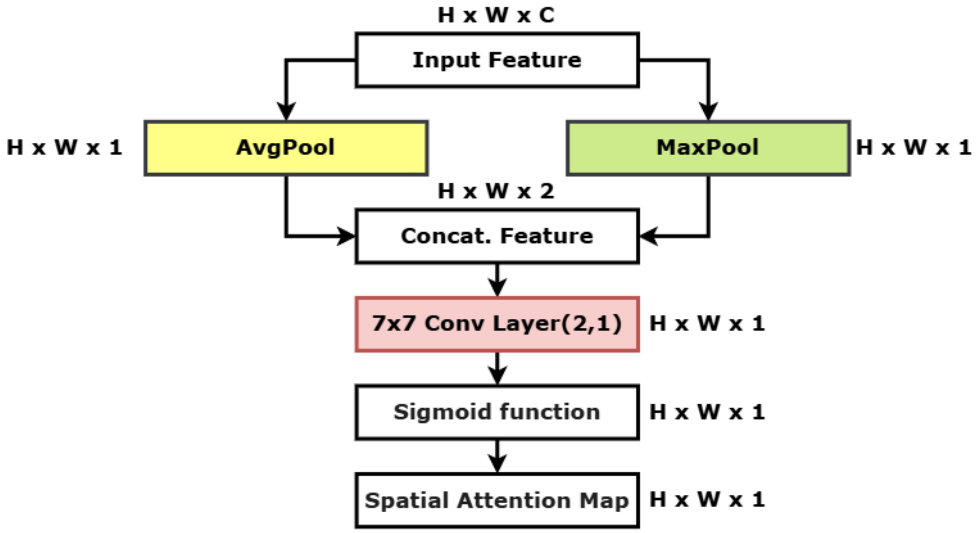


Figure 2.3: Spatial Attention Module (SAM)

$$\mathbf{F}^s_{\text{avg}} = \text{AvgPool}_C(\mathbf{F}') = \frac{1}{C}\sum_{k=1}^{C}\mathbf{F}'_{k,i,j} \tag{2.6}$$

$$\mathbf{F}^s_{\text{max}} = \text{MaxPool}_C(\mathbf{F}') = \max_{1\le k\le C}\{\mathbf{F}'_{k,i,j}\} \tag{2.7}$$

The Spatial Attention Map $\mathbf{M}_s : \mathbb{R}^{H\times W\times C} \to \mathbb{R}^{H\times W\times 1}$ is found using Equation (2.8), where $\text{Conv}_{7\times 7}(\cdot)$ is a convolutional layer with kernel size 7x7 and zero-padding of 3 pixels around the feature map. The 7x7 convolutional layer reduces the number of channels from 2 to 1.

$$\mathbf{M}_s(\mathbf{F}') = \sigma_s\left(\text{Conv}_{7\times 7}\left(\text{Concat}\left[\mathbf{F}^s_{\text{avg}};\mathbf{F}^s_{\text{max}}\right]\right)\right) \tag{2.8}$$

Finally, Sequential CBAM Output Feature Map tensor $\mathbf{F}''_{\text{seq}} \in \mathbb{R}^{H\times W\times C}$ is found by element-wise multiplication of Spatial Attention Map $\mathbf{M}_s$ and intermediate feature map tensor $\mathbf{F}'$ as shown in Equation (2.9).

$$\mathbf{F}''_{\text{seq}} = \mathbf{M}_s(\mathbf{F}') \otimes \mathbf{F}' \tag{2.9}$$

2.2 CBAM Integration

Woo et al. [20] proposed to place the CBAM between convolutional blocks as depicted in Fig. 2.4. The purpose of this design is to adaptively refine intermediate

feature map within the network. Moreoever, such design potentially aids gradient flow, which leads to better behaved convergence.Furthermore, such design could be used for "plug-and-play" for various standard CNNs.

However, it could be argued that placing CBAM in between convolutional blocks might disrupt the feature flow i.e. interfere with base CNN's learned hierarchical patterns. Furthermore, because of disruption, it might require longer training or make the it unstable. In addition, repeating CBAMs in Deep Neural Net adds more parameters to the model, which leads to a need for more compute power and longer training.

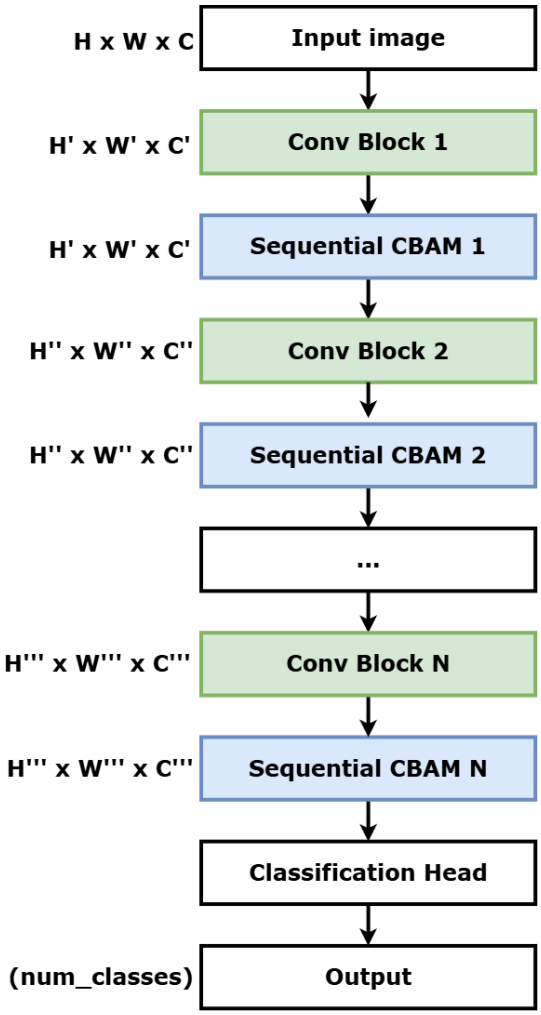


Figure 2.4: Convolutional Block Attention Module (CBAM) [20]

Chapter 3

Proposed Methodology

3.1 Proposed Parallel CBAM

The proposed parallel CBAM architecture and its integration method is shown in the Fig. 3.1. The proposed method builds upon the idea of Bottleneck Attention Mechanism (BAM) introduced by Park et al. [21]. While BAM introduced the parallel computation for CAM and SAM, the underlying funtions in CAM and SAM were different. Instead of concatenating the outputs of SAM and CAM, the authors summed them. Furthermore, the attention module placements are different. The implementation of SAM is significantly different in two works as BAM employs a bottleneck structure in spatial branch using 1x1 convolution with reduction ratio 'r', and then dilated convolutions to aggregate a wide contextual range.

The Combined Parallel Attention Map $\mathbf{M}_{\text{parallel}} : \mathbb{R}^{H\times W\times C} \rightarrow \mathbb{R}^{H\times W\times C}$ is found using Equation (3.1), where $\mathbf{M}_c$ and $\mathbf{M}_s$ are identical to the ones discussed in 2.1.

$$\mathbf{M}_{\text{parallel}}(\mathbf{F}) = \mathbf{M}_c(\mathbf{F}) \otimes \mathbf{M}_s(\mathbf{F}) \tag{3.1}$$

Parallel CBAM Output Feature Map $\mathbf{F}''_{\text{parallel}} \in \mathbb{R}^{H\times W\times C}$ is found by multiplying $\mathbf{M}_{\text{parallel}}$ by the input feature map tensor $\mathbf{F}$ element-wise as shown in (3.2).

$$\mathbf{F}''_{\text{parallel}} = \mathbf{M}_{\text{parallel}}(\mathbf{F}) \otimes \mathbf{F} \tag{3.2}$$

3.2 Proposed CBAM integration

Figure 3.1 shows the proposed Parallel CBAM and the Proposed integration method. In [20], the authors proposed to integrate CBAM between convolutional blocks and/or before residual connections. The rationale behind placing CBAM in between convolutional blocks is that it can refine intermediate features. By placing

it between convolutional blocks, it helps the network focus on the most relevant features before they are passed on to the next layer.

In this study, we propose integrating the parallel CBAM after the convolutional blocks and before classification head. The rationale behind this design decision is that classification head relies on the final feature maps. Therefore, the final and often most complex feature maps are refined. Furthermore, the architectural adaptability of this integration method is more flexible than the original CBAM integration technique.

The significance of the proposed method lies in its architectural adaptability, potential for efficient transfer learning and capture of complementary channel and spatial features. Post-Convolutional Parallel CBAM is adaptive in architectural sense as it is placed after the backbone, therefore, no internal block modification is needed. Furthermore, it could be used for Transfer Learning of Pretrained CNNs as shown in the Section 4.5. In addition, the Parallel CBAM block processes channel and spatial attention in parallel, therefore, provides more effective information filtering for the final classification head.

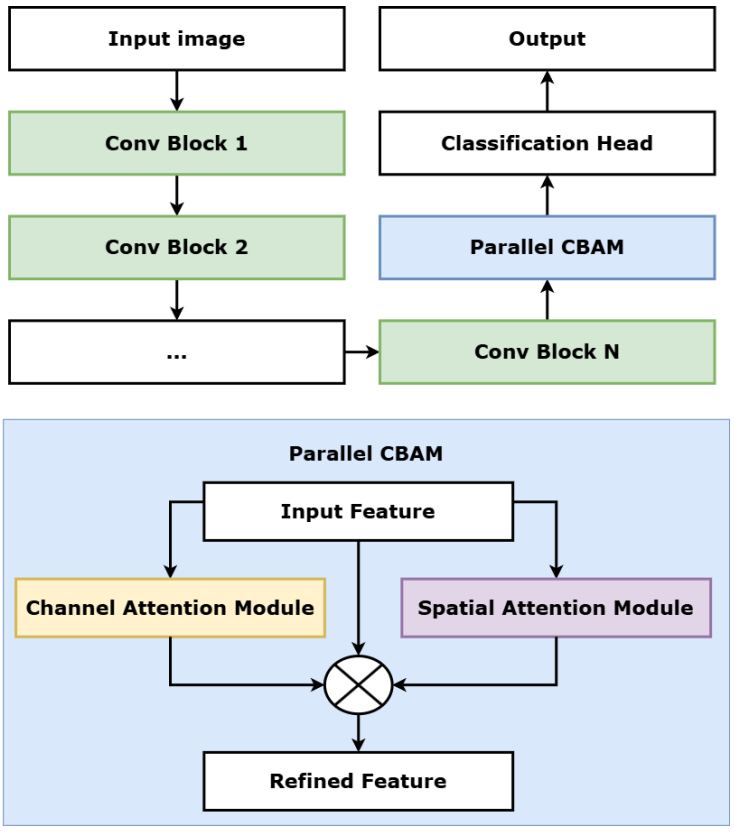


Figure 3.1: Proposed Parallel CBAM Integration method

Chapter 4

Findings and Analysis

4.1 Evaluation Metrics

Accuracy Accuracy is defined as the ratio of correctly classified samples to the total number of samples:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}} = \frac{\sum_{i=1}^{C} TP_i + TN_i}{\sum_{i=1}^{C} (TP_i + TN_i + FP_i + FN_i)}, \tag{4.1}$$

where C is the total number of classes, TP_i is the true positives, TN_i is the true negatives, FP_i is the false positives, FN_i is the false negatives for class i.

Precision (Weighted) accounts for the proportion of true positives among all predicted positives, weighted by the number of samples in each class:

$$\text{Weighted Precision} = \sum_{i=1}^{C} w_i \cdot \frac{TP_i}{TP_i + FP_i}, \tag{4.2}$$

where $w_i = \frac{\text{Number of Samples in Class } i}{\text{Total Number of Samples}}$ represents the weight for class i.

Recall (Weighted) measures the proportion of true positives identified out of all actual positives, weighted by the number of samples in each class:

$$\text{Weighted Recall} = \sum_{i=1}^{C} w_i \cdot \frac{TP_i}{TP_i + FN_i}. \tag{4.3}$$

F1 Score (Weighted) is the harmonic mean of precision and recall, providing a single measure of classification performance. The weighted F1 score is calculated as:

$$\text{Weighted F1} = \sum_{i=1}^{C} w_i \cdot \frac{2 \cdot \text{Precision}_i \cdot \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i}, \tag{4.4}$$

Sensitivity (Macro Average) The sensitivity for class i (True Positive Rate) is:

$$\text{Sensitivity}_i = \frac{TP_i}{TP_i + FN_i}. \tag{4.5}$$

The Macro Average Sensitivity is the unweighted mean across all C classes:

$$\text{Macro Avg Sensitivity} = \frac{1}{C} \sum_{i=1}^{C} \frac{TP_i}{TP_i + FN_i}. \tag{4.6}$$

Specificity (Macro Average) The specificity for class i (True Negative Rate) is:

$$\text{Specificity}_i = \frac{TN_i}{TN_i + FP_i}. \tag{4.7}$$

The Macro Average Specificity is the unweighted mean across all C classes:

$$\text{Macro Avg Specificity} = \frac{1}{C} \sum_{i=1}^{C} \frac{TN_i}{TN_i + FP_i}. \tag{4.8}$$

Area Under the ROC Curve (AUC ROC - Weighted Average) Calculated using the one-vs-rest (OvR) approach for each class i, yielding AUC_i. The final metric is the weighted average:

$$\text{Weighted AUC ROC} = \sum_{i=1}^{C} w_i \cdot \text{AUC}_i, \tag{4.9}$$

4.2 Data Collection and Preprocessing

The data set used in this study is the Kangbuk Samsung Medical Center (KBSMC) colon cancer dataset, which consists of 9,863 histopathological images annotated by pathologists for cancer classification. The dataset includes both whole slide images (WSIs) and tissue microarrays (TMAs) of colon cancer, scanned at 40x magnification. Each sample is classified into one of four categories: benign, well-differentiated (WD), moderately differentiated (MD), and poorly differentiated (PD) tumors [26].

The preprocessing pipeline for the dataset, illustrated in Figure 4.1, was designed to standardize image quality, enhance consistency, and prepare the images for input into deep learning models. The pipeline consists of the following steps:

1. **Input and Resizing**: The raw histopathology images are read in their original resolution ($3\times512\times512$ pixels, where 3 represents the RGB channels and 512×512 corresponds to the image width and height). These images are resized and cropped to a uniform dimension of $3\times224\times224$ pixels for compatibility with pre-trained deep learning models.

2. **Data Augmentation**: To simulate variability and improve model generalization, the following random transformations are applied during training:
    - *Random Resizing and Cropping*: Ensures the model learns invariant features by cropping regions of interest.
    - *Random Flipping*: Introduces spatial variation in the dataset.
    - *Random Rotation*: Applies rotation of up to 15 degrees to account for orientation variability.
3. **Normalization**: After augmentation, the images are normalized using the mean and standard deviation values of the ImageNet dataset. This step ensures that pixel values have a consistent range, allowing the deep learning model to converge effectively.
4. **Train-Validation Split**: The dataset is split into 75% training and 25% validation subsets to ensure robust model evaluation.
5. **Output for Model Input**: The preprocessed images ($3\times224\times224$) are fed into a pre-trained deep learning model for further analysis and classification.

Table 4.1: Sample Distribution Across Sets with Total Counts

Status	Training Set (75%)	Validation Set (25%)	Total
Benign	1200	400	1600
WD	1742	580	2322
MD	3079	1026	4105
PD	1377	459	1836

4.3 Implementation Details

For the experiments, Python programming language was used. Moreover, PyTorch deep learning framework was the backbone for GPU accelerated experiments. Furthermore, Git/GitHub was used for version control of the project. In addition, Weights and Biases (wandb) platform was used to store and visualize the results. Also, sklearn library was used to split the dataset "on the fly" and to calculate evaluation metrics. Google services such as Colab was used to access cloud computing using T4 GPUs. The key training hyperparameters are presented in the table 4.2.

The Cross-Entropy Loss ($\mathcal{L}_{CE}$), defined in Eq. (4.10), quantifies the dissimilarity between the true class distribution and the predicted probability distribution for a given sample. The summation runs over all K classes. For each class k, y_k is an indicator (1 if it's the true class, 0 otherwise), and $P(y = k|\mathbf{z})$ is the model's

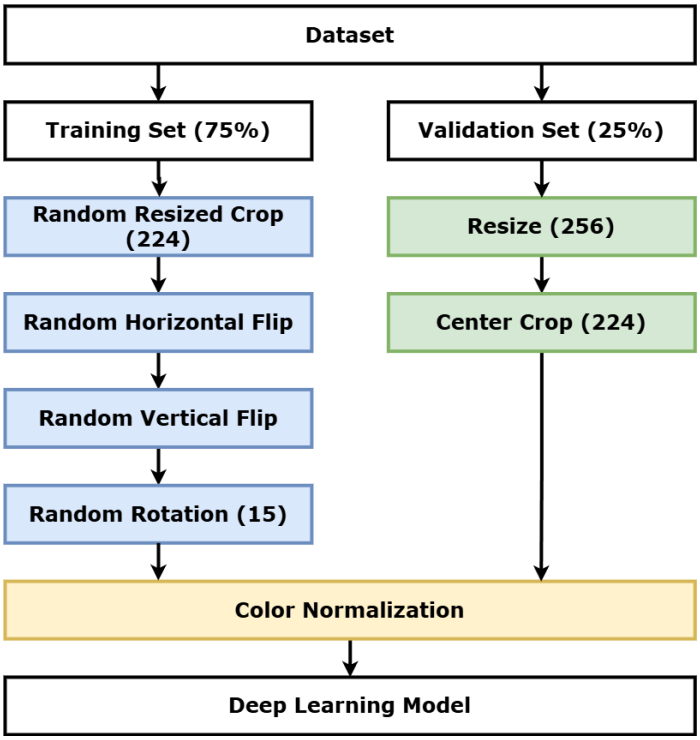


Figure 4.1: Preprocessing pipeline for histopathology images

predicted probability for that class from Softmax function (1.3) output based on logits $\mathbf{z}$. The logarithm term penalizes the model more heavily for low probabilities assigned to the correct class, thus minimizing $\mathcal{L}_{CE}$ encourages the model to assign high probability to the true class.

$$\mathcal{L}_{CE} = -\sum_{k=1}^{K} y_k \log(P(y = k|\mathbf{z})) \tag{4.10}$$

Table 4.2: Key Training Hyperparameters.

Parameter	**Value / Setting**
Learning Rate	0.001
Batch Size	256
Number of Epochs	20
Loss Function	Cross Entropy Loss (4.10)
Optimizer	AdamW [27]

4.4 Experiment 1: Investigation of Transfer Learning capabilities of Convolutional Neural Nets and Visions Transformers

The evaluation metrics for the Experiment are defined in 4.1. The results for the Experiment 1 is shown in the Table 4.3. Pretrained CNNs (VGG16 [14] , GoogLeNet [15], ResNet34) and Vision Transformers (ViT [16] , Swin Transformer [19]) were tested for their Transfer Learning capabilities in the task of Cancer Grading.

Table 4.3: Performance Comparison of Various Models on Cancer Grading. Parameters in Millions (M). Best performance per metric is **bold**.

Model	Params (M)	Acc.	F1	Prec.	AUC	Recall	Sens.	Spec.
VGG16 [14]	134.3	0.811	0.809	0.820	0.942	0.811	0.828	0.929
GoogLeNet [15]	**9.9**	**0.848**	**0.847**	**0.854**	**0.960**	**0.848**	**0.866**	**0.942**
ResNet34 [16]	21.3	0.817	0.816	0.828	0.948	0.817	0.841	0.932
ViT-Base [18]	85.8	0.664	0.656	0.663	0.860	0.644	0.771	0.886
Swin-Tiny [19]	27.5	0.494	0.392	0.245	0.584	0.494	0.243	0.757

GoogLeNet [15] model achieved the highest performance in all evaluation metrics, including the lowest parameter count, making it not only accurate, but efficient model for cancer grading. Furthermore, other CNNs like VGG16 [14] and ResNet34 [16] performed similarly, but below GoogLeNet's performance level. However, VGG16 has significantly more parameters, making it less efficient than GoogLeNet and ResNet34 models.

Vision Transformers, on the other hand, underperformed despite having higher parameters count than GoogLeNet and ResNet34. A potential reason for the underperformance might be that ViT [18] and Swin Transformer [19] require much more data than CNNs to perform well, while the KBSMC dataset [26] used for the experiment might not be enough for the Vision Transformers to converge. Furthermore, the Vision Transformers were fine-tuned for 20 epochs, which may not be enough for ViT and Swin Transformer to capture the necessary global context. Moreover, CNNs inherently are more capable of capturing local patterns, which are crucial for pathology images.

4.5 Experiment 2: Post-Convolution Parallel CBAM

For each CNN model, three types of experiments were conducted. First experiment involved no CBAM. Second experiment integrated sequential CBAM between Convolutional Blocks. Third experiment integrated Post-Convolutional Parallel CBAM into the model, which is the proposed method.

Table 4.4: VGG16 Performance and Parameter Count. VGG16 row indicates no CBAM. Original CBAM uses sequential CBAM integration. Proposed CBAM uses Parallel CBAM with Post-Convolutional integration. Parameters in Millions (M). Best performance per metric for this model is **bold**.

Model Variant	Params (M)	Acc.	F1	Prec.	AUC	Recall	Sens.	Spec.
VGG16 [14]	134.285	0.811	0.809	**0.820**	0.942	0.811	0.828	0.929
Original CBAM [20]	**134.513**	0.665	0.658	0.697	0.849	0.665	0.690	0.875
Proposed CBAM	134.375	**0.812**	**0.810**	0.819	**0.943**	**0.812**	**0.839**	**0.931**

The results for the CBAM integration for the VGG16 model [14] are shown in the Table 4.4. The baseline VGG16 performance metrics are solid across various performance metrics, especially Precision is the highest among 3 methods. On the other hand, Original CBAM integration significantly reduces the performance on evaluation metrics, despite increasing the number of parameters. In addition, Proposed CBAM has shown some improvement on the baseline VGG16 model performance. The general pattern is that the Original CBAM method is detrimental to the baseline CNN in this setup, while the Proposed CBAM method improves performance.

One of the potential reasons for the observed pattern is that Original CBAM placement between convolutional blocks interferes with the learned hierarchies of the model, this disrupting the feature flow. On the other hand, Proposed CBAM is placed after the convolutional blocks, which would act as a post-processing step. Therefore, the Proposed method is less disruptive. Furthermore, the proposed Parallel CBAM might capture broader context, which would make it more effective for the final feature refinement stage. In addition, the limitations of the setup like limited training epochs (20) and limited data might be one of the reasons for the observed pattern.

Table 4.5: GoogLeNet Performance and Parameter Count. GoogLeNet row indicates no CBAM. Original CBAM uses sequential CBAM integration. Proposed CBAM uses Parallel CBAM with Post-Convolutional integration. Parameters in Millions (M). Best performance per metric for this model is **bold**.

Model Variant	Params (M)	Acc.	F1	Prec.	AUC	Recall	Sens.	Spec.
GoogLeNet [15]	9.942	0.848	0.847	0.854	0.960	0.848	0.866	0.942
Original CBAM [20]	**10.417**	0.846	0.846	0.855	**0.961**	0.846	0.874	0.944
Proposed CBAM	10.073	**0.850**	**0.850**	**0.855**	0.960	**0.850**	**0.874**	**0.945**

The results for the CBAM integration for the GoogLeNet model [15] are shown in the Table 4.5. The baseline GoogLeNet performance, as shown in 4.3, is higher than other CNNs and Vision Transformers. Compared to VGG16, the effect of CBAM integration to GoogLeNet is not as significant. Still, the Original CBAM in-

tegration method results in minimal change from the Baseline performance, while Proposed CBAM results in consistent, small improvements over the Baseline. Furthermore, Proposed CBAM is more parameter-efficient than the Original CBAM as it adds more parameters to the Baseline model.

One of the potential reasons for the observed pattern is the fact that GoogLeNet's Inception blocks are complex and captures multiple scales of patterns. Therefore, placing CBAM between Inception blocks does not disrupt the feature flow as much as in the case with VGG16 as seen in the Table 4.4. Furthermore, Proposed Parallel CBAM might capture complementary information for the final classification than Sequential CBAM for more complex features.

Table 4.6: ResNet34 Performance and Parameter Count. ResNet34 row indicates no CBAM. Original CBAM uses sequential CBAM integration. Proposed CBAM uses Parallel CBAM with Post-Convolutional integration. Parameters in Millions (M). Best performance per metric for this model is **bold**.

Model Variant	Params (M)	Acc.	F1	Prec.	AUC	Recall	Sens.	Spec.
ResNet34 [16]	21.289	0.817	0.816	0.828	0.948	0.817	0.841	0.932
Original CBAM [20]	**21.605**	0.739	0.738	0.764	0.903	0.739	0.755	0.901
Proposed CBAM	21.319	**0.822**	**0.823**	**0.837**	**0.951**	**0.822**	**0.842**	**0.933**

The results for the CBAM integration for the ResNet34 model [16] are shown in the Table 4.6. The general patterns are similar to the ones found in the Table 4.4. Overall, the proposed Post-Convolutional Parallel CBAM consistently outperforms the Baseline (No CBAM) and the Original CBAM models across nearly all evaluation metrics. The proposed method achieved highest performance in respective architectures in accuracy, F1-score, Recall, Sensitivity and Specificity, though the proposed model has fewer parameters than the Original CBAM model. One of the potential reasons for this might be that Parallel computation of Channel and Spatial Attention Maps might capture complementary information more effectively than the sequential approach, leading to better feature refinement in Post-Convolution.

On the other hand, the Original CBAM integration was detrimental to ResNet34 and VGG16 architectures. A potential reason for this might be that the Original CBAM method significantly disrupt the information flow in the model. Therefore, 20 epochs may not be enough for the model to re-adapt to the new architecture. Thus, the proposed Post-Convolutional Parallel CBAM might offer more efficient Transfer Learning capabilities.

Chapter 5

Conclusion

The problem with manual cancer grading is that it is time-consuming, labor-intensive and subject to inter-observer and intra-observer variabilities [11]. Therefore, there is a need for an accurate and efficient Deep Learning system to automate the cancer grading of pathology images. The existing methods use Convolutional Neural Nets (CNNs) [12] for image classification and Convolutional Block Attention Modules (CBAM) [20] for intermediate feature map refinement.

The KBSMC dataset [26] was used for training and validation. Image preprocessing pipeline was applied to the dataset as shown in 4.1 with transformations like Resizing, Cropping, Random Vertical/Horizontal Flip, Random Rotation and Color Normalization

The main contribution of this work is the proposed Post-Convolutional Parallel CBAM method that has experimentally shown for 3 different CNN architectures to improve evaluation metrics such as Accuracy, F1-score, Recall, Sensitivity and Specificity just over 20 epochs of transfer learning outperforming baseline CNNs and Original CBAM methods. The architectures tested are VGG16 [14], GoogLeNet [15] and ResNet34 [16]. Furthermore, the proposed method has less parameters, hence, more efficient than the Original CBAM method.

The main implication of this work is that Parallel CBAM and Post-Convolutional placement potentially captures complementary information better than traditional methods. Another significance of this work might be the Efficient Transfer Learning capabilities of the proposed method as it has outperformed both Baseline and Original CBAM methods with only 20 epochs of training across various performance metrics. However, this study comes with limitations like time and resource constraints, models being tested on a single dataset and the training duration might not be enough. Therefore, the future work should include testing the models on more diverse dataset and longer training.

Bibliography

[1] Hyuna Sung et al. "Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries". In: *CA: A Cancer Journal for Clinicians* 71.3 (2021), pp. 209–249. DOI: 10.3322/caac.21660. URL: https://doi.org/10.3322/caac.21660.

[2] World Health Organization. *Global cancer burden growing, amidst mounting need for services*. News release, Lyon, France; Geneva, Switzerland. Available at: https://www.who.int/news/item/01-02-2024-global-cancer-burden-growing-amidst-mounting-need-for-services. 2024.

[3] C. Cuevas and D. Shibata. "Medical imaging in the diagnosis and management of cancer pain". In: *Current Pain and Headache Reports* 13.4 (2009), pp. 261–270. DOI: 10.1007/s11916-009-0042-9. URL: https://doi.org/10.1007/s11916-009-0042-9.

[4] Anna Maria Pavone et al. "Digital Pathology: A Comprehensive Review of Open-Source Histological Segmentation Software". In: *BioMedInformatics* 4.1 (2024), pp. 173–196. ISSN: 2673-7426. DOI: 10.3390/biomedinformatics4010012. URL: https://www.mdpi.com/2673-7426/4/1/12.

[5] Philippe Lambin et al. "Radiomics: Extracting more information from medical images using advanced feature analysis". In: *European Journal of Cancer* 48.4 (2012), pp. 441–446. ISSN: 0959-8049. DOI: https://doi.org/10.1016/j.ejca.2011.11.036. URL: https://www.sciencedirect.com/science/article/pii/S0959804911009993.

[6] International B. R. "Retracted: Developing an Efficient Cancer Detection and Prediction Tool Using Convolution Neural Network Integrated with Neural Pattern Recognition". In: *BioMed Research International* 2024 (2024), p. 9853939. DOI: 10.1155/2024/9853939. URL: https://doi.org/10.1155/2024/9853939.

[7] Irene Dankwa-Mullan and Dilanthi Weeraratne. "Artificial Intelligence and Machine Learning Technologies in Cancer Care: Addressing Disparities, Bias, and Data Diversity". In: *Cancer Discovery* 12.6 (2022), pp. 1423–1427. DOI: 10.1158/2159-8290.CD-22-0373. URL: https://doi.org/10.1158/2159-8290.CD-22-0373.

[8] Jonathan I. Epstein. "An update of the Gleason grading system". In: *The Journal of Urology* 183.2 (2010), pp. 433–440. DOI: 10.1016/j.juro.2009.10.046. URL: https://doi.org/10.1016/j.juro.2009.10.046.

[9] Jaeung Lee, Keunho Byeon, and Jin Tae Kwak. "Centroid-Aware Feature Recalibration for Cancer Grading in Pathology Images". In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2023: 26th International Conference, Vancouver, BC, Canada, October 8–12, 2023, Proceedings, Part II*. Vancouver, BC, Canada: Springer-Verlag, 2023, pp. 212–221. ISBN: 978-3-031-43894-3. DOI: 10.1007/978-3-031-43895-0_20. URL: https://doi.org/10.1007/978-3-031-43895-0_20.

[10] Emad A. Rakha et al. "Breast cancer prognostic classification in the molecular era: the role of histological grade". In: *Breast Cancer Research* 12.4 (2010). Epub 2010 Jul 30, p. 207. DOI: 10.1186/bcr2607. URL: https://doi.org/10.1186/bcr2607.

[11] Peter S. Ginter et al. "Histologic grading of breast carcinoma: a multi-institution study of interobserver variation using virtual microscopy". In: *Modern Pathology* 34.4 (2021). Epub 2020 Oct 19, pp. 701–709. DOI: 10.1038/s41379-020-00698-2. URL: https://doi.org/10.1038/s41379-020-00698-2.

[12] Yann LeCun and Yoshua Bengio. "Convolutional Networks for Images, Speech, and Time Series". In: *Handbook of Brain Theory and Neural Networks*. Ed. by Michael A. Arbib. MIT Press, 1995, p. 3361.

[13] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet classification with deep convolutional neural networks". In: *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'12. Lake Tahoe, Nevada: Curran Associates Inc., 2012, 1097–1105.

[14] K Simonyan and A Zisserman. "Very deep convolutional networks for large-scale image recognition". In: Computational and Biological Learning Society, 2015, pp. 1–14.

[15] Christian Szegedy et al. "Going deeper with convolutions". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 1–9. DOI: 10.1109/CVPR.2015.7298594.

[16] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[17] Ashish Vaswani et al. "Attention is all you need". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, 6000–6010. ISBN: 9781510860964.

[18] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: 2010.11929 [cs.CV]. URL: https://arxiv.org/abs/2010.11929.

[19] Ze Liu et al. *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows*. 2021. arXiv: 2103.14030 [cs.CV]. URL: https://arxiv.org/abs/2103.14030.

[20] Sanghyun Woo et al. "CBAM: Convolutional Block Attention Module". In: *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part VII*. Munich, Germany: Springer-Verlag, 2018, 3–19. ISBN: 978-3-030-01233-5. DOI: 10.1007/978-3-030-01234-2_1. URL: https://doi.org/10.1007/978-3-030-01234-2_1.

[21] Jongchan Park et al. *BAM: Bottleneck Attention Module*. 2018. arXiv: 1807.06514 [cs.CV]. URL: https://arxiv.org/abs/1807.06514.

[22] David Tellez et al. "Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology". In: *Medical Image Analysis* 58 (2019), p. 101544. ISSN: 1361-8415. DOI: https://doi.org/10.1016/j.media.2019.101544. URL: https://www.sciencedirect.com/science/article/pii/S1361841519300799.

[23] Rodrigo Escobar Díaz Guerrero et al. "A Data Augmentation Methodology to Reduce the Class Imbalance in Histopathology Images". In: *Journal of Imaging Informatics in Medicine* 37.4 (2024), pp. 1767–1782. ISSN: 2948-2933. DOI: 10.1007/s10278-024-01018-9. URL: https://doi.org/10.1007/s10278-024-01018-9.

[24] Abhijeet Patil et al. "Fast, Self Supervised, Fully Convolutional Color Normalization Of H&E Stained Images". In: *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)* (2020), pp. 1563–1567. URL: https://api.semanticscholar.org/CorpusID:227238956.

[25] Min Lin, Qiang Chen, and Shuicheng Yan. *Network In Network*. 2014. arXiv: 1312.4400 [cs.NE]. URL: https://arxiv.org/abs/1312.4400.

[26] Trinh Thi Le Vuong et al. "Joint categorical and ordinal learning for cancer grading in pathology images". In: *Medical Image Analysis* 73 (2021), p. 102206. ISSN: 1361-8415. DOI: https://doi.org/10.1016/j.media.2021.102206. URL: https://www.sciencedirect.com/science/article/pii/S1361841521002516.

[27] Ilya Loshchilov and Frank Hutter. *Decoupled Weight Decay Regularization*. 2019. arXiv: 1711.05101 [cs.LG]. URL: https://arxiv.org/abs/1711.05101.