

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

AI-based methods for cancer cells classification using histopathological images in a cross-cohort setup

Nuno Guilherme Amaral Santos



Mestrado em Engenharia Informática e Computação

Supervisors: Tânia Pereira, Hélder Oliveira

July 28, 2023

AI-based methods for cancer cells classification using histopathological images in a cross-cohort setup

Nuno Guilherme Amaral Santos

Mestrado em Engenharia Informática e Computação

July 28, 2023

Resumo

A histopatologia é o estudo das alterações nos tecidos causadas por doenças. O processo de análise de tecidos é de extrema importância para o diagnóstico, prognóstico e seleção de tratamento em pacientes com cancro. A capacidade de digitalizar lâminas histológicas para produzir imagens digitais (WSI - Whole Slide Images) permitiu que os histopatologistas fornecessem um diagnóstico preciso com base na classificação e quantificação de células cancerígenas nessas imagens.

Considerando que, nos dias atuais, essa metodologia é cada vez mais utilizada em laboratórios de patologia em todo o mundo, a aplicação de WSIs, o aumento dos casos de cancro ao longo dos anos e o tempo consumido para obter resultados precisos, é de extrema importância desenvolver um processo automatizado para melhorar a precisão, replicação dos resultados e reduzir o tempo consumido. Com isso em mente, o principal objetivo desta tese foi investigar e avaliar técnicas de inteligência artificial (IA) para a classificação de imagens histológicas. O objetivo final é criar uma metodologia que pudesse apoiar os patologistas no seu processo de diagnóstico, oferecendo informações adicionais para melhorar a precisão e eficiência dos diagnósticos. Aproveitando as capacidades de Deep Learning e métodos de visão computacional, esses modelos têm a capacidade de aprender padrões complexos e características distintas encontradas em imagens histológicas. Isso permite-lhes diferenciar entre vários tipos de tecidos, detectar células irregulares e, numa última instância, contribuir para a detecção e prognóstico de doenças. Dois conjuntos de dados foram estudados. Em primeiro lugar, o conjunto de dados PCAM foi utilizado para estudar o efeito dos hiperparâmetros nas métricas de avaliação usando um modelo VGG16 "do zero", usando esses valores como referência para a aplicação de métodos posteriores. O estudo de normalização, que se mostrou ineficaz, e de variação de cor, que teve um grande impacto positivo, também foi realizado. Por fim, devido às más características do conjunto de dados, três métodos foram implementados para tentar melhorar ainda mais os resultados. O redimensionamento foi realizado para mitigar o problema de baixa resolução do conjunto de dados, pesos de classe foram utilizados para reduzir a alta diferença entre o número de falsos negativos e falsos positivos nos resultados de teste, e, por último, um classificador de conjunto foi utilizado para combinar os modelos com melhor desempenho com diferentes arquiteturas, VGG19, ResNet50 e GoogleNet, a fim de combinar forças para obter um resultado ótimo. Em segundo lugar, o conjunto de dados MNIST de cancro colorretal foi explorado para estudar os efeitos das técnicas de normalização e aumento de cor que foram aplicadas ao conjunto de dados PCAM, medindo se a aplicação desses métodos teria o mesmo impacto positivo. Uma exploração da técnica de transferência de aprendizado também foi estudada, usando os melhores modelos testados no conjunto de dados PCAM, mas esse método teve um desempenho ruim, levando a crer que são necessárias mais experimentações, especialmente com hiperparâmetros.

Abstract

Histopathology is the study of changes in tissues caused by disease. The process of tissue analysis is highly important for the diagnosis, prognosis and treatment selection in cancer patients. The ability to scan histology slides to produce digital images (WSI - Whole Slide Images) has enabled histopathologists to provide an accurate diagnosis based on classification and quantification of cancer cells in these images.

Considering that in the current days this methodology is used more and more in pathology labs around the world, the application of WSI's, the increase of cancer cases over the years and the time consumed required to provide precise results, it is highly important to develop an automated process to improve accuracy, replication of results and reduce time consumed. With this in mind, the main purpose of this thesis was to investigate and assess artificial intelligence (AI) techniques for the classification of histology images. The ultimate goal was to create a methodology that could support pathologists in their diagnostic process by offering additional information to enhance the accuracy and efficiency of diagnoses. By leveraging the capabilities of deep learning and computer vision methods, these models have the ability to learn complex patterns and distinctive features found in histology images. This enables them to differentiate between various types of tissues, detect irregular cells, and ultimately contribute to the detection and prognosis of diseases. Two datasets were studied, firstly, the PCAM dataset was used to study the effect of hyper parameters on evaluation metrics using a VGG16 "from scratch" model, using these values as reference for further method application. The study of normalization, which proved ineffective, and color augmentation, which had great positive impact, was also studied. And finally, due to poor characteristics of the dataset, three methods were implemented to try and further improve outcome values. Resizing to try and mitigate low resolution problem of the dataset, class weights to try and lower the high ratio between False Negatives and False Positives in the test results, and lastly, an ensemble classifier to combine the best performing models with different architecture, VGG19, ResNet50 and GoogleNet, in order to combine strengths to obtain an optimal result. And secondly, the MNIST Colorectal Cancer dataset was explored in order to study the effects of normalization and color augmentation techniques that were applied on the PCAM dataset, measuring if the application of such methods would have the same positive impact. An exploration of transfer learning technique was also studied, using the best performing models tested on the PCAM dataset, but this method performed poorly, leading to believe that further experimentation is needed, specially with hyper parameters.

Acknowledgments

I would like to thank my supervisors Tânia Pereira and Hélder Oliveira for guiding me through this task, in spite of all challenges and set backs, i could not have found the right path that fit this assignment. I would also like to thank Francisco Silva for being understanding of my chaotic methods and lack of explaining capabilities, and still being help full in this journey. Many thanks to my Mom and Dad that listened to my complaints and calmed me down when things didn't go my way. Finally, i thank my friends, Zé, Amândio, Duarte, Pedro, Teresa and Marta, that gave me insight with their previous experience on similar assignments, which helped me find the correct attitude to face this task.

Nuno Santos

“A society grows great when old men plant trees whose shade they know they shall never sit in.”

Greek Proverb

Contents

Resumo	i
Abstract	ii
Acknowledgments	iii
1 Introduction	1
1.1 Context and Motivation	1
1.2 Objectives	1
1.3 Document Structure	2
2 Background	3
2.1 Histopathology	3
2.1.1 History	3
2.1.2 Modern Days	3
2.1.3 Whole slide imaging	4
2.2 Machine Learning	4
2.2.1 Brief History	4
2.2.2 Modern Days	4
2.3 Deep Learning	6
2.4 Convolutional Neural Networks	7
2.4.1 Hyper parameters	9
2.5 Data Preparation	9
2.5.1 Normalization	9
2.5.2 Data Augmentation	11
2.6 Summary	12
3 Related Work	13
3.1 Datasets	13
3.1.1 ImageNet Dataset	13
3.1.2 PCAM Dataset	14
3.1.3 MNIST Colorectal Cancer Dataset	16
3.2 Automated methods in Pathology	17
3.3 Summary	26
4 Methodology	27
4.1 Methodology applied on PCAM Dataset	27
4.1.1 Methods	27
4.1.2 Results	40

4.1.3	Discussion	45
4.2	MNIST Colorectal Cancer dataset	49
4.2.1	Results	49
4.2.2	Discussion	51
4.3	Summary	52
5	Conclusions	54
5.1	Future work	56
A	Image Classification - Detailed Results	57
A.1	Tests done on PCAM	57
A.1.1	Graphical analysis of table 4.2	57
A.1.2	Graphical analysis of table 4.4	57
	References	59

List of Figures

2.1	Traditional programming vs machine learning [34].	5
2.2	Left-most - labeled data on the left most, middle - correctly labeled data but useless for the task, right-most - unlabeled data [56].	5
2.3	Illustration on how Deep Learning is a sub-field of Machine Learning which in turn is a sub-field of Artificial Intelligence [71].	7
2.4	Neural Network architecture example, presenting the 3 types of layers [13]. . . .	8
2.5	Convolutional Neural Network Architecture Example	8
2.6	Data augmentation examples [4].	12
3.1	ImageNet image examples [1].	14
3.2	PCAM image examples. [9]	15
3.3	MNIST image examples presented and labeled by Kather <i>et al.</i> [29]:"(a) tumour epithelium, (b) simple stroma, (c) complex stroma (stroma that contains single tumour cells and/or single immune cells), (d) immune cell conglomerates, (e) debris and mucus, (f) mucosal glands, (g) adipose tissue, (h) background."	17
3.4	Visual representation of layers in the convolution process. [18]	19
3.5	Khened <i>et al.</i> proposed pipeline. [31]	21
4.1	VGG Architecture [42]	28
4.2	Horizontal/vertical flip and rotation	33
4.3	Examples of application of different intensities of Gaussian blur.	34
4.4	Example outcomes of applying grayscale method.	35
4.5	Variations in saturation, hue, brightness and contrast	36
4.6	Learning rate of 0.0001, SGD, BCE and 128 batch size	40
4.7	Learning rate of 0.0005, SGD, BCE and 128 batch size	41
4.8	Learning rate of 0.001, SGD, BCE and 128 batch size	41
4.9	Presentation of results of table 4.2 graphically.	42
4.10	Graphical presentation of results of table 4.5.	51
A.1	Analysis of AUC and accuracy values on table 4.2, the label of the method is presented above the point of it's classification on the graph. Tests 1 and 2 are overlapping.	57
A.2	Analysis of AUC and accuracy values on table 4.4, the label of the method is presented above the point of it's classification on the graph.	58

List of Tables

3.1	ImageNet Dataset.	14
3.2	PCam (PatchCamelyon) Dataset.	15
3.3	MNIST Colorectal Cancer Dataset	16
3.4	Parameters of the CNN architecture. [18]	19
3.5	Experimented methods and respective Accuracy. [18]	20
4.1	Test results with learning rate variation on a "from scratch" VGG16 model	41
4.2	Test results using . Lr(learning rate). DA(data augmentation). CA(color augmentation. b -brightness, c - contrast, s - saturation, h - hue), where values are varied between (min, max)	42
4.3	Test label. Label is assigned to what test was made. Ensemble(A+B+C) means an ensemble classifier made with the models from experiment A,B and C, giving the most weight(0.4) to the model that outperformed the others(in this case was VGG19 in all situations). Class weights are only used for training, in the case of the ensemble, class weights are N/A because that value is only used for training, and as such, it depends on the models used in the ensemble. Threshold is the value for which a class is classified true or false, a threshold of 0.5 means that if the output of a model is equal to or greater than 0.5, it is classified as belonging to the positive class, if the output is less than 0.5, it is classified as belonging to the negative class.	44
4.4	Test results.TN - True Negatives, FN - False Negatives, TP - True Positives, FP - False Positives.	45
4.5	Tests comparing the application of normalization, color augmentation and transfer learning	50
4.6	Test comparison with similar articles Rane <i>et al.</i> [50] and Zhong <i>et al.</i> [76] . . .	53

Abbreviations and symbols

AI	Artificial Intelligence
ANN	Artificial Neural Network
API	Application Programming Interface
AUC	Area Under the Curve
CNN	Convolutional Neural Network
DL	Deep Learning
FCN	Fully Convolutional Network
GPU	Graphics Processing Unit
H	E
Hematoxylin	Eosin
ML	Machine Learning
MNIST	Modified National Institute of Standards and Technology
PCAM	PatchCamelyon
ReLU	Rectified Linear Unit
ROC	Receiver Operating Characteristic
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
WHO	World Health Organisation
WSI	Whole Slide Images

Chapter 1

Introduction

1.1 Context and Motivation

Cancer is still one of the most deadly diseases currently on a global scale. Current incidences rise from year to year, worldwide the number of new cases is estimated to be above 14 million of new cases each year, despite of the decline of mortality rate[72]. A projection for cancer cases in the UK estimates most common types of cancer by the year 2035 are breast and prostate cancer[61]. In the USA for instance, by the year 2030, lung cancer will be the most deadly[49]. We must consider this to understand the demands, time wise, of a cancer diagnosis and treatment.

Histopathology is one of steps that are taken between manifestations of a disease and its diagnosis. It is the study and examination of manifestations of disease of the tissues under a microscope. After the removal of a tissue sample from the body, it is put in a formalin solution(most common) to prevent decay. This sample is then prepared for viewing under a microscope[36]. Staining of these processed histology slides can be done in order to reveal cellular components.Using digital cameras is becoming a more common approach to capture histopathological images. This approach, called virtual microscopy or Whole Slide Scanning, and although rather recent, allows the digitization of whole slides in high resolution. As a final stage, a pathologist will then examine the slides and provide a tissue report describing the histological findings[26]. It is important to note that this last stage of examination is done by a lab technician, and so the results may slightly vary based on experience, time frame and other possible variables since it is a classification task and therefor, subjective.

1.2 Objectives

Considering the number of active cancer cases and time consuming process of cancer classification and quantification done by a technician and the possibility of result variation due to external factors, it is a prominent target for an AI model implementation. there are 3 main objectives for this work. First off is to contribute to the research and exploration endeavour world-wide to the use of Artificial Intelligence models in modern medicine and laboratory procedures in order to facilitate,

enrich or increase efficiency in said procedures. Two final goals are to develop AI based methods in order to classify and quantify cancer cells in WSI's in a way to retrieve additional information to aid the lab technicians and healthcare professionals to make a more informed decision, and to be able to begin and conclude such process so that it does not hold back the already time consuming process of histopathology and so make it as efficient as possible. It is important to point out that none of the goals is to use an AI model as a substitute to a lab technician. There are no instances in modern medicine where a decision/conclusion is made solely using information from AI methods, there is always confirmation or exploration of results given by said methods, by the according professionals. On the other hand information from digital sensors is commonly used and accepted, although the stage of confirmation and/or exploration may still take place, a sensor result is of a different type since it is measured while a result by an AI model is a decision made by that model, and so it would be unethical to base treatment or diagnosis solely based on an AI model output.

1.3 Document Structure

Besides the introduction, this document contains 4 more chapters. In Chapter 2, the state of the art is explained. In Chapter 2, the literary work is explored. In Chapter 4, the methodology is explained, the results presented and discussed. In Chapter 5, conclusions and possible future work presented.

Chapter 2

Background

In this chapter, background needed for the understanding of this work is presented. It introduces two areas that are fundamental for histopathology image classification. It presents how histopathology started and evolved to the technology that allows us to analyse sample's images digitally, this meaning the Whole Slide Imaging. And also the evolution, importance and application of Machine Learning methods, focusing in deep learning, convolutional networks and the parameters that influence their classification outcome. It is also presented methods applied to datasets before and during the process of training a ML model.

2.1 Histopathology

2.1.1 History

Although it was not called Histopathology at the time, this specialty has been practiced since the beginning of the 19th century. It was only in 1838 that the one of the first mentions of the practice is found, with a book from Johannes Miller, On the Nature and Structure Characteristics of Cancer. In 1673 Anton van Leeuwenhoek [12] started the development of simple microscopes with single lenses that gave improved magnification and resolution. Different laboratory chemicals were investigated for use as fixatives. Formalin, the most common fixative used today, was first used in 1893[67].

2.1.2 Modern Days

After tissue removal or sampling from the patient, this being an autopsy, biopsy or surgery, and placed in a fixative to stabilize the sample and prevent decay. The most common fixative is 10% neutral buffered formalin[51].The sample must be fixated from 6 to 12 hours after the biopsy. After 24 hours after biopsy, the blade should be ready to examine. The sample is then prepared for viewing under a microscope. Modern techniques have provided tools for a more comprehensive view of disease and its effect on tissues, since the preparation process preserves the underlying tissue structure. With the recent advance in whole slide digital scanners, tissue histopathology

slides can now be digitized and stored in digital image form. Due to this, it provided the tool for the evolution of most automated histopathology image analysis techniques. Despite the progress made in this area thus far, this is still a large area of open research due to the variety of imaging methods and disease-specific characteristics[26].

2.1.3 Whole slide imaging

Whole slide imaging is the technology that enables a digital program to work with data from tissue samples. Enables the integration of digital models in the microscopic examination step of the histopathology process because it allows for the digitization of glass slides containing tissue samples [46]. The first slide scanner, used to create the WSIs, was created by James Bacus and called the BLISS. Over the years, several manufacturers have developed slide scanners with the purpose of addressing clinical needs. Since the start of utilization of WSIs in primary diagnosis, numerous validation studies have been published indicating that WSI is a reliable tool for routine diagnosis in surgical pathology [45]. Eventhough the usage of WSI is a tremendous breakthrough, some technical challenges still prove to be a major problem in the adoption to clinical area. The biggest challenge, one that will also be tackled in this dissertation, is data management [46]. A single compressed WSI, scanned at fourty times magnification for a single focal plane, is typically 1–4 GB or larger in size [45]. Considering the average size of a dataset,that will be shown later,it is easy to see how the size of each WSI can become a setback when dealing with large datasets.

2.2 Machine Learning

2.2.1 Brief History

It is uncertain where machine learning was born, but the first mention of "machine learning" is dated back to 1952 by Arthur Samuel of IBM [68]. He developed a program that was able to play checkers by using functions that calculated the probability of each side winning. Samuel also developed mechanisms for the program to become better such as remembering the previous positions played [54].

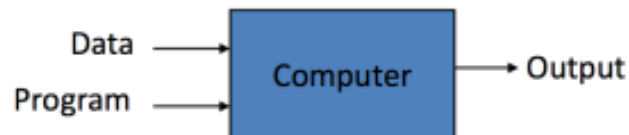
In the years 1957,1959,1960, a psychologist from the Cornell University, New York,named Frank Rosenblatt, created a machine for recognizing the letter of the alphabet, based of a human nervous system, became a prototype of modern artificial neural networks(ANN) [23].

2.2.2 Modern Days

In today's world, programs have advanced to perform a wide range of tasks using different approaches. While the basic principle remains consistent, the distinction between simple and complex programs lies in their input and output. Every program takes an input and produces an output, but this output is already predefined. This meaning that can only respond with answers it was programmed with. It receives an input(or dataset) and passing it through it's algorithm, returns an output as is sketched in 2.1. Here is where machine learning differs. Tom M. Mitchell [40] gave

an extensively acknowledged definition of machine learning algorithms in 1997, "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T , as measured by P , improves with experience E ". So machine learning algorithms are able to receive a dataset(or mentioned above as "experiences") and the expected output, returning the algorithm required to reach said output [68].

Traditional Programming



Machine Learning

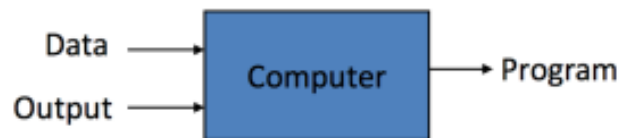


Figure 2.1: Traditional programming vs machine learning [34].

There are 2 types of datasets, labeled and unlabeled. Labeled data processes both the input and its classification(machine readable). Unlabeled data consists of only the input(the data) but not its classification in machine readable form [5].

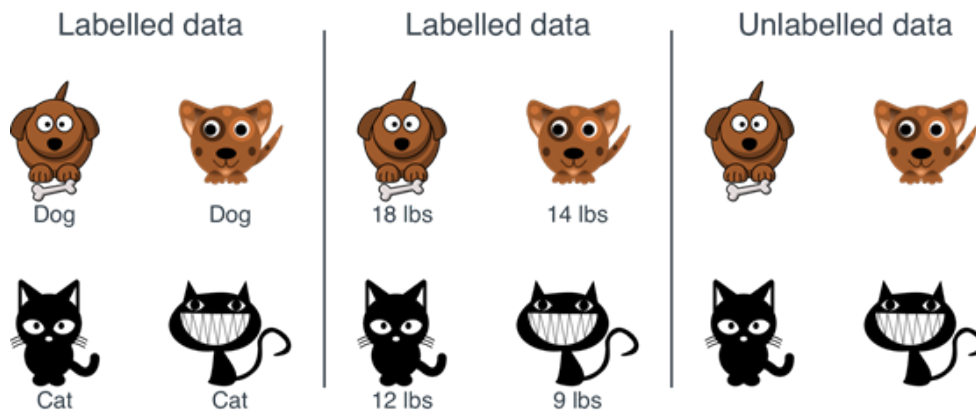


Figure 2.2: Left-most - labeled data on the left most, middle - correctly labeled data but useless for the task, right-most - unlabeled data [56].

Although there are various types of machine learning algorithms, they can be broadly classified into 3 categories:

- Supervised Learning: uses labeled data, presented in left most in ??, and extracts features from the dataset shaping a model based on those. Using this type of dataset for training,

throughout iteration, a model/algorithm is built that is able to predict the output of new inputs [41]. Supervised learning can be divided into 2 further types:

- Classification: assign data into specific categories. Common classification algorithms are linear classifiers.
- Regression: is used to understand the relationship between dependent and independent variables. Commonly used for projections.
- Unsupervised Learning: uses unlabeled data, presented in the right most in ??, and discovers hidden patterns or data clusterings or groupings and is more similar to the human way of thinking due to the absence of categorized data. It can be further categorized into 2 types:
 - Clustering: groups objects based on their similarities, and also differences to other groups. Finds common characteristics on data and categorizes them as per the presence or absence of those characteristics.
 - Association: finds association between objects in data in order to predict what events most likely happen together.
- Reinforcement Learning: uses a reward/penalty system where the optimal solution is not known. As such, reward functions are used depending on the task, and the agent must react in order to obtain the most desirable action with the objective of maximizing the reward. AI game models or simulation based intelligence commonly use reinforcement learning.

2.3 Deep Learning

Deep learning is part of machine learning and uses supervised learning in situations of image classification or object detection, which is the subject of this dissertation, and as such, uses labeled data. Although the current increase in use of deep learning algorithms, they are not recent, in fact it can be dated back to the 1940s. The machine created by Frank Rosenblatt [23] mentioned above, called the 'perceptron' was the first breakthrough in the area of neural network, with the training of a single neuron. The recent boost in performance in deep learning ideas is due to a variety of factors being some of them: the increase in number and size of available datasets, increase in computational power and the opportunity of model training on GPUs(Graphics Processing Unit). Neural Networks are a mathematical model inspired by the way human's brains work where the most simple unit is a neuron. A model of a neuron can be used as a binary classifier and can have multiple inputs. Connecting several neurons we obtain a neural network.

They are arranged in layers, one input layer, one output layers and N hidden layers between them as presented in 2.4. The number of layers in a network and the number of neurons per layer are called the hyperparameters of the architecture. Although there are several types of neural networks such as recurrent neural networks(RNNs), generative adversarial neural networks(GANs),

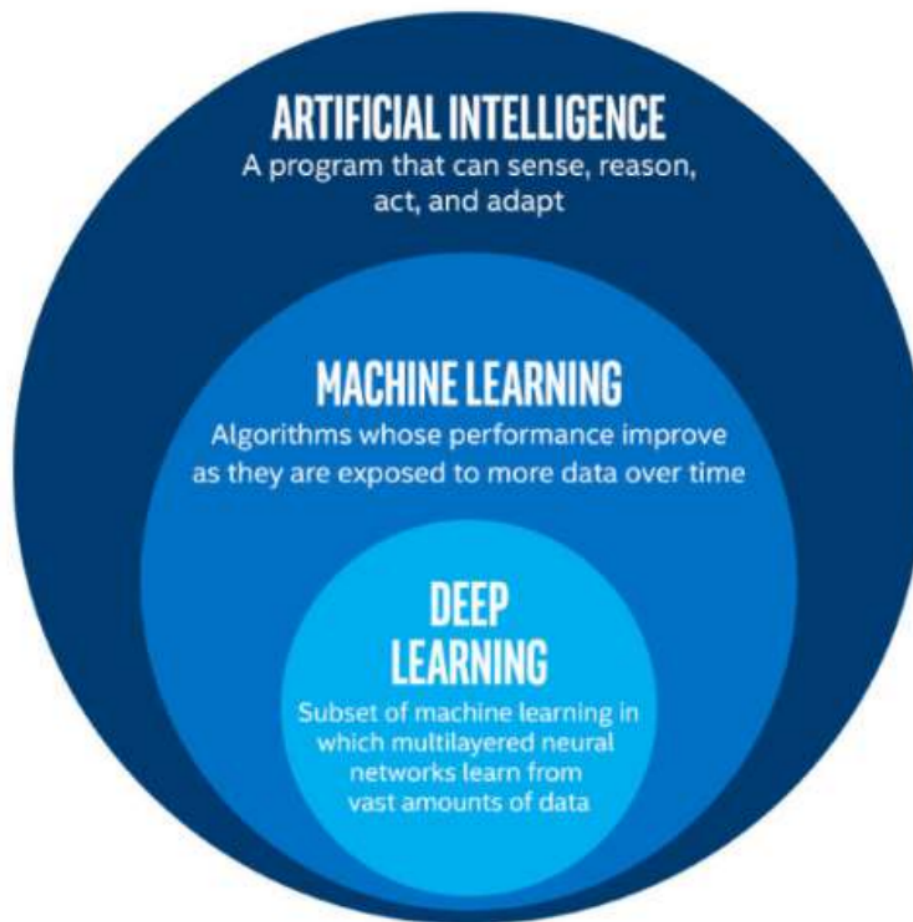


Figure 2.3: Illustration on how Deep Learning is a sub-field of Machine Learning which in turn is a sub-field of Artificial Intelligence [71].

the focus of this paper will be convolutional neural networks(CNNs) since they are most commonly use in image patter recognition tasks due to their performance and accuracy, specially in face recognition, object detection, vision components in robotics and self-driving cars [44] [70].

2.4 Convolutional Neural Networks

A CNN contains three main types of neutral layers:

- Convolutional layers: these make up the core building block of a CNN. It uses something called kernel or filter that moves along the image in order to identify features creating a feature map. These kernels are made up of weights, which are to be learned throughout the training. After each convolution operation, a CNN applies a Rectified Linear Unit (ReLU) transformation to the feature map, introducing nonlinearity to the model. Several sequential convolutional layers can be present, creating a feature hierarchy within the CNN [70].

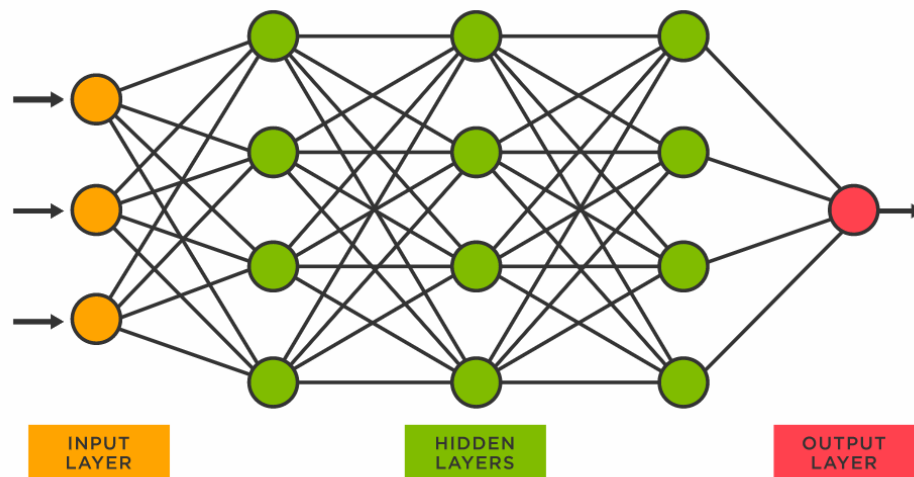


Figure 2.4: Neural Network architecture example, presenting the 3 types of layers [13].

- **Pooling layers:** when the input parameters are too numerous, pooling layers are used to perform dimensionality reduction. Similar to the convolutional layer, it applies a filter across the image but in this case an aggregation function is used. Although some information is lost, it is prejudicial to have too many input parameters [70]. There are 2 main types of pooling:
 - Max pooling selects the pixel with the maximum value to send to the output array.
 - Average pooling calculates the average value to send to the output array.
- **Fully connected layers** perform the task of classification based on the features extracted from convolutional layers, and then reduced by pooling layers. While these tend to use ReLu functions, fully connected layers usually use various types of activation functions to classify inputs appropriately [70].

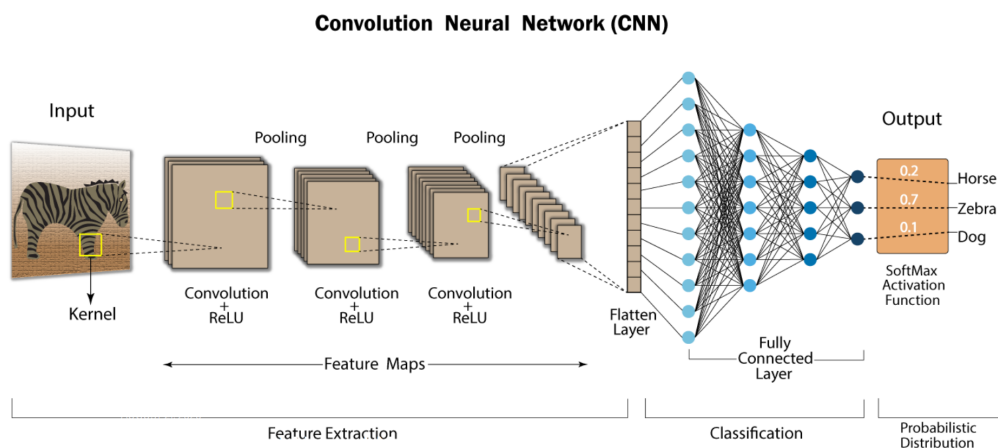


Figure 2.5: Convolutional Neural Network Architecture Example

2.4.1 Hyper parameters

In a convolutional neural network (CNN) project, several key factors significantly impact the training process and the performance of the model. The learning rate, which determines the step size for updating model parameters, plays a crucial role in the convergence and stability of the model. A higher learning rate may yield faster convergence but risks overshooting the optimal solution, while a lower learning rate can lead to slow convergence or getting stuck in sub optimal solutions [24] [10].

Another important consideration is the batch size, which refers to the number of samples processed in each iteration of the training algorithm. The batch size affects both the memory requirements during training and the speed of convergence. Larger batch sizes can provide more efficient computations but may require more memory, while smaller batch sizes yield a noisier estimate of the gradient but can result in faster convergence and better generalization [35].

Furthermore, the number of epochs, which represents the number of passes through the entire training dataset, is a critical factor in model training. Running multiple epochs allows the model to see the data multiple times, facilitating better convergence and improved generalization. However, training for too many epochs can lead to overfitting the training data, where the model becomes too specialized and fails to generalize well to unseen data [24] [75].

These considerations demonstrate the importance of carefully selecting appropriate values for the learning rate, batch size, and the number of epochs in a CNN project. Finding the right balance is crucial for achieving optimal training performance and ensuring the generalization capability of the model.

2.5 Data Preparation

Before starting training the model with our dataset, some data preparation is needed in order to find and resolve data quality issues that might be present and prepare data for training. In its raw format, data can be missing, inconsistent or erroneous, for example: noise and outliers, missing values, duplicated, inconsistent or incorrect data [41]. Data processing methods:

- Feature Extraction - extract features from data on which analysis can be performed.
- Data Cleaning - data may require extra parsing efforts in order to become readable.
- Data Transformation - it may be necessary to change some of the values of the data.
- Feature Engineering - the process of using domain knowledge of the data to create features that might help when solving the problem.

2.5.1 Normalization

Data normalization, also known as feature scaling or standardization, is a technique used to transform the values of different features in a dataset to a common scale. It involves adjusting the range

and distribution of the data so that they fall within a specific range or follow a specific distribution. The main goal of data normalization is to ensure that all features contribute equally to the analysis and modeling processes.

By normalizing the data, one can eliminate any potential bias or dominance that certain features may have due to their larger values or different scales. This ensures that each feature is treated fairly and equally in the analysis or machine learning models. It also helps to improve the stability and convergence of algorithms during the training process.

The primary purpose of data normalization is to bring the features of a dataset onto a common scale. By normalizing the data, it is possible to eliminate the influence of the magnitude of the features, allowing the learning algorithm to focus on the relative importance and relationships between the features.

Z-Score Normalization (Standardization): Z-score normalization, also known as standardization, transforms the data so that it has zero mean and unit variance. It subtracts the mean value of each feature and divides it by the standard deviation. This process ensures that the transformed data has a mean of zero and a standard deviation of one. Standardization is suitable when the distribution of the features is approximately Gaussian or when the algorithm assumes zero-centered data [28].

Min-Max Scaling: Min-max scaling, also called feature scaling, transforms the data to a fixed range, typically between 0 and 1. It subtracts the minimum value of each feature and divides it by the range (maximum value minus minimum value). Min-max scaling preserves the original distribution of the data while scaling it to a specific range. It is appropriate when the distribution of the features is not necessarily Gaussian and when preserving the relationships between the data points is important [2].

When considering machine learning, there are several benefits [8]:

- Normalized data helps the learning algorithm converge faster as it reduces the chances of getting stuck in local optima and accelerates gradient descent.
- Stable Learning: Normalization helps stabilize the learning process by preventing features with larger magnitudes from dominating the learning algorithm.
- Better Generalization: Normalization helps improve the generalization performance of the model by reducing the sensitivity to the scale of the input features.
- Compatibility with Distance Metrics: Normalized data ensures that distance-based similarity measures are not biased towards features with larger scales.

In this dissertation the focus will be on the images, and so if a value of a pixel, color, intensity is too extreme, it may change the analysis of the dataset. All values are re-scaled so that they have a mean equal to 0 and standard deviation equal to 1. Values will typically lie in range $[-3,3]$ or $[-1,1]$ [63].

2.5.2 Data Augmentation

Another possible step that can be made depending on the analysis of the data quality is data augmentation. Depending on the type of classes and their quantity and quality present in the dataset. Due to class imbalance or presence of outliers, the model can be accurate with the training data but perform poorly with test data(overfitting) or perform poorly in both situations(underfitting). Data augmentation is a crucial technique in training convolutional neural networks (CNNs) that aims to increase the diversity and size of the training dataset by applying various transformations to the existing images. It helps address the challenge of limited training data, which can lead to overfitting and poor generalization of the model. By generating augmented samples, the model becomes more robust and can better capture the underlying patterns and variations in the data. [5] [33]. These are the most commonly used data augmentation techniques used in CNN projects [59]:

- **Horizontal and Vertical Flips:** This technique involves randomly flipping the image horizontally or vertically. It exploits the symmetry and invariance properties of many objects, such as faces or animals, and effectively increases the size of the training set.
- **Random Rotations:** By applying random rotations to the images within a certain range, such as -10 to +10 degrees, the model becomes more tolerant to different orientations of objects, improving its ability to recognize them from different viewpoints.
- **Image Translations:** Shifting the image horizontally or vertically introduces variations in the object's position within the image. This technique helps the model learn spatial invariance and improves its ability to locate objects irrespective of their exact position.
- **Zooming and Scaling:** Zooming in or out and rescaling the images introduces variations in object sizes. This augmentation technique helps the model become more robust to changes in scale and better generalize across different object sizes present in the dataset.
- **Brightness and Contrast Adjustments:** Modifying the brightness and contrast of the images helps the model handle variations in lighting conditions. It enhances the model's ability to recognize objects under different illumination levels.
- **Noise Injection:** Adding random noise to the images simulates the presence of noise or artifacts commonly encountered in real-world scenarios. It helps the model learn to distinguish relevant features from noise and improves its overall robustness.

By applying these data augmentation techniques, also illustrated in 2.6, the training dataset is effectively expanded, providing the CNN model with a more diverse and representative set of examples. This helps prevent overfitting, reduces the risk of memorization, and improves the model's generalization performance [48].

It's worth noting that the choice and extent of data augmentation techniques depend on the specific characteristics of the dataset and the problem domain. Some domains may benefit from

specific augmentation techniques tailored to their unique characteristics. Therefore, it is important to carefully select and evaluate the impact of different augmentation strategies on the model's performance [16].

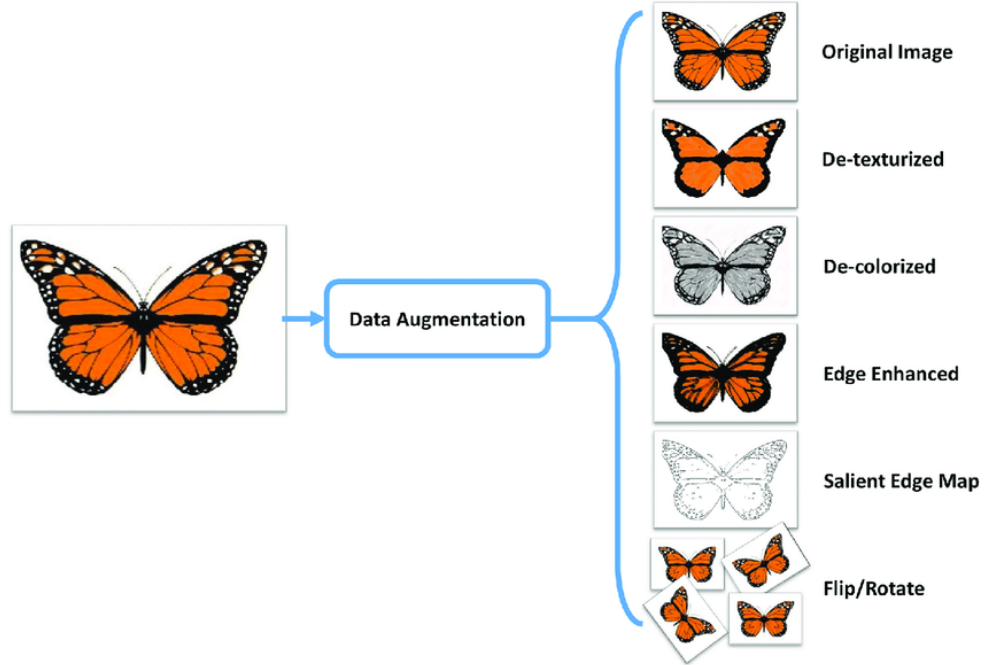


Figure 2.6: Data augmentation examples [4].

2.6 Summary

When considering an image classification task there are several variables that can influence the outcome of the classification depending on the pipeline chosen. Beginning at the dataset used for the task, exploring its characteristics and employing methods to deal with any inconsistency, such as analysing and pre-processing the data, verify the class balance and quality of the images and if needed, apply methods of normalization and data augmentation. When advancing to the next phase of the pipeline, which is exploring the model architecture and hyper parameters such as learning rate, number of epochs and batch size, there is no certain solution to obtain optimal results, as will be explored in 3.

Chapter 3

Related Work

This chapter includes the presentation and discussion of some methods used in various sorts of image classification problems. Documented experiments and articles that provide insightful methodologies that might be used, adapted or studied in order to better develop a methodology that better suits this dissertation. It encompasses presentation of different datasets, data processing methods, pipeline architecture, data transformation, and CNN architecture.

3.1 Datasets

The characteristics of a dataset affect the outcome of the model using it, being size, class balancing or overall quality. As explained before 2.1.3, the development of WSI technology allowed for the introduction of deep learning algorithms in histopathology but with an obstacle: image size. Some public datasets were found, to be used for model training and testing but they differ in image size and magnification, and may differ in cancer type. But due to their structure, it was only possible to use 3 datasets: ImageNet [19], PCAM [7], and MNIST [30].

3.1.1 ImageNet Dataset

The ImageNet [19] dataset is a widely recognized benchmark dataset in the field of computer vision. Introduced in the seminal work "ImageNet: A large-scale hierarchical image database" by Deng et al. [19], it has significantly contributed to the advancement of deep learning and image recognition tasks. ImageNet is known for its vast collection of labeled images, covering a comprehensive range of object categories. The original ImageNet dataset consisted of over 14 million images, as detailed in table 3.1, but for practical purposes, researchers commonly refer to the ILSVRC2012 subset. This subset contains approximately 1.2 million images, with each image associated with one of the 1,000 object classes. The images in ImageNet exhibit a wide variety of visual characteristics, including diverse objects, scenes, and backgrounds, captured under varying lighting conditions, viewpoints, and scales.

With such an extensive collection of labeled images, ImageNet provides a rich resource for training and evaluating deep learning models. Its large-scale nature enables researchers to develop

robust algorithms that can handle real-world variations and complexities, thereby pushing the boundaries of image classification, object detection, and other computer vision tasks.

Table 3.1: ImageNet Dataset.

Property	Description
Name	ImageNet
Number of Images	Over 14 million
Number of Classes	21,841
Number of Images per Class	Varies per class
Size	Varies (images of various resolutions)



Figure 3.1: ImageNet image examples [1].

3.1.2 PCAM Dataset

The PCAM dataset [7] presented in table 3.2, also known as PatchCamelyon, is a specialized dataset designed for histopathology image analysis in the context of cancer detection. It focuses on the detection of metastatic tissue in lymph node sections obtained from breast cancer patients. PCAM offers researchers an opportunity to explore the application of deep learning and image

analysis techniques in the field of pathology. Comprising a total of 327,680 image patches, PCAM derives from 400 whole-slide images that consisted the CAMELYON16 challenge, but due to the size of each WSI, it's usage depended on high computational requirements and as such the authors of PCAM were able to create an algorithm where several patches were taken from each WSI. Each image patch measures 96x96 pixels and is labeled as either positive (containing metastatic tissue) or negative (without metastatic tissue). This binary classification setup allows researchers to develop and evaluate machine learning algorithms specifically tailored for accurate cancer detection.

The PCAM dataset provides a unique insight into the microscopic characteristics of cancerous tissue and the visual indicators of metastasis. Researchers can leverage PCAM to train models capable of recognizing subtle visual cues associated with metastatic tissue, contributing to the development of more accurate and efficient cancer detection systems. It is important to note that the algorithm used to retrieve the patches from WSI allowed for some images to be classified as cancerous when very little information was present. As mentioned by the authors, the image was classified as cancerous even if only one pixel of tumor tissue is present in the center of the image.

Table 3.2: PCam (PatchCamelyon) Dataset.

Property	Description
Name	PCam (PatchCamelyon) dataset
Number of Images	327,680
Number of Classes	2
Classes	Positive or Negative
Number of Images per Class	Equal distribution
Size	96xp by 96xp

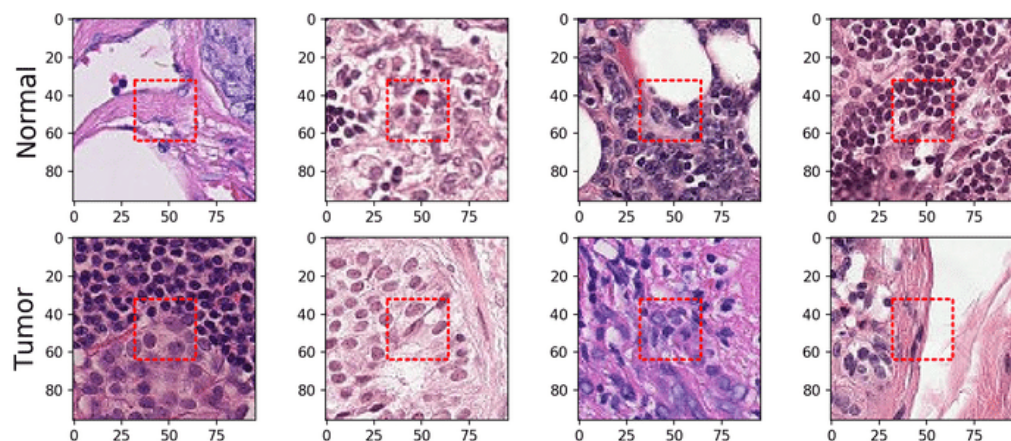


Figure 3.2: PCAM image examples. [9]

3.1.3 MNIST Colorectal Cancer Dataset

The MNIST Colorectal Cancer dataset [30] extends the well-known MNIST dataset to the domain of colorectal cancer detection. It focuses on histopathology images of colorectal tissue samples, aiming to facilitate research in machine learning-based cancer classification. The MNIST colorectal cancer dataset offers a comprehensive collection of histology images containing multiple classes representing different histological textures and patterns found in colorectal cancer. It contains 5000 images with 625 of each class, with size of 150px by 150px. The several classes present in the dataset are empty, debris, adipose, mucosa, lymphoma, complex, stroma and tumor.

Table 3.3: MNIST Colorectal Cancer Dataset

Property	Description
Name	MNIST Colorectal Cancer dataset
Number of Images	5,000
Number of Classes	8
Number of Images per Class	62
Classes	empty, debris, adipose, mucosa, lymphoma, complex, stroma, and tumor
Size	150px by 150px

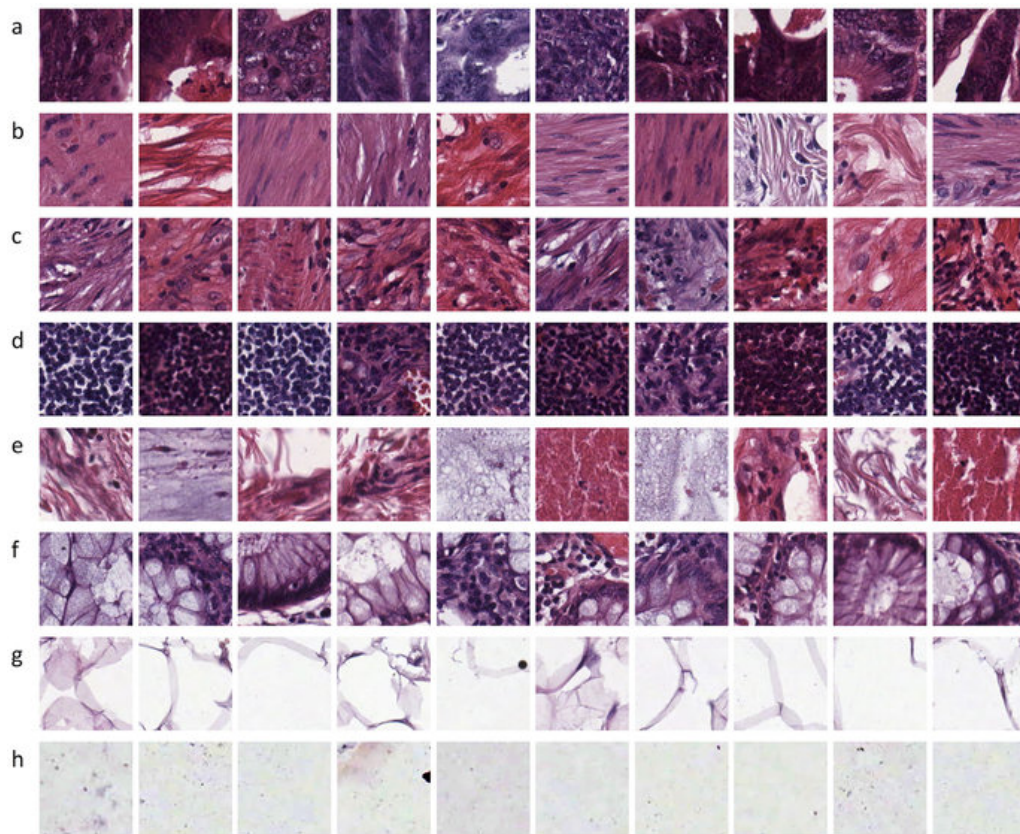


Figure 3.3: MNIST image examples presented and labeled by Kather *et al.* [29]: "(a) tumour epithelium, (b) simple stroma, (c) complex stroma (stroma that contains single tumour cells and/or single immune cells), (d) immune cell conglomerates, (e) debris and mucus, (f) mucosal glands, (g) adipose tissue, (h) background."

3.2 Automated methods in Pathology

Ayyad *et al.* [6] explores recent developments in combining the use of artificial intelligence and histopathological images to detect prostate cancer. The authors review several studies that used different methods to detect prostate cancer in histopathological images using image processing and found several common challenges such as:

- Large image size leads to higher computational and memory requirements, higher difficulty in applying data augmentation methods, more complex data management solutions and worse model performance and results overall. Some solutions are based on dividing the image into partial regions called patches and used to train the model. Reducing the dimension of the image was also tried but this proved to be less effective measure that lead to loss of information and subsequently lower model accuracy.
- Insufficient labeled data for training of a deep learning model can result in poor generalization performance on new, unseen data and can't generalize well to real-world scenarios (overfitting). This was alleviated by data augmentation or applying transfer learning tech-

niques or even the use of weak and unlabeled data. In spite of these methods, this challenge depends greatly on the amount of labeled data available.

- Difficulty in standardizing data due to different imaging techniques and stains that result in different image characteristics that are unassociated with the underlying biological factors. This can result in several obstacles such as color variation and noise, where possible solutions are data preprocessing techniques that include conversion to grayscale, color normalization and color augmentation.

Several solutions for this challenges are explored and presented in the papers below, as well as neural networks architectures chosen, although the authors noted that various neural network models such as Convolutional Neural Networks (CNNs) were the most common and yielded better results when the input data were images.

Shao *et al.* [58] objective was to use histopathological images in conjunction with MRI scans. One of the ideas used was the application of similar methods to both types of images in order to overlap them. However for this paper the methods that may be interesting were the ones until feature extraction from the histopathological images. For preprocessing the authors tried to verify if any data augmentation methods were used, such as horizontal flipping or angle rotation. Both images (MRI and WSI) were fed into a 2 stage registration architecture using a CNNGeometric and ProsRegNet networks that consisted of feature extraction, feature matching, and transformation parameter estimation.

Sali *et al.* [52] proposed a pipeline that applied patch extraction, data augmentation, normalization, standardization methods and experimented with several types of architecture. The authors devised three different experiments using fully supervised, unsupervised and weakly supervised feature learning. For fully and weakly supervised learning, a ResNet34 architecture was used although the fully connected layers were removed for the feature extractor and a dense layer receiving that output. A softmax function was also used to calculate the probability of each class. As for unsupervised learning, a ResNet18 architecture was used for the feature extractor and convolutional and up-sampling layers for the decoder. Due to the image size present in the dataset, a sliding window method was used in order to generate smaller patches from a WSI. Considering the color variation challenge, several solutions were proposed such as color balancing, gray-scaling and stain normalization methods proposed by Vahadane *et al.* [69].

Vahadane *et al.* [69] presents two important methods for processing histopathology images: Structure-Preserving Color Normalization (SPCN) and Sparse Stain Separation (SNMF). Sparse stain separation involves representing the image in a sparse format to preserve the structural information while reducing the impact of noise and then adding the constrain of sparseness and separating the different stains present in the image using a sparse representation approach. Final step is involves recombining the stain-separated channels to obtain the final images with improved representation of tissue structures. SPCN has the purpose of obtaining an accurate stain density map. It corrects for non-uniformities in the intensity values across the image, adjusts the color

intensity and hue of the image to correct for differences in staining protocols and illumination conditions, and adds the constraint of sparseness by using SNMF.

Dabeer *et al.* [18] proposes a pipeline that starts with preprocessing data by removing redundant pixels in the image using compression techniques that are available in the OpenCV library. As for feature extraction, the authors propose a CNN architecture like the one presented in figure 3.4 and table 3.4.

Layer attribute	L1	L2	L3	L4	L5	L6
Type	conv	pool	conv	pool	conv	pool
Channel	32	–	64	–	128	–
Filter Size	5x5	–	5x5	–	5x5	–
Conv. stride	1x1	–	1x1	–	1x1	–
Pooling size	–	3x3	–	3x3	–	3x3
Pooling stride	–	1x1	–	1x1	–	1x1
Padding size	same	none	–	none	–	none
Activation	ReLu	–	ReLu	–	ReLu	–

Table 3.4: Parameters of the CNN architecture. [18]

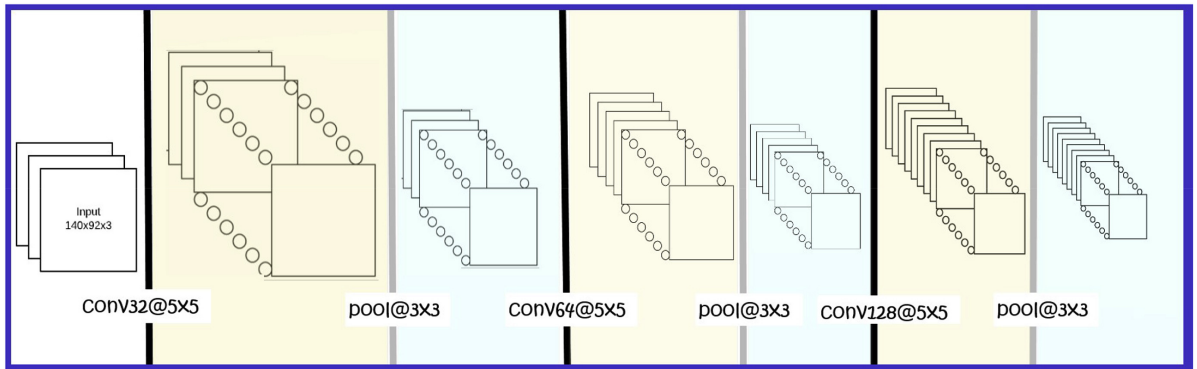


Figure 3.4: Visual representation of layers in the convolution process. [18]

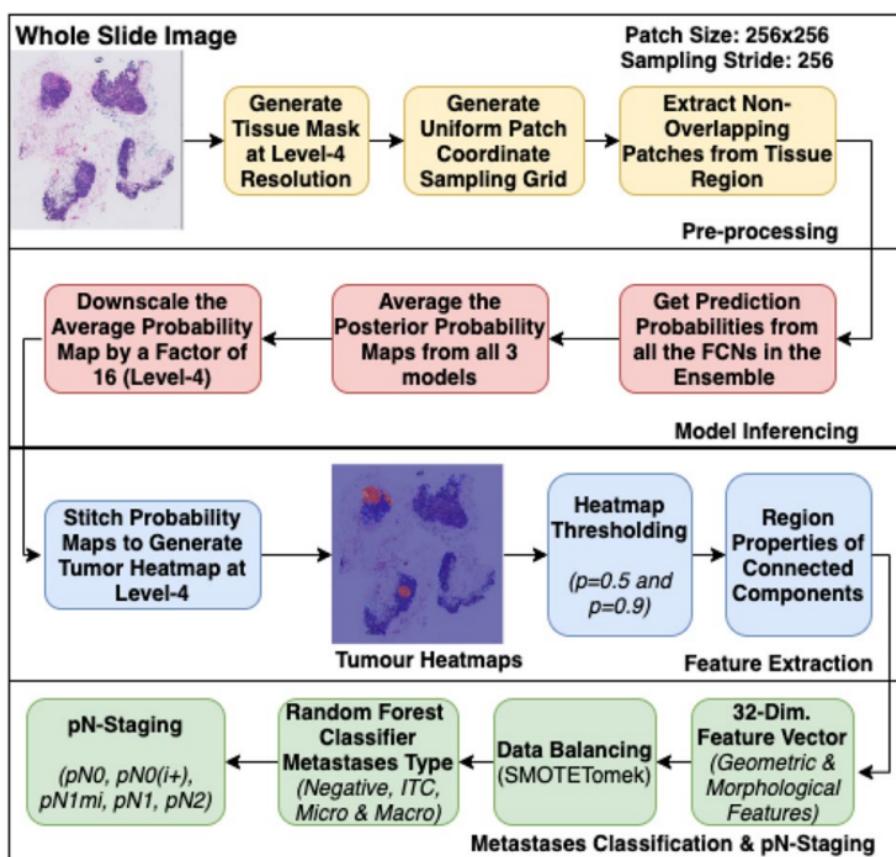
The authors then use two ReLu activation layers and a Softmax activation layer to calculate the loss and alter the weights of the internal hidden nodes. The final results obtained proved that deep convolutional neural networks had the best accuracy out of all methods experimented 3.5.

Year	Method Used	Validation Accuracy Range (in %)	Error Rate
2017	K-Nearest Neighbor[53]	83 to 86	19.28
2017	Pre-Trained Networks[64]	80 to 89	4.74
2017	Feature Extracted Using CNN[62]	83 to 90	4.28
2017	Deep Convolution Neural Network [3]	91.54	8.54

Table 3.5: Experimented methods and respective Accuracy. [18]

Shahidi *et al.* [57] is a comparison paper that studies several approaches when dealing with deep learning in histopathology. Describing the pipeline, it starts with analyzing databases with labeled data. Next step described is the application of data augmentation and preprocessing methods such as rotation, flipping, resizing and normalization. The authors compared several deep learning architectures and optimization functions and studied their results on the following tasks: for a 2 class problem (benign and malignant cancers) on the BreakHis database, for a 4 class problem (normal, benign, carcinoma in situ, and invasive carcinoma) and finally a 8 class problem (adenosis, fibroadenoma, phyllodes-tumor, tubular-adenoma, ductal carcinoma, lobular carcinoma, mucinous-carcinoma and papillary-carcinoma). It is important to note that in binary classification task and eight class task, the highest accuracy (second highest in the 2 class task) achieved was 99.87% by a pre-trained SENet-154 network after being applied data augmentation techniques, normalization and Adam's optimization method. The authors mentioned that accurate multi-classifications provided higher value information for breast cancer diagnosis, or prognosis, comparing to binary classifications. In the four class task, the highest accuracy of 97.08% was reached by a Inception-V3 model after employing the Adam's optimization function, data augmentation methods and the transfer color between images was utilized to normalize images. Not only that but all models' accuracy was improved by the employment of these methods.

Khened *et al.* [31] proposes an ensemble strategy where 4 different models were used as encoders and 4 Random Forest classifiers would vote, and in case of a tie, the higher metastases category was selected for that patch of the image. As first steps, the authors removed the background from the WSI using the ground truth image (regions labeled in the WSI) or threshold techniques. Patches were then removed randomly, using their dimensions as hyperparameters. The patches that were obtained underwent a process where they were subjected to a segmentation method pipeline. This pipeline generated a downsized version of the patch probability heatmap. The heatmaps underwent a binarization process by applying a threshold. After binarization, the connected components were identified, and various properties of these regions were measured. And finally to predict the metastases type, an ensemble of Random Forest classifiers was trained using the extracted features as demonstrated in 3.5.

Figure 3.5: Khened *et al.* proposed pipeline. [31]

Cruz-Roa *et al.* [15] performed a study to compare how two neural networks, with the same architecture, trained on different datasets would perform when classifying the same test dataset. The study analyzed the detection sensitivity of the method to the data used for training and found highly correlated performance measures between classifiers trained with different data cohorts. However, there were some differences in prediction performance, possibly due to "batch effects" originating from the ground truth annotation or slide digitization process. The results of the experiment show that the ConvNet is highly successful in identifying areas of invasive breast cancer in whole slide histopathology images, even when tested on cases from a different group of samples than the one used for training. Additionally, the ConvNet exhibited a high level of accuracy, with a low rate of false positives, and its performance improved as the number of training samples increased. However, the model faced difficulties in cases where invasive breast cancer coexisted with *in situ* disease. To overcome this challenge, training a more advanced network that includes examples of these precursor lesions could potentially enhance the model's performance in such cases.

The method utilized a convolutional neural network (CNN) as a patch-based classifier, using a learned feature approach rather than hand-crafted features. The CNN classifiers showed better performance compared to other visual features and machine learning methods tested.

Although the method achieved a high level of accuracy, there were limitations. It struggled

to detect rare subtypes of invasive breast cancer, such as mucinous carcinoma, and incorrectly classified some in situ breast cancer regions as invasive. The inclusion of these rare variants and additional training data could improve the model's performance and generalizability.

Batch effects, including stain variability and variations in the digitization process, were identified as sources of variation. Color normalization and expansion of the classifier into a multiclass predictor were suggested as potential approaches to address these issues.

In conclusion, the presented method offers an automatic and accurate solution for detecting invasive breast cancer on whole slide histopathology images, with potential applications in digital pathology and tumor grading, even when considering different datasets with different characteristics.

Macenko *et al.* [39] proposes two approaches to address the existing inconsistencies in the staining process. By employing these mechanisms, histology slides that were subjected to different processing or storage conditions can be effectively transformed into a standardized and normalized space. This normalization facilitates enhanced quantitative analysis by ensuring comparability and consistency across the slides. The two methods are:

- The stain vector identification method is an algorithm designed to identify stain vectors in an image. Stains on a slide can vary due to different staining procedures, manufacturers, and storage times. The goal of this method is to determine the specific stain vectors corresponding to the stains present in the image.

The algorithm assumes that the resulting color of each pixel, represented in optical density (OD) space, is a linear combination of these stain vectors. To identify the stain vectors, the algorithm follows a step-by-step process. Firstly, a thresholding method, where pixels with low OD values, indicating little or no stain, are thresholded to ensure stability in the analysis. This step helps in removing noise and unwanted variations. A fringe detection method is then applied, instead of searching for peaks, the algorithm focuses on finding the fringe of the pixel distribution. This approach is chosen to account for the presence of noise in the image. By finding the fringe, the algorithm aims to capture the minimum and maximum values along the direction of interest. Using geodesic path projection, the algorithm then focuses on finding the shortest path between two color vectors with unit-norm on a sphere, which is known as the geodesic path. By identifying this particular geodesic direction, the algorithm can effectively project the pixels transformed through optical density (OD) onto this path. This projection plays a crucial role in determining the endpoints that correspond to the stain vectors. With plane calculation, the algorithm then calculates the plane formed by the stain vectors. To do this, it utilizes the singular value decomposition (SVD) of the OD-transformed pixels. By considering the two largest singular values, the algorithm determines the plane that the stain vectors form. By then applying angle calculation, for each point in the plane, the algorithm calculates the angle with respect to the first SVD direction. This angle mapping helps in characterizing the directions in the plane and provides valuable information about the stain vectors. And finally, a histogram analysis is made, the algorithm

analyzes the histogram of the calculated angles. This analysis allows for the identification of the specific stain vectors based on their distribution characteristics. The algorithm aims to find stain vectors that form tight clusters in the histogram. The effectiveness of this stain vector identification method has been demonstrated through experiments on different slides with considerable variation. The automatically generated stain vectors exhibit tight clusters, although they may not perfectly coincide with manually chosen vectors. This method enables repeatable and automatic identification of stain vectors, which is crucial for consistent analysis in the field of digital pathology.

- The Confounding Factors Removal Method aims to address the factors that can degrade the signal in stained samples, thereby enabling direct analytical analysis of the samples. These confounding factors include the original stain strength, the staining procedure, fading over time, and the presence of the cellular substance of interest.

The method operates under the assumption that the quantity of protein or nucleic acid, which is the actual target for measurement, is a random variable that is scaled by these confounding factors. To facilitate the accurate analysis of the samples, the following steps are involved: Firstly, an intensity histogram calculation is made, for each specific stain of interest, the method calculates the intensity histograms for all the pixels that predominantly exhibit that stain. By focusing on the pixels dominated by a particular stain, the method captures the distribution of intensities associated with that stain. A robust approximation is then used. To provide a robust approximation of the maximum intensity associated with a stain, the method identifies the 99th percentile of the intensity values within the corresponding histogram. Experimental evidence supports the effectiveness of this percentile as a simple and reliable descriptor of the histogram. By analyzing multiple patches from each slide, the method verifies that the 99th percentile adequately represents the stain's maximum intensity. Finally, a histogram scaling method is applied, to enable meaningful comparisons between different stains, the intensity histograms are scaled to have a common pseudo-maximum. The pseudo-maximum is determined using the robust approximation obtained in the previous step. By scaling the histograms, they can be directly compared with each other, mitigating the impact of variations in staining procedures and other confounding factors.

The results of this method demonstrate that each stain exhibits its own distinct distribution, as expected based on the underlying theory. The method assumes that fading affects each stain equally, with the only difference being the amount of stain present. Ongoing investigations are being conducted to assess the impact of fading and to validate the assumption of equal fading effects. By effectively removing the confounding factors, this method facilitates direct analytical analysis of stained samples, enhancing the accuracy of measuring the cellular substance of interest.

The Confounding Factors Removal Method is an essential technique in digital pathology that focuses on eliminating factors that can degrade the signal in stained samples. By addressing these confounding factors, the method enables more accurate and direct analysis of

the target cellular substance, leading to improved results in quantitative measurements and analysis.

Gupta *et al.* [25] explores the effect and necessity of stain normalization in breast cancer histopathology. The authors propose a method that aims to standardize histopathological images by reducing staining variations that exist due to different staining protocols. It addresses the challenge of not having a predefined target template by automatically selecting a target image from the dataset.

The procedure begins by converting the images from RGB to HSV color-space and focuses on the chromatic information represented by the Hue (H) and Saturation (S) components. K-means clustering is then applied to group the images into clusters based on their chromatic characteristics.

The cluster with the highest number of images, indicating the most commonly occurring chromatic information, is selected. From this cluster, the image with mean H and S values closest to the centroid of the cluster is chosen as the target image. This target image represents the desired staining characteristics.

The procedure is applied separately to images of different magnifications, resulting in a target image for each magnification group. Once the target images are selected, a stain normalization method is used to align the color distributions of the entire dataset based on the target images.

Overall, the proposed procedure, automatically selects a representative target image for stain normalization by considering the most common chromatic information in the dataset. This approach helps reduce staining variations and enables more consistent analysis of histopathological images.

The authors point out that using joint color-texture features has the ability to obtain better results than other most common methods for breast cancer histopathology image classification. The authors compared cases with and without stain normalization and, although with such method the results are better, the difference is at the most 3% (using patient recognition rate as the evaluation metric). It is important to mention that the classifiers used in this paper were a Linear SVM (Support Vector Machine), a NN (Nearest Neighbors) classifier and a RF (Random Forest) classifier.

Ciampi *et al.* [14] also explores the need of normalization in a cancer image classification task. The authors used a ConvNet (Convolutional Network) with the architecture showed in ??, to evaluate the impact of stain normalization on the classification accuracy of histopathological images. Initially, the ConvNet is trained on a rectal cancer (RC) dataset and tested on a colorectal cancer (CRC) dataset without stain normalization, resulting in a low accuracy of 50.96%.

To investigate the role of staining in the classification task, a representative image from the RC dataset is selected as the template image. Two stain normalization algorithms, A proposed by Bejnordi *et al.* [21], and B, that was proposed by Macenko *et al.* [39] already mentioned above, are used to normalize patches in the CRC dataset based on the template image. The normalized patches are then classified using the same ConvNet. The results show a significant improvement in accuracy after stain normalization, with values of 75.55% for A and 73.99% for B.

However, it has been observed that B, one of the stain normalization techniques, fails to properly normalize certain patches, particularly those containing adipose tissue. Additionally, B tends

to force color normalization regardless of the specific tissue type. In contrast, technique A successfully preserves a significant portion of the red channel for blood cells while accurately normalizing the tumor patch.

Based on the positive outcomes from these findings, the experiment delves into the question of whether stain normalization should be applied exclusively to the test data or also to the training data. Technique A, which includes stain normalization, is integrated into both the training and testing processes, leading to four different configurations that are evaluated. The results demonstrate that incorporating stain normalization into the training phase enhances the accuracy by approximately 5% compared to the previous experiments. This improvement indicates that the ConvNet can benefit from learning the stain variability present in the training set.

In contrast, the final experiment, where stain normalization is not applied to either the training or test data, yields the lowest performance. This clearly demonstrates the importance of capturing and understanding the stain variability in the training set in order to effectively handle the expected variability in the test set.

Overall, this experiment highlights the effectiveness of stain normalization in significantly improving the accuracy of classifying histopathological images. It reveals that adjusting the test data to match the stain distribution of a reference image can greatly reduce variability and enhance the performance of the ConvNet.

Tellez *et al.* [66] explores the impact of stain augmentation compared to the use of stain normalization. Stain color augmentation, including HSV(Hue, Saturation, Value) and HED(Hematoxylin-Eosin-DAB) transformations, was found to be crucial for achieving top performance, regardless of the stain color normalization technique used. The inclusion of color augmentation, whether in HSV or HED space, played a key role in enhancing the convolutional neural network's (CNN) ability to generalize to unseen stain variations.

Among the color augmentations, strong and light variations showed similar performance, with a slight advantage towards light augmentation. It was observed that heavy augmentation could potentially reduce performance on images similar to those in the training set. Non-color augmentations, such as basic, morphology, and brightness-contrast (BC), were also explored, with BC augmentation yielding the best results, followed by morphology and basic augmentation.

The results of the experiments demonstrated the significant impact of stain color augmentation on classification performance, as it enhanced the ability of the CNN to handle variations in stains. Interestingly, the CNN showed a general insensitivity to the specific type and intensity of color augmentation, as long as some form of augmentation was applied. However, it was observed that CNNs trained with simpler stain color normalization techniques were more sensitive to the intensity of color augmentation, requiring stronger augmentation for optimal performance.

Relatively low performance was observed when using grayscale images, indicating that color information provided valuable insights to the model. Surprisingly, it was found that stain color normalization methods alone did not guarantee superior classification performance. In fact, the best-performing method did not involve any form of normalization, challenging the common belief that color normalization is a necessary step for achieving top classification performance in

histopathology.

Based on these findings, the authors concluded that stain color augmentation played a crucial role in reducing generalization errors caused by stain variation, regardless of the specific stain color normalization technique employed. The transformations of HSV and HED colors were identified as key factors for improving performance.

3.3 Summary

Due to the importance of classification of histopathology images to the diagnosis of a patient, several automated methods have been explored and employed in cancer classification. Comparing all methods analysed in this chapter, the use of Deep Learning seems to be more accurate and efficient in relation to other machine learning methods. The major objective of this section was the review of data augmentation, data normalization, selection of model architecture or important image alteration that could influence the outcome of classification. It is important to note that the computational resources available to each author was not disclosed and as such cannot be deducted how it could've impacted their decisions.

Chapter 4

Methodology

In this chapter, the methods used will be presented and explained, followed by the description of tests performed, chronologically, and results obtained. Finally a discussion involving all trade-offs made, considering the best and worst results obtained. It is important to mention that in the first section 4.1, the tests were only executed on the PCAM dataset.

4.1 Methodology applied on PCAM Dataset

In this initial phase, the PCAM dataset, detailed in a previous chapter 3.1.2, was one of two datasets used, for every train and test. Considering that pretrained models were also used, the ImageNet dataset is also present in this section 3.1.1.

4.1.1 Methods

4.1.1.1 VGG16

The first architecture chosen and experimented with in this thesis was a VGG16 based architecture, due to its primary application is image classification, where it has achieved high accuracy rates in various benchmark datasets. The architecture's design is straightforward and easy to understand, and as such a great choice for the initial part of this task. But is computationally expensive due to its depth and large number of parameters. The VGG16 architecture was proposed by Karen Simonyan and Andrew Zisserman in their 2014 paper titled "Very Deep Convolutional Networks for Large-Scale Image Recognition." [60]. At its core, VGG16 is composed of 13 convolutional layers and 3 fully connected layers. The convolutional layers are responsible for extracting features from input images. These layers apply convolutions using small receptive fields (3x3) and use the Rectified Linear Unit (ReLU) activation function to introduce non-linearity. The use of ReLU helps alleviate the vanishing gradient problem and promotes sparse activations. Between the convolutional layers, VGG16 incorporates max-pooling layers with 2x2 windows and a stride of 2. This downsamples the feature maps and reduces the spatial dimensions, enabling the network to learn hierarchical representations while achieving translation invariance.

Following the convolutional layers, VGG16 includes three fully connected layers, which capture high-level semantic information by integrating features from the previous layers. The fully connected layers are typically flattened to a 1-dimensional vector before being connected to subsequent layers.

One notable characteristic of VGG16 is its ability to extract meaningful and discriminative features from images. The earlier layers capture low-level features such as edges, corners, and textures, while the deeper layers learn more complex and abstract features. By leveraging these features, VGG16 can effectively discriminate between different objects and classes in image classification tasks.

It's worth mentioning that VGG16 has a large number of model parameters, approximately 138 million, due to its depth and extensive use of convolutional filters. This makes training VGG16 computationally expensive and needs a substantial amount of labeled training data to prevent overfitting.

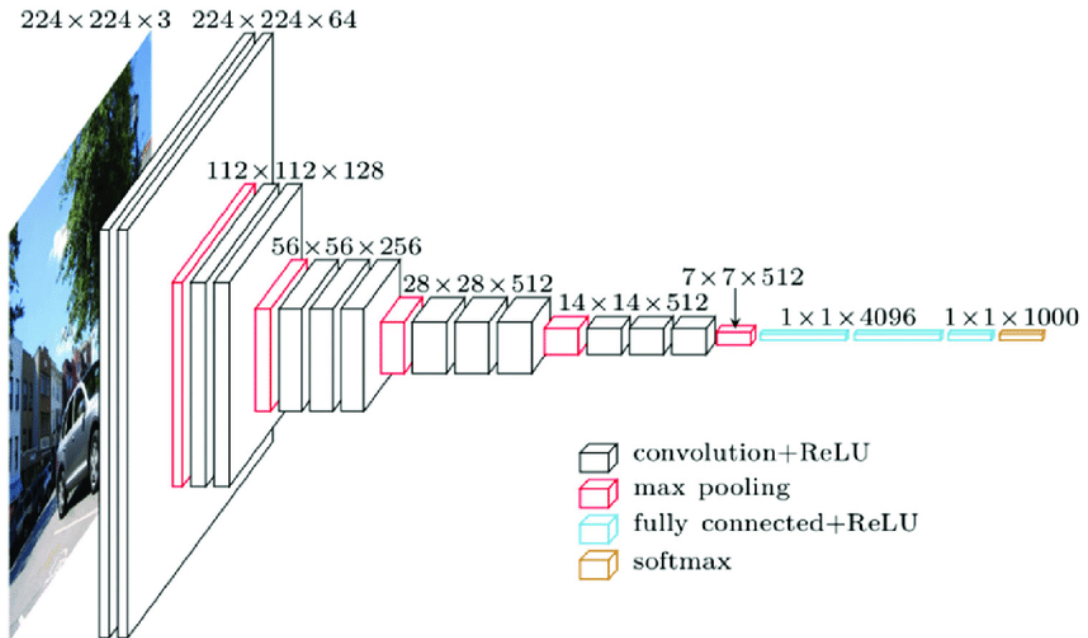


Figure 4.1: VGG Architecture [42]

The output layers of the VGG16 architecture are responsible for producing the final predictions or classifications. After the last fully connected layers, VGG16 typically employs a softmax activation function, which converts the outputs of the network into probability distributions over the different classes.

The number of output units in the final layer corresponds to the number of target classes in the specific classification task. For example, in the original VGG16 model trained on the ImageNet dataset, there are 1000 output units representing the 1000 object categories in ImageNet, as can be seen in figure 4.1. But for the PCAM dataset task, the last output layer is changed in order to accommodate the number of classes in the problem, since this is a binary classification task, the number of classes is 2, instead of the 1000 represented here.

4.1.1.2 VGG19

While VGG16 served as the initial architecture choice in this thesis, it is essential to explore the VGG19 model to further enhance the understanding of deep convolutional neural networks for this image classification task. VGG19, also proposed by Karen Simonyan and Andrew Zisserman in their 2014 paper [60], is an extension of VGG16 and shares similar design principles.

The architecture of VGG19 builds upon the foundation of VGG16 by incorporating additional convolutional layers, resulting in a deeper network. Specifically, VGG19 consists of 16 convolutional layers and 3 fully connected layers, offering a more complex feature extraction capability.

Like VGG16, VGG19 employs 3x3 convolutional filters throughout its convolutional layers, accompanied by the Rectified Linear Unit (ReLU) activation function. The cascaded convolutional layers enable the network to capture and learn intricate patterns and structures from the input images, ranging from low-level features to high-level semantic representations.

To downsample the feature maps and introduce translation invariance, VGG19 includes max-pooling layers with 2x2 windows and a stride of 2. This spatial dimension reduction aids in hierarchical feature learning while maintaining the network's ability to generalize across different image variations.

The fully connected layers of VGG19, similar to VGG16, capture high-level semantic information by integrating features from the preceding layers. These layers are typically flattened into a 1-dimensional vector before being connected to subsequent layers, allowing for the extraction of meaningful representations that contribute to accurate image classification.

As with VGG16, VGG19 boasts an impressive capacity to discriminate between objects and classes in image classification tasks. The additional layers in VGG19 further enhance its ability to capture complex and abstract features, providing an even more robust representation learning capability.

While VGG19 offers improved depth and feature extraction, it comes at a computational cost. With approximately 143 million parameters, training VGG19 requires substantial computational resources and a sufficient amount of labeled training data to prevent overfitting.

As well as is the case with the VGG16 model, the last output layer is also changed in order to accommodate the number of classes in the problem, since this is a binary classification task, the number of classes is 2.

4.1.1.3 ResNet50

In addition to VGG16 and VGG19, another prominent architecture that has significantly impacted the field of deep learning is ResNet50. ResNet50, short for Residual Network with 50 layers, was introduced by Kaiming He *et al* [27], in their 2015 paper titled "Deep Residual Learning for Image Recognition."

ResNet50 is designed to address the challenge of training very deep neural networks by utilizing residual connections. These connections allow information to bypass certain layers, facilitating

the flow of gradients during backpropagation and alleviating the degradation problem encountered in extremely deep networks.

The core idea behind ResNet50 is the introduction of residual blocks, which consist of skip connections that bypass one or more convolutional layers. By using identity mappings, ResNet50 enables the network to learn residual functions that capture the difference between the input and output of a block. This residual learning facilitates the training of deep networks and enables the extraction of highly discriminative features.

The architecture of ResNet50 comprises 50 layers, consisting of convolutional layers, residual blocks, and fully connected layers. The initial layers of ResNet50 perform convolution operations with small receptive fields, similar to VGG16 and VGG19, to extract low-level features from input images. These features are progressively refined and enriched as they pass through the residual blocks.

The skip connections in ResNet50 allow gradients to flow directly across multiple layers, enabling the network to effectively learn both shallow and deep features. This unique design contributes to the impressive performance of ResNet50 in various image recognition tasks.

One notable characteristic of ResNet50 is its ability to mitigate the vanishing gradient problem commonly encountered in very deep networks. By propagating gradients through shortcut connections, ResNet50 facilitates the training of networks with increased depth, ultimately leading to improved accuracy and performance.

With approximately 25 million trainable parameters, ResNet50 strikes a balance between model complexity and computational efficiency. It achieves state-of-the-art results on a range of image classification benchmarks, demonstrating its effectiveness in capturing intricate visual patterns and discriminating between different object classes.

The output layer of this model is also changed to correctly output the number of classes in the task at hand, which is 2.

4.1.1.4 GoogleNet

GoogleNet, also known as Inception-v1, is a notable deep convolutional neural network architecture introduced by Christian Szegedy et al. in their 2014 paper titled "Going Deeper with Convolutions." [65] GoogleNet was developed to address the challenges of depth and computational efficiency in deep neural networks.

The key innovation in GoogleNet is the inception module, which enables the network to capture features at multiple scales and abstraction levels without significantly increasing the computational cost. The inception module consists of parallel convolutional layers of different sizes, allowing the network to extract information at various receptive field sizes simultaneously.

The core idea behind the inception module is the utilization of 1x1, 3x3, and 5x5 convolutions within the same module. The 1x1 convolutions are responsible for dimensionality reduction, reducing the number of input channels and computational complexity. This facilitates the subsequent 3x3 and 5x5 convolutions, which capture spatial patterns and provide more expressive power to the network.

GoogLeNet also incorporates the use of max-pooling layers and auxiliary classifiers to enhance training and improve gradient flow. The auxiliary classifiers, positioned at intermediate depths of the network, introduce additional supervision during training and help alleviate the vanishing gradient problem. This architectural design enables GoogLeNet to achieve high accuracy while maintaining computational efficiency.

With 22 layers, GoogLeNet strikes a balance between depth and performance. It was the first architecture to demonstrate the effectiveness of the inception module and the benefits of parallel convolutions in deep networks.

One notable characteristic of GoogLeNet is its significantly reduced number of parameters, approximately 6.8 million trainable parameters, compared to the other architectures presented, VGG16, VGG19 or ResNet50. This reduction is achieved through the utilization of 1x1 convolutions, which reduce the computational burden while maintaining expressive power.

GoogLeNet achieved state-of-the-art results in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2014, outperforming previous architectures with a significantly lower computational cost. Its success showcased the potential of parallel convolutions and inspired further research on efficient and scalable deep network architectures.

As well as the other models used, the last output layer is also changed to correctly output the task number of classes, which is 2.

4.1.1.5 Optimizer

An optimizer, in the context of neural networks, refers to an algorithm that adjusts the model's parameters during the training phase to minimize the loss function and improve the model's performance. The primary objective of an optimizer is to find the optimal set of weights that lead to the most accurate predictions. This process involves iteratively updating the weights based on the gradients calculated through the backpropagation algorithm. Taking the task in this thesis into account, since the two datasets are for histology image classification, the chosen optimizers were AdamW and SGD.

Stochastic Gradient Descent (SGD) is a classic and widely used optimizer in the field of deep learning. It updates the model's weights by taking small steps in the direction of the negative gradient of the loss function. SGD performs this update for each training sample or a batch of samples, making it computationally efficient. It is a simple yet effective optimizer that can converge to a good solution, especially in large-scale datasets. However, SGD has some limitations, such as being sensitive to the learning rate choice and potentially getting stuck in local minima [11] [74].

AdamW, short for Adaptive Moment Estimation with Weight Decay, is an optimizer that combines the benefits of both AdaGrad and RMSprop. It maintains separate adaptive learning rates for each parameter and also incorporates weight decay, which helps prevent overfitting. AdamW utilizes momentum-based techniques to estimate the first and second moments of the gradients and adjusts the learning rates accordingly. This optimizer has gained popularity due to its ability

to handle sparse gradients, deal with noisy data, and converge faster than traditional optimization algorithms[32] [38] and also provide better results when the task is a binary classification task, although SGD can also be a good option, ADAMW is commonly a better and more efficient option[37].

4.1.1.6 Loss function

In the case of the loss function, for this type of task, only 2 loss functions were explored, the Cross-Entropy Loss function and the Binary Cross-Entropy Loss function.

The Binary Cross-Entropy Loss is specifically designed for binary classification tasks, where there are only two classes. It measures the dissimilarity between the predicted probabilities and the true labels of the positive class. The loss is computed using the formula:

$$-L(y, p) = y * \log(p) + (1 - y) * \log(1 - p)$$

Here, "y" represents the true labels (0 or 1), and "p" represents the predicted probabilities for the positive class. The loss is minimized when the predicted probabilities align well with the true labels.

The Cross-Entropy Loss, also known as Categorical Cross-Entropy Loss, is commonly used for multi-class classification tasks. It extends the concept of Binary Cross-Entropy Loss to handle scenarios with more than two classes. The loss measures the dissimilarity between the predicted class probabilities and the true class labels. It calculates the loss using the formula:

$$-L(y, p) = -\sum(y * \log(p))$$

Here, "y" represents the one-hot encoded true labels, and "p" represents the predicted class probabilities. The loss is minimized when the predicted probabilities match the true labels across all classes.

4.1.1.7 Normalization

For normalization, three different mean and standard deviation were calculated and used.

As the function from Pytorch library was used [47], it involves subtracting the mean and dividing by the standard deviation of the data. In this case, the value of the three channels in each pixel of the image(RGB).

The value in each of the three channels, in every pixel, in every image of the dataset was added and divided by the total of pixels added from all images.

$$\text{number of images} * 96 * 96 * R$$

Being R, the value for the Red channel. Same was done for Green and Blue channel. The same process was followed to calculate the standard deviation.

Considering this, the values for mean and standard deviation vary depending on which images they are being calculated on. In order to test the effects on the use of normalization, three cases were studied: one with the whole dataset being considered for the calculation of the mean and std(standard deviation), another were only the images from the validation set were used, and the final one, where the values from the ImageNet dataset, which are commonly used in cases of normalization, were used.

The values for the RGB channels were required to be in a positive interval due to the data augmentation function explained bellow. The method of varying the hue of the image can only be possible if the pixel values of the input image are non-negative, due to the function used available in the Pytorch library [47].

4.1.1.8 Data augmentation

The PCAM dataset is quite large and balanced when considering the examples of both classes present. As such, data augmentation methods were applied to the training dataset in order to reduce overfitting and increase the model's generalization ability. As mentioned before, the process of providing a WSI from a biopsy can have some different image details depending on the technician performing this process. Because of this, the methods used were a tentative to generalize the training set of images considering all possible types of WSI at the end of the process. Some of the methods are common in image classification tasks such as: horizontal and vertical flip, and rotation, from 90 to 270 degrees, their possible outcomes can be seen in figure 4.2.

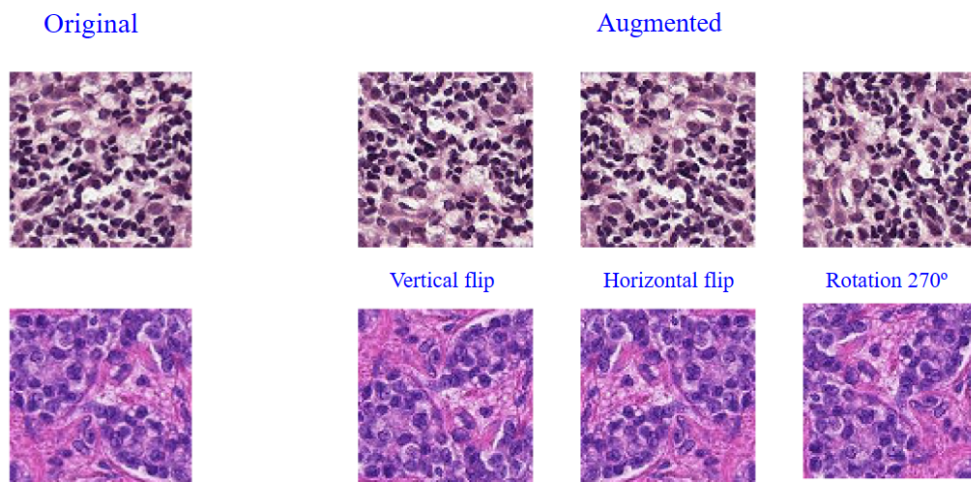


Figure 4.2: Horizontal/vertical flip and rotation

A gaussian blur filter was also applied with a 5x5 kernel size and a sigma value between 0.5 and 1.5 which can be seen in figure 4.3. This was explored in order to try and reduce noise and background features, highlighting details that help classification accuracy [20].

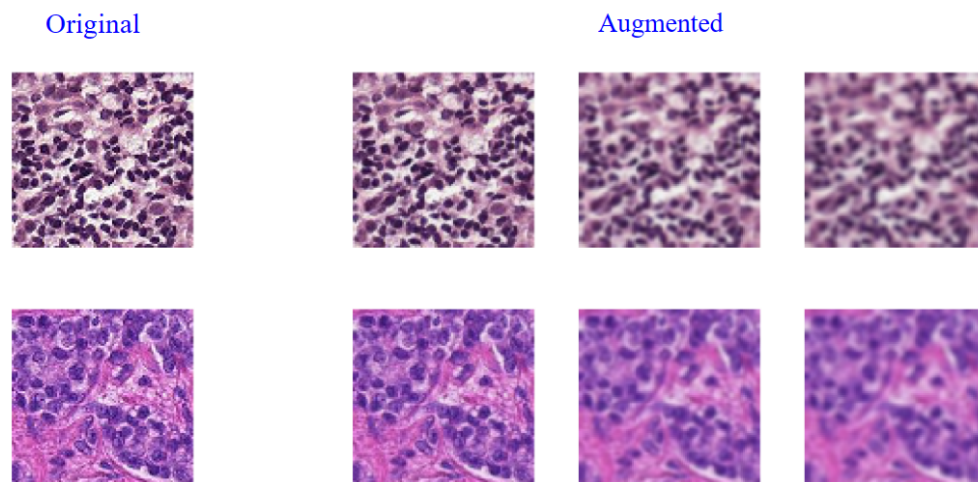


Figure 4.3: Examples of application of different intensities of Gaussian blur.

Grayscale methods were also experimented but due to the fact that histopathology images often contain important diagnostic and discriminating details that are encoded in the color variations, such as the presence of specific cellular or tissue components that are identified through distinct staining techniques, as seeing as grayscale methods remove this information as can be seen in figure 4.4, this line of thinking was quickly abandoned. .

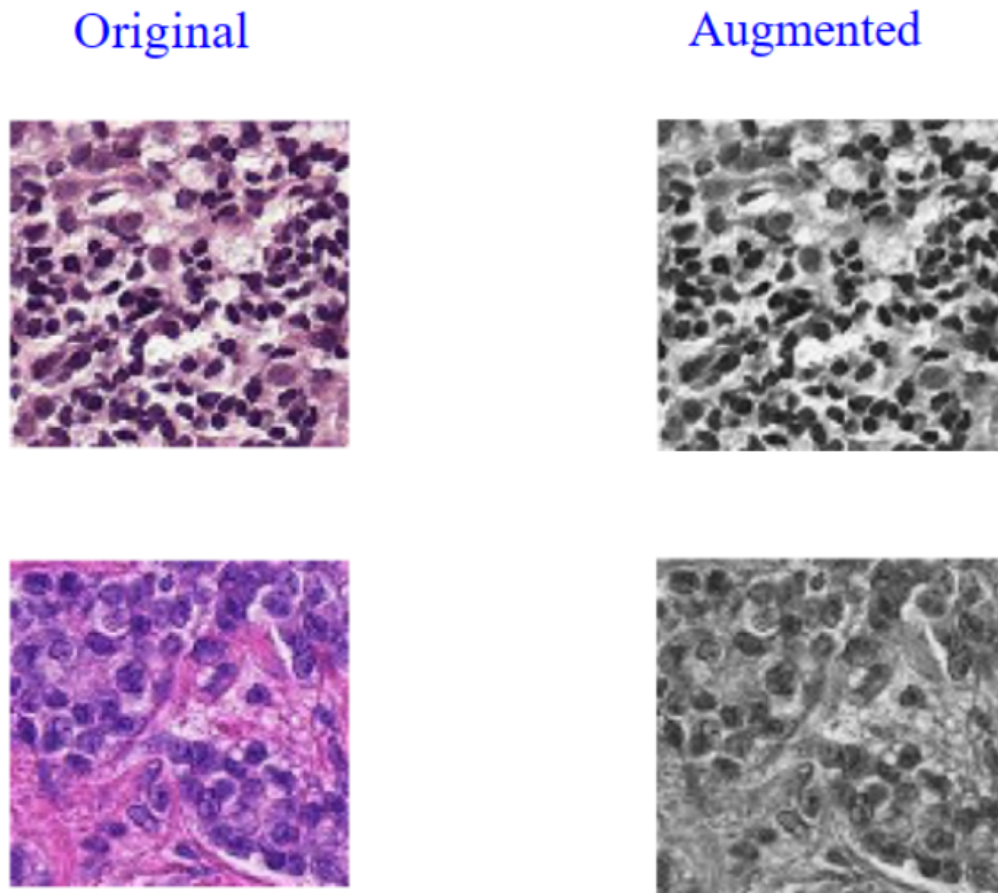


Figure 4.4: Example outcomes of applying grayscale method.

The most complex methods implemented were the color augmentation methods which altered the hue, saturation, contrast and brightness values of the images. As mentioned in [2](#), a WSI can have different color values and image properties due to several factors:

- Staining inconsistencies can occur across different regions within a WSI due to variations in the staining process or tissue properties. This can lead to variations in hue, saturation, and brightness across the WSI.
- Slide Preparation: Similar to individual samples, the preparation steps for WSIs involve tissue processing and slide creation. Variations in the tissue preparation and staining process can result in differences in color and intensity across different areas of the WSI.
- WSIs are typically captured using slide scanning systems that digitize the entire slide at high resolution. Factors such as the choice of scanner, image acquisition settings, and lighting conditions during scanning can affect the color representation and intensity levels across the WSI.

- WSIs are often large in size and require compression to store and transmit efficiently. Compression algorithms can introduce artifacts and affect the visual properties of the image, including hue, saturation, contrast, and brightness..

As such the idea was to implement color augmentation methods that could mimic the possible different types of WSI that could be present on a histology dataset, like the examples in 4.5.

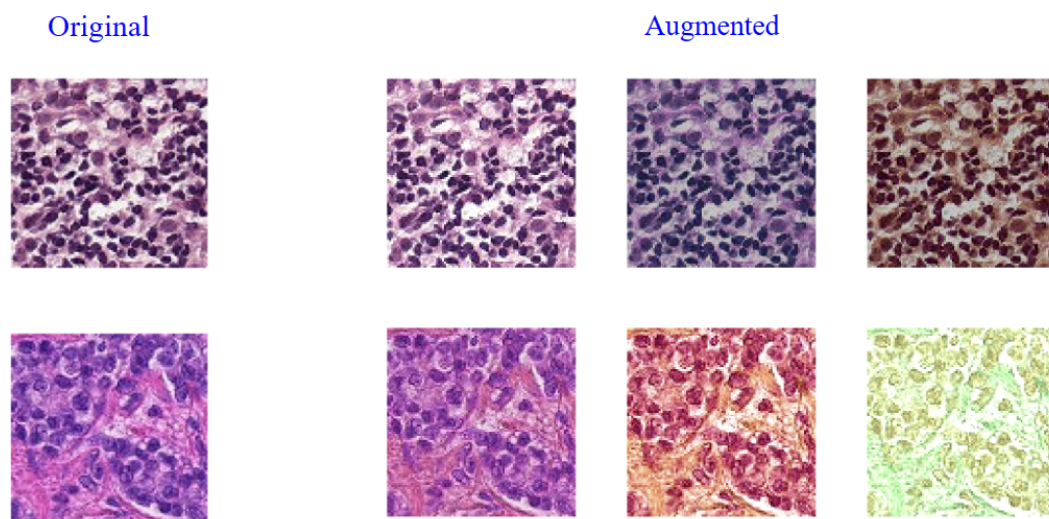


Figure 4.5: Variations in saturation, hue, brightness and contrast

And finally, the resizing of the images of the dataset. This was largely due to the new models that were going to be used, specifically the VGG19 that requires an input size image of 224px by 224px, but also due to a tentative to better identify patches where there was small number of pixels of tumor tissue. The resizing was done with the aid of the PyTorch library, and to be able to calculate the value of new pixels, bilinear interpolation was used.

Bilinear interpolation Bilinear interpolation is a popular technique used in image processing to resize images. It offers a way to estimate pixel values at non-grid locations by considering the neighboring pixels. The name "bilinear" arises from the fact that it involves a linear combination of values from two dimensions.

When resizing an image using bilinear interpolation, each new pixel's value is determined by taking into account the surrounding pixels in the original image. Specifically, the algorithm identifies the four nearest neighboring pixels to the desired location in the input image and calculates their contributions based on their distances. These contributions are then combined using weighted averages to determine the new pixel value [43].

One of the primary advantages of bilinear interpolation is its ability to produce smoother results compared to other interpolation methods. By taking into account the intensity values of nearby pixels, it reduces the presence of jagged edges or blocky artifacts that can be introduced during image resizing. This can lead to visually appealing images with improved visual quality.

In the context of image classification models, bilinear interpolation can be particularly beneficial when working with low-resolution images. Low-resolution images often lack fine details and can pose challenges for accurate classification. By resizing these images using bilinear interpolation, more information is introduced into the image, resulting in a higher-resolution representation. The additional details captured through the interpolation process can provide valuable cues for the classification model, enhancing its ability to discern important features and make more accurate predictions [22].

Several studies have demonstrated the effectiveness of bilinear interpolation in improving the classification performance of models trained on low-resolution images. By applying bilinear interpolation during preprocessing, researchers have observed enhanced feature extraction and improved discrimination between classes, leading to higher classification accuracy [73].

4.1.1.9 Pretrained vs scratch

In regards to the option of using a VGG16 pretrained model or using VGG16 from scratch model incurred some differences in the best situation they are to be applied and their expected performance [60].

Model Initialization: A "from scratch" VGG16 model is initialized with random weights for all its layers. In contrast, a pretrained VGG16 model has weights that are initialized using a large-scale dataset (such as ImageNet) during pretraining. The pretrained model's weights already capture meaningful features learned from the training dataset.

Training Process: A "from scratch" VGG16 model requires training from the beginning on the specific task or dataset of interest. It typically involves training the model on a large labeled dataset, adjusting the weights through backpropagation and gradient descent. On the other hand, a pretrained VGG16 model has already undergone extensive training on a large dataset, so it can be fine-tuned or used as a feature extractor for a new task. This process is often referred to as transfer learning.

Training Time and Resource Requirements: Training a VGG16 model from scratch requires significant computational resources and time. It involves training the model for multiple epochs on a large dataset to achieve good performance. In contrast, using a pretrained VGG16 model reduces training time and computational requirements since the initial weights are already tuned for feature extraction.

Performance and Generalization: A pretrained VGG16 model, having been trained on a large and diverse dataset like ImageNet, has learned general features that are applicable to various computer vision tasks. It can leverage this general knowledge to achieve good performance on new datasets with limited labeled samples. On the other hand, a "from scratch" VGG16 model's performance heavily depends on the quality and size of the dataset it is trained on.

Dataset Size and Task Complexity: If the task at hand involves a small dataset or a complex task that requires a deeper understanding of the data, a "from scratch" VGG16 model may be preferred. This allows the model to learn task-specific features from scratch without relying on pretrained weights that might not be as relevant.

Considering this, a VGG16 from scratch model was chosen for this initial tests as a tentative to train the model to better extract features from the PCAM dataset, without the pretrained weights from the ImageNet dataset. But in later tests both types of model were used and compared in 4.1.2.

4.1.1.10 Metrics

Due to the objective of this task being able to compare final results with similar articles, since the most common test metric used are precision, recall, AUC, accuracy and confusion matrix, these were the ones applied to all testing phases in this thesis.

Confusion matrix A confusion matrix is a table that visualizes the performance of a classification model by displaying the counts of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions. It provides a detailed breakdown of the model's performance for each class, allowing the calculation of various metrics like recall, precision, and accuracy. The confusion matrix helps identify any imbalances or patterns in the model's predictions, such as misclassifications or biases towards certain classes.

Recall Recall measures the percentage of actual positive instances that are correctly identified as positive by a classification model. It is calculated as the ratio of true positives (TP) to the sum of true positives and false negatives (FN):

$$Recall = TP / (TP + FN)$$

A high recall indicates that the model is effectively identifying positive instances, minimizing false negatives. Considering the task at hand where the an image is being classified as cancerous or not, it is highly important that this metric has high outputs, since the false negatives incur a higher negative impact than false positives.

Precision Precision measures the percentage of predicted positive instances that are correctly identified as positive by a classification model. It is calculated as the ratio of true positives (TP) to the sum of true positives and false positives (FP):

$$Precision = TP / (TP + FP)$$

A high precision indicates that the model has a low rate of false positives and is accurate in predicting positive instances.

Accuracy Accuracy measures the overall correctness of a model's predictions across all classes. It is calculated as the ratio of the number of correct predictions (true positives and true negatives) to the total number of predictions:

$$Accuracy = (TP + TN) / (TP + TN + FP + FN)$$

Accuracy is a widely used metric, especially when the classes are balanced. However, it can be misleading when dealing with imbalanced datasets.

AUC(Area Under the Curve) AUC is a metric used in binary classification tasks to evaluate the performance of a model's Receiver Operating Characteristic (ROC) curve. The ROC curve plots the true positive rate (TPR) against the false positive rate (FPR) at various classification thresholds. The AUC represents the overall performance of the model, with higher values indicating better discrimination between positive and negative instances. An AUC of 1 represents a perfect classifier, while an AUC of 0.5 indicates a random classifier.

4.1.1.11 Class Weights

After reviewing some of the results obtained, as will be shown in 4.1.2, and due to the fact of unbalance between FP(false positives) and FN(false negatives) obtained, weights were applied to the negative class of the binary classification task of the PCAM dataset, that in this case corresponds to a negative presence in cancerous cells.

By assigning a higher weight to the negative class, more emphasis is being placed on correctly classifying negative instances and minimizing false negatives, i.e. , the loss function gives more significance to the miss classification of instances from the weighted class. This weighting scheme encourages the model to be more cautious in predicting negative instances and helps reduce the occurrence of false negatives, but also may lead to an increase in false positives.

4.1.1.12 Ensemble Classifier

An ensemble classifier is a machine learning technique that combines multiple individual classifiers to make predictions or classifications. The idea behind ensemble classifiers is to leverage the diversity and collective decision-making capabilities of multiple models to improve overall performance and robustness.

In the context of this thesis, the model architectures that constitute the ensemble classifier are 3 of the ones presented above: the ResNet50, VGG19 and GoogleNet. Each of these models has its own strengths and weaknesses, and by combining them, the aim is to capitalize on their complementary abilities to enhance the accuracy and reliability of the predictions.

The ensemble classifier method involves leveraging the predictions of multiple individual models, which operate independently on the input data. In this particular case, a weighted averaging fusion rule was employed, assigning greater weight (0.4) to the model that exhibited superior standalone performance, while the other models were assigned weights of 0.3 each. Weighted averaging allows for a weighted combination of the individual model predictions, resulting in a final ensemble prediction. By giving more weight to the model with better performance, the ensemble benefits from its strengths and contributes to an enhanced overall prediction. This fusion process captures the collective decision-making of the ensemble, effectively integrating the insights from

multiple models into a single prediction. As will be explained bellow, the model with better results was the VGG19 and as such was always the model with the highest voting weight(0.4).

4.1.2 Results

At a first stage, tests were done to verify the impact of the chosen hyperparameter values, data augmentation methods and optimizer. This was only studied on a VGG16 model trained from scratch. Since the task at this first stage, where only the PCAM dataset is being used and as such is a binary classification problem, the BCE was used. When considering the MNIST dataset, the loss function was changed to CE due to it being a multi-class problem. Regarding the hyperparameters such as learning rate, number of epochs and batch size, the following was done:

- 3 different experiments were done using different learning rates of 0.0001, 0.0005 and 0.001, and the respective train losses, train accuracy, validation loss and validation accuracy graphs can be seen in figures 4.6, 4.7 and 4.8, respectively.
- The batch size was fixed on 128 due to the memory limitations. Since several data augmentation methods were to be applied to the dataset, including the chance of resizing, the value of 128 was chosen so that it would not need to be altered in a final stage. Instead, the learning rate was chosen accordingly with the knowledge of this fixed batch size value.
- The number of epochs was experimented, between 10 and 100 but after testing it was concluded that the last 50 epochs had little influence in the results in all tests, as such, the training was done with 50 epochs in all tests and any need to add epochs to the training stage was documented.

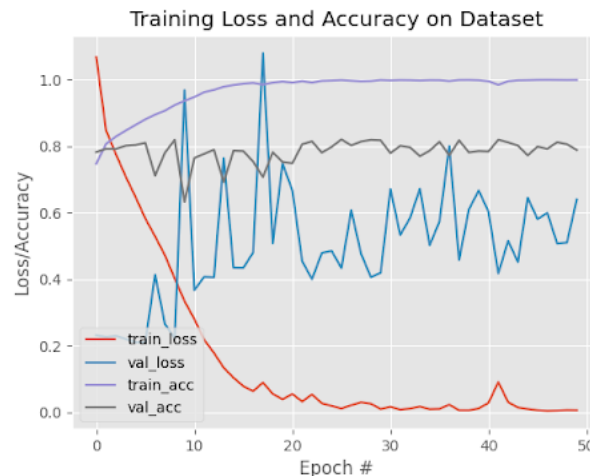


Figure 4.6: Learning rate of 0.0001, SGD, BCE and 128 batch size

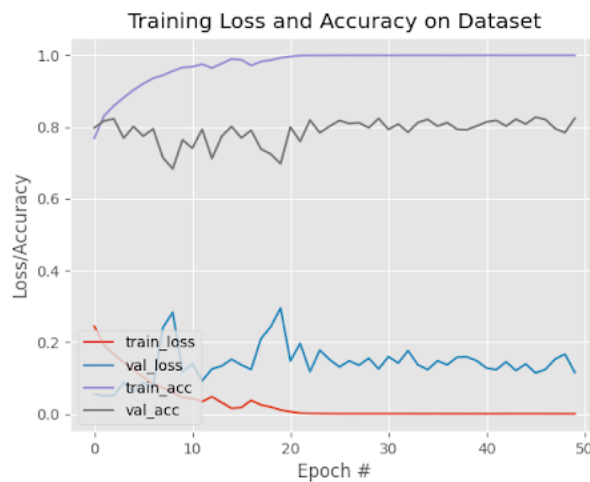


Figure 4.7: Learning rate of 0.0005, SGD, BCE and 128 batch size

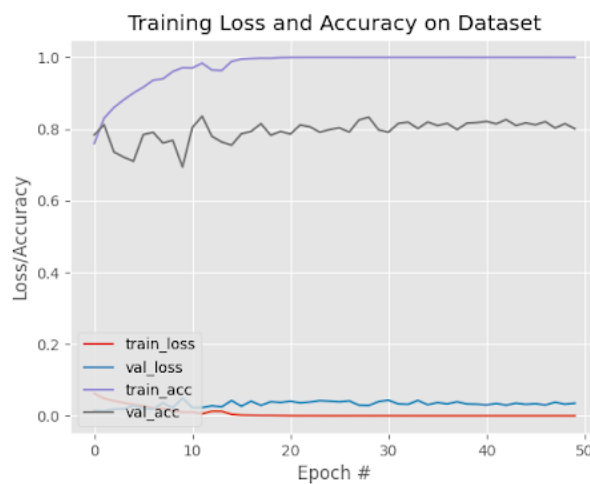


Figure 4.8: Learning rate of 0.001, SGD, BCE and 128 batch size

Table 4.1: Test results with learning rate variation on a "from scratch" VGG16 model

Test	Recall	Precision	AUC	Accuracy
Learning rate of 0.0001	0.669	0.878	0.883	0.792
Learning rate of 0.0005	0.689	0.865	0.886	0.791
Learning rate of 0.001	0.623	0.885	0.881	0.771

As can be seen in 4.1 all result metrics are similar but, after reviewing the behaviour of the loss curve, due to the fact that there is less oscillatory behaviour, the value chosen for the initial learning rate was 0.001 for subsequent tests. Since the learning rate was locked, more tests were done in order to learn the impact of several methods in the test result as can be seen in 4.2. In A.2 the graphic representation of the results is presented.

Table 4.2: Test results using . Lr(learning rate). DA(data augmentation). CA(color augmentation. b -brightness, c - contrast, s - saturation, h - hue), where values are varied between (min, max)

Label	Test	Recall	Precision	AUC	Accuracy
1	Lr(0.001), SGD, BCE	0.623	0.885	0.881	0.771
2	Lr(0.001), SGD, BCE + Normalization + DA(flip, rotation, gaussian(0.5 sigma))	0.585	0.939	0.882	0.773
3	Lr(0.001), AdamW, BCE + Normalization + DA(flip, gaussian(0.5 sigma))	0.649	0.961	0.888	0.811
4	Lr(0.0001), AdamW, BCE + Normalization + DA(flip, gaussian(1.0 sigma))	0.707	0.906	0.891	0.835
5	Lr(0.0001), AdamW, BCE + DA(flip, gaussian(0.5 sigma))	0.777	0.893	0.905	0.835
6	Lr(0.0001), AdamW, BCE + DA(flip, gaussian(1.5 sigma))	0.422	0.970	0.830	0.705
7	Lr(0.0001), AdamW, BCE + DA(flip, CA(b(0.1,0.8),c(0.5,2.0),s(0.5,2.0),h(-0.2,0.2)))	0.701	0.911	0.920	0.835
8	Lr(0.001), SGD, BCE + Grayscale	0.421	0.515	0.551	0.492

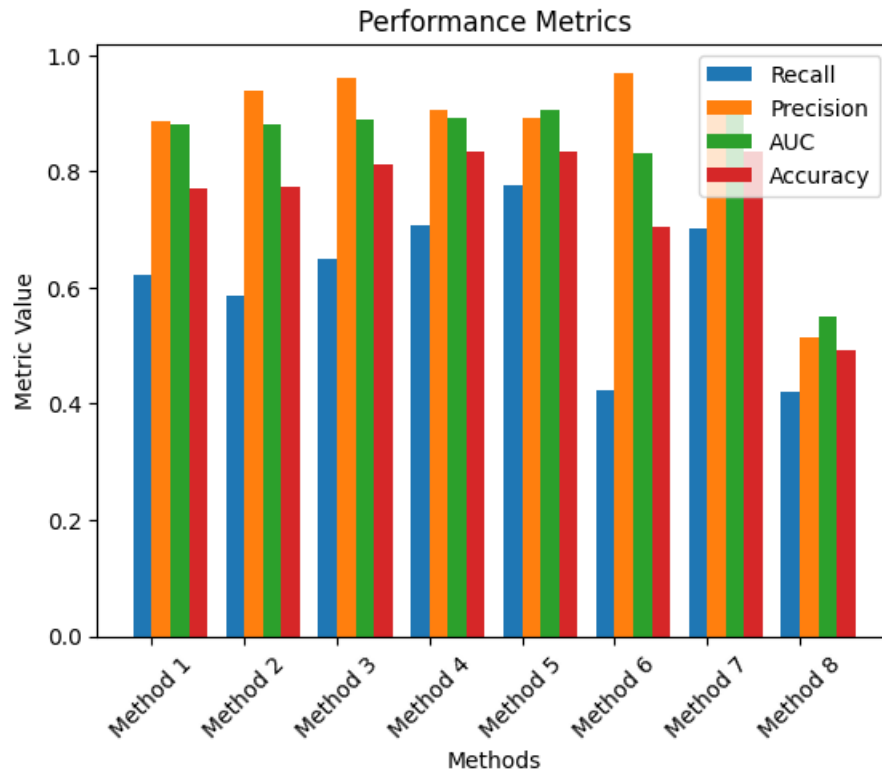


Figure 4.9: Presentation of results of table 4.2 graphically.

As it is possible to note, the best result was obtained with an AdamW optimizer, data augmentation, including color augmentation, with a learning rate of 0.0001 and without a gaussian filter. It is important to note that after exploring these results, one can come to some conclusions:

- The optimal learning rate varies with the optimizer used.
- Due to the low size of the images, a gaussian filter reduces the model's efficacy due to removing necessary details that were already few to start with (in some positive labeled cases).
- In this specific dataset, normalization did not have impact on the results.

Taking these results into account, some methods were dropped for subsequent tests due to their negative impact. Grayscale, gaussian blur, normalization and the SGD optimizer were removed. Color augmentation methods were implemented in 3 intensities, light (variations between -0.1 and 0.1), medium (variations between -0.3 and 0.3) and strong (variations between -0.5 and 0.5) values in brightness, saturation, hue and contrast. The selection of which intensity of color augmentation to apply to each image was random between the 3, this method was implemented for all subsequent tests. The VGG16 model architecture was dropped, and VGG19, and a ResNet50 and GoogleNet models were also trained and tested in order to be able to compare different model architectures in the final results. Not only that, but since there was a compelling idea to explore the if an ensemble classifier would achieve optimal evaluation metrics, three models with different architecture, that also achieved good results in similar tasks, as is shown in section 3, these three model architectures were chosen, the VGG19, the ResNet50 and the GoggleNet. Considering all methods that were going to be experimented and applied, in order to avoid having three times the tests to study the effect of the learning rate, and considering the usage of the AdamW optimizer with weight decay, a fixed learning rate of 0.001, which is also the default value for the the learning rate of the AdamW, was used for all models.

Due to the fact that all changes led to 22 subsequent tests, to better analyse the results obtained, only pretrained models versions of the three (ResNet50, GoogleNet and VGG19) were used to understand the impact of the applied methods and the "from scratch" models were only used to compare the metrics difference between pretrained and "from scratch".

In order to simplify the analysis of the results and the methods that were applied, the table 4.3 assigns the label to the experiment made, in order to better refer a specific test, and the table 4.4 presents the results obtained, according to their label. In the discussion section, the tests will be referred by their label.

Table 4.3: Test label. Label is assigned to what test was made. Ensemble(A+B+C) means an ensemble classifier made with the models from experiment A,B and C, giving the most weight(0.4) to the model that outperformed the others(in this case was VGG19 in all situations). Class weights are only used for training, in the case of the ensemble, class weights are N/A because that value is only used for training, and as such, it depends on the models used in the ensemble. Threshold is the value for which a class is classified true or false, a threshold of 0.5 means that if the output of a model is equal to or greater than 0.5, it is classified as belonging to the positive class, if the output is less than 0.5, it is classified as belonging to the negative class.

Label	Model	Image size	Threshold	Class weights
A	VGG16(scratch)	96x96	0.5	1 to 1
B	ResNet50(Pretrained)	96x96	0.5	1 to 1
C	ResNet50(Pretrained)	224x224	0.7	1 to 3.5
D	ResNet50(Scratch)	224x224	0.7	1 to 2
E	VGG19(Pretrained)	96x96	0.5	1 to 1
F	VGG19(Pretrained)	224x224	0.7	1 to 2
G	VGG19(Scratch)	224x224	0.5	1 to 2
H	GoogleNet(Pretrained)	96x96	0.5	1 to 1
I	GoogleNet(Pretrained)	96x96	0.5	1 to 2
J	GoogleNet(Pretrained)	96x96	0.5	1 to 3
K	GoogleNet(Pretrained)	96x96	0.5	1 to 5
L	GoogleNet(Pretrained)	224x224	0.5	1 to 1
M	GoogleNet(Pretrained)	224x224	0.7	1 to 1
N	GoogleNet(Scratch)	224x224	0.7	1 to 1
O	GoogleNet(Pretrained)	224x224	0.5	1 to 5
P	GoogleNet(Pretrained)	224x224	0.7	1 to 5
Q	Ensemble(B+E+H)	96x96	0.5	N/A
R	Ensemble(B+E+H)	96x96	0.9	N/A
S	Ensemble(B+E+H)	96x96	0.93	N/A
T	Ensemble(C+F+O)	224x224	0.5	N/A
U	Ensemble(C+F+O)	224x224	0.9	N/A
V	Ensemble(C+F+O)	224x224	0.7	N/A

Table 4.4: Test results. TN - True Negatives, FN - False Negatives, TP - True Positives, FP - False Positives.

Label	Recall	Precision	AUC	Accuracy	TN	FN	TP	FP
A	0.7391	0.9511	0.9350	0.8506	15769	4272	12105	622
B	0.7698	0.9472	0.9455	0.8635	15689	3769	12608	702
C	0.8625	0.9419	0.9637	0.9047	15521	2251	14126	870
D	0.9761	0.7028	0.9542	0.7818	9633	391	15986	6758
E	0.8362	0.9455	0.9582	0.8941	15603	2682	13695	788
F	0.9327	0.8857	0.9708	0.9062	14420	1101	15276	1971
G	0.8820	0.8668	0.9468	0.8733	14173	1931	14446	2218
H	0.7496	0.9422	0.9397	0.8518	15638	4100	12277	753
I	0.8259	0.8880	0.9313	0.8609	14686	2851	13526	1705
J	0.8503	0.8792	0.9412	0.8668	14478	2450	13927	1913
K	0.8589	0.8611	0.9340	0.8602	14122	2310	14067	2269
L	0.7889	0.9550	0.9506	0.8759	15783	3456	12921	608
M	0.8099	0.9442	0.9506	0.8811	15608	3112	13265	783
N	0.8083	0.9336	0.9536	0.8754	15450	3139	13238	941
O	0.8790	0.9204	0.9609	0.9015	15147	1981	14396	1244
P	0.9020	0.8931	0.9609	0.8971	14624	1604	14773	1767
Q	0.7944	0.9682	0.9644	0.8842	15965	3367	13010	426
R	0.8708	0.9392	0.9644	0.9072	15468	2115	14262	923
S	0.8846	0.9313	0.9644	0.9097	15323	1889	14488	1068
T	0.8979	0.9460	0.9752	0.9234	15553	1671	14706	838
U	0.9534	0.8766	0.9752	0.9096	15323	1889	14488	1068
V	0.9218	0.9268	0.9752	0.9245	15200	1280	15097	1191

4.1.3 Discussion

In this section, it is discussed the key findings and observations from the experiments on histology image classification dataset, PCAM. The goal was to investigate the impact of various methods on the performance of the models. From initial tests showed in 4.2, some methods were clearly impactful:

- Grayscale removes information that is crucial for a histology image classification task.
- Gaussian blur, although meant to mimic possible WSI's, removes information already crucial in a low resolution image, leading to worse results, the higher the intensity of the blur.
- Normalization did not have an impact in the PCAM dataset.

But as was seen in subsequent tests, among the different approaches explored, four main methods emerged as having a significant influence on the evaluation results: color augmentation, resizing, threshold selection, and ensemble classification.

4.1.3.1 Impacts of color augmentation

One aspect that was investigated was the impact of color augmentation techniques on the classification outcomes. By applying various color augmentation methods, such as brightness adjustments, contrast and saturation modifications, and random color jittering, the goal was to enhance the models' ability to generalize to different color distributions and improve their robustness to variations in staining techniques. As it is possible to understand from table 4.2, the value where AUC and accuracy is greater is where color augmentation techniques are applied, in experiment 7, although the values applied in an interval of $[0.5, 2]$ for brightness, saturation and contrast, and $[-0.2, 0.2]$ for hue, were larger than the variations applied in experiment A in table 4.4. Here the method of color augmentation was changed to light, medium and strong variations applied randomly, which obtained slightly better results. The findings indicated that color augmentation had a positive effect on the models' performance, as reflected in increased accuracy, precision, and recall scores in by comparing these 2 results. This suggests that color augmentation helps the models learn more discriminating features and adapt to variations in color appearance, contributing to improved classification outcomes.

4.1.3.2 Resizing

When considering the resizing of the 96x96 dataset images to 224x224, it is important to highlighting the impact of such method on the performance metrics, for example, in the transition from experiment H to L, and from K to O. In experiment H, the utilization of GoogleNet (Pre-trained) with an image size of 96x96 pixels resulted in a reasonable accuracy of 0.8518 and an AUC score of 0.9397. However, by increasing the image size to 224x224 pixels in experiment L, there was a noticeable improvement in the model's performance metrics. The accuracy increased to 0.8759, and the AUC score improved to 0.9506. This enhancement suggests that the larger image size allowed the model to capture more intricate features and details, resulting in improved discriminating power.

Similarly, comparing experiments K and O sheds light on the impact of resizing on model performance. Although in these two experiments a class weight of $[1, 2]$ was used, the final metrics are comparable. Experiment K utilized VGG19 (Pretrained) with an image size of 96x96 pixels, achieving a reasonable accuracy of 0.8602 and an AUC score of 0.9340. However, by increasing the image size to 224x224 pixels in experiment O, there was a noticeable improvement in performance metrics. The accuracy increased to 0.9015, and the AUC score improved to 0.9609. This improvement can be attributed to the larger image size, which allowed the model to capture more intricate details and enhance its ability to distinguish between classes.

These transitions from smaller to larger image sizes demonstrate the positive impact of resizing on the metrics of accuracy and AUC score. The increase in image resolution enabled the models to capture finer details and improve their discriminating power. It is important to note that the improvement in metrics comes at the cost of increased computational complexity, as larger images require more computational resources for training and inference, and because of this, a fixed batch size of 128 was chosen at the beginning of the task, as mentioned above. Another important note to consider is that the pretrained models used in this thesis were trained on the ImageNet dataset with samples of size 224pixels by 224 pixels, making this image resolution the optimal when testing. This improvement in metric results using pretrained models can be explained due to this fact.

4.1.3.3 Threshold Selection and Class Weights

When analysing initial test results, it became apparent that the number of False Negatives was substantially higher than the number of False Positives, as can be seen in experiments B, E and H. In order to mitigate this problem and at the same time try to increase metric values, threshold adjustment and class weights methods were experimented.

The experiments in the table provided valuable insights into the effectiveness of using class weights and to address the high false negative rates observed. Notably, experiments H and I, with class weights of 1 to 1 and 1 to 2 respectively, demonstrated improvements in key metrics. Experiment H, with balanced class weights and a smaller image size of 96x96, achieved a recall of 0.7496 and an accuracy of 0.8518. However, in experiment I, with the same image size but a modified class weight ratio of 1 to 2, the recall increased significantly to 0.8259, indicating a reduction in false negatives. Moreover, the accuracy also improved to 0.8609, showcasing the effectiveness of class weight adjustment in improving the model's ability to capture positive cases.

Actually if we consider experiments H, I , J and K, designed to test the influence of class weights, it is observable that the recall increases with the weight applied, being K the experiment with the most severe weights(1 to 5) but also has the highest recall of 0.8589, on the other hand, the number of number of True Negatives decreased and the number of False Positives increased, leading to the worst Precision value of the 4 experiments. The AUC and accuracy score in the experiment J were the highest with 0.9412 and 0.8668, respectively, with 1 to 3 class weights.

When considering threshold adjustment, the results from experiment L and M give some insight on its impact, the trade off between Recall and Precision occurs, this meaning that L, with a threshold of 0.5 has higher Precision but lower Recall. Due to this trade off between False Negatives and False Positives, the AUC score remains the same but the Accuracy score increases with a threshold of 0.7 on experiment M.

These results highlight the importance of considering class weights and threshold adjustment as strategies to mitigate false negatives and improve overall metric values. The combination of adjusting class weights to place higher emphasis on the positive class and fine-tuning the threshold allows for a more balanced trade-off between false negatives and false positives. By leveraging these techniques, the experiments demonstrated notable enhancements in recall and AUC(when

considering class weights), reflecting the improved ability to detect positive cases and reducing the risk of missing critical findings.

4.1.3.4 Ensemble Classification

Even-though threshold adjustment and class weights mitigated the low recall scores and difference between the number of False Negatives and False Positives, an ensemble classifier was experimented in order to combine the strengths of the three models to further decrease the gap between False Negatives and False Positives.

Ensemble classifiers were employed in experiments Q, R, S, T, U, and V, further exploring the potential of combining multiple models to enhance performance. These ensembles were constructed by combining the predictions of multiple base models, namely experiments B, E, and H in the case of experiments Q, R, and S, and experiments C, F, and O for experiments T, U, and V. Notably, certain ensemble experiments also incorporated other optimization techniques such as threshold adjustment, and class weights to optimize performance.

For instance, experiments Q, R, and S utilized the ensemble of models B, E, and H on 96x96 images with varying thresholds of 0.5, 0.9, and 0.93, respectively. Notably, class weights were dependant on the models used for the ensemble, since they were only applied in the training phase. The results demonstrated the effectiveness of ensemble classifiers, with experiment Q achieving a recall of 0.7944, precision of 0.9682, and an AUC of 0.9644. Similarly, experiments R and S exhibited high recall values of 0.8708 and 0.8846, precision values of 0.9392 and 0.9313, and AUC values of 0.9644, highlighting the improved performance achieved through ensemble methods.

On the other hand, experiments T, U, and V incorporated the ensemble of models C, F, and O on larger 224x224 images. These experiments employed a threshold of 0.5 for experiment T, 0.9 for experiment U, and 0.7 for experiment V. Class weights were not utilized in experiments T and U, while experiment V employed a class weight ratio of 1 to 5. The ensemble classifiers in these experiments demonstrated robust performance, with experiment T achieving a recall of 0.8979, precision of 0.9460, and an impressive AUC of 0.9752. Similarly, experiments U and V exhibited high recall values of 0.9534 and 0.9218, precision values of 0.8766 and 0.9268, and AUC values of 0.9752, showcasing the benefits of combining models and incorporating additional optimization techniques.

These ensemble experiments showcase the potential of combining multiple models to improve performance in detecting positive cases. The results demonstrate the efficacy of ensemble classifiers in achieving high recall, precision, and AUC values, indicating enhanced capability in capturing positive instances. Furthermore, the utilization of resize, threshold adjustment, and class weights in some of these experiments further highlights the benefits of incorporating multiple optimization strategies to maximize the overall performance of the ensemble classifiers.

4.2 MNIST Colorectal Cancer dataset

In this section, it will be explored what is the impact of methods applied in the previous dataset(PCAM), in the MNIST Colorectal Cancer dataset, more specifically, normalization and color augmentation, usage of "from scratch" models and transfer learning from the highest performing models.

4.2.0.1 Transfer Learning

A common approach in transfer learning is to unfreeze layers of the pre-trained model, in this case from experiments O, F and C, gradually during the fine-tuning process. This allows the model to adapt and learn task-specific features from the target dataset while still benefiting from the generic features learned by the pre-trained model. One strategy for unfreezing layers is to unfreeze an additional layer every few epochs, a similar approach to the one purposed by Sara Alves in [5].

During the initial stages of fine-tuning, only the last few layers of the pre-trained model are typically unfrozen while the remaining layers are kept frozen. This is because the early layers capture low-level features such as edges, textures, and basic shapes that are common across different image domains. These features are often generic and transferable, so it is beneficial to keep them fixed to prevent overfitting and to retain the pre-trained knowledge [55].

As the fine-tuning progresses, more layers can be gradually unfrozen. In this case, every 5 epochs, an additional layer will be unfrozen, allowing the model to learn more task-specific features. This incremental unfreezing approach helps the model to adapt to the target dataset progressively and avoid catastrophic forgetting, where previously learned information is overwritten by new information.

In addition to unfreezing layers, altering the learning rate during the fine-tuning process is another common practice. Initially, a 0.001 learning rate will be used to prevent drastic changes to the pre-trained weights and ensure a smooth transition between the pre-trained and fine-tuned layers. This helps in preserving the learned representations and preventing significant degradation in performance. As the training progresses, the learning rate will be increased by a factor of 1.1, also every 5 epochs, to facilitate faster convergence and fine-tuning of the unfrozen layers.

4.2.1 Results

In order to study the effect of normalization, color augmentation and image resizing, studied in the previous section 18 tests were made with the following characteristics and resulting in the metric values presented in 4.5:

- A learning rate of 0.001 was used for all models and an AdamW optimizer was chosen due to the fact that an optimal learning rate was not searched.
- Cross-entropy loss function was used for training of all models.
- The training consists of 25 epochs and the batch size is 128.

- The color augmentation method is the same as the last section, meaning random selection of light, medium and strong variations on contrast, brightness, saturation and hue, consisting of intervals of $[-0.1, 0.1]$, $[-0.3, 0.3]$, $[-0.5, 0.5]$ respectively.
- Normalization method was used considering the whole dataset, this meaning that the mean and standard deviation were calculated using the RGB values from the images from the whole dataset and then applied to the whole dataset.
- Transfer learning was only used on the last 3 tests due to the fact that the models were trained on PCAM after a 224x224 resize, and the impact of image resizing was not studied on other tests.

Table 4.5: Tests comparing the application of normalization, color augmentation and transfer learning

Test	AUC	Accuracy
VGG19(From Scratch)	0.7013	0.6959
ResNet50(From Scratch)	0.6905	0.6345
GoogLeNet(From Scratch)	0.7311	0.6248
VGG19(From Scratch) + Normalization	0.8212	0.8022
ResNet50(From Scratch) + Normalization	0.8331	0.7991
GoogLeNet(From Scratch) + Normalization	0.8152	0.8045
VGG19(From Scratch) + Normalization + Color Augmentation	0.8233	0.8174
ResNet50(From Scratch) + Normalization + Color Augmentation	0.8312	0.8077
GoogLeNet(From Scratch) + Normalization + Color Augmentation	0.8030	0.7954
VGG19(transfer learning) + Normalization + Resize(224x224)	0.9181	0.5575
ResNet50(transfer learning) + Normalization + Resize(224x224)	0.9056	0.5334
GoogLeNet(transfer learning) + Normalization + Resize(224x224)	0.9105	0.5876

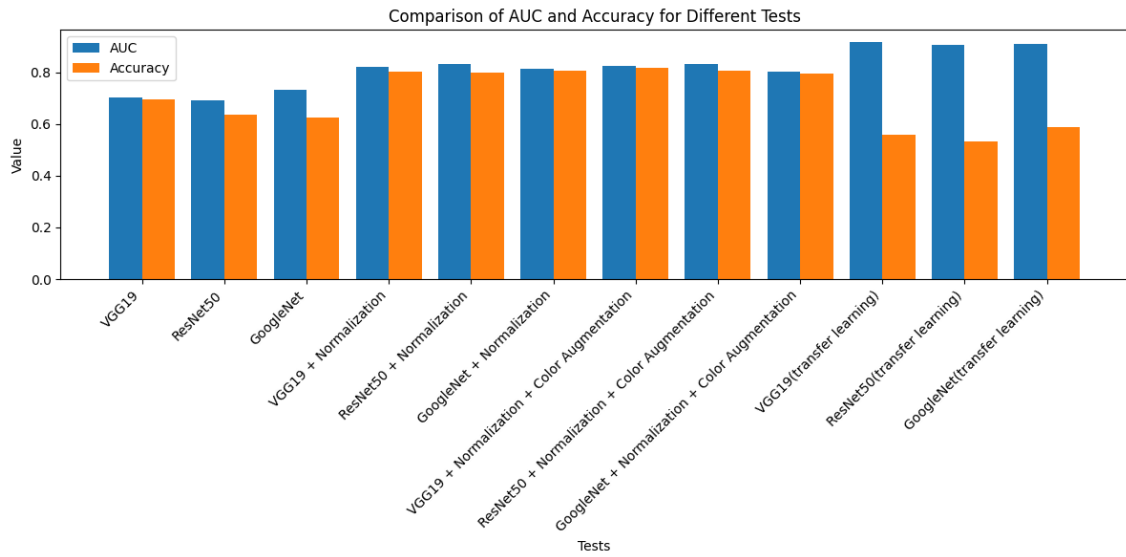


Figure 4.10: Graphical presentation of results of table 4.5.

4.2.2 Discussion

Based on the provided results in Table 4.5 and graphically presented in A.2, we can discuss the performance of different models and configurations, focusing on normalization, color augmentation, and transfer learning techniques applied to the MNIST colorectal dataset.

When the models (VGG19, ResNet50, GoogleNet) are trained from scratch without any additional techniques, we observe relatively low AUC and accuracy values. This outcome suggests that starting with randomly initialized weights alone may not be sufficient to effectively capture the complexities present in the dataset. Consequently, the models struggle to achieve high discriminating power and accuracy.

However, the introduction of normalization to the models trained from scratch yields noticeable improvements in performance. Normalization helps standardize the pixel values and mitigate the impact of variations in lighting conditions. As a result, the AUC and accuracy values for VGG19, ResNet50, and GoogleNet models trained from scratch with normalization demonstrate a significant increase compared to their respective counterparts without normalization.

Moreover, the incorporation of color augmentation further enhances the models' performance. By introducing variations in hue, brightness, saturation and contrast, color augmentation diversifies the training data and helps the models generalize better. The models trained from scratch with both normalization and color augmentation exhibit higher AUC and accuracy values, indicating improved classification performance.

Interestingly, when transfer learning is applied to the VGG19, ResNet50, and GoogleNet models in combination with normalization and resizing to 224x224 pixels, we observe a different pattern of results. In this case, the accuracy values show a significant drop compared to the models trained from scratch with normalization and color augmentation. This unexpected decrease in

performance suggests that the pre-trained model's features might not align well with the dataset's characteristics.

These results imply that while normalization alone improves the models' performance, the inclusion of color augmentation is crucial for achieving better generalization and classification accuracy. Furthermore, when using transfer learning, it is important to carefully consider the differences between the dataset where the models were originally trained, PCAM in this case, and the dataset where the transfer learning is being applied, this meaning the MNIST Colorectal dataset.

4.3 Summary

Considering PCAM dataset array of tests, several phases and objectives occurred in order to determine the best course of action to achieve the highest metric values. In an initial phase, the impact of hyperparameters, data augmentation techniques, optimizer and loss function choices were tested with a VGG16 model trained from scratch. The data augmentation technique that proved to have the most impact was color augmentation, considering that normalization was ineffective when the values for the RGB channels already were in a $[0,1]$ interval.

Taking into account the last phase of tests on the PCAM dataset 4.3 and 4.4, the influence of a smaller interval variations in color augmentation proved effective. The impact of image resizing was also tested and concluded that almost all metric values improved when the training and testing was done with higher image sizes. This can be explained due to the small resolution images that comprise the PCAM dataset lead to low number of features that are crucial for a positive label. Taking this into account, the identification of cancerous information in the image becomes harder with a image with 96x96 resolution. The resizing of the images to 224x224 mitigates this problem by using bilinear interpolation to fill new pixels with the existing one's information. This can lead to the increase of cancerous information in the image, and in return better metric values, in the case of a positive label.

The problem of the small image resolution of the dataset led to a high difference between False Negatives and False Positives, and Recall values, fortifying the idea that due to the low image size, it is harder to identify features that correctly lead to a positive label. As such, threshold adjustment, class weights and an ensemble classifier were experimented, yielding promising results. While the threshold adjustment keeps the AUC score the same, the value trade-off between recall and precision values always occurred. In some cases the accuracy increased, concluding that there is an optimal value for the threshold that increases accuracy values even-though the value for either recall or precision may be lower. Class weights behaviour is similar to threshold adjustments in the sense that there is an optimal value where the AUC score increases (instead of accuracy in threshold adjustment cases), while the recall and precision trade-off still occurs. The ensemble classifier methods yielded the best results of all tests, while combining the methods of class weights and threshold adjustments. It is important to note that, considering that histology image classification plays a vital role in predicting disease prognosis and assessing treatment response, the value of

a False Negative is much more prejudicial than that of a False Positive, hence the tentative to decrease the number of False Negatives without decreasing AUC or accuracy scores.

Comparing now the best value obtained from this test to results from other articles, tested on a similar version of the PCAM dataset 4.6. Although the experiment V from this thesis has the highest AUC score, the accuracy is lower than the one in Zhong *et al.* [76], this can be justified due to the version of the PCAM dataset used. Due to the PCAM's probabilistic sampling, the original PCam dataset contains duplicates which lower metric values. The articles compared here, and also explored in the literary review chapter 3, use a version of the PCAM dataset available on kaggle that does not contain duplicates [17], which can explain the higher accuracy values.

Table 4.6: Test comparison with similar articles Rane *et al.* [50] and Zhong *et al.* [76]

Label	AUC	Accuracy
Rane <i>et al.</i> [50]	0.9607	N/A
Zhong <i>et al.</i> [76]	0.971	0.989
Experiment V	0.9752	0.9245

When analysing the results of the tests done on the MNIST Colorectal Cancer dataset, it is possible to understand that, even-though normalization methods were ineffective on the PCAM dataset, they yielded solid results on the MNIST Colorectal Cancer, this due to the fact that this dataset wasn't previously normalized. Additionally, when applying color augmentation methods in conjunction with normalization, the AUC and Accuracy improve for all 3 models, which is the same conclusion from the PCAM datasets, even when considering it was already normalized.

When considering the application of transfer learning in this instance, and the high difference between AUC and accuracy values, a high AUC but low accuracy could indicate that the models are struggling with the complexity of the classification task. Even though the AUC is high, suggesting good separability between classes, the model may still struggle to make precise predictions. The low accuracy could be due to difficulties in correctly assigning samples to the correct class boundaries. Additionally, the number of samples per class being equal, in this case being 625 for each class, does not necessarily guarantee that the classes are equally challenging to classify. Some classes may have inherent complexities or variations that make them more difficult to distinguish accurately, even with a balanced dataset.

In summary, the results indicate that the inclusion of normalization and color augmentation techniques improves the performance of models trained from scratch on the MNIST colorectal dataset. However, transfer learning experiments resulted in decreased performance, suggesting that the classification tasks might be too different in order to apply transfer learning.

Chapter 5

Conclusions

The accurate classification of histology images plays a vital role in diagnosing and treating various diseases, including cancer. Histopathological analysis, which involves examining tissue samples under a microscope, is an important method in determining the presence and severity of malignant tumors. However, the manual interpretation of histology images can be time-consuming, subjective, and prone to human error. To address these challenges, the field of histology image classification has witnessed remarkable advancements with the integration of Artificial Intelligence (AI) methods.

The objective of this thesis was to explore and evaluate AI methods for histology image classification, aiming to develop models that can assist pathologists by providing more information to aid in making accurate and efficient diagnoses. By harnessing the power of deep learning and computer vision techniques, these models can learn intricate patterns and features present in histology images, enabling them to distinguish between different tissue types, identify abnormal cells, and ultimately aid in disease detection and prognosis.

For this task, several variables were studied in different stages. Initially, only the PCAM dataset was considered, due to its large size it was felt that high sample size could lead to better understanding of methods impact on evaluation metrics. Three different values for the learning rate were tested on a VGG16 model "from scratch" as a tentative to find the optimal learning rate and corresponding metrics. The results of the experiments with different learning rates showed similar performance metrics. However, after reviewing the loss curves, a learning rate of 0.001 was chosen for subsequent tests due to its less oscillatory behavior. Further tests were conducted to explore the impact of additional methods on the test results. The tests included variations in data augmentation (flip, rotation, and Gaussian augmentation), optimizer (SGD and AdamW), and image preprocessing (normalization and grayscale). The results of these tests were evaluated based on recall, precision, AUC, and accuracy.

Based on the results, the best performance was achieved with an AdamW optimizer, data augmentation (flip and Gaussian augmentation), and a learning rate of 0.0001 without a Gaussian filter. It was observed that the use of a Gaussian filter reduced the model's efficacy, potentially due to the removal of necessary details in some positively labeled cases, when taking into account the

low resolution images that consist the PCAM dataset. Normalization did not significantly impact the results, on the other hand, color augmentation methods to vary the contrast, hue, brightness and saturation proved effective, even with a considerable large interval. Due to the fact that stain normalization and color augmentation methods focus on color values present in dataset samples, a deeper exploration of normalization techniques is needed in order to understand why the application of these two methodology differ by a large margin when considering their impact on evaluation metrics.

Following tests were set to further prove the impact of color augmentation techniques, the interval was adjusted in a way that light, medium and strong variations were applied at random. At this stage, the major obstacle of the image classification task of the PCAM dataset was the image resolution that comprised this dataset. As the results prove, the ratio of False Negatives and False Positives was large, this can be explained by the fact that the models struggled to identify pixels with cancerous information, due to said image resolution. To try and mitigate this problem, 3 different model architectures, all pretrained on ImageNet, a VGG19, a GoogleNet and a ResNet50, due to their popularity in various computer vision tasks, including histology classification, were combined with 3 different methods: a resizing method using bilinear interpolation, class weights in model training, giving more weight to the positive class, and ensemble classification. All three methods proved having a positive impact on the metric values in all models. The principal observations were that class weights led to a specifically higher AUC and recall value, even-though there was a decrease in precision. Resizing allowed for improvement in accuracy and AUC, due to having more pixels with cancerous information that the model can identify, even-though new pixels are generated with information from it's neighbours, but due to the fact that the optimal image size for these pretrained models is 224x224(due to being trained on ImageNet with samples with 224x224 size), further exploration is needed in order to fully understand the impact of image resizing. Finally, an ensemble classifier was employed using the best performing models, giving the most weight to the VGG19, produced the best metric values of all tests, although there were cases where the recall value was higher. When comparing to other articles that also used the PCAM dataset, [50] and [76], it is important to mention that said articles used a version of the PCAM dataset that didn't contain duplicates, which can account for the higher accuracy values, even-though the AUC score is lower.

Finally, normalization, color augmentation and transfer learning methods were applied on the MNIST Colorectal Cancer dataset in order to observe if they had the same impact. The effect of normalization improved significantly the outcome of AUC and accuracy scores, while considering the whole dataset for the calculation of mean and standard deviation. Applying the same color augmentation method that was used on the PCAM dataset, proved to have the same positive impact on the MNIST Colorectal Cancer dataset's tests. On the other hand, when transfer learning was applied to the VGG19, ResNet50, and GoogleNet models in combination with normalization and resizing to 224x224 pixels(because they were trained after the resized PCAM dataset), there was an unexpected decrease in performance suggests that the pre-trained model's features might not align well with the dataset's characteristics and a further exploration of the hyper parameters

might be needed.

5.1 Future work

During the exploration of this thesis, there were a lot of obstacles that limited the amount of tests made and datasets explored. As seen before, when comparing the best result in the PCAM dataset with metric values of papers that used a different version of the PCAM dataset, it can be deduced that if the same methodology used in this thesis, was used on a PCAM dataset without duplicates, the accuracy value may increase. The usage of this dataset [17], proved to be an obstacle due to the computational costs that came with its management. As such, a possible future work would be measure the difference between the best performed methods, tested on this new version and on the original version of the PCAM dataset. Furthermore, due to the impact of image resizing in evaluation metrics, as future work, it would be interesting to explore different resizing methods to deal with low image resolution problems, such as super-resolution algorithms to enhance the resolution, patch-based approaches, multi-resolution training and progressive resizing. When considering the tests done on the MNIST Colorectal Cancer dataset, due to time limitations, some tests were not possible to be made, as such, as future work ,it would be interesting to see more complex comparisons between the methods implemented. Not only that, but a better exploration of hyper parameters when using transfer learning and their impact to the metric outcome would also be captivating.

Appendix A

Image Classification - Detailed Results

This Appendix contains additional graphics portraying the results obtained in the tests, in order to better compare the impact of the methods when considering the position of the evaluation results.

A.1 Tests done on PCAM

A.1.1 Graphical analysis of table 4.2

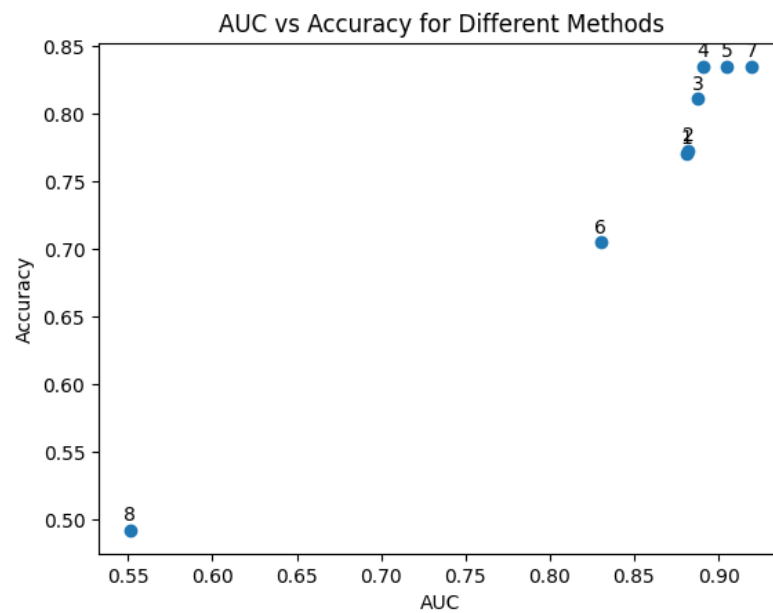


Figure A.1: Analysis of AUC and accuracy values on table 4.2, the label of the method is presented above the point of it's classification on the graph. Tests 1 and 2 are overlapping.

A.1.2 Graphical analysis of table 4.4

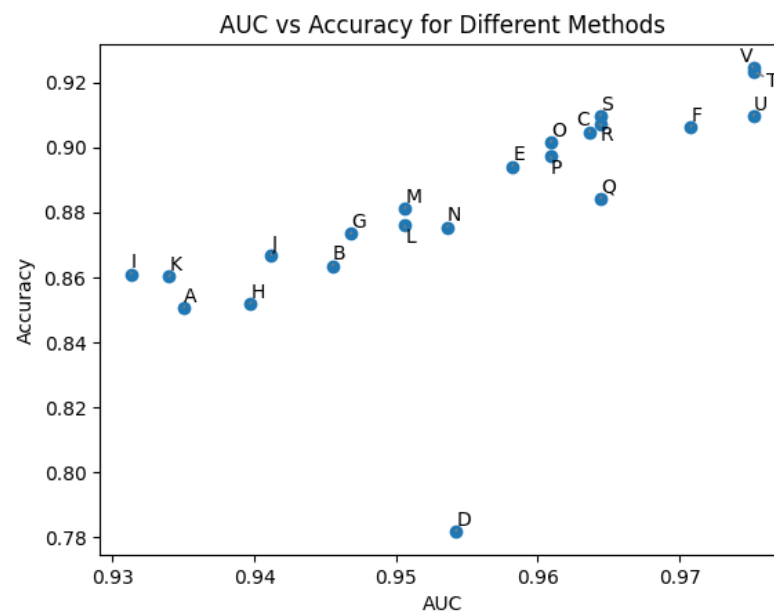


Figure A.2: Analysis of AUC and accuracy values on table 4.4, the label of the method is presented above the point of it's classification on the graph.

References

- [1] Datasets catalog. <https://www.tensorflow.org/datasets/catalog/imagenet2012.html>. Accessed: 30-04-2023.
- [2] Scikit-learn documentation: Preprocessing data. <https://scikit-learn.org/stable/modules/preprocessing.html>.
- [3] Steve A. Adeshina, Adeyinka P. Adedigba, Ahmed A. Adeniyi, and Abiodun M. Aibinu. Breast cancer histopathology image classification with deep convolutional neural networks. In *2018 14th International Conference on Electronics Computer and Computation (ICECCO)*, pages 206–212, 2018.
- [4] Jamil Ahmad, Khan Muhammad, and Sung Baik. Data augmentation-assisted deep learning of hand-drawn partially colored sketches for visual search. *PLOS ONE*, 12:e0183838, 08 2017.
- [5] Sara Carvalho Alves. Ai-based methods for cancer cells quantification using whole slide imaging. Master’s thesis, Faculdade de Engenharia da Universidade do Porto, October 2022.
- [6] Sarah M. Ayyad, Mohamed Shehata, Ahmed Shalaby, Mohamed Abou El-Ghar, Mohammed Ghazal, Moumen El-Melegy, Nahla B. Abdel-Hamid, Labib M. Labib, H. Arafat Ali, and Ayman El-Baz. Role of ai and histopathological images in detecting prostate cancer: A survey, 2021.
- [7] J. Winkens T. Cohen M. Welling B. S. Veeling, J. Linmans. Rotation equivariant cnns for digital pathology, September 2018.
- [8] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [9] Eric Bonnet. Using convolutional neural networks for the classification of breast cancer images. 08 2021.
- [10] Léon Bottou. Stochastic gradient descent tricks. In Grégoire Montavon, Geneviève B. Orr, and Klaus-Robert Müller, editors, *Neural Networks: Tricks of the Trade*, pages 421–436. Springer, Berlin, Heidelberg, 2012.
- [11] Léon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning, 2018.
- [12] Britannica, The Editors of Encyclopaedia. Antonie van leeuwenhoek. <https://www.britannica.com/biography/Antonie-van-Leeuwenhoek>. Encyclopedia Britannica, 23 Mar. 2023.

- [13] Emily Chen and Angela Zhao. Quantum neural networks.
- [14] Francesco Ciompi, Oscar Geessink, Babak Ehteshami Bejnordi, Gabriel Silva de Souza, Alexi Baidoshvili, Geert Litjens, Bram van Ginneken, Iris Nagtegaal, and Jeroen van der Laak. The importance of stain normalization in colorectal tissue classification with convolutional networks, 2017.
- [15] Angel Cruz-Roa, Hannah Gilmore, Ajay Basavanahally, and et al. Accurate and reproducible invasive breast cancer detection in whole-slide images: A deep learning approach for quantifying tumor extent. *Scientific Reports*, 7:46450, 2017.
- [16] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 113–123, 2019.
- [17] Will Cukierski. Histopathologic cancer detection. Available at: <https://kaggle.com/competitions/histopathologic-cancer-detection>, 2018.
- [18] Sumaiya Dabeer, Maha Mohammed Khan, and Saiful Islam. Cancer diagnosis in histopathological image: Cnn based approach. *Informatics in Medicine Unlocked*, 16, 2019.
- [19] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.
- [20] Tulasi Gayatri Devi, Nagamma Patil, Sharada Rai, and Cheryl Sarah Philipose. Gaussian blurring technique for detecting and classifying acute lymphoblastic leukemia cancer cells from microscopic biopsy images. *Life*, 13(2), 2023.
- [21] Babak Ehteshami Bejnordi, Geert Litjens, Nadya Timofeeva, Irene Otte-Höller, André Homeyer, Nico Karssemeijer, and Jeroen AWM van der Laak. Stain specific standardization of whole-slide histopathological images. *IEEE Transactions on Medical Imaging*, 35(2):404–415, 2016.
- [22] Abbas Elazhari. Interpolation of low resolution images for improved accuracy in human face recognition. Electronic Theses and Dissertations, 2014. <https://scholar.uwindsor.ca/etd/5059>.
- [23] Alexander L. Fradkov. Early history of machine learning. volume 53, 2020.
- [24] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [25] Vibha Gupta, Apurva Singh, Kartikeya Sharma, and Arnav Bhavsar. Automated classification for breast cancer histopathology images: Is stain normalization important? In *Computer Assisted and Robotic Endoscopy and Clinical Image-Based Procedures*, pages 160–169. Springer International Publishing, 2017.
- [26] Metin N. Gurcan, Laura E. Boucheron, Ali Can, Anant Madabhushi, Nasir M. Rajpoot, and Bulent Yener. Histopathological image analysis: A review. *IEEE Reviews in Biomedical Engineering*, 2:147–171, 2009.

- [27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- [28] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning*. Springer, 2013.
- [29] Jakob Kather, Cleo-Aron Weis, Francesco Bianconi, Susanne Melchers, Lothar Schad, Timo Gaiser, Alexander Marx, and Frank Zöllner. Multi-class texture analysis in colorectal cancer histology. *Scientific Reports*, 6:27988, 06 2016.
- [30] Jakob Nikolas Kather, Frank Gerrit Zöllner, Francesco Bianconi, Susanne M Melchers, Lothar R Schad, Timo Gaiser, Alexander Marx, and Cleo-Aron Weis. Collection of textures in colorectal cancer histology, May 2016.
- [31] Mahendra Khened, Avinash Kori, Haran Rajkumar, Ganapathy Krishnamurthi, and Balaji Srinivasan. A generalized deep learning framework for whole-slide image segmentation and analysis. *Scientific Reports*, 11, 12 2021.
- [32] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.
- [33] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.
- [34] Ajitesh Kumar. Difference between machine learning & traditional software. October 2019.
- [35] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
- [36] Kyubum Lee, John H. Lockhart, Mengyu Xie, Ritu Chaudhary, Robbert J.C. Slebos, Elsa R. Flores, Christine H. Chung, and Aik Choon Tan. Deep learning of histopathology images at the single cell level, 2021.
- [37] Zechun Liu, Zhiqiang Shen, Shichao Li, Koen Helwegen, Dong Huang, and Kwang-Ting Cheng. How do adam and training strategies help bnns optimization?, 2021.
- [38] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.
- [39] Marc Macenko, Marc Niethammer, J. S. Marron, David Borland, John T. Woosley, Xiaojun Guan, Charles Schmitt, and Nancy E. Thomas. A method for normalizing histology slides for quantitative analysis. In *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1107–1110, 2009.
- [40] Tom M. Mitchell. *Machine Learning*. McGraw-Hill Science/Engineering/Math, March 1997.
- [41] João Mendes Moreira, André C.P.L.F. de Carvalho, and Tomáš Horváth. *A General Introduction to Data Analytics*. John Wiley Sons, Inc., 2019.
- [42] Will Nash, Tom Drummond, and Nick Birbilis. A review of deep learning in the study of materials degradation. *npj Materials Degradation*, 2, 12 2018.
- [43] OpenCV Documentation. Image interpolation. Available at: https://docs.opencv.org/3.4/da/d54/group__imgproc__transform.html#ga5bb5a1fea74ea38e1a5445ca803fff121.

- [44] Andrea Palazzi. Deep learning and temporal data processing 1-deep neural networks, 2017.
- [45] Liron Pantanowitz, Ashish Sharma, Alexis B. Carter, Tahsin Kurc, Alan Sussman, and Joel Saltz. Twenty years of digital pathology: An overview of the road travelled, what is on the horizon, and the emergence of vendor-neutral archives. *Journal of Pathology Informatics*, 9:40, 1 2018.
- [46] Liron Pantanowitz, Paul N. Valenstein, Andrew J. Evans, Keith J. Kaplan, John D. Pfeifer, David C. Wilbur, Laura C. Collins, and Terence J. Colgan. Review of the current state of whole slide imaging in pathology. *Journal of Pathology Informatics*, 2:36, 1 2011.
- [47] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An open-source deep learning framework. <https://pytorch.org/>, 2019.
- [48] Lukas Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.
- [49] Lola Rahib, Benjamin D. Smith, Rhonda Aizenberg, Allison B. Rosenzweig, Julie M. Fleshman, and Lynn M. Matrisian. Projecting cancer incidence and deaths to 2030: The unexpected burden of thyroid, liver, and pancreas cancers in the united states, 2014.
- [50] Chinmay Rane, Rishabh Mehrotra, Sourav Bhattacharyya, and et al. A novel attention fusion network-based framework to ensemble the predictions of cnns for lymph node metastasis detection. *Journal of Supercomputing*, 77:4201–4220, 2021.
- [51] Geoffrey Rolls. Fixation and fixatives - popular fixative solutions.
- [52] Rasoul Sali, Nazanin Moradinasab, Shan Guleria, Lubaina Ehsan, Philip Fernandes, Tilak U. Shah, Sana Syed, and Donald E. Brown. Deep learning for whole-slide tissue histopathology classification: A comparative study in the identification of dysplastic and non-dysplastic barrett’s esophagus. *Journal of Personalized Medicine*, 10, 2020.
- [53] Afiqah Abu Samah, Mohammad Faizal Ahmad Fauzi, and Sarina Mansor. Classification of benign and malignant tumors in histopathology images. In *2017 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pages 102–106, 2017.
- [54] A. L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3:210–229, 7 1959.
- [55] Dipanjan Sarkar and Raghav Bali. *Hands-On Transfer Learning with Python*. Packt Publishing, 2019.
- [56] Luis Serrano. Grokking machine learning.
- [57] Faezehsadat Shahidi, Salwani Mohd Daud, Hafiza Abas, Noor Azurati Ahmad, and Nurazeen Maarop. Breast cancer classification using deep learning approaches and histopathology image: A comparison study. *IEEE Access*, 8, 2020.
- [58] Wei Shao, Linda Banh, Christian A. Kunder, Richard E. Fan, Simon J.C. Soerensen, Jeffrey B. Wang, Nikola C. Teslovich, Nikhil Madhuripan, Anugayathri Jawahar, Pejman Ghannouni, James D. Brooks, Geoffrey A. Sonn, and Mirabela Rusu. Prosregnet: A deep learning

- framework for registration of mri and histopathology images of the prostate. *Medical Image Analysis*, 68, 2021.
- [59] Chris Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, 2019.
 - [60] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.
 - [61] C. R. Smittenaar, K. A. Petersen, K. Stewart, and N. Moitt. Cancer incidence and mortality projections in the uk until 2035. *British Journal of Cancer*, 115, 2016.
 - [62] Yang Song, Ju Jia Zou, Hang Chang, and Weidong Cai. Adapting fisher vectors for histopathology image classification. In *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, pages 600–603, 2017.
 - [63] V. V. Starovoitov and Yu. I. Golub. Data normalization in machine learning. *Informatics*, 18, 2021.
 - [64] Jiamei Sun and Alexander Binder. Comparison of deep learning architectures for he histopathology images. In *2017 IEEE Conference on Big Data and Analytics (ICBDA)*, pages 43–48, 2017.
 - [65] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, 2014.
 - [66] David Tellez, Geert Litjens, Péter Bándi, Wouter Bulten, John-Melle Bokhorst, Francesco Ciompi, and Jeroen van der Laak. Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology. *Medical Image Analysis*, 58:101544, dec 2019.
 - [67] Michael Titford. A short history of histopathology technique, 2006.
 - [68] Adya Trisal and Dheeraj Mandloi. Machine learning: An overview. *International Journal of Research -GRANTHAALAYAH*, 9:343–348, 8 2021.
 - [69] Abhishek Vahadane, Tingying Peng, Amit Sethi, Shadi Albarqouni, Lichao Wang, Maximilian Baust, Katja Steiger, Anna Melissa Schlitter, Irene Esposito, and Nassir Navab. Structure-preserving color normalization and sparse stain separation for histological images. *IEEE Transactions on Medical Imaging*, 35(8):1962–1971, 2016.
 - [70] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopadakis. Deep learning for computer vision: A brief review, 2018.
 - [71] Karan Wagh. What is machine learning?
 - [72] Yawen Wu, Michael Cheng, Shuo Huang, Zongxiang Pei, Yingli Zuo, Jianxin Liu, Kai Yang, Qi Zhu, Jie Zhang, Honghai Hong, Daoqiang Zhang, Kun Huang, Liang Cheng, and Wei Shao. Recent advances of deep learning for computational histopathology: Principles and applications, 2022.
 - [73] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1395–1403, 2015.

- [74] Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the Twenty-First International Conference on Machine Learning*, page 116, New York, NY, USA, 2004. Association for Computing Machinery.
- [75] Zhilu Zhang and Mert R. Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels, 2018.
- [76] Ziliang Zhong, Muhang Zheng, Huafeng Mai, Jianan Zhao, and Xinyi Liu. Cancer image classification based on densenet model. *Journal of Physics: Conference Series*, 1651(1):012143, nov 2020.