Segregation Similarity Loss In Morphological Ranking Of Image Search In Histopathology

by

Pooria Mazaheri

A thesis submitted to the
School of Graduate and Postdoctoral Studies in partial
fulfillment of the requirements for the degree of

Master of Applied Science in Electrical and Computer Engineering

Department of Electrical, Computer, and Software Engineering (ECSE)

University of Ontario Institute of Technology (Ontario Tech University)

Oshawa, Ontario, Canada

April 2022

Thesis Examination Information

Submitted by: **Pooria Mazaheri**

Master of Applied Science in **Electrical and Computer Engineering**

Thesis title:
Segregation Similarity Loss In Morphological Ranking Of Image Search In Histopathology

An oral defense of this thesis took place on April 21, 2022 in front of the following examining committee:

Examining Committee:

Chair of Examining Committee	Dr. Khalid Elgazzar
Research Supervisor	Prof. Shahryar Rahnamayan
Examining Committee Member	Dr. Massoud Makrehchi
Thesis Examiner	Dr. Mehran Ebrahimi

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

Abstract

Histopathology is the study of changes in tissue caused by diseases such as cancer. It plays an important role to diagnose the cancers. Regarding the large variation of many cancers types, and the large size of Whole Slide Images (WSIs), the analysis of histopathology images is challenging. To come up with this challenge, AI algorithms, such as deep learning (DL) are used to automate image analysis efficiently and accurately. In this study, some DL methods are developed on medical images focusing on the following goals.

Firstly, we propose a model that can help us classify cancers better and faster and achieve good results compared to other models. Most of the present models for the classification of histopathology images are very large and accordingly have many parameters to be learned/optimized and require enormous computational times to achieve reliable results. We propose a more compact network that is tuned to classify cancer subtypes with less computation time and memory complexity to overcome these issues. This model, namely custom EfficientNet, is based on EfficientNet topology, but it is tailored for classifying histopathology images. The utilized model is evaluated over three-tumor-type brain, lung, and kidney from TCGA repository. The results show that the proposed model, compared to state-of-the-art models, i.e., KimiaNet, can classify cancer subtypes more accurately and provides superior results. Besides, the proposed model achieves memory and computational efficiency in the training phase and is a more compact deep topology compared to KimiaNet.

More recently, deep learning was applied for the challenging task of image search on the TCGA repository. Researchers can use the image search results to compare data of current and previous patients and learn from cases that have been clearly treated and diagnosed. However, there is no way to train a model using image search, hence image search must be used on the outcomes of a model that was trained using classification. Moreover, it can be seen that the obtained results from the classification method suffer from bias, and the classification loss function cannot make it possible for us to reduce bias during the network's training phase. Secondly, the research proposes a new loss function, Similarity Loss (SL), to address these problems. This loss function allows us to train the model based on image search, removing the requirement for us to use other approaches for training image search models. Besides, unlike the classification loss function, the modified version of this loss function, Segregation Similarity Loss (SSL), helps us reduce the adverse effect of one of the major problems in this field called bias and obtain better and more reliable results. By utilizing SSL, we achieve promising results to classify histopathology images. SSL function achieved up to 5% and 9% improvement compared with the state-of-art models for Lung and Brain dataset, respectively.

Keywords: EfficientNet; Deep Learning; Loss Function; Histopathology Images; Classification; Content-based Image Search; Bias Reduction.

Author's Declaration

I hereby declare that this thesis consists of original work authored by me. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize the University of Ontario Institute of Technology (Ontario Tech University) to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize University of Ontario Institute of Technology (Ontario Tech University) to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis will be made electronically available to the public.

_____ Pooria Mazaheri

Statement of Contributions

The following contributions are presented in this thesis:

- Proposing an effective approach to select randomly part of each patch as network's input to make the model faster and in addition reduce the need for expensive processing resources
- Utilizing mean method to calculate the feature vector for each WSI with extracted features from the related patches
- Utilizing WSI's feature vector to predict its label based on the other WSIs found in the neighbors
- Customizing a smaller network to achieve better results and reduce the computational power compared with the state-of-art model to classify cancers.
- Creating a new loss function based on the image search method for search-based model and eliminating their needs from using other methods such as classification
- Creating a new loss function to tackle one of the significant problems in histopathology models, bias, and reduce the effect of it on the network during the training phase to improve accuracy and achieve better and more reliable results

Sole authorship

I hereby certify that I am the sole author of this thesis and that no part of it has been published or submitted for publication. I have used standard referencing practices to acknowledge ideas, research techniques, or other materials that belong to others. Furthermore, I hereby certify that I am the sole source of the creative works and/or inventive knowledge described in this thesis.

Acknowledgments

I am overwhelmed in all humbleness and gratefulness to acknowledge my gratitude to all those who have helped me put these ideas well above the level of simplicity and into something concrete.

I would like to express my special thanks to my supervisor, Prof. Shahryar Rahnamayan, as well as my co-supervisor, Prof. Hamid Tizhoosh, who gave me the golden opportunity to do this project on the topic Histopathology Images, and helped me in doing a lot of research and I came to know about so many new things. I am thankful to them.

I would also like to thank Dr. Azam Asilian for his valuable advice and knowledge and for all her help in writing papers and conducting the research.

I would like to thank my parents and my sisters for supporting me throughout my education and making many sacrifices to ensure that I could pursue higher education and achieve my goals. I would like to dedicate this thesis to them, who meant the world to me, and always patiently supported me and guided me during tough days.

Pooria Mazaheri

Table of Contents

List of Tables

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

List of Figures

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

List of Abbreviations and Symbols

AI…….Artificial Intelligence

ANN……..Artificial Neural Network

CNN…….……Convolutional Neural Network

CAE…….Convolutional AutoEncoders

DL……...Deep Learning

FNN……...Feedforward Neural Networks

LRN……...Local Response Normalization

LSTM……...Long Short Term Memory

MSE…….....Mean Square Error

NAS……Neural Architecture Search

NLP…….Natural Language Processing

RNN……...Recurrent Neural Networks

SGD……...Stochastic Gradient Descent

SL……. Similarity Loss

SSL…….……Segregation Similarity Loss

TCGA……...The Cancer Genome Atlas

WSI……..Whole Slide Image

Chapter 1

Introduction

1.1 Introduction to Histopathology

Histopathology is the study and diagnosis of diseases of the tissues such as cancer and involves examining tissues or cells under a microscope. Diagnosing a disease in a patient using histopathology has some steps, which start with performing a biopsy. It means removing a small part of tissues called a specimen from the patient's body, mainly from a mass or tumor. There are different ways for doing the biopsy, such as performing surgery, an endoscope, and using a needle. In the second step, the specimen is analyzed by a pathologist. They describe how it looks by features such as morphology. In the next step, for further diagnosis, they cut and put it under the microscope. To cut the specimen into thin slices, it should be firm enough. Two ways have been presented for doing that.

- Paraffin-embedded (permanent)
- Frozen sectioning

In the permanent way, it is placed into a fixative for several hours, and then the water inside the specimen is replaced with paraffin wax. When it is firm, they cut it into very thin slices. Then, they put a slice on a glass slide and replace the paraffin with water. In the last step, to stain different parts of the cells in the tissue, they utilize dyes. As a result of using dyes, the cell nuclei turn to dark blue, and the cytoplasm to be turn into pink. In the second way, the specimen is frozen and cut into thin layers. The staining process in this way is the same as permanent sectioning. In the last step, the prepared specimen slide is put under a microscope by a pathologist, and then a diagnosis is made based on the investigation.

Both of these ways have some advantages and disadvantages. Although the permanent sectioning process takes many days, it has the best quality for the examination. The frozen sectioning is completed after 20 minutes, but its quality is lower compared to permanent sectioning. Since quality is important, the permanent slides are used in this study [1]–[4]. The prepared specimen is scanned in very high magnifications. The magnification that is provided by special scanners can be different, like 10x, 20x, and 40x. These special scanners provide the results that are gigapixels images, as large as 100K * 100K pixels named WSI [5]. Figure 1.1 shows a sample WSI.

During the last decade, due to the advancement of the whole slide digital scanners, digital pathology has attracted more attention among researchers. Using digitized WSI has many benefits [6]–[9]:

(i) Fast search over the WSIs in the hospitals and speeding up the procedure

(ii) Make collaboration among pathologists easier and get different ideas from pathologists around the world.

(iii) Provide the opportunity to apply image processing algorithms on data to extract useful information

(iv) Provide the opportunity to utilize Machine Learning and Deep learning for decreasing inter-observer variability and increasing consistency.

Handcrafted methods were used in the past to analyze digital pathology slides. The handcrafted methods have some disadvantages, such as not generalizing well and reducing performance when facing new data [10], [11]. So, Deep Learning attracts attention to solve these problems. Deep Learning is a good solution that digital pathology can benefit from. Since the large size of WSIs is a big problem in digital pathology, using extracted features from a deep network can be an excellent candidate to represent an image. These extracted features help us to find similar WSIs for a query WSI in an extensive archive. This finding method, Image search, plays a crucial role in returning similar WSIs and predicting the labels. It can help and assist medical professionals in various tasks such as diagnostic and research. It is essential to implement Image Search very well because experts can benefit from it to recommend well-informed

diagnoses and treatments.

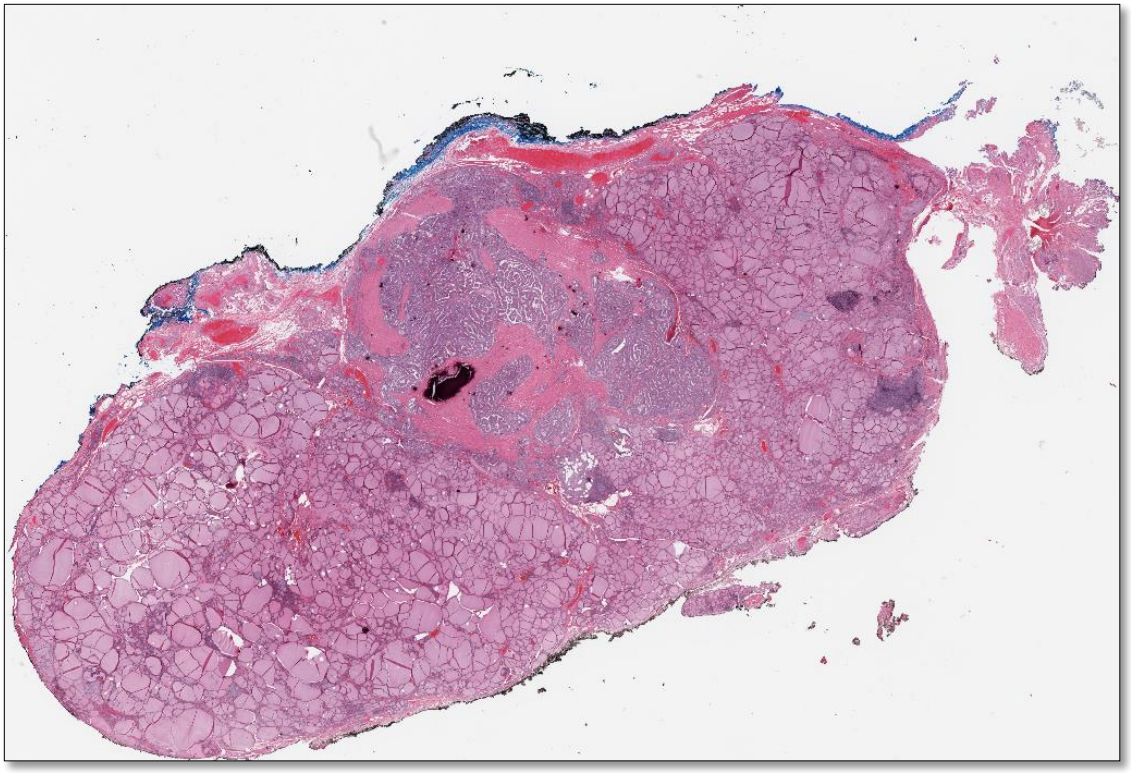


Figure 1. 1 : Illustration of a gigapixel WSI of Kidney Renal Clear Cell Carcinoma

1.2 Contribution

The first objectives of the thesis are to implement an efficient classifier that is faster and more accurate for different types of cancers. Some other models, like KimiaNet [12] that is the state-of-art model to classify histopathology images, but it has many disadvantages and problems. Despite good classifying histopathology images, the KimiaNet is a huge model that needs many resources, such as GPUs to train the network. Moreover, training the network takes many hours, and tweaking the hyper-parameters takes many days. Furthermore, because it requires a lot of resources and takes a long time to compute, it is only trained for all cancer images, not for a specific type of cancer. Motivated to address

these challenges and limitations, we present a customized “smaller” network, custom EfficientNet, to achieve better results compared to state-of-art models, and reduce the time and resource budget for training the model. As well, we need more compact DNNs compared to state-of-the-art DNNs, and the capability of training with smaller datasets. These features enable us to train our model multiple times for different forms of cancer, such as lung and brain cancer, and then swiftly retrain it for fresh cancer images. Pathologists can benefit from our classifier since it uses a method that detects the most comparable cases to a query slide and predicts a label for them. To implement this model, we customize an EfficientNet model [13] and train different models on that in the first step. These models are trained for two datasets, Lung, and Brain. These datasets involve patches of size 1000 by 1000 pixels extracted at 20x magnification from 7375 WSIs of the TCGA dataset. The first model related to Lung dataset is trained on around 25000 patches depicting two different tumor subtypes. The second model is trained on around 36000 patches on Brain dataset with two different subtypes. Finally, the obtained results from these models are compared with one of the state-of-art models, KimiaNet, that trained on histopathology images. The models have high F1-Scores and show promising results. Despite the positive results, we do not stop there; as the thesis' second goal, we attempt to develop a new loss function that will aid in the training of an image search model based on the search technique rather than the classification approach.

The second objective of the thesis is to create and develop a new technique to train a model based on the image search method and reduce bias during the training phase. Since the image search model is an important technique in histopathology images, it is necessary to have a model that focuses on it and trains a network based on that. Because there is

currently no approach for training models using image search, our proposed technique could be a useful way to bridge this gap and allow researchers to use image search to train their models. Besides, as a result of effecting bias on the result, it is crucial to consider the bias and attempt to reduce its effects to achieve better and more accurate results. To achieve these goals, we create a new loss function and train an EfficientNet model based on that. This new loss function allows us to utilize the image search technique for training the model and helps us consider and reduce the effect of bias to achieve promising and more reliable results.

The contributions of this thesis are:

- Proposing an effective approach to adjust patch size for the feeding network to make the model faster and reduce computational power and hardware costs
- Customizing the EfficientNet to obtain better results and reduce the computational power compared with the state-of-art model to classify cancers.
- Implementing a model called leave-one-out to find similar WSIs for a query WSI in an extensive archive
- Creating a new loss function based on the image search method for search-based models and eliminating their needs from using other methods such as classification
- Creating a new loss function to tackle one of the significant problems in histopathology models, bias, and reduce its effect on the network during the training phase to improve accuracy and achieve better and more reliable results

1.3 Thesis Organization

The remainder of this thesis is structured as follows. Chapter 2 will review the relevant literature, research papers, and methods in deep learning and digital pathology. Next, this chapter provides background information and concepts used in the thesis. After reviewing the other research and previous works, the next chapter discusses how the datasets are created and explains the details of each dataset. This process of creating datasets and data preparation is described in Chapter 3. Then, the custom EfficientNet model is implemented to utilize the created datasets to train. This custom EfficientNet is compiled in Chapter 4. Next, Chapter 5 explains the procedure of creating the new loss function based on image search. This chapter provides the details of the function, Similarity Loss (SL), and shows how to create and develop it. Next, another type of proposed new loss function for reducing bias, Segregation Similarity Loss (SSL) is compiled in Chapter 6. This chapter helps to show the effect of reducing bias on training models. Finally, future direction and Conclusion are stated in Chapter 7.

Chapter 2

Background Literature

This chapter covers the core concepts used in this thesis and reviews the relevant literature. Firstly, Section 2.1 overviews deep learning, Convolutional Neural Networks (CNN) network and presents background information about network architectures that have been utilized in this thesis for histopathology feature learning and classification. In Section 2.2, we review other researches that utilized deep learning and various networks in histopathology images. In Section 2.3, concise background information regarding image search and retrieval is provided. Moreover, this section briefly reviews the researches that work on image search in digital pathology. At the end of this chapter, in Section 2.4, we discuss bias and its effects on histopathology images, and we review some researches on that.

2.1 Introduction to Deep Learning

Deep learning is a sub-filed of machine learning methods and is based on the algorithms inspired by functions and structure of the biological brain. To be more detailed, a neuron takes electric inputs from other neurons. If a neuron takes input with high electric potential, it will send electric potentials to others. Firstly, in 1962, F. Rosenblatt [14] simulated a single-layer Artificial Neural Network (ANN). As a result of development in hardware and theory, the ANN became more practical in the next decade [15], [16]. Figure 2.1 shows an example of a two-layer perceptron. Early work illustrates that a linear perceptron is unable to be a universal classifier, but a network that has one hidden layer with a nonpolynomial activation function can be. So, researchers were willing to use more layers in their models and utilize deep learning. The adjective deep in deep learning refers to the use of multiple layers in a neural network. Some reasons led to the succeeding of deep learning in the past years, like achieving state-of-art results in Natural Language Processing and Computer Vision, using learned parameters in the various domains (Transfer Learning), and reducing the cost of computational hardware and storing the data. The last reason results in different ANN architectures being developed.

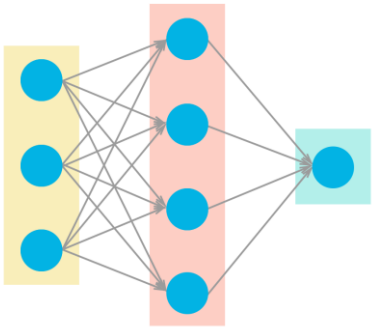


Figure 2. 1 : Two-Layer Perceptron

Due to hardware and ANN architecture development, new methods are applied in various applications [17]–[20]. These applications utilize advanced ANN architecture such as weight-sharing [21], batch-normalization [22], and ReLu [23], variants of ANN such as CNN [24], Recurrent Neural Networks (RNN) [25], Convolutional AutoEncoders (CAE) [26]. Due to the increasing number of layers in ANN models, researchers specifically refer to deep neural network-based methods. Researchers proposed different deep learning models for less training time and better generalization. For instance, deeper network architectures are built to capture higher-order concepts in images [27], [28]. RNN models are proposed to handle temporal patterns and benefit from internal memory [29], [30]. The CNNs achieved excellent results in multiple applications and became arguably the most widely used single-image classification and detection model. Figure 2.2 shows some of the existing architectures.

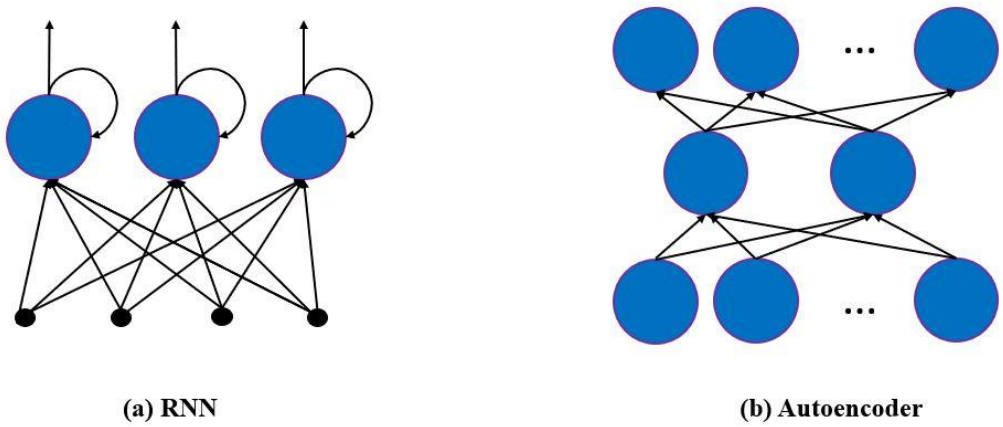


Figure 2. 2 : Two examples of neural network architectures. (a) Recurrent Neural Network (RNN) Structure; (b). Autoencoder Structure

2.1.1 Architectures

Different types of deep learning models have been proposed during the last decades, such as RNNs and CNNs. These different networks have been proposed for less training time, better generalization, and a broader domain of applications. In the following, it is discussed about some architecture and theoretical analysis of ANNs with emphasizes on CNN.

2.1.1.1 Feedforward and Recurrent Neural Network

In Feedforward Neural Networks (FNN), the relation between the input and output of two neurons is fixed. It means the status of the sending neuron does not depend on the status of the receiving neuron. For better simulation of neural activities, an undirected network known as the Hopfield network was proposed [31]. This kind of network is capable of holding content-addressable memory. In contrast, a recurrent network is directed but has recurrent links. The Recurrent network is applied to leverage temporal information [32], [33] in Natural Language Processing (NLP) and object recognition [34], [35]. The Long Short Term Memory (LSTM) [36], [37] is an RNN unit widely used in speech and audio-related applications [36], [38].

2.1.1.2 Convolutional Neural Network

The CNN model is one of the architectures that is used in different works [39]–[41]. A typical architecture with explanations is shown in Figure 2.3. A CNN is a feedforward, nonstochastic, supervised deep neural network with weight sharing in most applications.

The CNNs achieved state-of-the-art results during experiments in multiple applications [42] such as video classification [43], speech recognition [44], and image classification [45], [46], object detection [47], and action recognition [48].

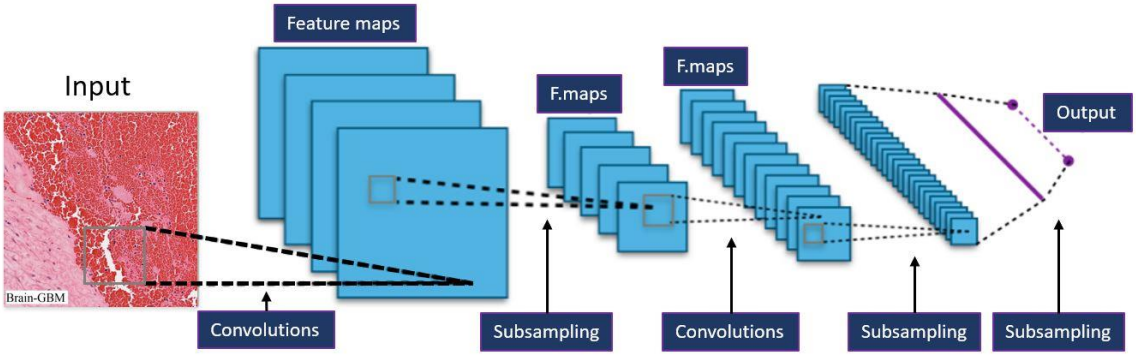


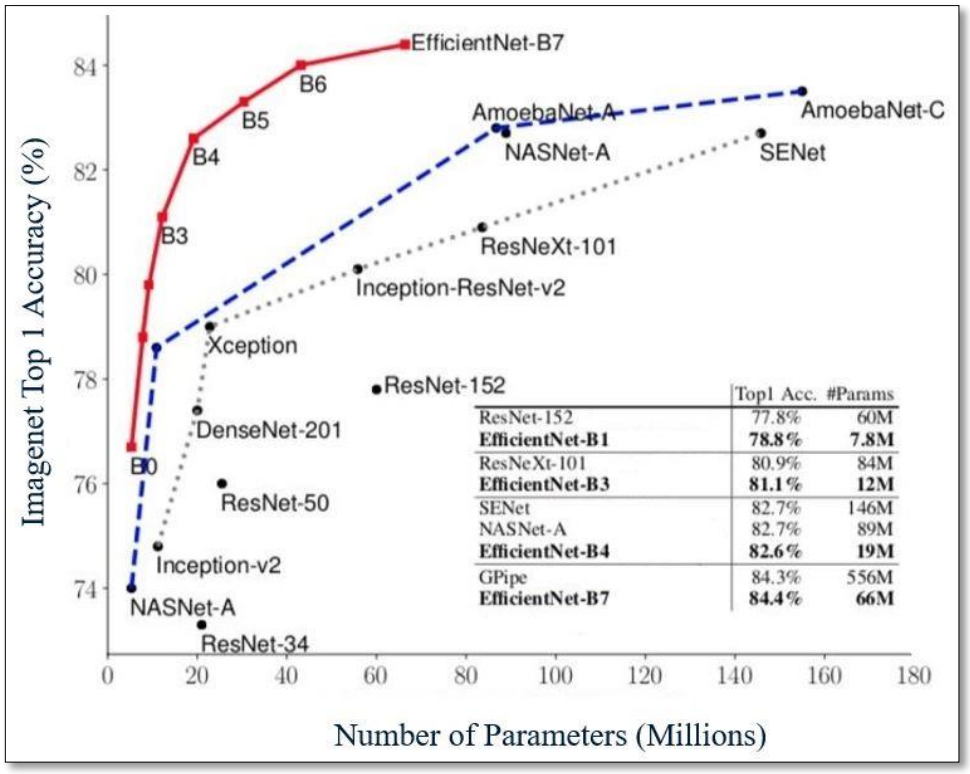
Figure 2. 3 : A typical CNN architecture. First, a group of layers is repeated several times. There are several optional layers such as Local Response Normalization (LRN), applying an activation function after convolution, batch normalization [49]*, weight normalization* [50]*, and dropout* [51]*. Two widely used downsampling methods are max-pooling and average-pooling, i.e., extracting the maximum or average value in nearby convolutional responses. After the convolutional layers, several fully connected layers can be attached. Finally, for classification problems, a multi-class logistic regression model is applied to the last fully connected layer to generate the probability values of the predicted classes. The final multi-class logistic regression is also called the Softmax layer. The image is taken from wikimedia.org.*

2.1.2 EfficientNet

During this research, we choose the EfficientNet architecture for our model to classify the patches. This model architecture was designed by Tan and Le [52] in 2020 and has later been commonly used in computer vision problems due to their amazing feature learning ability. There are eight CNN models in the EfficientNet family, including EfficientNet-B0,

EfficientNetB1, ..., and EfficientNet-B7. A larger index number shows that the corresponding model has a larger network size.

We choose EfficientNet-B0 for our model to classify the patches instead of DenseNet that used as base architecture for KimiaNet for many reasons. EfficientNet-B0 has fewer parameters and also a smaller size than the DenseNet. This feature results in our model training faster. In addition, as a well-designed base architecture, EfficientNet is successful in many datasets like ImageNet compared with other famous networks such as VGG [53], ResNet [47], and DenseNet [55]. The efficiency of EfficientNets compared to other state-of-the-art models is shown in Figure 2.4.



	Top1 Acc.	#Params
ResNet-152	77.8%	60M
EfficientNet-B1	**78.8%**	**7.8M**
ResNeXt-101	80.9%	84M
EfficientNet-B3	**81.1%**	**12M**
SENet	82.7%	146M
NASNet-A	82.7%	89M
EfficientNet-B4	**82.6%**	**19M**
GPipe	84.3%	556M
EfficientNet-B7	**84.4%**	**66M**

Figure 2. 4 : The efficiency of EfficientNets compared to other methods [52]

2.1.2.1 A Better Way to Scale Up CNNs

Convolutional neural networks are usually built at a fixed resource cost and then scaled up to obtain better accuracy when more resources are made available. For instance, with increasing the number of layers, Resnet-18 can be scaled up to ResNet-200 [56]. In the conventional ways for model scaling, researchers increase the CNN width or depth or use larger input image resolution for training. Although these methods can improve the accuracy, they usually need tedious manual tuning. Tan and Le [52] presented a novel model scaling method that uses an effective compound coefficient to scale up CNNs in a more structured manner. By using this method and AutoML [57]–[59], they developed EfficientNets models that are small and fast. Figure 2.5 shows a comparison between conventional scaling methods and compound scaling method. In the first figure, the baseline of a network can be seen, and in the following three figures, the conventional scaling way is used, and just one dimension of network resolution, depth, or width is considered to increase. In the last figures, the compound scaling method is used to uniformly scale all three dimensions with a fixed ratio.

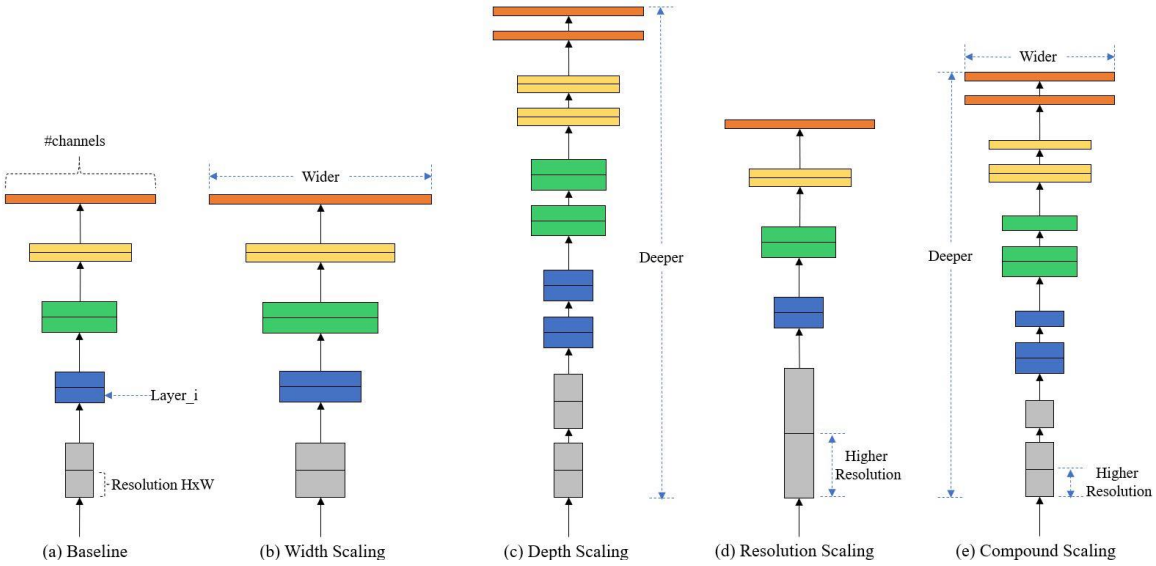


Figure 2. 5 : Model Scaling. (a) is an example for baseline network. (b), (c), (d) are conventional scaling that only try to increase one dimension of network resolution, depth, or width. (e) is the compound scaling method that is proposed to uniformly scales all 3 dimensions with a fixed ratio [52].

2.1.2.2 Compound Model Scaling

The compound model scaling method is based on wise customization of the components to scale up the network. Generally, in the compound scaling method, firstly, a grid search is performed to find the relationship between various scaling dimensions of the baseline network when resource constraint is fixed. Then, those coefficients are applied to scale up the baseline network to the desired target model size or computational budget.

2.1.2.3 EfficientNet Architecture

The baseline network is a vital part of the effectiveness of model scaling, and it should have good architecture. Good architecture means the baseline network architecture should

have already achieved suitable accuracy in order to further changes can improve it. Therefore, a baseline architecture selection plays an important role in model scaling. The researchers that work on EfficientNet have done their experiments on some baseline architectures, and then they have developed a new baseline network using Neural Architecture Search (NAS) and AutoML that optimizes accuracy and efficiency. This developed architecture uses the mobile inverted bottleneck convolution (MBConv) both for the resulting architecture and the baseline network scale-up to obtain a family of EfficientNet models. In MBConv, the key idea is first to use a 1*1 Conv layer to increase the number of channels to three times the initial and then apply a Depthwise Convolution [60] to get the feature maps. Finally, the second 1*1 Conv layer downsamples the number of channels to the initial value. The structure inside an MBConv module is shown in Figure 2.6. It reformulates a standard convolutional operation into a sequence of operations, including expansion, depthwise convolution, and residual connection layers, which allows an MBConv module to use much fewer network parameters to express comparable feature learning capacity compared to a standard convolutional layer. In an EfficientNet, there are two types of MBConvX: MBConv1 and MBConv6, which show the use of ReLu and ReLu6 activation functions in the corresponding MBConvX module, respectively. The other types of EfficientNet models have similar structures to the EfficientNet-B0, except for a different number of MBConvX modules used in the CNN blocks. The EfficientNet with a smaller index uses fewer MBConvX modules, so the EfficientNet-B0 is the smallest, and EfficientNet-B7 is the largest model among the EfficientNet models. The network architecture of EfficientNet-B0 illustrated in Figure 2.7 has a different number of MBConv

blocks as a basic building block of the network. It can be divided into seven blocks in different colors based on striding, filter size, and the number of channels.

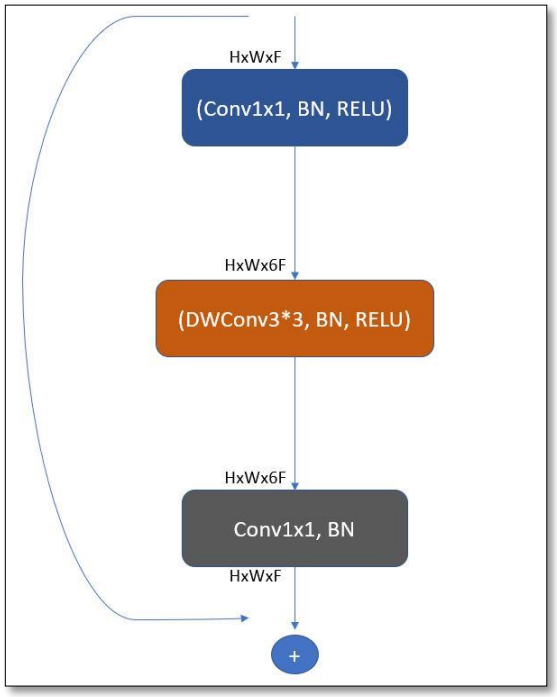


*Figure 2. 6 : MBConv Block. DWConv stands for depthwise Conv, k3*3 / k5*5 defines the kernel size, BN is the batch norm, H*W*F is tensor shape (Height, Width, Depth)*

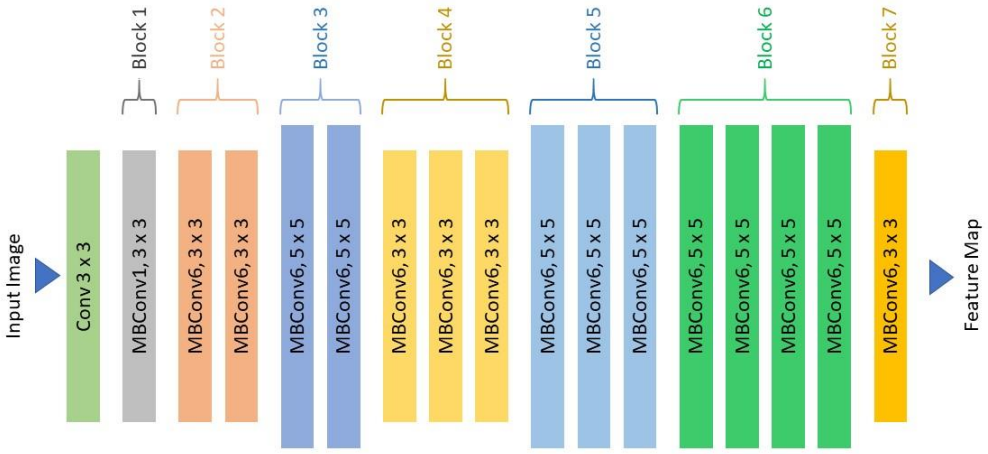


Figure 2. 7 : The simple architecture of the baseline network EfficientNet-B0 helps scale and generalize easily. The picture shows using different blocks, including Conv and MBConv, to create the EfficientNet baseline.

2.1.2.4 Architectural Details of EfficientNet Model

EfficientNet has eight models, and they start from B0 to B7. The first model, EfficientNet-B0, has 237 layers, and the last model, EfficientNet-B7, has 813.

During this work, the EfficientNet-B0 was used as a network to classify the histopathology images. The first thing in EfficientNet-B0 is the stem, and the last layer is the final layer. Figure 2.8 shows the Stem and Final layer in EfficientNet architecture. These layers include padding to basically extend the area of an image, activation to learn complex patterns in data, and batch normalization to allow every layer of the network to do learning more independently.

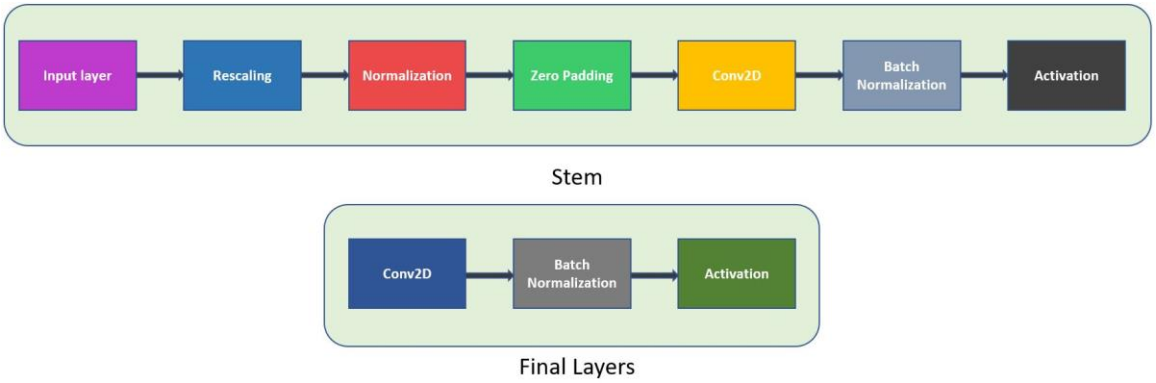


Figure 2. 8 : Stem and Final layer in EfficientNet Architecture.

After this, it contains seven blocks. As mentioned, EfficientNet-B0 has 237 layers, and all these layers can be made from the five modules shown below in Figure 2.9 and the stem above, shown in Figure 2.8.

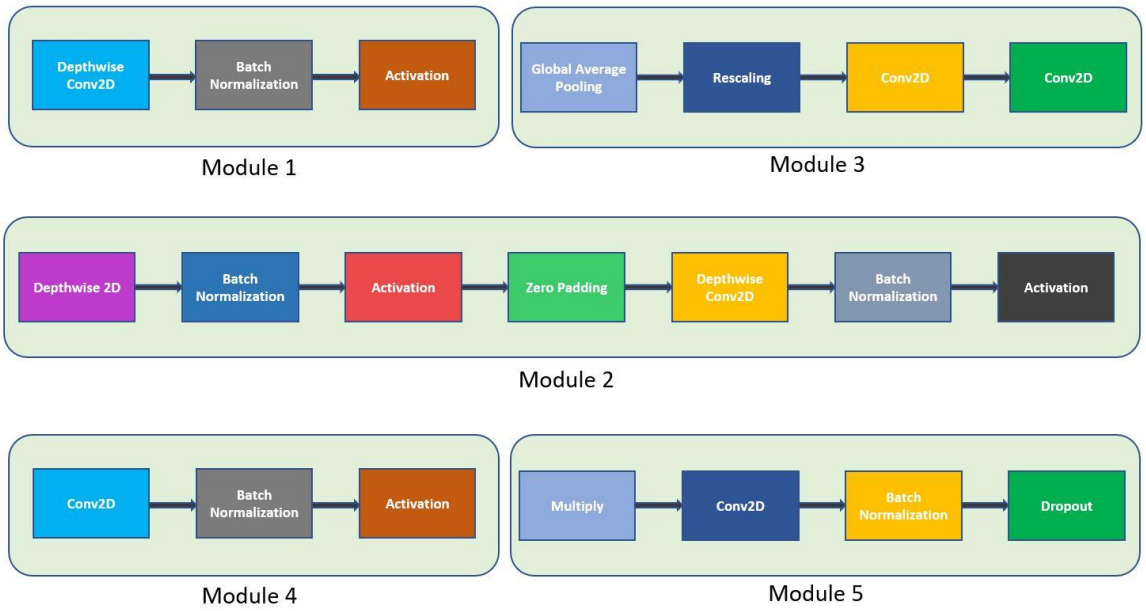


Figure 2. 9 : Different Modules in EfficientNet Architecture.

Module 1: It was utilized as a starting point for sub-blocks.

Module 2: It was utilized as a starting point for the first sub-block of all the seven main blocks except the first one.

Module 3: It was connected as a skip connection to all the sub-blocks.

Module 4: It was utilized for combining the skip connection in the first sub-blocks.

Module 5: Each sub-block was connected to its previous sub-block in a skip connection, and they were combined using Module 5.

These modules are further combined to form sub-blocks that will be utilized in a determined way in the blocks. These created sub-blocks are shown in Figure 2.10. The first sub-block is made from modules 1,3 and 4 and combines batch normalization, average pooling, and rescaling that is utilized as the first sub-block in the first block. The second sub-block is created from modules 2,3 and 4 that includes zero padding and depthwise 2d to process the data in the first step of each block except the first one. The last sub-block used to create the EfficientNet model many times in all blocks except the first one is combined from batch normalization and dropout to improve the generalization of the model and reduce the overfitting. The last sub-block plays an important role in the EfficientNet model to improve the generalization of the model and helps it to obtain promising results on the test data.

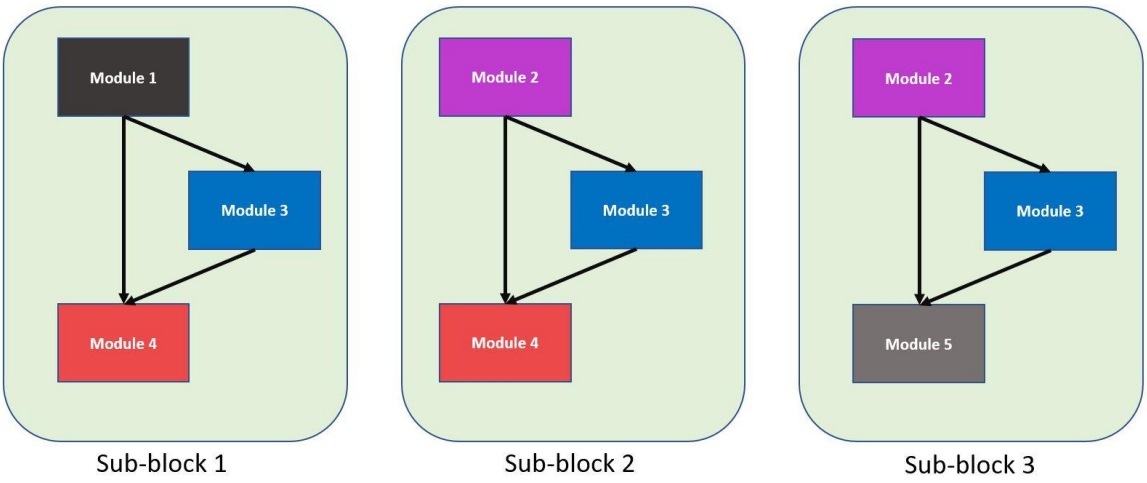


Figure 2. 10 : Different Sub-blocks in EfficientNet Architecture.

Sub-block 1: It was utilized only as the first sub-block in the first block.

Sub-block 2: It was utilized as the first sub-block in all the other blocks.

Sub-block 3: It was utilized for any sub-block except the first one in all the blocks.

Now everything combines to create the EfficientNet model, Figure 2.11.

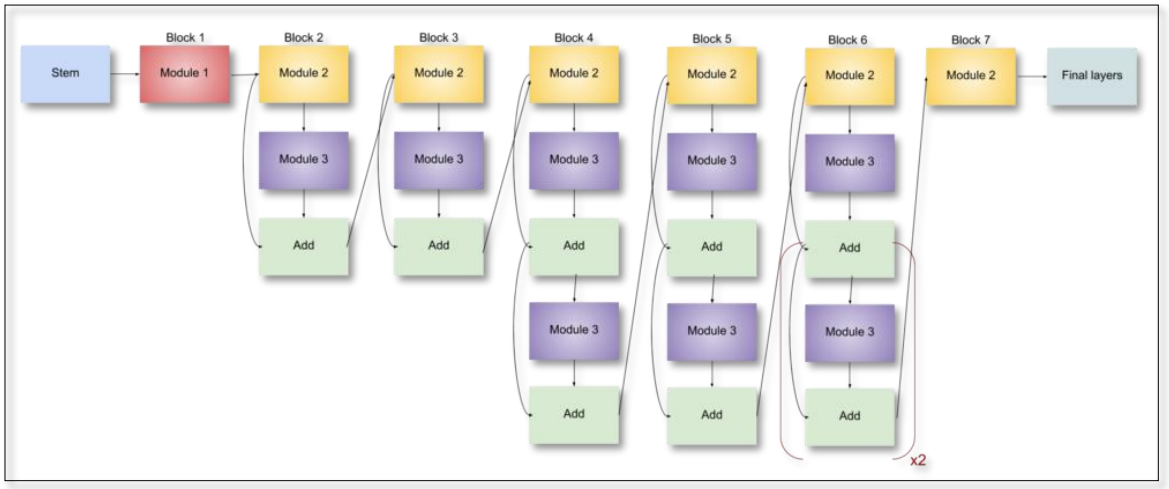


Figure 2. 11 : Combined Sub-blocks to create EfficientNet-B0 model [61]

The last thing that should be mentioned here is kernel size for convolution operations along with the channels, layers, and resolution in EfficientNet-B0 [52]. The first part in Table 2.1, the network baseline, shows the number of Channels, Layers, and Resolution for the EfficientNetB0.

Table 2. 1: Custom EfficientNet Architecture

Stage i	**Operator** F_i	**Resolution** H_i*W_i	**#Channels** C_i	**#Layers** L_i
EfficientNetB0 Architecture, the network baseline				
1	Conv 3x3	224*224	32	1
2	MBConv1, k3x3	112*112	16	1
3	MBConv6, k3x3	112*112	24	2
4	MBConv6, k5x5	56*56	40	2
5	MBConv6, k3x3	28*28	80	3
6	MBConv6, k5x5	14*14	112	3
7	MBConv6, k5x5	14*14	192	4
8	MBConv6, k3x3	7*7	320	1
9	Conv1x1 & Pooling &FC	7*7	1280	1
Additional Layer				
10	BN/Dropout	7*7	1280	1
11	FC/BN/Swish/Dropout	1	512	1
12	FC/BN/Swish	1	128	1
13	FC/Softmax	1	NC	1

2.1.3 Transfer Learning

For training a neural network, the big problem is to collect adequate data. Solving this problem is too hard since labeled data can only be obtained through a manual process, which is both time-consuming and prone to error [62]. To this end, Huang et al. [55] in

2017 presented transfer learning as an effective way to export the knowledge extracted from a mature source domain to a novice target domain. With using transfer learning, existing parameters, such as convolution weights from a model trained on extensive datasets, are used for training new models with a relatively fewer number of labeled images. In this work, we used weights pre-trained from the ImageNet dataset as it is beneficial for classifying images with EfficientNet.

To adapt the network architecture with the histopathology data, we extended the EfficientNet-B0 architecture by adding additional layers and a fully connected layer before a softmax function at the top of the classifier. The batch normalization constrains the output of the last layer in range, forcing standard deviation one and zero mean. This regulation improves the stability of the model and decreases training time. The second part in Table 2.1, Additional layers, shows the additional layers to the EfficientNetB0 baseline. This part includes Batch normalization (BN), Dropout, and Fully connected layer (FC). Now, the created pre-trained custom EfficientNet is utilized for Histopathology images.

2.2 Deep Learning Approaches in Digital Pathology

Researchers utilize various types of Networks in histopathology images. These networks can be categorized into three categories. The first category is the Pre-trained network. Spanhol et al. [63], in 2017, worked on patches of breast cancer, BreaKHis dataset, and tried to extract features from the deepest layers of the BVLC CaffeNet model, reusing the pre-trained ImageNet weights. To classify these features, they used Logistic Regression. They compared their promising results to a CNN trained from scratch. The second category

is the Fine-Tuned networks. Faust et al. [53], in 2019, worked on a VGG_19 model and tried to fine-tune the last two blocks of it with an average pooling added at the end, initialized with pre-trained weights of the ImageNet. To find out the relationships of CNNs' deep features and human recognizable morphologic patterns, they extracted features from 1656 WSIs. The last category is the Trained Networks. In 2019, Fu et al. [65] worked on an Inception-V4 network. They tried to fine-tune this network to classify around 8000 WSIs available on the TCGA repository. They utilized this network as a histopathology patch feature extractor. They called these features computational histopathological features. During the research, they tried to find out the relationship between the computational histopathological features and genomic driver alterations, as well as the whole transcriptomes and survival. In another work, Liu et al. [66], in 2017, trained an Inception model on the Camelyon16 dataset. They trained this network with three configurations, random weights, ImageNet weights, and downsize models. They found out that although the pre-trained weights speed up the convergence, they cannot improve the results. Next, in 2019, Wei et al. [67], [68], worked on a Resnet model to classify the Lung cancer dataset. They used this model to find major and minor histologic patterns in a WSI. These patterns consider essential information that could help pathologists with preparing the documentations needed for each patient. To prepare the dataset, the three pathologists manually labeled the images. They implemented four Resnet models with different sizes and obtained the same performance. So, they chose the smallest one, a ResNet with 18 layers. Their model obtained promising results compared with the three pathologists' work.

The mentioned studies only focus on classification methods for their work, while another method, Image search, can be beneficial for histopathology images.

2.3 Image Search in Digital Pathology

More recently, deep learning has been applied for the challenging task of image search on the TCGA repository. Image search is an application of deep learning in histopathology that requires salient, discriminative, and representative features. These features, which are descriptive of the content of images, are obtained by the feature extraction method, i.e., feeding images into a pre-trained model and using the deep feature output of a specific layer for image representation. The idea of image search is to compare a feature extracted from a query image to all whole slide images in a dataset in a computationally efficient manner and find the most similar matches. The results can provide an opportunity for researchers to match records of current patients and past patients and learn from evidently treated cases and also diagnosed cases.

These days, some researches can be found on content-based image search in digital pathology. Kalra et al. [68] introduced Yottixel that is a search engine for real-time WSI retrieval in digital pathology. The name of Yottixel is a combination of “yotta” which is the largest decimal unit prefix in the metric system, and “pixel”. In this work, an unsupervised color-based clustering method is used to extract a set of images from each WSI called Mosaic. Each Mosaic covers around 5 percent of the tissue specimen. Next, the Mosaic is used as input for pre-trained CNNs to extract deep features. Finally, the extracted feature vectors are barcoded to create a bunch of barcodes for fast indexing of WSIs. As a result of the barcoding of gigapixel WSIs, Yottixel can search millions of times in real-time. In another work, Hedge et al. [69] utilized a pre-trained network to convert an input

image into a feature vector. This network is pre-trained on five billion images and is able to extract discriminative features by computing the embeddings of input images. This network adopted a dataset that was annotated manually by pathologists to evaluate the search performance in finding patches with the same histologic features. In another recent research, Riasatian et al. [12] proposed image representation for search in digital pathology. They built a network called KimiaNet based DenseNet topology with several configurations. In the first configuration, they retrained only the last layer, while other layers froze. In the second, they retrained two last layers, and in the third and fourth configuration, they tried to retrain the third and fourth last layers, respectively. They utilized the TCGA dataset and extracted around 240,000 histopathology images from 7000 WSI with a clustering-based approach at 20X magnification. Their approach worked based on a high-cellularity metric and extracted images with the size of 1000*1000 pixels. After training the network, a Min-Max barcoding algorithm was used to convert the feature vectors to binary codes in the test phase [68]. The authors test the KimiaNet for image search on three histopathology datasets for multi-organ WSI search. They reported two types of search, Horizontal search and Vertical Search. During Horizontal search, they searched images across the dataset to find the WSI with a similar tumor type to the query WSI among all WSIs. During Vertical search, they searched to find a similar type of malignancy in an anatomical site, i.e., search for similar WSI with the similar tumor subtype between all whole slide images of the same tumor site in their test dataset. Other works can be mentioned that focus on content-based image search in digital pathology [70]-[79]. One of the recent research in this field attracts attention to a problem that it might not consider many times. This research [80] shows that the

TCGA dataset that was used in the mentioned studies suffers from a problem that can affect the results. This problem is bias.

2.4 Bias in Histopathology Images

Although the obtained results from image search can be helpful for pathologists, they may suffer from a bias. An approach could be subject to bias if the feature extractor is trained on special institutional datasets and potential hidden biases are not accounted for. This bias affects any other operation like segmentation, classification, and prediction. In recent work, Howard et al. [81] reported that the distribution of institutional data in the TCGA dataset, such as survival and gene expression patterns, remarkably differ among samples provided by various clinics and laboratories. They showed that usually, some models detect source sites instead of predicting prognosis or mutation states. In another research, DeGrave et al. [82] showed that the trained models on radiographic images are more likely to learn medically irrelevant shortcuts and are usually attributable to biases in data acquisition instead of the actual underlying pathology. Recently, Dehkharghanian et al. [80] showed that tissue source site (TSS) specific patterns of TCGA images could be used to identify contributing hospitals and institutions without any explicit training. In addition, they observed that a trained model for classification cancer subtype was able to discover such tissue source site-specific patterns within digital slides to classify cancer types. The factors such as digital scanner configuration and noise, tissue stain variation and artifacts, and source site patient demographics are more likely to account for the observed bias.

There have been many types of research to tackle the bias. These techniques that were proposed fall into one of these categories:

- Techniques that utilize data preprocessing before training
- In-processing during training
- Post-processing after training

In this research, we create an in-processing technique that helps us reduce the effect of bias during the training phase.

In the above parts, we talked about core concepts that we used in this thesis and overviewed Deep Learning and the network architecture that is utilized for histopathology feature learning. We showed the structure of the EfficientNet model, methods used in digital pathology, and concepts that affect histopathology images. Now we discuss how to create and prepare datasets for our EfficientNet model and implement our techniques on it to obtain a promising result.

Chapter 3

Data Preparation and Methods

In this chapter, firstly, we talk about the datasets and the methods that are proposed to prepare them for our network. Next, we investigate the EfficientNet network and its procedure training to obtain results.

3.1 Image Datasets

There are many public datasets in the area of histopathology, such as TCGA and CAMELYON17, that are used in various researches. The public datasets have some advantages, like the possibility of evaluating improvements and comparing with other methods [83], [84]. The dataset that is used in this research is TCGA. The TCGA repository is a publicly available repository that contains 30,072 WSIs [85]–[88]. These WSIs are obtained from 11,007 cases and depict primary sites with 32 cancer subtypes that can be seen in Table 3.1. Each case is associated with much information such as primary diagnosis, tissue of origin, morphology, patient age, tumor stage, race, and gender.

Table 3. 1: The codes of primary diagnoses in the TCGA dataset

#	TCGA Code	Primary Diagnosis
1	*ACC*	Adrenocortical Carcinoma
2	*BLCA*	Bladder Urothelial Carcinoma
3	*BRCA*	Breast Invasive Carcinoma
4	*CESC*	Cervical squamous cell carcinoma Endocervical adenocarcinoma
5	*CHOL*	Cholangiocarcinoma
6	*COAD*	Colon Adenocarcinoma
7	*DLBC*	Lymphoid Neoplasm Diffuse Large B-cell Lymphoma
8	*ESCA*	Esophageal Carcinoma
9	*GBM*	Glioblastoma Multiforme
10	*HNSC*	Head and Neck Squamous Cell Carcinoma
11	*KICH*	Kidney Chromophobe
12	*KIRC*	Kidney Renal Clear Cell Carcinoma
13	*KIRP*	Kidney Renal Papillary Cell Carcinoma
14	*LGG*	Brain Lower Grade Glioma
15	*LIHC*	Liver Hepatocellular Carcinoma
16	*LUAD*	Lung Adenocarcinoma
17	*LUSC*	Lung Squamous Cell Carcinoma
18	*MESO*	Mesothelioma
19	*OV*	Ovarian Serous Cystadenocarcinoma
20	*PAAD*	Pancreatic Adenocarcinoma
21	*PCPG*	Pheochromocytoma and Paraganglioma
22	*PRAD*	Prostate Adenocarcinoma
23	*READ*	Rectum Adenocarcinoma
24	*SARC*	Sarcoma
25	*SKCM*	Skin Cutaneous Melanoma
26	*STAD*	Stomach Adenocarcinoma
27	*TGCT*	Testicular Germ Cell Tumors
28	*THCA*	Thyroid Carcinoma
29	*THYM*	Thymoma
30	*UCEC*	Uterine Corpus Endometrial Carcinoma
31	*UCS*	Uterine Carcinosarcoma
32	*UVM*	Uveal Melanoma

Now we talk about the procedure of creating the datasets by Riasatian [12] and then explain the Grid method that we implemented to prepare datasets for EfficientNet. For creating a general dataset, some constraints are applied to TCGA dataset by Riasatian [12]. Because low quality leads to adverse effects on the network's learning process, some frozen section biopsy WSIs are filtered. This means that the dataset involves only permanent section biopsy WSIs. This is done by choosing the diagnostic slides option that can find under the experimental strategy bar on the GDC repository website [89]. To choose validation and test samples, the cases with only one whole slide image for the sake of simplicity of performance calculation are chosen. For each class of the dataset, cases that have a single WSI are separated. Then, they are shuffled, and two groups of them with a 10% size of that class are chosen in order to add to validation and test datasets. After this procedure, the test dataset has 744 slides that involved 10% of the data. In addition, the validation dataset has 741 slides that involve around 10% of the data. The rest of the data is assigned to the training set. The training set contained 7126 WSIs that involved 80% of the whole dataset. Table 3.2 shows the number of WSIs in each set.

Table 3. 2 : Dividing the WSIs by Riasatian [12]

	Training	**Test**	**Validation**
Number of WSI	7126	744	741

3.2 Patch Extraction

Since the WSIs are too large to be fed to the networks, images with small sizes should be extracted from WSIs. These small size images are called patches. The size of the patches chosen, 1000*1000 at 20x magnification, is the largest size that can be fed to a network considering available computational resources. Figure 3.1 shows sample images for the TCGA dataset.

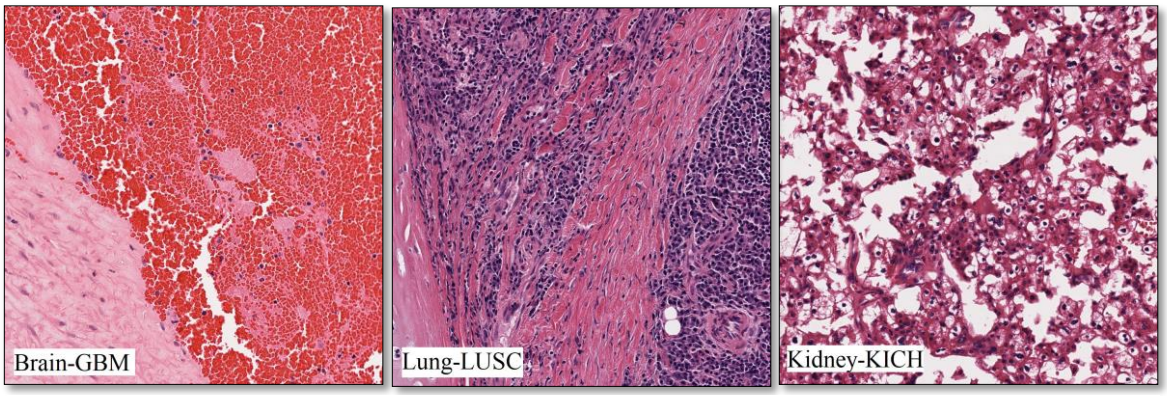


Figure 3. 1 : Sample patches from TCGA dataset. A simple patch for Brain Blioblastoma Multiforme (Right), Lung Squamous Cell Carcinoma (Middle), and Kidney Chromophobe(Left)

For patch extraction from the test WSIs, patches with 1x magnification are extracted for preprocessing. Firstly, the markers and then background pixels are removed from them. Finally, the patches are extracted by moving through the tissue background mask. During this moving, it is checked that 90 percent of the patch contains tissue and has less than 10 percent background. After this processing, around 116000 patches are extracted for the test dataset. This method is not suitable to extract patches for training and validation datasets due to the large number of WSIs in them. Therefore, Riasatian et al. [12] proposed another method for that. They firstly applied the mosaic generation of the

Yottixel search engine [68]. This algorithm partitions each WSI into nine different regions using a K-means algorithm based on color decompositions. Next, considering spatial diversity implemented by another k-means algorithm [90], 15% of the patches of each partition are extracted randomly. These patches can be a good idea to represent WSI with less amount of data. Since most images found in the TCGA dataset show high-grade carcinomas, the top 20 percent of the patches with respect to their cell nuclei amount is used by a nuclei segmentation function. Then they tried to calculate the cell nuclei ratio in each patch. To do so, first, they converted the color space to hematoxylin and eosin from RGB using color deconvolution. Then, the nuclei mask of the patch was obtained from the binarized hematoxylin channel by an empirical threshold, Figure 3.2. Next, they calculated the number of positive pixels in the nuclei mas divided by the patch area to obtain the cell nuclei ratio of each patch. In the end, the final dataset was created from the top 20 percent of sorted patches based on their cell nuclei ratio. The final training dataset and validation dataset include around 242000 and 24000 patches, respectively. Figure 3.3 shows extracted patches by the Yottixel algorithm.

To solve the problem of patch labeling, the tumor type information of the WSI that the patches are extracted from, was presented as a way. This way is a good solution instead of manually annotating the WSI, which only a pathologist can do and is a time-consuming and expensive process.

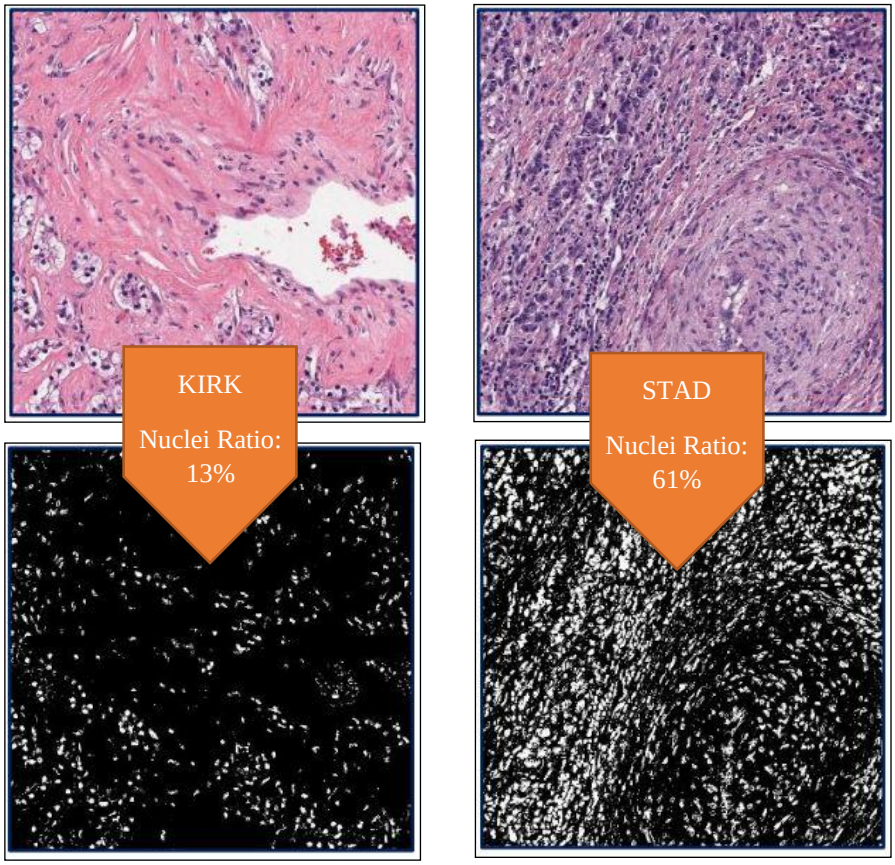


Figure 3. 2 : Two examples for cell nuclei Segmentation, Kidney Renal Papillary Cell Caricinoma(left), Stoch Adenocarcinoma(right)

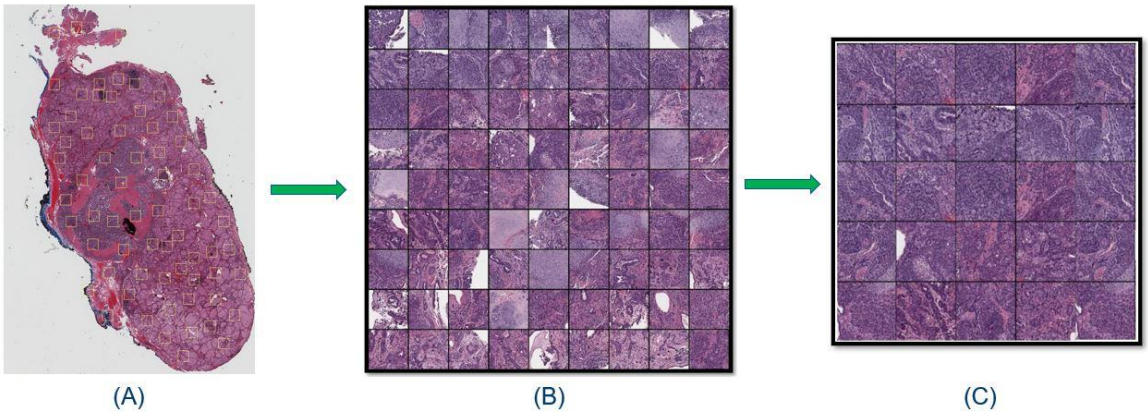


Figure 3. 3 : A WSI and its selected mosaic patches (A), Yottixel mosaic with 80 patches (B), modified cell Mosaic with 16 patches (C)

3.3 Creating Datasets

The discussed dataset consists of around 242000, 24000, and 11000 patches for training, validation, and test dataset. These patches are extracted at 20x magnification from 7126, 741, and 744 WSI that belongs to the TCGA repository, Table 3.3. A clustering-based mosaic method is used to assign a label to each patch based on its WSI. Now, we categorize them based on the types of cancer to create two datasets, the Brain and Lung.

Table 3. 3 : Number of WSIs and Patches in each set

Type	Patches	WSIs
Training	242000	7126
Test	24000	744
Validation	11000	741

3.3.1 Brain Dataset

To create a dataset for the brain, all the patches that belong to Brain cancer are extracted. The extracted patches are 34629, 8068, 1830 patches for training, test, and validation dataset. The training dataset of Brain cancer involves 1324 WSI that 750 of them are related to the first label, GBM, and 574 of them are related to another label, LGG. The test dataset involves 74 WSI that 35 of them are related to the first label, GBM, and 39 of them related to another label, LGG. The validation dataset involves 76 WSI that 36 of them are related to the first label, GBM, and 40 of them are related to another label, LGG, Table 3.4.

Table 3. 4 : Number of patches in Brain dataset

Type	GBM WSI	LGG WSI	Total WSI	Patches
Training	750	574	1324	34629
Test	35	39	74	8068
Validation	36	40	76	1830

3.3.2 Lung Dataset

The Lung dataset involves three types of cancer, LUAD, LUSC, and MESO. To create a dataset for it, all the patches that belong to Lung cancer are extracted. The extracted patches are 23321, 11535, 2758 patches for the training, test, and validation dataset. The training dataset of Lung cancer involves 719 WSI that 307 of them are related to the first label, LUAD, and 362 of them are related to the second label, LUSC, and 50 of them are related to the third label, MESO. The test dataset involves 86 WSI that 38 of them are related to the first label, LUAD, 43 of them related to the second label, LUSC, and 5 of them related to the third label, MESO. The validation dataset involves 84 WSI that 38 of them are related to the first label, LUAD, and 41 of them are related to the second label, LUAC, and 5 of them are related to the third label, MESO, Table 3.5.

Table 3. 5 : Number of patches in Lung dataset

Type	LUAD WSI	LUSC WSI	MESO WSI	Total WSI	Patches
Training	307	362	50	719	23321
Test	38	43	5	86	11535
Validation	38	41	5	84	2758

3.4 Grid Method for Preparing Datasets

A portion of data equal to batch size should be fed to the network for training, but there is a problem here. The size of images in datasets is 1000*1000, but the ImageNet weight is used during the Transfer learning. Since the size of images in the ImageNet dataset is 224*224, for getting the best results, the image size should be changed to 224*224. Now, there are two ways to handle this problem. The first way is to resize images from 1000*1000 to 224*224. Although this way helps us to solve the problem, it has a disadvantage. When we resize an image, we may lose some information and details that are necessary to diagnose cancer. The second solution is to crop the images. To do so, we create the Grid method.

3.4.1 Grid method for the training phase

First, we mesh each image into 25 parts with the size of 224*224. Then, we randomly choose one of them and feed the network in each epoch during the training phase, Figure 3.4. Finally, each randomly selected part is fed into a pre-trained EfficientNet for feature extraction. This solution helps us not only resize images without losing information but also cover most part of the original images to feed the network.

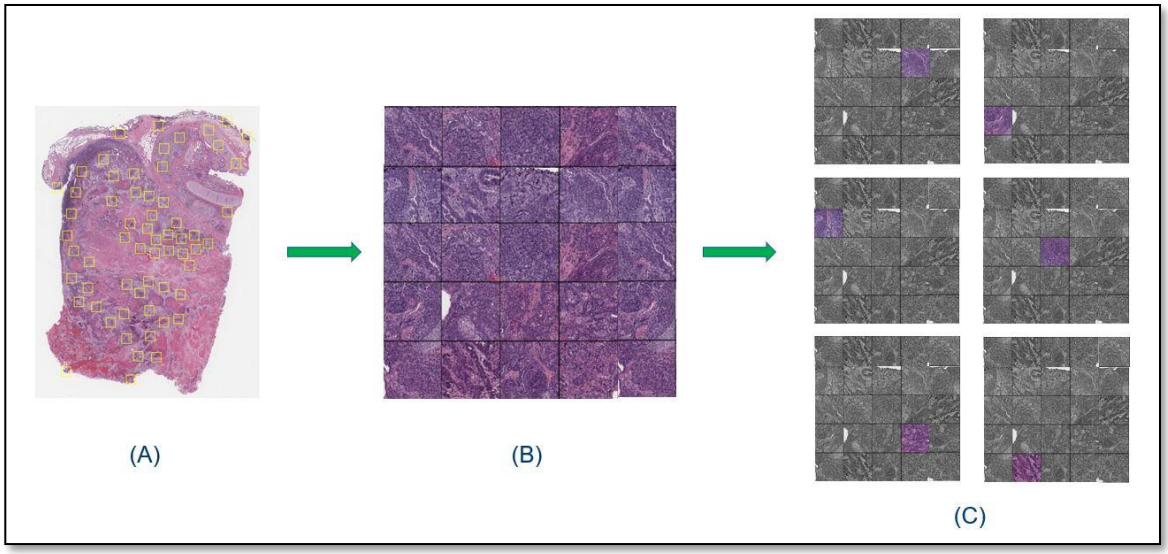


Figure 3. 4 : Patches from a whole slide image(left), create a mosaic from extracted patches (middle), randomly choose one part from 25 parts for feeding to the network (right).

3.4.2 Grid method for the test phase

In the test phase, we mesh each image into 25 parts with the size of 224*224. Next, instead of choosing a random part in the training phase, all the 25 parts of an image were fed to the network. We obtain 25 feature vectors and consider the average of these 25 vectors as a feature vector for the selected image.

Now, after creating two datasets and implementing the Grid method on that, the inputs are ready to feed to the network. We have this opportunity to train our models based on that and show how our model can be good for classifying different types of cancers (Chapter 4). Then in the next chapter (Chapter 5), we use these datasets to train a model with a new loss function that helps us classify cancers better. Finally, the effect of reducing bias to create an improved model on these datasets is shown (Chapter 6). Also, to better

evaluate the model in that experiment, we modify these datasets and use a part of them as external validation.

Chapter 4

Custom EfficientNet

This chapter investigates the custom EfficientNet network and its procedure training to obtain results. First, we talk about how to customize the EfficientNet, and then we discuss the training procedure in Section 4.1. This section talks about network details, input, and settings. Next, Section 4.2 explains the testing procedure. We show how to use test data to evaluate our model. Then, In Section 4.3, the results are analyzed. We show model results for each dataset and illustrate the advantages of our model compared with other models. At the end of this chapter, the Conclusion is stated in Section 4.4.

4.1 Training Custom EfficientNet

We trained the custom EfficientNet network with different settings to be able to compare our results with a state-of-art network, KimiaNet. As we can see in Table 2.1, we added four layers to the EfficientNet to customize that for our datasets. We added Batch Normalization (BN) [91] to normalize layers' inputs by re-centering and re-scaling. It helps the model to be faster and more stable. Because our model has tens of layers, its training is challenging as it can be sensitive to the configuration of the learning algorithm and the initial random weights. One of the reasons for this challenge is that the inputs distribution

of deep layers in the network may change after each mini-batch when the weights are updated. This reason can cause the learning algorithm to chase a moving target forever. Thus, we used batch normalization as a technique for the training model to standardize the inputs to a layer for each mini-batch. It helps the model train faster and dramatically reduces the number of training epochs required to train the model. In addition, we added dropout as a regularization technique to control overfitting [92]. One of the major aspects that should be considered during the training of machine learning models is avoiding overfitting. To prevent that, during training the custom EfficientNet, we used dropout as a regularization technique to discourage learning a more complex or flexible model to avoid the risk of overfitting. Moreover, we added three Fully Connected layers (FC) that the size of the last one is equal to the number of classes that we have. The number of classes for the Lung dataset is three (LUAD, LUSC, MESO), and for the Brain dataset is two (LGG, GBM). The Pytorch framework was utilized to implement the training and testing of the networks. The model was trained on one P100 GPU with 16GB memory. The size of the batch was set to 64. The epoch time for our model was around 20 minutes. Adam optimizer was used for our model with an initialized learning rate of 0.1. The model was initialized with pre-trained weights of the ImageNet. The input of the network was batches of 224×224 patches of 20X magnification, and one of the two classes for the brain and lung and four classes for the kidney was assigned to each patch as its label.

4.2 Experiments

For evaluating our network, we fed the network with all 25 parts of an image, generated using the Grid method. We obtained 25 feature vectors with the size of 1,280 from the feature extractor layer for each patch in our dataset. Then, we calculated the mean of these feature vectors and considered it as the representative for the selected patch. Figure 4.1 shows this process.

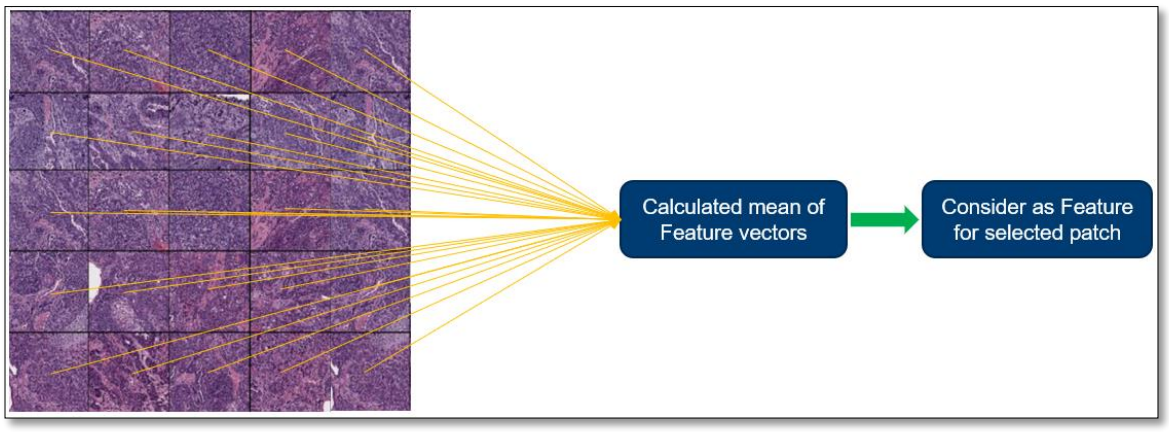


Figure 4. 1: Calculate the mean of 25 feature vectors and consider it as a feature vector for the selected patch

As mentioned previously, the extracted feature vectors are employed for image search to find the most similar images for a query WSI. However, to evaluate the performance of the search, the primary diagnosis labels are used. For this purpose, to predict the label for each WSI, we passed each patch feature vector to the leave-one-out method. In the leave-one-out method, we tried repeatedly iterating over all WSIs, taking one as the query WSI and the rest as the database. Firstly, we calculated the average of all patches features vectors that belong to one WSI. Then, we used the Euclidean distance to calculate the dissimilarity between the query WSI and the rest of the samples. Next, the three-Nearest Neighbours

method is applied to predict the label of query WSI. The label of the majority determined the label for the query WSI. Figure 4.2 shows an example of this experiment.

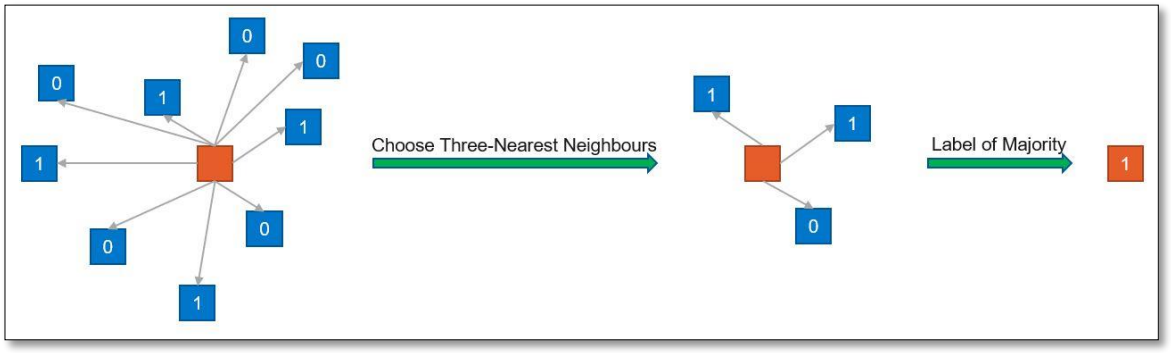


Figure 4. 2 : Find the label of WSI. Calculate the distance between the query WSI and the rest of the WSIs (left). Find the three-Nearest Neighbours of the query WSI (middle). Consider the label of the majority for the query WSI (right). Blue squares are WSIs feature vectors, and the orange square is the query WSI feature vector.

4.3 Analysis of Results

We trained our network for two datasets. In each network, we measure the model's performance in finding the WSI with a similar tumor subtype to the query WSI between all WSIs in the dataset (Slide-based Search). For instance, a query WSI is given with the subtype label "LUSC", which is a Lung tumor. Therefore, the algorithm is supposed to search between all WSIs in the lung dataset and suggest WSIs with the same class, namely LUSC.

To verify the performance of our model, we compared the results against two other models. The first model is the original DenseNet [55] trained with 1.2 million natural images from ImageNet [93]. The pre-trained DenseNet is utilized to extract the feature

vectors for histopathology images. The resulted feature vectors are in the size of 1,024 features. The second model is a state-of-the-art model in histopathology images, namely KimiaNet, that is based on the DenseNet architecture and is trained with a large number of histopathology patches [12]. Figure 4.3 represents the results of image search on two datasets obtained using the custom EfficientNet, KimiaNet, and DenseNet. We used F1-score [94] to evaluate the performance of our model instead of accuracy. Accuracy is not a good metric here because if our model is biased and classifies all WSIs of cancer to one label, the accuracy for that label would be 100%, whereas the accuracy for other labels is 0%.

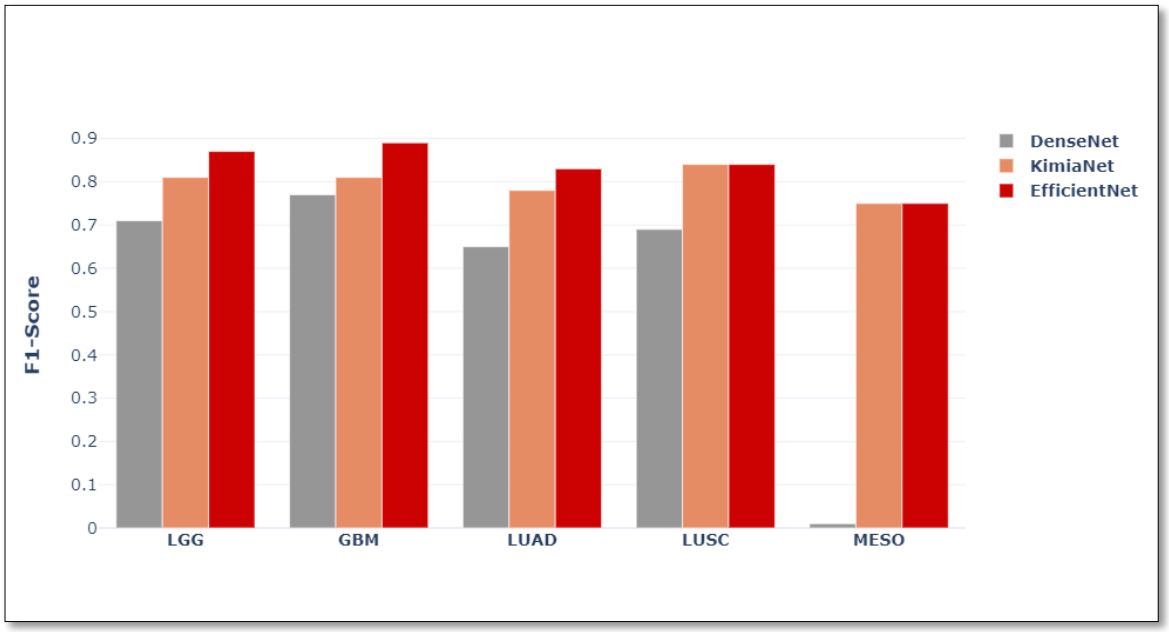


Figure 4. 3 : A comparison among the result of the search through representations generated by the DenseNet, KimiaNet and custom EfficientNet on 5 tumor subtypes. The two classes of the Brain dataset are LGG and GBM. The tumor subtypes of Lung dataset are LUAD, LUSC, MESO. The results show that in all classes the custom EfficientNet is better than or equal to KimiaNet and DenseNet

The results for each dataset are shown in the following tables. The results for the Lung dataset show that our model can improve the accuracy by 5%. Table 4.1 shows the F1-score for different sub-types of Lung cancer.

Table 4. 1 : The obtained results for Custom EfficientNet that was trained on the Lung cancer

Dataset	Measure	#Training	#Test	Sub-type	DenseNet	KimiaNet	Custom EfficientNet
Lung	F1-Score	22144	11041	LUAD	0.65	0.78	**0.83**
				LUSC	0.69	0.84	**0.84**
				MESO	0.00	0.75	**0.75**

In another experiment on the Brain dataset, our model performs well on both classes, and the results have significantly enhanced from 81% to 87% for LGG and from 81% to 89% for GBM. Table 4.2 shows the F1-score for two types of Brain cancer.

Table 4. 2 : The obtained results for Custom EfficientNet that was trained on the Brain cancer

Dataset	Measure	#Training	#Test	Sub-type	DenseNet	KimiaNet	Custom EfficientNet
Brain	F1-Score	34629	8068	LGG	0.71	0.81	**0.87**
				GBM	0.77	0.81	**0.89**

In addition to better accuracy, the size of input images is nearly 20 times smaller than that of KimiaNet. The reduced size of inputs makes our model very fast compared to competitors. Our proposed model requires just around 14 minutes for each epoch, whereas the state-of-the-art, KimiaNet, model took approximately 110 minutes for the training phase. Moreover, since the number of parameters was reduced to nearly 40%, fewer computational resources are needed to train our network. The proposed model demands 8 GB memory of a GPU while the state-of-art model used 4 GPUs with 128 GB memory for

training. Table 4.3 compares the number of parameters, epoch, batch-size, and the size of images for the DenseNet, KimiaNet and the custom EfficientNet.

Table 4. 3 : Compare the size of Custom EfficientNet and KimiaNet

Model	#Epoch	#Batch_size	Image Size	#Parameters	#Epoch time
DenseNet	20	64	1000*1000	7.1 Million	-
KimiaNet	20	64	1000*1000	7.1 Million	110 minutes
Custom EfficientNet	20	64	**224*224**	**4.3 Million**	**14 minutes**

These advantages, such as reducing parameters, reducing input size, and improving accuracy, provide this opportunity to classify histopathology images accurately and efficiently. Furthermore, it addresses some major challenges in digital pathology, such as high computational time and heavy computational resources.

4.4 Conclusion

In this chapter, we propose our model, custom EfficientNet, that helps us categorize cancers very well. We trained the model for each dataset and then evaluated it with the test dataset. During the evaluation, we calculate the distance between the query WSI and others and, based on that, predict the label for query WSI. To show the model's performance, we used the F1-score method and then compared our model with the competitors in this field. It can be seen that the custom EfficientNet can improve the accuracy and achieve better results with fewer parameters. In addition, due to reducing the size of input images, the custom EfficientNet can train much faster than KimiaNet. Generally, the custom EfficientNet is smaller and more robust compared to KimiaNet and DenseNet.

Although the custom EfficientNet helps us achieve good results, we propose a new method in the following chapters to train a model based on the objective of the network, image search, instead of classification and in addition tackle one of the major problems that can affect the results.

Chapter 5

Similarity Loss

This chapter explains the performed experiments to implement and evaluate the proposed new loss function, Similarity Loss (SL) that helps us implement search-based loss function instead of classification loss. The reasoning and intentions that led to the creation of this loss function are discussed in this chapter. We show the steps involved in creating this loss function as well as the results achieved after applying it to the training network. Section 5.1 explains the motivation. The motivation for this chapter comes from the desire to train the model based on the image search instead of the classification method because our objective from training the model is image search. Section 5.2 explains how to feed extracted patches to the network and discusses the implemented methods that help us reduce the size of the input and make the model training faster. Section 5.3 describes the proposed method and the different steps to create it. Then, in Section 5.4, experiments and quantitatively assess SL model performance are described. In this part, to illustrate the model performance, we compare the model with the state-of-art model in histopathology images. At the end of this chapter, conclusion is stated in Section 5.5.

5.1 Motivation

Image search has been one of the most important fields in computer vision for the past decade. It lets a client to search for a certain image and then retrieve related images from a large database. This approach is useful for medical and histopathological imaging and has a practical application in the real world. One of the benefits of image search is that it may take advantage of the rich hidden information included in the pixel values of photos without requiring any additional data. Since there is no additional information regarding the new biopsy sample in the clinical pathology setting, keyword-based searching to find diagnosis-relevant cases is not an option. Hence, the image search method can be used because it does not demand any external information, and the output of the search is determined based on the content of the images. As a result of this advantage, many researchers utilized the image search method [95]-[100]. These researches have a significant limitation. Their objective is image search, but they cannot train their network based on that. To be more precise, in terms of the lack of a technique to train networks for the goal of image search, researchers have to use alternative ways such as the classification method instead. Most of these research works goals are searching images in a repository, but their models cannot learn the parameters based on the primary goals. Hence, we propose a method that helps different researchers to train their model based on their objective, images search, instead of classification and open the door for us to customize that for other goals such as reducing the bias. This method, called Similarity Loss (SL), allows us to train a model based on the image search method.

5.2 Extract patches for feeding to network

WSI files are much larger than other types of medical images [101], and their size is more than 50K * 50K. As we discussed in Chapter 3, because the WSIs are too large, we should extract patches with small sizes from them to create the datasets. After creating datasets, they are ready to use for our models. In the first step, we choose one of the datasets to train our network. We utilize the Grid method and choose one part of each patch for feeding to the network. To be more precise, like the custom EfficientNet model, we mesh each image into 25 parts with the size of 224*224. Then, we randomly choose one of them and feed a pre-trained EfficientNet for feature extraction. The extracted features vector with dimension 1280 is passed to the SL function to calculate the loss value.

5.3 Proposed Method - Similarity Loss Function

This loss function helps researchers to train their models based on the image search method. During the training phase, in each epoch, it gets a batch of files one by one and returns the value of loss for that batch. Then this value is used for the backpropagation phase in the network. To be more specific, this SL function receives one input, i.e., the extracted patches The loss value is calculated using a similarity matrix built using the Cosine Similarity measure. The cosine similarity metric is used to determine how similar two vectors are, regardless of their size. By more details, it measures the cosine of the angle between the two vectors projected in a space with multi-dimension. At the first step to create a Cosine Similarity matrix, a matrix F with dimension batch size * 1280 was created by images

passed into the loss function. The matrix F includes the images feature vectors. At the next step, Cosine Similarity matrix S among the feature vectors is calculated:

$$S = F \cdot F \tag{1}$$

The matrix S with dimensions batch size * batch size includes the calculated similarity between the feature vectors. Since a vector has the most similarity with itself, the Cosine Similarity between each vector and itself should not be considered. To remove this similarity, we change the matrix diagonal to 0. Therefore, we create matrix S′ with diagonal 0 by the following product:

$$S' = S - \text{diag}(S) \tag{2}$$

The matrix S′ proposed the similarity between every two vectors but did not consider the similarity of each vector and itself and put 0 in that place.

In the next step, each row of S′ normalized to bring all the values in each feature vector to the same range and make the process less sensitive to the scale of values::

$$S' = Normalize(S')\,, dim = 1 \tag{3}$$

Since each row should normalize one by one, we put dim equal to 1. Then we build the prediction vector P in the size of batch-size to use for loss calculation. To predict a label for each image, the labels of other images are used, but with no equal effect. For predicting the query feature vector label, the most similar vector to our query feature vector has the greatest effect, while the least similar vector has the smallest effect. We utilize the similarity matrix and the actual label vector L to create a prediction vector. The vector L

includes the real labels of images in a batch, and it is in the size of batch-size too. To be more specific, as we mentioned, to predict a label for image *i*, we use other images that belong to that batch size. Whatever an image similar to *i*, it has more effect for predicting the label for i:

$$P = S' \cdot L \tag{4}$$

In the last step, the Mean Square Error (MSE) is used as the evaluation metric to obtain the loss value between the predicted labels and actual labels. MSE is the most commonly used loss function that measures the average of the squares of the errors, that is, the average squared difference between the actual and predicted labels. In this step, this metric is utilized to obtain the loss value between the true and predicted labels:

$$loss\ value = MSE(P\ , L) \tag{5}$$

Finally, after calculation of the loss value, it passes to the Stochastic Gradient Decent (SGD) for backpropagation and training of the weights. In this step, hyperparameters like learning-rate are tuned to help obtain better results. As mentioned, in this loss function, the more a feature vector is similar to the query input feature vector, the more it affects the result to predict the corresponding label, while the classification method only uses the query input label for prediction. Figure 5.1 shows the training process for SL function.

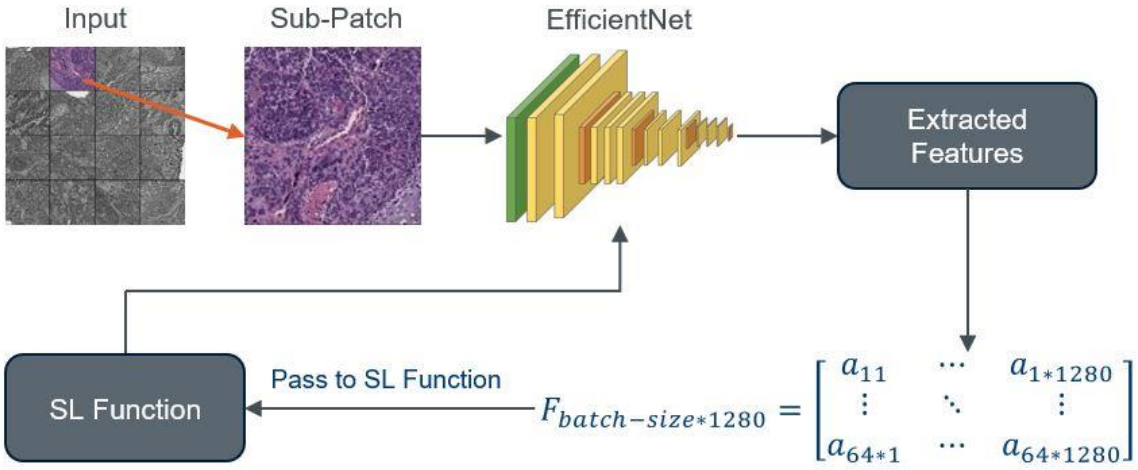


Figure 5. 1 : Overall structure of training process. Firstly, sub-patches are randomly extracted and fed to the EfficeintNet network. Then feature vectors are extracted and passed to the SL function to calculate the loss value. Finally, the RLL function returns loss value to the network for backpropagation.

5.4 Experiments

To show the efficiency of our proposed model, we evaluate it with two datasets and compare that with some other classification methods. To be more specific, we have some models with the same goal that is image search. The three of them are trained based on classification, and one of them is trained based on image search (proposed method). Now, to show the performance of the proposed model, we have done two experiences on both datasets from the TCGA repository,

1. 26021 Lung images from 81 slides
2. 36000 Brain Cancer images from 74 slides.

Both Lung and Brain datasets have two classes, and experiments have been done to find the actual class for each input.

5.4.1 Experiment Procedure

In this experiment, we evaluated the performance of our model without considering bias and compared our results with DenseNet, custom EfficientNet, and KimiaNet which is a famous network for classifying histopathology images [12]. Grid method is used to feed the images to the network in both the training and test phases of the proposed method. We first train the network with training data, and then feed the test data to the model to determine performance and results. After evaluating the model on test data, it can be seen that our proposed method achieves a state of art accuracy on both Lung and Brain datasets. Table 5.1 shows the F1-score results for four models, DenseNet, KimiaNet, custom EfficientNet, and our proposed model. Figure 5.2 compares the results between models for both Lung and Brain datasets.

Table 5. 1 : Compare results of four models, DenseNet, KimiaNet, Custom EfficientNet, and SL. It can be seen that in all the classes, the proposed model is better than others. The best result shows with green color.

Site	Subtype	n_{slides}	n_{Images}	DenseNet	KimiaNet	Custom EfficientNet	SL
Brain	LGG	39	36000	71	81	87	**91**
	GBM	35	36000	77	81	89	**92**
Lung	LUAD	38	26000	38	78	83	**84**
	LUSC	43	26000	43	84	84	**86**

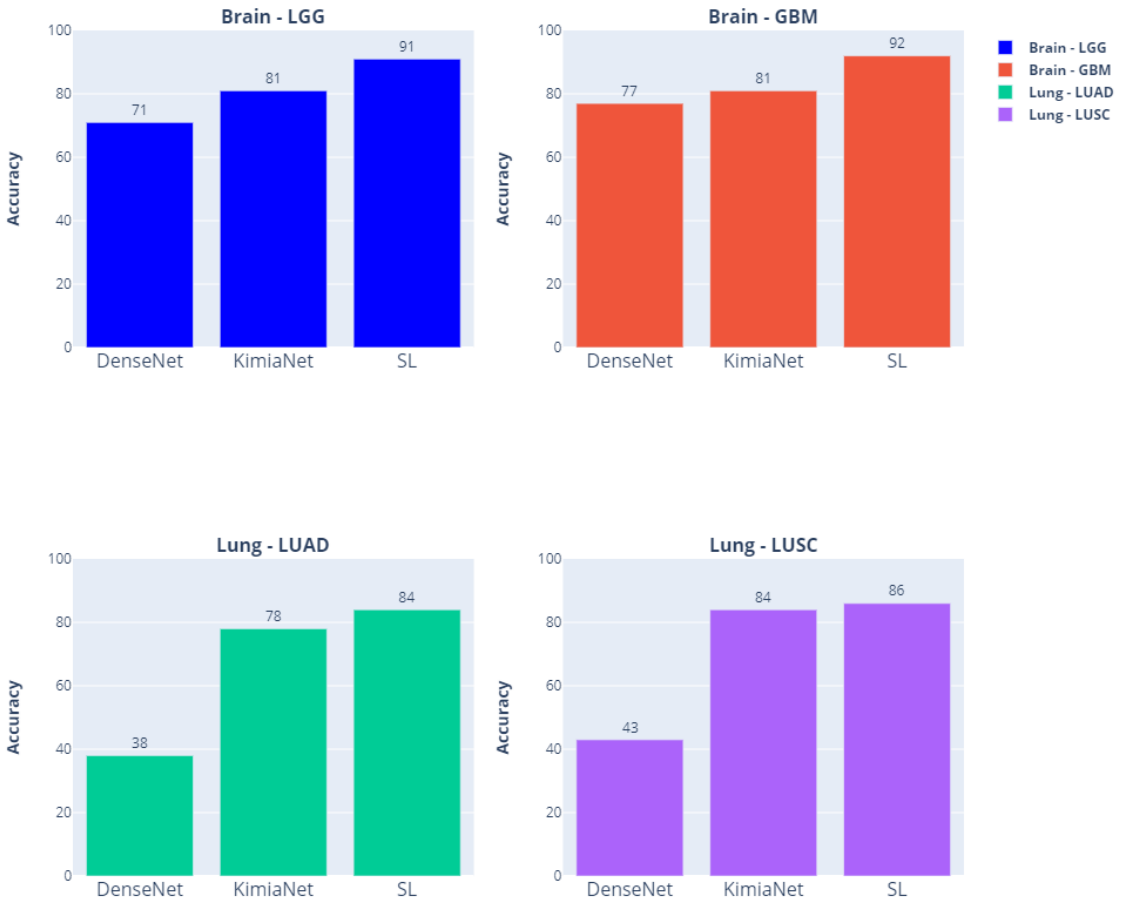


Figure 5. 2 : A comparison among the result of the search through representations generated by the DenseNet, KimiaNet and SL on 4 tumor subtypes. The two classes of the Brain dataset are LGG and GBM. The tumor subtypes of Lung dataset are LUAD, LUSC. The results show that in all classes the RLL is better KimiaNet and DenseNet.

In addition to better accuracy, the image size that is used for our model is nearly 20 times smaller than that of KimiaNet. The reduced size of inputs makes our model very fast compared to KimiaNet. Our proposed model requires just around 20 minutes for each

epoch, whereas the state-of-the-art, KimiaNet, model took approximately 110 minutes for the training phase.

Moreover, since the number of parameters was reduced to about 40%, fewer computational resources are needed to train our network. The proposed model demands 8 GB memory of a GPU while the KimiaNet model uses 4 GPUs with 128 GB memory for training. Our network benefits from these advantages and trains much faster than the other method. Table 5.2 shows the difference among the number of parameters, epoch, batch-size, and the size of images for the KimiaNet, and the proposed model.

Table 5. 2 : The number of parameters between KimiaNet and Our proposed model

Model	#Epoch	#Batch size	Image Size	#Parameters	#Epoch time
KimiaNet	20	64	1000*1000	7.1 Million	110 min
Proposed Model	20	64	**224*224**	**4.3 Million**	**20 min**

5.4.2 Analysis of results

We evaluated our model for both Lung and Brain datasets in the first experiment. To evaluate the model's performance on Lung dataset, we fed around 26000 images that belong to 38 and 43 LUAD and LUSC's slides to our network. The network was initialized with ImageNet weights and trained for 20 epochs with a batch size of 64. We could improve the accuracy to classify LUAD and LUAC near 6 and 2 percent compared to the state of art model KimiaNet. Besides, our model has much better accuracy near 46 and 43 percent for LUAD and LUAC than the DenseNet model. Moreover, our network is smaller than

the two other networks. In total, we can achieve more accuracy and fewer parameters compared with other models.

To evaluate the model on the Brain dataset, we fed around 36000 images that belong to 39 and 35 LGG and GBM's slides to the network The network parameter is identical to the one used in the Lung dataset model. For both types of the Brain dataset, we were able to improve accuracy by around 10% and 11%. Same as the Lung model, we achieved better accuracy and trained a model with fewer parameters.

Totally, our model performs better than other methods because it focuses on the final goal of the network, which is image search. This focus allows the suggested model to obtain superior outcomes despite the fact that it is smaller, has fewer parameters, and was trained with 25 times smaller images. In fact, we have a network that has the following characteristics but has more accuracy because it trains the network using the image search method.

- 40% smaller than the state-of-art model
- It is fed with 25 times smaller images that make the model much faster than others.
- Achieve better results to classify cancers better than the state-of-art model

5.5 Conclusion

The purpose of current chapter was to implement a model based on the images search. The model helped us be independent of the classification method for searching images. The results support the idea that the model based on image search can be more efficient for

searching an image in a dataset. It shows that a model focusing on the image search for training the network can achieve better results, faster training, and less computation. This model can be used in future study to help researchers reach their image search goals by reducing computation and training time and achieving good results.

After proposing this model, we try to use another ability that this model provides to achieve better results. We attempt to use it to reduce the effect of bias that greatly impacts the obtained results. Since some datasets suffer from this problem, we try to make our method more applicable to prevent and reduce it in the next chapter.

Chapter 6

Segregation Similarity Loss

This chapter explains the performed experiments to implement and evaluate the proposed Segregation Similarity Loss (SSL). It is discussed the motivation and goals which lead to creating another type of Similarity Loss. Moreover, we show the difference between these two types of loss functions and details of creating the new version. In addition, we show the obtained results from implementing it on many external validation datasets. Section 6.1 explains the motivation. The motivation for this chapter comes from the desire to train the model that helps us reduce the bias during the training phase. Section 6.2 shows how to feed extracted patches to the network and explains the implemented Grid method to reduce the input size and make model training faster. Section 6.3 provides information about the bias label matrix and how we create it using one-hot vectors. Section 6.4 describes the proposed method and the steps to create it. Then, In Section 6.5, experiments and quantitatively assess SSL model performance are described. At the end of this chapter, conclusion is stated in Section 6.6.

6.1 Motivation

The image search method, as explained in the previous chapter, is critical in histopathology imaging. Image search is advantageous because it may extract useful hidden information from image pixel values without requiring any further information. In previous chapter, to use the advantages of this method, we proposed a new loss function, Similarity Loss (SL), to train the model based on this method. In this chapter, we propose another type of SL function that helps us reduce the bias in the dataset. Based on some recent studies [70], TCGA dataset suffers from bias. This internal bias originates from the hospitals and institutions that contributed WSIs to the TCGA dataset. The models that train based on the classification loss function are not able to prevent this bias during the model training phase and reduce the effect of that on the results. To address this problem, we propose a new loss function, Segregation Similarity Loss (SSL), that can help us reduce bias and train a model based on the image search method. To be more precise, using SSL, we can reduce bias during the training phase and prevent its effect of it on the results.

6.2 Extract patches for feeding to network

Since WSIs are too large, we should find a way to feed them to the network. So, we extract patches of small sizes from them and use them for training our model. Then, same as the previous chapter, we utilize the Grid method and choose one part of each patch to feed the network. We mesh each image into 25 parts with the size of 224*224. Then, we randomly choose one of them and feed a pre-trained EfficientNet for feature extraction. The extracted

features vector with dimension 1280 is passed to the SSL function for calculating loss value. The process until this step is the same as the SL function, but now we have a Bias Label Matrix that helps us prevent conflict of interest voting.

6.3 Bias Label Matrix

There are two inputs to the SSL function. The extracted feature from patches is the first, while the Bias Label Matrix is the second. The TCGA dataset suffers from internal bias emanating from the institutions that submitted WSIs to the TCGA dataset, according to research [80], [81]. Since the feature extractor is trained on specific institutional datasets and potential hidden biases are not accounted for, we create a Bias Label Matrix that involves hospitals' information which images belong. So, we change the hospital label to a one-hot vector. Since we have 10 and 12 hospitals in Brain and Lung datasets, the size of this one hot vector for the Brain and Lung dataset is 10 and 12, respectively. For each batch 64, we obtain a Bias Label Matrix B with dimensions 64(batch-size) * 10 for Brain and 64 (batch-size) *10 for the Lung dataset. The number 64 for batch size chosen by users can be changed based on their memory and resources. After creating the matrix, we pass it to the SSL function. Figure 6.1 illustrates an example for matrix (B) and (1-B) which is used in the next step.

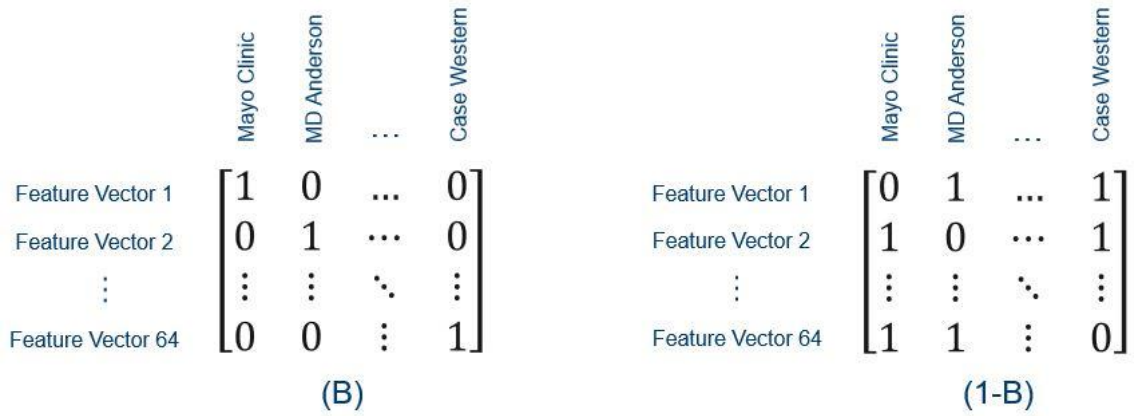


Figure 6. 1 : The Bias Label Matrix B and the inverse Bias Label Matrix (1-B). The columns show the hospitals, and the rows show the feature vectors. The number of rows is equal to batch-size, and the indexes show the feature vector belongs to a specific hospital or not.

6.4 Segregation Similarity Loss Function

This SSL function receives two inputs, Bias Label Matrix and extracted patches. For calculating loss value, a matrix F with dimension batch size * 1280 is created by images passed into the loss function. At next step, a similarity matrix S among the feature vectors is calculated as:

$$S = F \cdot F \tag{1}$$

To avoid the bias effect, images from the same hospital should not be able to predict labels for each other. To be more specific, during the training phase, photos from the same hospital should not be allowed to vote for each other. To implement that, the similarity value in similarity matrix S between two images from a specific hospital changes to 0. Then, the matrix S´ which prevents bias is created by the following product:

$$S' = S \cdot (1 - B) \quad (2)$$

In the next step, each row of S´ normalized.

$$S' = Normalize(S')\,, dim = 1 \quad (3)$$

Then we build the prediction vector P with dimension batch-size * 1 to use for loss calculation. For creating a prediction vector, we use the similarity matrix and actual label vector L. To be more specific, to predict a label for image *i*, we use other images that belong to that batch size. The more an image is similar to *i*, the more effect it has for predicting the label for *i*.

$$P = S' \cdot L \quad (4)|$$

In the last step, Mean Square Error is used as the evaluation metric to obtain the loss value between predicted labels and actual labels:

$$loss\ value = MSE(P\,, L) \quad (5)$$

The loss value passes to the optimizer, Stochastic Gradient Descent, for backpropagation and weights training.

6.5 Experiments

In the previous section, the process of creating the SSL function was discussed. It showed that creating this function aims to prevent conflict of interest voting. Now, to show the efficiency of SSL, we evaluate it with 17 and 12 external validation datasets created from Lung and Brain datasets. For creating each training dataset, we separate the data from a

specific hospital and use the test data of the specific hospital as test data. Then we compare each model's performance with other methods. All models have two classes for both Lung and Brain datasets, and experiments were done to find the actual class for each input.

6.5.1 Experiment Procedure

In the second experiment, we evaluated the performance of our model for both types of our loss function and classification model using external validation. We utilized the Grid method, same as the other experiments, to train and evaluate our model. For the SSL function in the training phase, we first segregated each hospital's images from the dataset and used that hospital's test data as external validation to show network performance. In the Brain and Lung datasets, we've done this for all hospitals. We collected all hospitals with one WSI and got them as an external validation dataset to cover all the hospitals. We trained three different networks. First, we trained our model for each hospital based on the classification loss. Then we trained our network based on the SL method without considering bias. Finally, we trained our model for each hospital with the SSL method and tried to prevent the effect of bias during training the network. Figure 6.2 and Figure 6.3 illustrate the performance plots for three models on four Lung and Brain datasets, respectively.

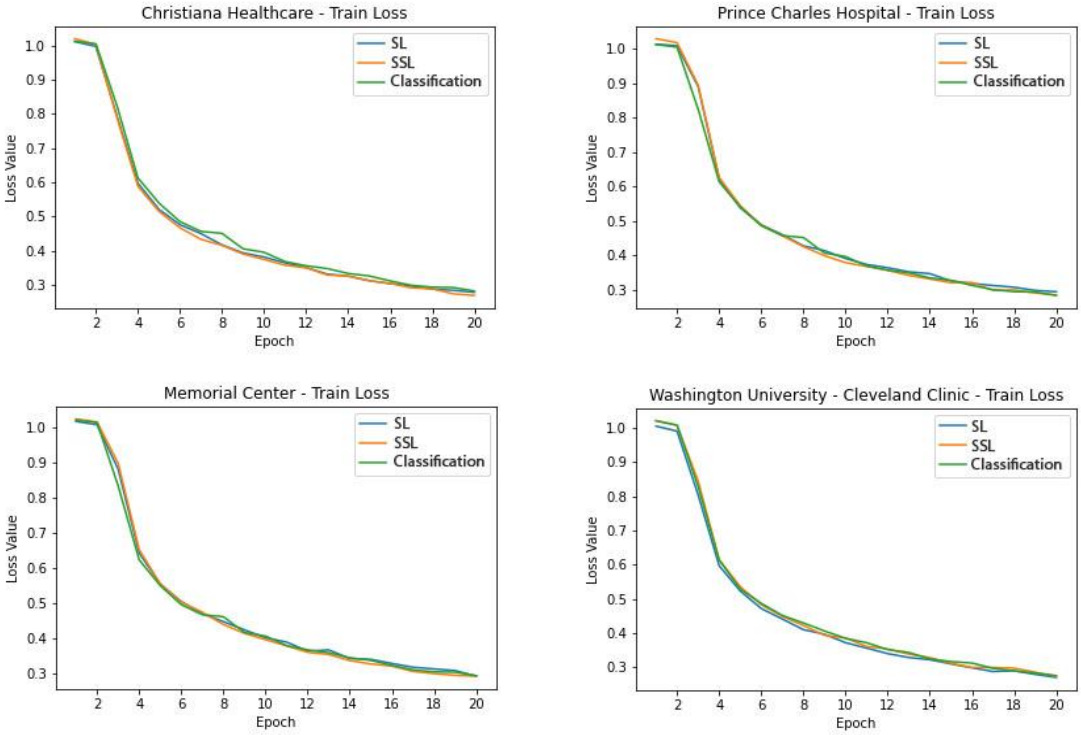


Figure 6. 3 : Performance plots for 4 example the Lung dataset

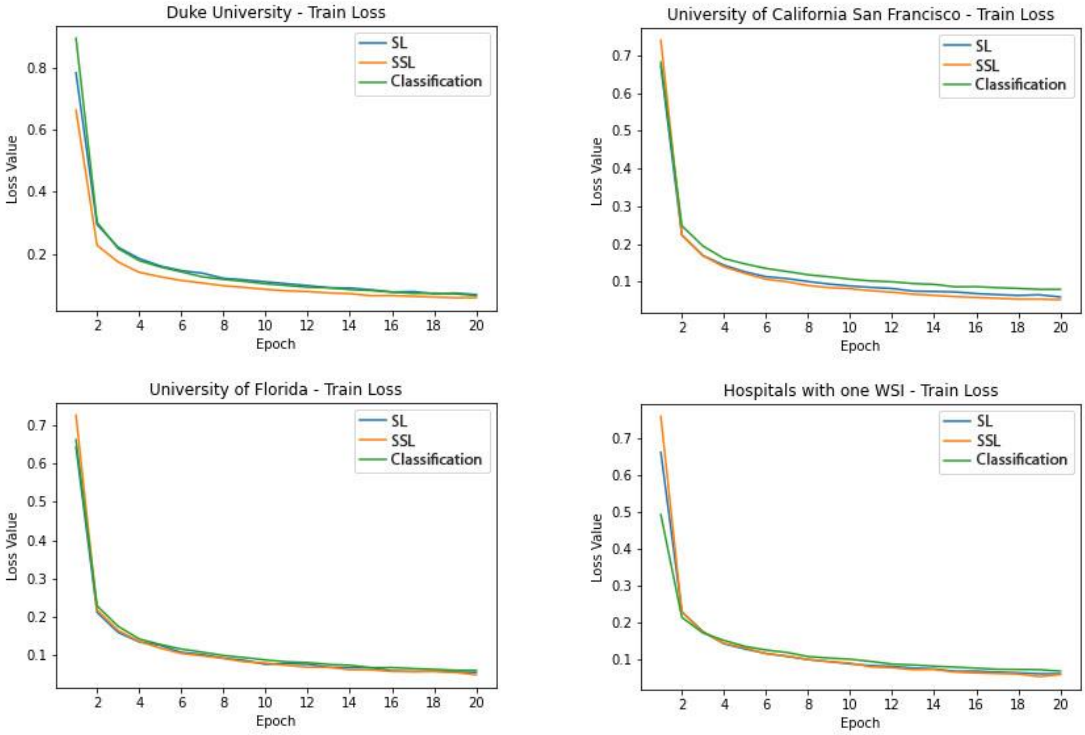


Figure 6. 2 : Performance plots for 4 example the Brain dataset

6.5.2 Analysis of results on the Lung dataset

To evaluate our model, SSL, first, we tested the model trained for Lung dataset. We fed the test images of one specific hospital that we did not use in the training phase as external validation. We have done that for 17 different models and compared the results with the classification model and SL with bias. The results show that our model performances in the 11 and 7 experiments are equal to or better than two other methods, the SL and classification model. Table 6.1 shows the obtained results for 17 hospitals on the Lung dataset.

Table 6. 1 : Compare the F1-score for 17 hospitals on the Lung dataset. In each dataset, the mentioned hospital data is separate from training data, and the mentioned hospital test data use for validation

#	Hospitals	N_{WSI}-Test set	Classification	SL	SSL
1	International Genomics Consortium	16	(0.67 , 0.57)	**(0.78 , 0.71)**	(0.59 , 0.53)
2	Indivumed	10	**(0.91 , 0.89)**	(0.77 , 0.57)	(0.83 , 0.75)
3	Christiana Healthcare	8	(0.60 , 0.33)	(0.80 , 0.67)	**(0.91 , 0.80)**
4	Asterand	9	(0.50 , 0.86)	**(1.00, 1.00)**	(0.00 , 0.94)
5	Mayo Clinic Rochester	5	(0.00 , 0.89)	(0.00 , 0.75)	**(1.00 , 1.00)**
6	Washington University - Alabama	3	**(1.00 , 1.00)**	**(1.00 , 1.00)**	(0.00, 0.80)
7	Roswell Park	3	**(0.67 , 0.67)**	(0.00, 0.50)	(0.67 , 0)
8	Ontario Institute for Cancer Research	3	(0.67 , 0.67)	**(1.00 , 1.00)**	(0.67 , 0.67)
9	University of North Carolina	2	(0.00 , 0.00)	(0.00 , 0.00)	(0.00 , 0.00)
10	Prince Charles Hospital	2	(1.00 , 1.00)	(1.00 , 1.00)	(1.00 , 1.00)
11	University of Pittsburgh	2	(0.67 , 0.00)	(0.67 , 0.00)	(0.67 , 0.00)
12	Washington University - Emory	4	(0.50 , 0.50)	**(0.67 , 0.80)**	(0.00 , 0.40)
13	Memorial Sloan Kettering Center	2	(0.00 , 0.67)	**(0.00 , 1.00)**	(0.00 , 0.67)
14	Washington Uni - Cleveland Clinic	2	(0.67 , 0.00)	(0.67 , 0.00)	**(0.00 , 1.00)**
15	Thorax klinik at Uni Hospital	2	**(0.67 , 0.00)**	(0.00 , 0.00)	(0.00 , 0.00)
16	Candler	2	(0.00 , 0.67)	(0.00 , 0.67)	(0.00 , 0.67)
17	Hospital with one WSI	6	(0.67 , 0.89)	(0.67 , 0.89)	(0.67 , 0.89)

In the next step, we obtained the number of WSI that each model can predict correctly for each hospital. We summed these numbers and calculated the accuracy for each model. SSL method can improve the accuracy by near 5% compared to the two other methods. In other words, we calculated the number of WSIs that each model predicts correctly from all WSIs that we have in the selected dataset. SSL shows its efficiency by improving the results by near 5% compared to other methods. It showed that reducing the effect of bias on the model can help us improve the model's ability for prediction. Table 6.2 and Figure 6.4 show the results and compare models.

Table 6. 2 : Obtained accuracy for each model on the Lung dataset

Model	Classification	SL	SSL
Accuracy	67.90%	67.90%	**72.80%**

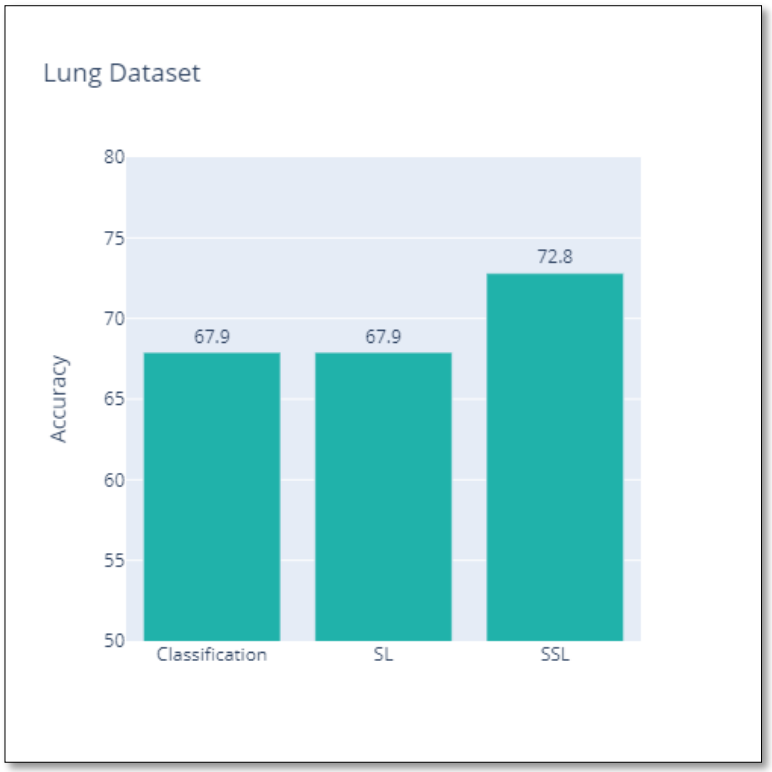


Figure 6. 4 : Compare the accuracy for each model on the Lung dataset. It can be seen that accuracy of the Lung dataset model experience near 5 percent improvement with using the SSL function

6.5.3 Analysis of results on the Brain dataset

At the second step, we tested the SSL model trained on the Brain dataset. Similar to the search on Lung cancer images, our model receives the test images of one specific hospital and outputs two classes for GBM or LGG. We evaluated 12 different models with this dataset and compared the results to other approaches. We were successful in 10 of the 12 experiments on external validation datasets, achieving equal or greater accuracy. Table 6.3 shows the obtained results for 12 hospitals on the Brain dataset.

Table 6. 3 : Compare the F1-score for 12 hospitals on the Brain dataset

#	Hospitals	N_{WSI}-Test set	Classification	SL	SSL
1	University of Sao Paulo	2	(0.00 , 1.00)	(0.00 , 1.00)	(0.00 , 1.00)
2	Henry Ford Hospital	15	(0.57 , 0.63)	(0.50,**0.67**)	(**0.62** , 0.57)
3	Case Western - St Joes	6	(0.00 , 0.80)	**(0.00 , 1.00)**	**(0.00 , 1.00)**
4	Duke University	1	(0.00 , 1.00)	(0.00 , 1.00)	(0.00 , 1.00)
5	Dept of Neurosurgery at UofH	6	(0.00 , 0.67)	**(0.00, 0.80)**	**(0.00 , 0.80)**
6	University of Florida	7	(0.75 , 0.67)	**(0.86 , 0.86)**	(0.75 , 0.67)
7	Mayo Clinic	1	**(0.00 , 1.00)**	**(0.00 , 1.00)**	(0.00 , 0.00)
8	MD Anderson	7	(0.60 , 0.00)	(0.73 , 00)	**(1.00 , 1.00)**
9	Uni of California San Francisco	7	(0.80 , 0.50)	(0.80 , 0.50)	(0.80 , 0.50)
10	Case Western	11	(0.80 , 0.57)	**(0.93 , 0.86)**	(0.80 , 0.57)
11	Memorial Kettering Center	3	(0.00 , 0.80)	**(0.00 , 1.00)**	(0.00 , 0.50)
12	Hospital with one WSI	8	**(**0.00 **,** 1.00)	(0.00 **, 1.00)**	(**0.75** , 0.75)

In the next step, we obtained the number of WSIs that each model can predict correctly for all 12 hospitals. We summed these numbers and calculated the accuracy for each model. SSL method achieved the best accuracy compared with other methods. It improves the results by 6% compared to the two other methods. Same as the results obtained on the Lung dataset, we can see that reducing the effect of bias on the model can help us enhance the model's ability for prediction. Table 6.4 and Figure 6.5 show the results and compare models.

Table 6. 4 : Obtained accuracy for each model on the Brain dataset

Model	Classification	SL	SSL
Accuracy	71.60%	70.27%	**79.70%**

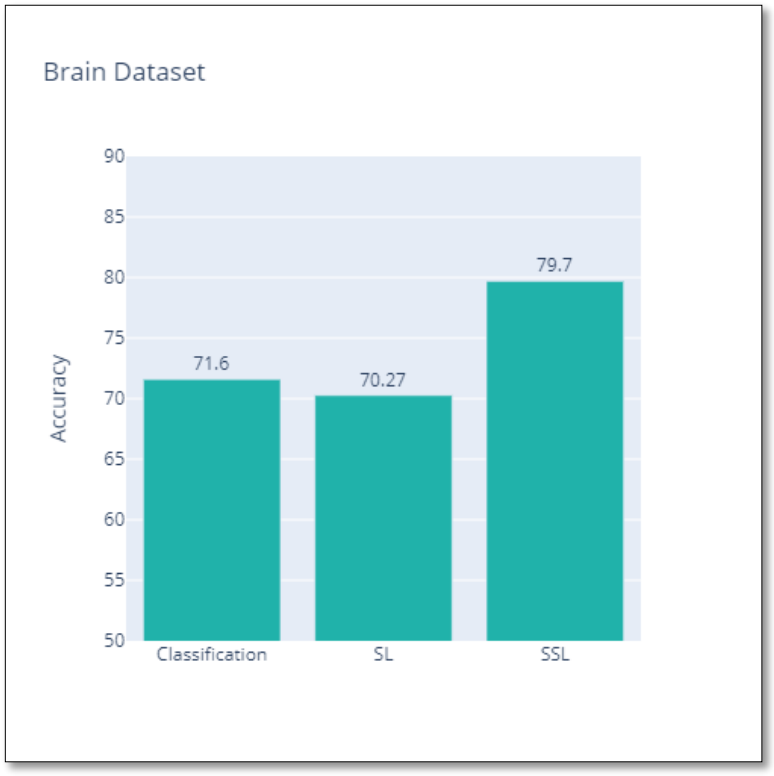


Figure 6. 5 : Compare the accuracy for each model on the Brain dataset. It can be seen that accuracy of the Brain dataset model experience near 8 percent improvement with using the SSL function

6.6 Conclusion

The purpose of the current chapter was to implement a model based on the images search that helped us be independent of the classification method for searching images and allowed us to prevent the bias effect of desired elements during the training network. To

do a strong evaluation on our method, many external validation data were employed. The results support the idea that the model based on image search can be more efficient for searching an image in a biased dataset.

Chapter 7

Summary and Conclusion

Digital histopathology image analysis is a new field because proper storage, transfer, process, and visualization of digital histopathology images were not available until the last century due to the high resolution and complexity of those images. However, it is an active research field because histopathology plays a crucial role in diagnosing, studying, and treating diseases such as cancer. One of the techniques that plays a crucial role in digital pathology is image search. This technique in extensive archives of digital pathology slides provides an opportunity for researchers to match records of past and current patients and learn from evidently diagnosed and treated cases. To implement this technique, a state-of-the-art network, EfficientNet, and a method to prepare the data for training the model are developed in this thesis.

EfficientNet is a neural network architecture and scaling method that can uniformly scale a network's depth, width, and resolution using a compound coefficient. This new baseline network achieves much better accuracy and efficiency on the ImageNet while being smaller and faster on inference than other convolutional

networks. Same as training other deep neural networks, there is a big challenge to train EfficientNet with WSIs. Since WSIs have gigapixel size, they cannot be processed by the networks that existed. To address this problem, a mosaic approach is applied to WSIs to divide them into smaller parts called patches. Then, since pre-trained EfficientNet can have better performance when the size of input images is the same as the ImageNet images' size, we proposed a new method, i.e., the random Grid method, that enabled us to reduce the size of the inputs (i.e., 20 times) and make the training procedure very fast. After using the Grid method, a network-based classification method called custom EfficientNet is trained with the images.

In the first experiment, we trained a custom EfficientNet model on different cancer training sets that are created from the TCGA repository. The model is evaluated on two tumor types using the image search in digital pathology. The results are compared with those from the state-of-the-art models specialized for histopathology images. From the results, it can be seen that the custom EfficientNet is more accurate and compact compared to competitors and capable of training with a smaller dataset. While most of the existing deep neural networks in digital pathology require excessive computational time and expensive resources, the proposed model reduces the computational complexity significantly. Despite achieving better accuracy with custom EfficientNet, we developed a new method, Similarity Loss Function (SL), to train a network based on our primary objective.

During the last years, image search has become one of the important techniques in computer vision. This technique involves finding images and patches

that share the same visual characteristics as the query image. Identification and analysis of the same images can help pathologists to find a diagnosis by making a baseline for comparison. Generally, the image search technique can assist pathologists in diagnosis quickly and accurately. These advantages change the image search to a popular method in histopathology images. Despite these beneficial features, there is no method to train the model based on the image search, and researchers have to use the classification method for training the model and utilize it for searching images. In the second experiment, we tried to solve this problem and developed a technique, Similarity Loss (SL), that helps us train a model based on the primary objective, images search. During the evaluation of this model, it was compared with three other methods and showed promising results. It can improve the accuracy near 10 and 4 percent for the Brain and Lung datasets compared with the state-of-art model, KimiaNet. In addition, it is size near 40 percent, and the size of the input images to feed this network is 25 times smaller than the KimiaNet. Totally, better accuracy and faster training make this work a well-designed technique. Our work does not stop here, and we utilized this technique to solve one of the major problems in this field, bias.

In the last work, we propose another type of SL function that helps us to reduce the bias in the dataset. Based on recent researches, the TCGA dataset suffers from bias originating from the hospitals and institutions that contributed WSIs to the TCGA dataset. To address this problem, we propose a new loss function, Segregation Similarity Loss (SSL), that can help us to reduce bias and prevent the effect of that on

the results. This function works the same as the SL function, but it prevents conflict of interest voting during the training phase by using a Bias Label Matrix created from institutions. It only allows the model to predict the label of a query image based on the same images from other hospital images. The results show a near 8 and 5 percent improvement compared with the state-of-the-art model. Same as the SL function, the SSL function also has 40 percent fewer parameters and is fed with 25 times smaller images, making this technique faster. Totally, this investigation shows that the SSL technique can help us prevent the effect of bias on the result and achieve better accuracy, predict the labels more accurately, and train the model much faster.

Future work can include applying the proposed SL method in other archives of WSIs in digital pathology. In addition, the SSL technique can be used for different datasets to generalize the methods better with reducing different elements bias. Moreover, the future direction of achieving more reliable and accurate results might be driven by discovering the source of bias in different databases.

References

[1] E. Brender, A. Burke, and R. M. Glass, “Frozen section biopsy,” *Jama*, vol. 294, no. 24, p. 3200, 2005.

[2] R. Levenson, “Histopathology is ripe for automation.” NATURE PUBLISHING GROUP MACMILLAN BUILDING, 4 CRINAN ST, LONDON N1 9XW, ENGLAND, 2017.

[3] “The American Cancer Society,” 2022. https://www.cancer.org (accessed Jan. 15, 2022).

[4] “The American Society of Clinical Oncology (ASCO).” Cancer.net (accessed Jan. 15, 2022).

[5] N. Hegde *et al.*, “Similar image search for histopathology: SMILY,” *NPJ digital medicine*, vol. 2, no. 1, pp. 1–9, 2019.

[6] A. BenTaieb and G. Hamarneh, “Deep learning models for digital pathology,” *arXiv preprint arXiv:1910.12329*, 2019.

[7] M. N. Gurcan, L. E. Boucheron, A. Can, A. Madabhushi, N. M. Rajpoot, and B. Yener, “Histopathological image analysis: A review,” *IEEE reviews in biomedical engineering*, vol. 2, pp. 147–171, 2009.

[8] H. Y. Chang *et al.*, “Artificial intelligence in pathology,” *Journal of pathology and translational medicine*, vol. 53, no. 1, p. 1, 2019.

[9] H. R. Tizhoosh and L. Pantanowitz, “Artificial intelligence and digital pathology: challenges and opportunities,” *Journal of pathology informatics*, vol. 9, 2018.

[10] A. BenTaieb and G. Hamarneh, “Deep learning models for digital pathology,” *arXiv preprint arXiv:1910.12329*, 2019.

[11] A. Janowczyk and A. Madabhushi, “Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases,” *Journal of pathology informatics*, vol. 7, 2016.

[12] A. Riasatian *et al.*, “Fine-Tuning and training of densenet for histopathology image representation using TCGA diagnostic slides,” *Medical Image Analysis*, vol. 70, p. 102032, 2021.

[13] M. Tan and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” in *International Conference on Machine Learning*, 2019, pp. 6105–6114.

[14] F. Rosenblatt, “Principles of neurodynamics. perceptrons and the theory of brain mechanisms,” Cornell Aeronautical Lab Inc Buffalo NY, 1961.

[15] J. J. Hopfield, “Neural networks and physical systems with emergent collective computational abilities,” *Proceedings of the national academy of sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

[16] “History of neural network.http://cs.stanford.edu/people/eroberts/ courses/soco/projects/neural-networks/History/history2.html.”

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[18] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[19] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, “Return of the devil in the details: Delving deep into convolutional nets,” *arXiv preprint arXiv:1405.3531*, 2014.

[20] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural networks*, vol. 61, pp. 85–117, 2015.

[21] S. J. Nowlan and G. E. Hinton, “Simplifying neural networks by soft weight-sharing,” *Neural computation*, vol. 4, no. 4, pp. 473–493, 1992.

[22] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International conference on machine learning*, 2015, pp. 448–456.

[23] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” 2010.

[24] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[25] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning internal representations by error propagation,” California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[26] M. A. Kramer, “Nonlinear principal component analysis using autoassociative neural networks,” *AIChE journal*, vol. 37, no. 2, pp. 233–243, 1991.

[27] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[28] C. Szegedy *et al.*, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[29] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, “Recurrent neural network based language model.,” in *Interspeech*, 2010, vol. 2, no. 3, pp. 1045–1048.

[30] A. Graves, A. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in *2013 IEEE international conference on acoustics, speech and signal processing*, 2013, pp. 6645–6649.

[31] J. J. Hopfield, “Neural networks and physical systems with emergent collective computational abilities,” *Proceedings of the national academy of sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

[32] K. Funahashi and Y. Nakamura, “Approximation of dynamical systems by continuous time recurrent neural networks,” *Neural networks*, vol. 6, no. 6, pp. 801–806, 1993.

[33] R. J. Williams and D. Zipser, “A learning algorithm for continually running fully recurrent neural networks,” *Neural computation*, vol. 1, no. 2, pp. 270–280, 1989.

[34] T. Mikolov, S. Kombrink, L. Burget, J. Černocký, and S. Khudanpur, “Extensions of recurrent neural network language model,” in *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2011, pp. 5528–5531.

[35] M. Liang and X. Hu, “Recurrent convolutional neural network for object recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3367–3375.

[36] P. Zhou *et al.*, “Attention-based bidirectional long short-term memory networks for relation classification,” in *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)*, 2016, pp. 207–212.

[37] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[38] H. Sak, A. W. Senior, and F. Beaufays, “Long short-term memory recurrent neural network architectures for large scale acoustic modeling,” 2014.

[39] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, “Face recognition: A convolutional neural-network approach,” *IEEE transactions on neural networks*, vol. 8, no. 1, pp. 98–113, 1997.

[40] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[41] P. Y. Simard, D. Steinkraus, and J. C. Platt, “Best practices for convolutional neural networks applied to visual document analysis.,” in *Icdar*, 2003, vol. 3, no. 2003.

[42] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural networks*, vol. 61, pp. 85–117, 2015.

[43] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale video classification with convolutional neural networks,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[44] T. N. Sainath *et al.*, “Deep convolutional neural networks for large-scale speech tasks,” *Neural networks*, vol. 64, pp. 39–48, 2015.

[45] R. Wu, S. Yan, Y. Shan, Q. Dang, and G. Sun, “Deep image: Scaling up image recognition,” *arXiv preprint arXiv:1501.02876*, vol. 7, no. 8, 2015.

[46] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[47] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[48] S. Ji, W. Xu, M. Yang, and K. Yu, “3D convolutional neural networks for human action recognition,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221–231, 2012.

[49] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International conference on machine learning*, 2015, pp. 448–456.

[50] T. Salimans and D. P. Kingma, “Weight normalization: A simple reparameterization to accelerate training of deep neural networks,” *Advances in neural information processing systems*, vol. 29, pp. 901–909, 2016.

[51] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[52] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*, 2019, pp. 6105–6114.

[53] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[55] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[56] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[57] M. Tan *et al.*, "Mnasnet: Platform-aware neural architecture search for mobile," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2820–2828.

[58] B. Zoph, V. Vasudevan, J. Shlens, and Q. Le, "AutoML for large scale image classification and object detection," *Google AI Blog*, vol. 2, p. 2017, 2017.

[59] M. Tan, "MnasNet: Towards Automating the Design of Mobile Machine Learning Models." 2018.

[60] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[61] Vardan Agarwal, "Complete Architectural Details of all EfficientNet Models," 2020.

[62] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Computers and electronics in agriculture*, vol. 147, pp. 70–90, 2018.

[63] F. A. Spanhol, L. S. Oliveira, P. R. Cavalin, C. Petitjean, and L. Heutte, "Deep features for breast cancer histopathological image classification," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2017, pp. 1868–1873.

[64] K. Faust *et al.*, “Intelligent feature engineering and ontological mapping of brain tumour histomorphologies by deep learning,” *Nature Machine Intelligence*, vol. 1, no. 7, pp. 316–321, 2019.

[65] A. J. R. T. S. G. H. V. Y Fu, “Pan-cancer computational histopathology reveals mutations, tumor composition and prognosis,” *Nature Cancer*, 2020.

[66] Y. Liu *et al.*, “Detecting cancer metastases on gigapixel pathology images,” *arXiv preprint arXiv:1703.02442*, 2017.

[67] J. W. Wei, L. J. Tafe, Y. A. Linnik, L. J. Vaickus, N. Tomita, and S. Hassanpour, “Pathologist-level classification of histologic patterns on resected lung adenocarcinoma slides with deep neural networks,” *Scientific reports*, vol. 9, no. 1, pp. 1–8, 2019.

[68] S. Kalra *et al.*, “Yottixel–an image search engine for large archives of histopathology whole slide images,” *Medical Image Analysis*, vol. 65, p. 101757, 2020.

[69] H. Narayan *et al.*, “Similar image search for histopathology: SMILY,” *NPJ Digital Medicine*, vol. 2, no. 1, 2019.

[70] M. Babaie *et al.*, “Classification and retrieval of digital pathology scans: A new dataset,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 8–16.

[71] N. Hegde *et al.*, “Similar image search for histopathology: SMILY,” *NPJ digital medicine*, vol. 2, no. 1, pp. 1–9, 2019.

[72] S. Hemati, S. Kalra, C. Meaney, M. Babaie, A. Ghodsi, and H. Tizhoosh, “CNN and Deep Sets for End-to-End Whole Slide Image Representation Learning,” 2021.

[73] S. Kalra *et al.*, “Pan-cancer diagnostic consensus through searching archival histopathology images using artificial intelligence,” *NPJ digital medicine*, vol. 3, no. 1, pp. 1–15, 2020.

[74] N. Mehta, A. Raja’S, and V. Chaudhary, “Content based sub-image retrieval system for high resolution pathology images using salient interest points,” in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2009, pp. 3719–3722.

[75] X. Qi *et al.*, “Content-based histopathology image retrieval using CometCloud,” *BMC bioinformatics*, vol. 15, no. 1, pp. 1–17, 2014.

[76] Y. Zheng *et al.*, “Histopathological whole slide image analysis using context-based CBIR,” *IEEE transactions on medical imaging*, vol. 37, no. 7, pp. 1641–1652, 2018.

[77] X. Qi *et al.*, “Content-based histopathology image retrieval using CometCloud,” *BMC bioinformatics*, vol. 15, no. 1, pp. 1–17, 2014.

[78] L. Zheng, A. W. Wetzel, J. Gilbertson, and M. J. Becich, “Design and analysis of a content-based pathology image retrieval system,” *IEEE transactions on information technology in biomedicine*, vol. 7, no. 4, pp. 249–255, 2003.

[79] A. Sridhar, S. Doyle, and A. Madabhushi, “Content-based image retrieval of digitized histopathology in boosted spectrally embedded spaces,” *Journal of pathology informatics*, vol. 6, 2015.

[80] T. Dehkharghanian *et al.*, “Biased Data, Biased AI: Deep Networks Predict the Acquisition Site of TCGA Images,” 2021.

[81] F. M. Howard *et al.*, “The impact of site-specific digital histology signatures on deep learning model accuracy and bias,” *Nature communications*, vol. 12, no. 1, pp. 1–13, 2021.

[82] A. J. DeGrave, J. D. Janizek, and S.-I. Lee, “AI for radiographic COVID-19 detection selects shortcuts over signal,” *Nature Machine Intelligence*, pp. 1–10, 2021.

[83] H. A. Piwowar and W. W. Chapman, “Public sharing of research datasets: a pilot study of associations,” *Journal of informetrics*, vol. 4, no. 2, pp. 148–156, 2010.

[84] T. Skripcak *et al.*, “Creating a data exchange strategy for radiotherapy research: towards federated databases and anonymised public datasets,” *Radiotherapy and Oncology*, vol. 113, no. 3, pp. 303–309, 2014.

[85] D. Komura and S. Ishikawa, “Machine learning methods for histopathological image analysis,” *Computational and structural biotechnology journal*, vol. 16, pp. 34–42, 2018.

[86] D. A. Gutman *et al.*, “Cancer Digital Slide Archive: an informatics resource to support integrated in silico analysis of TCGA pathology data,” *Journal of the American Medical Informatics Association*, vol. 20, no. 6, pp. 1091–1098, 2013.

[87] L. A. Cooper, E. G. Demicco, J. H. Saltz, R. T. Powell, A. Rao, and A. J. Lazar, “PanCancer insights from The Cancer Genome Atlas: the pathologist’s perspective,” *The Journal of pathology*, vol. 244, no. 5, pp. 512–524, 2018.

[88] K. Tomczak, P. Czerwińska, and M. Wiznerowicz, “The Cancer Genome Atlas (TCGA): an immeasurable source of knowledge,” *Contemporary oncology*, vol. 19, no. 1A, p. A68, 2015.

[89] “GDC repository web site.”

[90] S. Lloyd, “Least squares quantization in PCM,” *IEEE transactions on information theory*, vol. 28, no. 2, pp. 129–137, 1982.

[91] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International conference on machine learning*, 2015, pp. 448–456.

[92] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” *arXiv preprint arXiv:1207.0580*, 2012.

[93] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*, 2009, pp. 248–255.

[94] N. Chinchor and B. M. Sundheim, “MUC-5 evaluation metrics,” 1993.

[95] S. Kalra *et al.*, “Pan-cancer diagnostic consensus through searching archival histopathology images using artificial intelligence,” *NPJ digital medicine*, vol. 3, no. 1, pp. 1–15, 2020.

[96] M. Jain and D. Singh, “A survey on CBIR on the basis of different feature descriptor,” *Journal of Advances in Mathematics and Computer Science*, pp. 1–13, 2016.

[97] T. M. Lehmann *et al.*, “Content-based image retrieval in medical applications,” *Methods of information in medicine*, vol. 43, no. 04, pp. 354–361, 2004.

[98] L. R. Long, S. Antani, T. M. Deserno, and G. R. Thoma, “Content-based image retrieval in medicine: retrospective assessment, state of the art, and future directions,” *International Journal of Healthcare Information Systems and Informatics (IJHISI)*, vol. 4, no. 1, pp. 1–16, 2009.

[99] D. Markonis *et al.*, “A survey on visual information search behavior and requirements of radiologists,” *Methods of information in Medicine*, vol. 51, no. 06, pp. 539–548, 2012.

[100] H. Müller, N. Michoux, D. Bandon, and A. Geissbuhler, “A review of content-based image retrieval systems in medical applications—clinical benefits and future directions,” *International journal of medical informatics*, vol. 73, no. 1, pp. 1–23, 2004.

[101] L. Pantanowitz, J. Szymas, Y. Yagi, and D. Wilbur, “Whole slide imaging for educational purposes,” *Journal of pathology informatics*, vol. 3, 2012.