KimiaNet: Training a Deep Network for Histopathology using High-Cellularity

by

Abtin Riasatian

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2020

Author's Declaration

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

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

Abstract

With the recent progress in deep learning, one of the common approaches to represent images is extracting deep features. A primitive way to do this is using off-the-shelf models. However, these features could be improved through fine-tuning or even training a network from scratch by domain-specific images. This desirable task is hindered by lack of annotated or labeled images in the field of histopathology.

In this thesis, a new network, namely *KimiaNet*, is proposed that uses an existing dense topology but is tailored for generating informative and discriminative deep features from histopathology images for image representation. This model is trained based on the existing DenseNet-121 architecture but by using more than 240,000 image patches of 1000 × 1000 pixels acquired at 20× magnification.

Considering the high cost of histopathology image annotation, which makes the idea impractical at the large scale, a high-cellularity mosaic approach is suggested which could be used as a weak or soft labeling method. Patches used for training the KimiaNet are extracted from 7,126 whole slide images of formalin-fixed paraffin-embedded (FFPE) biopsy samples, spanning 30 cancer sub-types and publicly available through TCGA repository.

The quality of features generated by KimiaNet are tested via two types of image search, (i) given a query slide, searching among all of the slides and finding the ones with the tissue type similar to the query's and (ii) searching among slides within the query slide's tumor type and finding slides with the same cancer sub-type as the query slide's. Compared to the pre-trained DenseNet-121 and the fine-tuned versions, KimiaNet achieved predominantly the best results for both search modes.

In order to get an intuition of how effective training from scratch is on the expressiveness of the deep features, the deep features of randomly selected patches, from each cancer sub-type, are extracted using both KimiaNet and pre-trained DenseNet-121 and visualized after reducing their dimensionality using t-SNE. This visualization illustrates that for KimiaNet, the instances of each class can easily be distinguished from others while for pre-trained DenseNet the instances of almost all of the classes are mixed together. This comparison is another verification to show that how discriminative training with domain-specific images has made the features. Also, four simpler networks, made up of repetitions of convolutional, batch-normalization and ReLU layers, (CBR networks) are implemented and compared against the KimiaNet to check if the network design could still be further simplified. The experiments demonstrated that KimiaNet features are by far better than CBR networks which validates the DenseNet-121 as a good candidate for KimiaNet's architecture.

Acknowledgements

I would like to thank all the people who helped me in times of need.

Dedication

This is dedicated to the ones who stay humble while being in the position of power.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

CNN Convolutional Neural Network 4–8, 14, 22

ReLU Rectified Linear Unit iv, 7, 8, 16, 21, 44, 51

t-SNE t-distributed Stochastic Neighbor Embedding iv, xi, 39, 41–43, 50

TCGA The Cancer Genome Atlas iv, x, 3, 22, 25–29, 31, 32, 46, 50, 51

WSI Whole Slide Image x, xi, 2, 3, 6, 7, 10–12, 25–28, 30–33, 35, 36, 38, 39, 49–51

1. Introduction

Histopathology is the gold-standard method to diagnose many diseases. The process of diagnosing a disease in a patient using histopathology starts with performing a biopsy, which means removing a small part of tissue, called specimen, from patient's body, mostly from a tumor or mass. The most common ways to do a biopsy are using a needle, an endoscope or performing surgery. A pathologist first analyzes the specimen with naked eyes and describes how it looks by features such as color and size. After that, the specimen is prepared to be cut and put under the microscope for further diagnosis. There are two ways to make the specimen firm enough so it can be cut into thin slices: paraffin-embedded (permanent) sectioning and frozen sectioning. The paraffin-embedded sections are made by first placing the specimen in a fixative (usually formalin) for several hours to preserve the tissue. Next, the water inside the fixed specimen is replaced with paraffin wax. Then the specimen is put inside a paraffin block. When this block is firm, the specimen is cut into very thin slices. Afterwards, one thin slice is put on a glass slide, the paraffin is removed from the tissue and water is added again. Finally, to stain different parts of the cells in the tissue, dyes are used which results in the cell nuclei to turn to dark blue and cytoplasm to be colored pink. In frozen sectioning, the specimen is quickly frozen and cut into very thin layers. The staining process is the same as permanent sectioning. Finally, the pathologist investigates the prepared specimen slide under a microscope and makes a diagnosis in addition to writing a report. It is worth mentioning that the permanent sectioning process usually takes several days and it has the best quality for the examination by the pathologist while frozen sectioning is completed after 15 to 20 minutes but its quality is lower. In this study, the permanent slides are used due to their high quality [24, 4]. [1,2]

In digital pathology, the mentioned prepared specimen slides are scanned in very high magnifications, like 20× or 40× using special scanners which results in gigapixel images, as large as $100,000 \times 100,000$ pixels [17]. over the last few years, digital pathology has

[1] cancer.net

[2] cancer.gov

gained more attention among pathologists and researchers thanks to advancements of the whole slide digital scanners. Using digitized WSI has benefits such as (i) Easier collaboration among pathologists over the world through telepathology, (ii) Speeding up the workflow in the hospitals, e.g. fast search over the WSIs instead of manually looking for a glass slide, (iii) Providing the opportunity of applying image processing algorithms on a valuable set of data to obtain useful information and (iv) Increasing the consistency and decreasing/removing the inter-observer variability in diagnosis with the help of machine learning, specifically deep learning [3, 14, 40, 6].

Before the popularity of deep learning, handcrafted methods were used to analyze the digital pathology slides. The drawbacks of these methods are that having domain knowledge is a prerequisite for their development which is costly and time-consuming. In addition, handcrafted algorithms do not generalize well and usually drop in performance when facing new data. Deep networks are a solution to these challenges.

In deep learning, only simple preprocessings are needed on the raw data which is then fed to a network. Unlike the handcrafted methods, the task of feature extraction will be enabled by the parameters (weights) of the network, acting as a function estimator. That is why many of the recent research papers on digital pathology have benefited from either off-the-shelf models or customized models [3, 20].

The features generated by deep networks are good candidates to represent images. As one major problem in digital pathology is the large size of WSIs, e.g., 100,000 by 100,000 pixels, deep features of patches (sub-images) from images are extracted as compact, lightweight and expressive representations. These features can be used for finding similar WSIs for a query WSI in a large archive. The content-based histopathology image retrieval can be quite useful for the daily workflow of pathologists. As well, studying the relationships between the visual patterns in WSIs and genome information will be enabled by image representation [13, 20]. However, there are still unanswered questions regarding these matters. Do the advantages of fine-tuning/training a network with domain-specific data outweigh the costs or features extracted from the pre-trained models? What are the challenges and benefits of training a network from scratch in histopathology?

In this thesis, the focus will be on fine-tuning and training different models using the DenseNet-121 topology [19, 45, 46] by freezing the weights of different blocks. These models were trained on around 240,000 patches of size 1024 by 1024 pixels extracted at 20× magnification from 7,375 TCGA WSIs depicting 30 different tumor sub-types. Considering the experimental results, changes in quality of deep features generated by the trained models are investigated with respect to their performance for WSI search. The contributions of this thesis are:

(i) Creating a manually labeled dataset for the task of tissue segmentation, consisting of 244 thumbnails chosen from the publicly available TCGA dataset.
(ii) Comparing the performance of different encoders as the backbone of U-Net topology for the tissue segmentation task and proposing a tailored algorithm for this problem.
(iii) Proposing a clustering-based mosaic approach for WSI representation with a constraint on high cellularity to employ the WSI label as a soft label for its patches to facilitate the training.
(iv) Investigating the effect of training and fine-tuning the DenseNet-121 topology on the representativeness of its deep features for the retrieval task. The training set for this study contains 242,202 patches of 1024 × 1024 pixels extracted from the WSIs of 30 different tumor sub-types available on TCGA dataset.

In the following chapters, first the relevant literature will be reviewed. Afterwards, the tissue segmentation task is studied as an important step in the required preprocessing of histopathology images. Then the process of creating a dataset from the TCGA repository as well as fine-tuning/re-training a network, which we have named KimiaNet, is described. Next, the performance of features generated by different versions of KimiaNet and the pre-trained DenseNet-121 is compared. After that, the idea of using simpler and smaller network architectures instead of DenseNet-121 is tested and the superiority of KimiaNet is shown.

2. Literature Review

In this chapter, we will first review the research done for the problem of tissue segmentation for histopathology images. After that, the role of deep learning in histopathology image representation will be investigated. These approaches could be categorized into three types of pre-trained, fine-tuned and trained. Finally, the DenseNet architecture and its strengths are discussed as it is the utilized topology for training KimiaNet.

2.1 Histopathology Image Background Removal

There is not much research done on histopathology image background removal as a standalone problem and it is often addressed as a part of a larger processing pipeline. But this problem is generally solved using one of the following approaches:
(i) **Machine-vision-based** methods which include performing thresholding on features such as image saturation and intensity and applying methods such the Otsu algorithm on different color spaces. One of the competitive algorithms for this problem is the FESI (Foreground Extraction from Structure Information) algorithm [5] which uses a combination of basic methods, such as median filtering, thresholding, erosion and dilation to address this problem. Also, HistomicsTK library, which is one of the well-known and widely used open-source libraries in digital histopathology, provides a tissue detection function and its algorithm contains a series of Gaussian smoothing and Otsu thresholding.
(ii) **Learning-based** methods which use labeled data to train machine learning algorithms. For example Alomari et al. have fed four features including color, appearance, texture and spatial features to a two-layer neural network to classify the pixels into background and foreground classes [1]. Also, Bándi et al. trained a CNN with 7 convolutional layers as well as a UNet for tissue segmentation. The input for the networks were patches with a single label of “tissue” or “background” based on its central pixel [2].

2.2 Deep Learning Approaches in Histopathology Image Representation

2.2.1 Pre-Trained Networks

Spanhol et al. extracted features of patches of breast cancer, BreaKHis dataset, from the deepest layers of the BVLC CaffeNet model, e.g. fc6, fc7 and fc8, reusing the pre-trained ImageNet weights. Then they used Logistic Regression to classify these features which helped them achieve comparable results to a CNN trained from scratch [36].

Mormont et al. have compared the performance of several commonly used architectures, including VGG-16, VGG-19, Inception-V3, ResNet-50, Inception-ResNet-V2, DenseNet-201, and MobileNet, with 6 different strategies for preparing the features for the classifier:
(i) Extracting feature from the last layer, before the first fully connected layer, of the networks.
(ii) Selecting a subset of features extracted from the last layer.
(iii) Merging all the features extracted from the last layer of all of the networks together.
(iv) Merging the feature maps of several layers of each network.
(v) Extracting features from an inner layer instead of the last layer.
(vi) Fine-tuning by training the whole network on the specific dataset.
In the next step, the extracted features are fed to each of the three (i) linear support vector machines, (ii) Extremely Randomized Trees and (iii) Fully Connected Layer classifiers. These approaches were evaluated on 8 classification datasets, 4 histopathology and 4 cytology. The histopathology datasets were Necrosis, Lung, Breast and Glomeruli. The analysis of their results showed that ResNet-50 and DenseNet-201 were the best among the tested architectures. And concerning the methods of feature extraction, fine-tuning the weights by training the whole network and using the fully connected classifier could achieve the best results in general with considerable increase for multi-class datasets. The best method for using the off-the-shelf networks seemed to be extracting features from an inner layer, which should be determined separately for each individual case. This approach could result with result slightly worse than the trained networks while having the advantage of not needing any training. Some interesting conclusions that were drawn from the experiments were that (i) There was always an inner layer with features better than the last layer, however, the position of the layer was not the same for different networks. (ii) None of the feature merging approaches could improve the results. (iii) Many of the features are redundant and very few of them are informative for the classification. [28].

2.2.2 Fine-Tuned Networks

Faust et al. fine-tuned the last two blocks of a VGG-19 model, with a global average pooling added at the end, initialized with ImageNet pre-trained weights. The dataset used was made up of 838,644 1024 by 1024 patches extracted from 1,656 WSIs scanned at 20× consisted of 74 tissue classes. They used the features extracted from this model, having 512 dimensions, to investigate the relationships of CNN deep features and human recognizable morphologic patterns. [11]

2.2.3 Trained Networks

Liu et al. proposed a framework to detect and localize breast cancer metastasis in lymph nodes. They trained the Inception (V3) architecture with 3 different settings: (i) The network with weights initialized randomly. (ii) The pre-trained network with ImangeNet weights. (iii) A down-sized model by decreasing the number of the filters. Comparing the performance of each individual network, it was observed that using pre-trained weights did not improve the performance of the model and just speeded up the convergence while the smaller network could achieve the same results, even better in some metrics. The ensemble of the three networks slightly improved the results. Finally, they tried to mimic the workflow of the pathologists by feeding patches at different magnifications, e.g. 40× and 20×, which did not benefit the model. Their evaluations were done on the Camelyon16 dataset, achieving the state of the art results in lesion-level tumor detection task, and an independent dataset of 110 WSIs. [26]

Fu et al. fine-tuned an Inception-V4 network with 42 classes comprised of 28 tumor types and 14 classes normal tissue. The source of their dataset was 17,396 fresh-frozen whole slides available on TCGA repository. They extracted 512 by 512 pixels patches with 50 pixels overlap in 20× magnification which resulted in 6,564,045, 1,357,892 and 6,641,462 patches from 8,067, 1,687 and 7,672 whole slides for the training, validation and test datasets, respectively. To avoid any bias caused by differences in image preparation process, they randomly selected 80% slides of each laboratory for the training dataset. This network was trained with 6,564,045 patches extracted from 8,067 wholes slides in which were cleaned by removing the uninformative and blurry patches. They trained the mentioned network as a classifier but the ultimate goal was utilizing it as a histopathology patch feature extractor to study the relationship between the deep features extracted by the networks, which they called computational histopathological features, and genomic driver alterations, whole transcriptomes and survival. This was done by extracting 1,536-

dimensional features from the last layer of their trained model by feeding 14 Million patch to it. [13]

Wei et al. trained a ResNet model with 18 layers to classify a patch of Lung Adenocarcinoma to one of the 5 histologic patterns: Lepidic, Acinar, Papillary, Micropapillary, and Solid or label it as Benign. Using this model, they could identify major and minor histologic patterns in a whole slide which are both important information considering the fact that a WSI usually contains a mix of these patterns and could help pathologists with preparing the documentations needed for each patient. Their dataset was made up of 422 Formalin-Fixed Paraffin-Embedded (FFPE) whole slides scanned at 20× magnification which were randomly divided into 245, 34 and 143 slides for training, validation and test sets. To create the dataset, they divided each slide in the training set into 4,161 variable size crops and the validation slides into 1,068 224 by 224 pixel patches. Then these images were manually labeled by three pathologists. Due to the variable size of the training crops, a sliding window was used to produce fix sized patches from each crop at the training time. To find the right model size, they did tests on ResNets with 18, 34, 50, 101, and 152 layers which resulted in the same performance for all of the models so they chose the 18-layer version. Finally, comparing the results of their trained model against the performance of the three pathologists showed the model could outperform the pathologists using three different metrics, "Average Kappa Score", "Average Agreement" and "Robust Agreement".[43]

2.3 DenseNet Architecture

DenseNet was introduced based on the idea that using shorter connections between the input and output layers of CNNs, could give it the potential to get deeper, more accurate and more efficient. The general architecture of the DenseNet is made up of several, generally four, dense blocks (see Figure 2.1). Each of these dense blocks contains a number of layers where each layer is connected to all of the subsequent layers in a block. This means if a dense block contains l layers, there would be a total of $\frac{l(l+1)}{2}$ connections and the reason behind this design is maximizing the flow of information between the layers. Having the same feature map size for the layers in the dense block, all of the feature maps generated by the proceeding layers are concatenated to the output feature maps of a layer.

Each layer is comprised of a sequence of batch normalization, ReLU and a 3 by 3 convolutions. If each layer generates k feature maps, the number features that the layer l would take as input would be $k_0 + k \times (l - 1)$. The hyperparameter k is called “growth rate”. Due to the densely connected design of the network, each layer takes a large number of

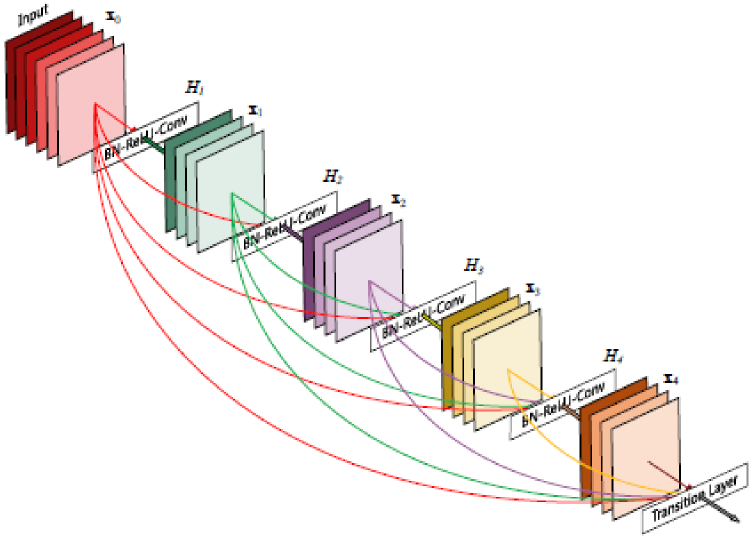


Figure 2.1: A 5-layer dense block with a growth rate of k =4. Each layer takes all preceding feature-maps as input [19].

feature maps as input. That is why a layer composed of a sequence of batch normalization, ReLU and a 1 by 1 convolution, called "bottleneck layer" is added before each layer with 3 × 3 convolution to reduce to the number of layer input feature maps, generally to $4k$, and decrease the computational cost.

"Transition layers", consisted of a sequence of batch normalization, a 1×1 convolution and a 2×2 average pooling, have been used between the consecutive dense blocks, to perform down-sampling of the feature maps as it is an important part of CNNs. Also, to make the network more compact, a hyperparameter β, where $0 < \beta \leq 1$, is used to decrease the m output feature maps of the dense block to $\lfloor \beta m \rfloor$. β is generally set to 0.5. The experiments have shown that this approach results in avoiding the generation of redundant features by the network.

The DenseNet architecture, Figure 2.2, brings a number of benefits such as:
(i) By utilizing the densely connected layers pattern as a method to resolve the vanishing gradient problem, this architecture can get deeper and more accurate without a decrease in performance.
(ii) In addition, this architecture needs fewer parameters compared to the other networks considering the fact that instead of relearning necessary feature maps at each layer, it reuses the features by passing the information taken from the previous layers and concatenating

them to the new feature maps generated at the current layer. For example, DenseNet could achieve the same level of accuracy requiring around $\frac{1}{3}$ of the parameters used in the ResNet architecture.
(iii) Due to the fact that there is a short path between the loss function and each layer in this network, it can be said that there is more supervision on the features produced by each layer. This "deep supervision" can result in learning more discriminative features by the intermediate layers which is highly desirable for image representation tasks.
(iv) DenseNet can achieve competitive results with very narrow layers, set by the growth rate. The reason could be the accessibility of the preceding features at each layer of a block which helps producing a "collective knowledge" in the network. Also, the growth rate controls the amount of new information produced at each layer and can avoid overfitting by acting as a regularizer.

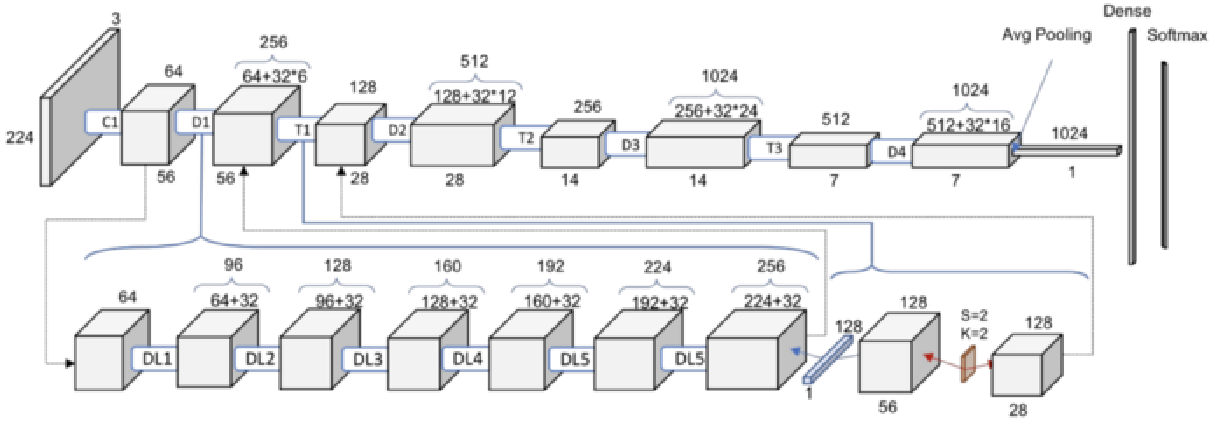


Figure 2.2: A detailed architecture of DenseNet with input size of 224 × 224, Image Source.

A well-known example of using DenseNet in a medical imaging task is CheXNet [27] which is a DenseNet-121 network architecture trained with over 100,000 X-ray images. CheXNet is able to detect all 14 diseases in the ChestX-ray14 dataset with a higher performance than radiologists.

3. Tissue Region Extraction

3.1 Introduction

WSIs can not be directly fed to the network as they are too large to be processed (a typical WSI is usually larger than 50000 × 50000 pixels). A common solution to this problem is to extract patches from the slide as its representatives. To perform the patch selection, the question arises *"where should the patches be extracted from?"* which is the problem of specifying the regions of interest. One of the ways to tackle this problem is to get the regions of interest of the slides annotated by histopathologists. However, this is a tedious and time-consuming task for the pathologists and is also subject to errors and variability. Another way that currently seems to be more popular is automating this task. The problem of automatic WSI annotation can be broken down into steps, all starting with removing the background. The background is part of the glass slides that do not contain any tissue. This background appears very bright in digital images and its segmentation might be considered as a rather easy task. However, to stain variations, debris and many artifacts (tissue folds, air bubbles, etc.) background segmentation, as the first major step for WSI processing, is practically a difficult task.

In this chapter, the performance of the U-Net topology with different backbones in the task of WSI background removal is investigated and an algorithm with close to 100% sensitivity and specificity is proposed which can be used as a highly improved substitute for the tissue mask generator function and also added to the Yottixel algorithm to increase its performance (both explained in 4.3.2).

3.1.1 Problem Definition

As the name suggests, the goal in the problem of background removal or tissue segmentation in histopathology images is to remove irrelevant and uninformative pixels as much as possible with the minimum damage to the tissue pixels. Since histopathology image analysis is generally the last step for the diagnosis of many diseases such as cancer, inflammation and infection, it is crucial to avoid losing tissue pixels. This adds an important consideration for the evaluation of the algorithm proposed for this problem which states that the sensitivity of the segmentation algorithm has more importance than its specificity and should be very high.

3.1.2 Importance and Applications

In many histopathology image analysis tasks, such as patch extraction for training a network, specifying the tissue parts is the first step. So segmenting the tissue precisely is a prerequisite for an effective algorithm. On the other hand, if this step is not performed well, irrelevant parts may confuse the algorithm (see Figure 3.1).

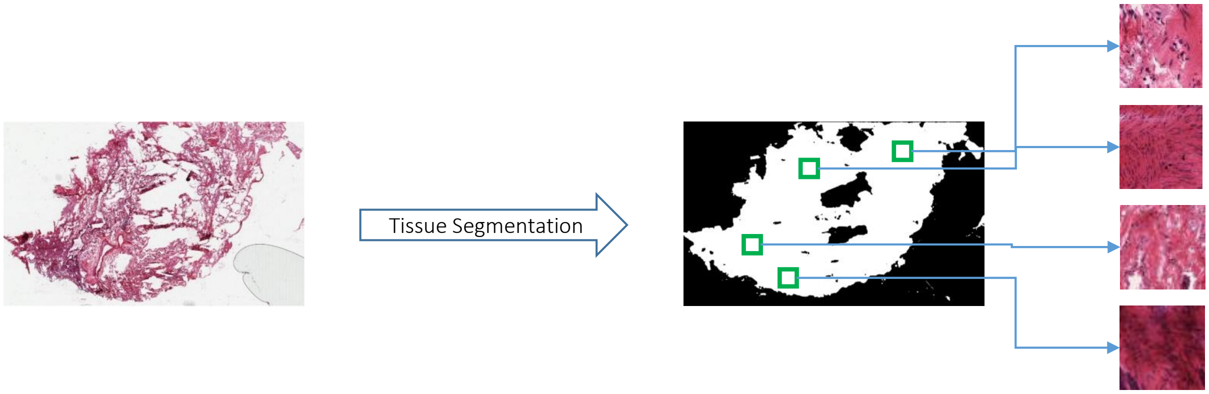


Figure 3.1: Using tissue segmentation for patch extraction

Tissue segmentation can be used to speed up the digital pathology slide scanners. These scanners are used to digitize glass slides containing tissue specimens and generate WSIs. While scanning, the focus depth of the scanner must be adjusted for different tissue regions due to variable tissue types. This creates the need for scanners to identify the tissue areas on the glass slide. This step should be done precisely; if a mistake occurs during scanning, some parts of the slide will be scanned blurry which has the potential for causing problems in the further analysis tasks since the data is lost, there is no way to fix it in the following steps of the workflow (see Figure 3.2). Currently, a lab technician manually checks every slide after scanning, and re-scans the corrupted slides which is a tedious and expensive procedure. Extracting the tissue parts of the slide before starting the scanning, can reduce the time and cost of this process while doing it at a higher precision [2, 29].

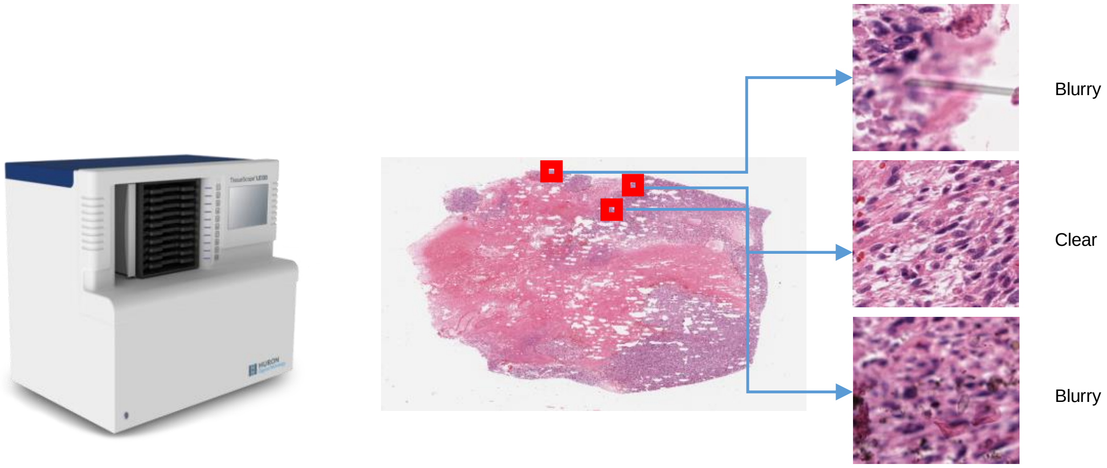


Figure 3.2: Different parts of a scanned slide having different clarities

3.1.3 Challenges

The problem of tissue segmentation may seem easy in some cases (Figure 3.3) but it has its own challenges. These challenges can be divided into two types: (i) The ones that are related to the tissue type. For instance, air sacs in the lung, and fatty tissue which could appear in many tissue types, may confuse algorithms due to their similarity to the background (i.e., lack of complex texture). (ii) The other category is artifacts which include extra or weak stain, dirt, air bubbles, broken glass and marker traces (Figure 3.4).

3.2 Dataset Creation and Training

3.2.1 Dataset

To create a dataset for tissue background segmentation, we used 244 WSIs randomly selected from different organs such as brain, breast, kidney, and lung. As one of the challenges of histopathology image analysis is the WSI size, a WSI could be as large as

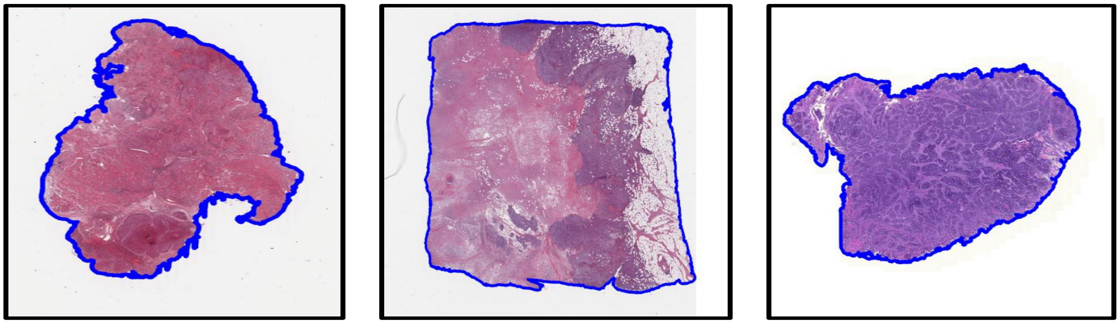


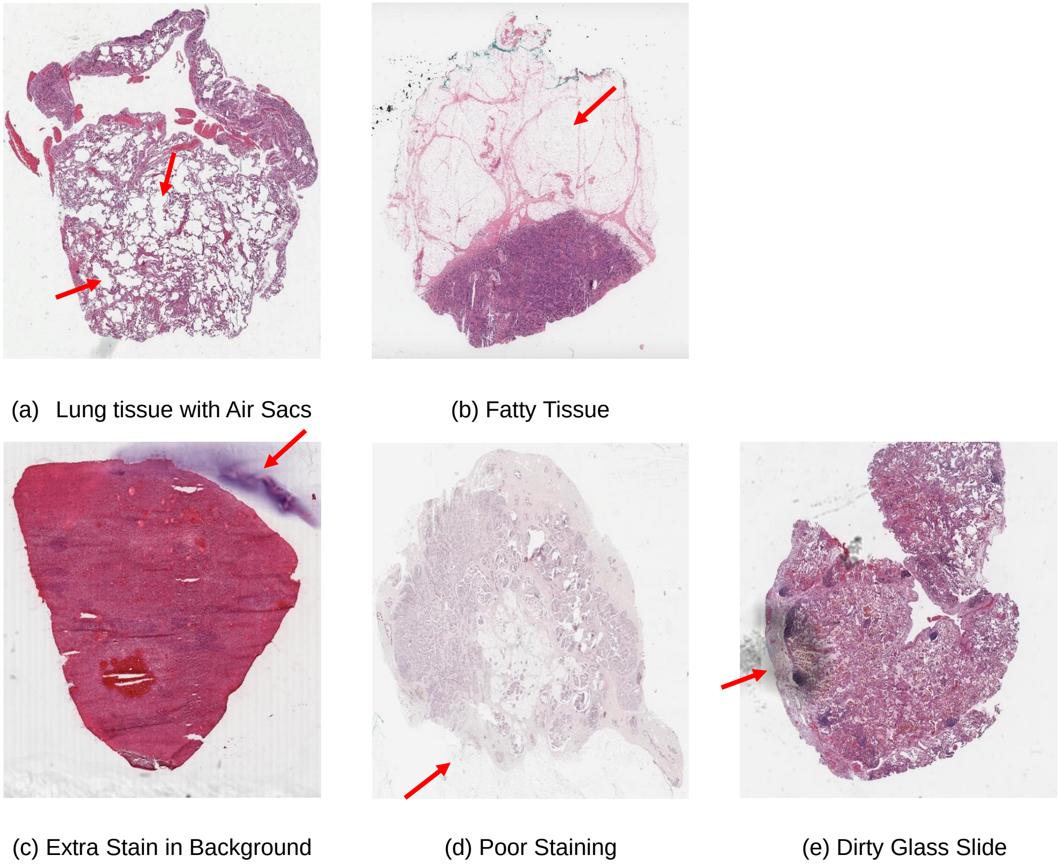
Figure 3.3: Simple cases for tissue segmentation. These samples are segmented by a simple handcrafted method.

100,000 by 100,000 pixels, labeling each pixel was not practical. To overcome this problem, we chose to work with the thumbnails, which are generally around 1× magnification. In addition to easier labeling, this has the advantage of faster computation.

Creating tissue masks was performed in 2.5x magnification as we found it to be the lowest magnification that still allows us to distinguish the tissue parts from the background in challenging cases such as poor staining. The labeling process is comprised of 3 steps (Figure 3.5):
(i) First, we developed a handcrafted algorithm to create initial masks for every thumbnail. This algorithm applies binary thresholding on the gray-scale thumbnails and does some processing based on the size of contours and their distances[1] (see Algorithm 1).
(ii) After that, the initial masks were refined manually to make sure that all tissue regions are selected and noise and artifacts are removed as much as possible. At this step, morphological dilation was performed on difficult cases, to make sure all tissue parts are preserved [2].
(iii) Finally, each pair of mask and thumbnail was resized to make each dimension less than 1024 pixels, preserving the aspect ratio, to make the images small enough to be fed to the network.

[1] Implemented by Abtin Riasatian

[2] Done by Maral Rasoolijaberi


(a) Lung tissue with Air Sacs

(b) Fatty Tissue

(c) Extra Stain in Background

(d) Poor Staining

(e) Dirty Glass Slide

Figure 3.4: Common challenges of the tissue segmentation task. (a) and (b) are examples for challenges caused by tissue type. (c), (d) and (e) are instances of challenges caused by artifacts.

3.2.2 Training - Model Architecture

U-Net, which is a CNN with a U-shape architecture, is made up of two parts, called encoder and decoder. The first sub-network, known as the encoder, extracts high-level features to capture the image content. The decoder subnetwork, also known as the expansion part, creates the desired segmentation map. Figure 3.6 shows the proposed network architecture. U-Net-based deep networks, the same as U-Net, include two encoder and decoder sub-networks. As the input image passes through the first sub-network, higher-level features are extracted. In the next sub-network, deep feature maps are combined with low-level

Algorithm 1 Handcrafted Masking Method

```
$chosenContours \leftarrow []$
$rgbThmb \leftarrow$ readInput()
$binThmb \leftarrow$ binaryThresholding($rgbThmb$)
$contours, hierarchy \leftarrow$ findContours($binThmb$)
$fatherContours \leftarrow$ getContours($contours, hierarchy, 0$)    ▷ Get the values is the first level of the hierarchy tree as the fatherContours
append($chosenContours, fatherContours$)
$firstLevelChildren \leftarrow$ getContours($contours, hierarchy, 1$)    ▷ Get the values is the second level of the hierarchy tree as the fatherContours
$firstLevelChildren \leftarrow$ sort($firstLevelChildren, key =' area'$)
append($chosenContours, firstLevelChildren[0]$)
$i \leftarrow 1$
while $firstLevelChildren[i].area >$ min($firstLevelChildren[i-1].area \times ratioThreshold, areaThreshold$) do    ▷ Add contours until the ratio of the contour's area to the next larger contour's is greater than a threshold
    append($chosenContours, firstLevelChildren[i]$)
    $i \leftarrow i + 1$
for $xinfirstLevelChildren$ do
    $distanceCondition \leftarrow$ distanceToClosest($x, chosenContours$) $< distanceThreshold$    ▷ Add contours closer than a threshold to one of the currently chosen ones
    if $distanceConditionand(xnotinchosenContours)$ then
        append($chosenContours, x$)
for $xinfirstLevelChildren$ do
    $areaCondition \leftarrow$ getArea($x$) $> areaThreshold$
    if $areaConditionand(xnotinchosenContours)$ then
        append($chosenContours, x$)
drawContours($chosenContours, finalMask,' white'$)    ▷ Specifying the tissue areas with the while color
for $holeintextrminvert(binThmb)$ do
    if $holeMinThreshold < hole.area < holeMaxThreshold$ then
        drawContours($hole, finalMask,' black'$)    ▷ Specifying the holes with the black color
```

feature maps from the encoder sub-network. The spatial resolution of feature maps is increased in the second sub-network so the output mask has the same size as the input image. The connections between the encoder and decoder in U-Net architecture facilitate the information propagation. In terms of connections in the U-Net architecture, feature maps from the encoder part are cropped and concatenated to feature maps in the decoder

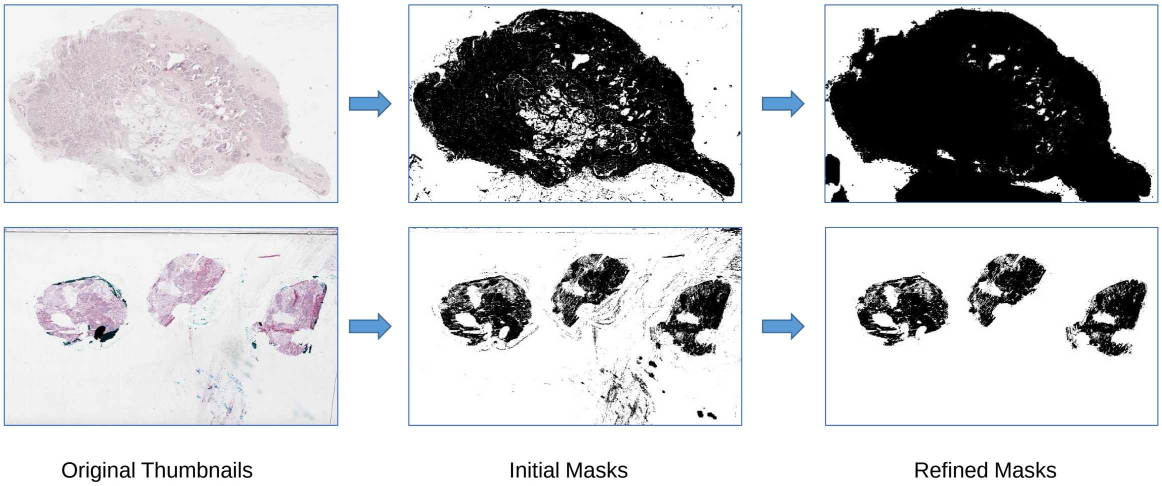


Figure 3.5: Steps of generating masks for two sample slides: (i) creating the initial masks using a handcrafted algorithm (ii) refining the initial masks manually.

sub-network to retrieve local information. These connections enable the network to learn from a few number of samples [33]. To improve the performance of U-Net, we applied custom backbones on its architecture using Segmentation Models python library. The encoder part of these customized networks is the feature extractor, i.e., complete network architecture except the last fully connected layer, of a chosen network, e.g., MobileNet. The decoder part consists of 5 decoder blocks with filters of size 256, 128, 64, 32 and 16 as it gets deeper. The structure of each decoder block is made up of one 2d-upsampling layer and two repetitions of 2d-convolution, batch-normalization and ReLU activation. Four skip connections connect layers from the encoder part, usually the output of ReLU activation at a certain layer of each encoder block, to the last four decoder blocks, after the up-sampling layer. The last layer of the network is a 2d-convolution layer with Sigmoid activation. We experimented with six different backbones (topologies) for U-Net-based solutions for tissue segmentation which are introduced in Section 3.3.1.

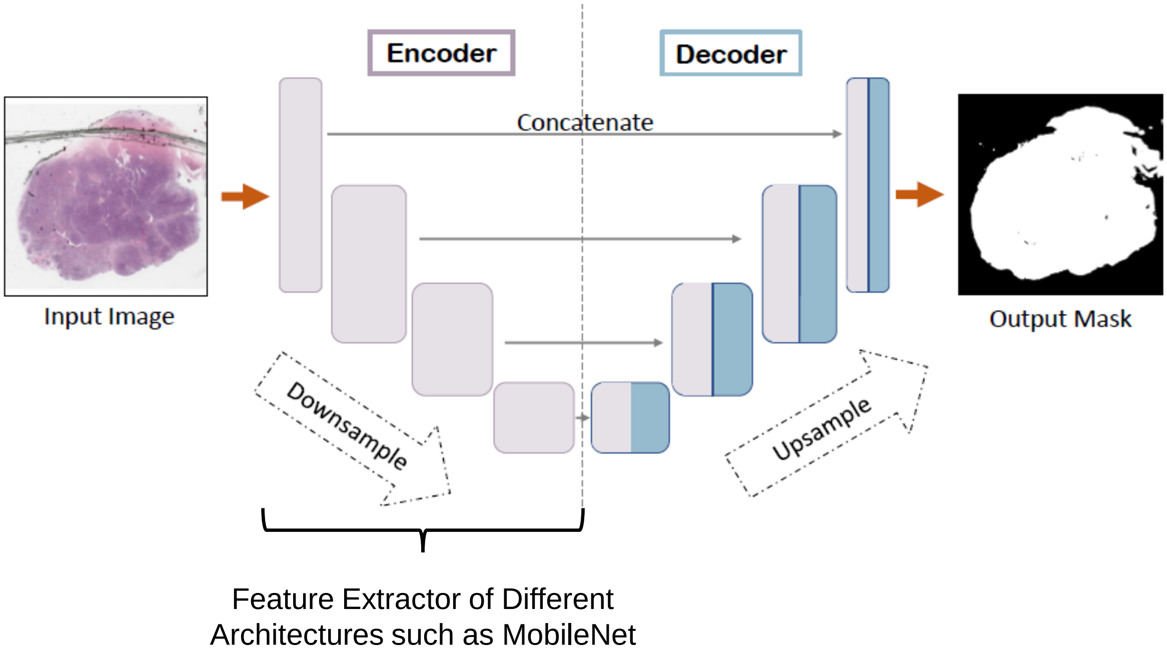


Figure 3.6: U-Net Topology

3.3 Experiments

3.3.1 Topologies and Training Process

We have experimented with 6 different network topologies including MobileNet [18], VGG16 [34], EfficientNetB3 [38], ResNet50 [16], ResNext101 [44], and DenseNet121 [19] as the backbone of U-Net model to find the most suitable ones for tissue segmentation. All networks were trained for 50 epochs, with no early stopping, using Adam optimizer with the learning rate of $1e-4$ on one NVIDIA Tesla V100 GPU with 32GB memory. After running the experiments with two loss functions, (i) Jaccard Index and (ii) sensitivity plus specificity. We chose the latter so the network tries to avoid the misclassification of tissue parts as background while having a good performance at recognizing background. This is due to the importance of the sensitivity in this problem (we have to find all tissue pixels). The drawback of using Jaccard Index as the loss function was the relatively low sensitivity of the results. The networks were initialized with ImageNet weights and were trained and evaluated with five-fold cross-validation [3]. For each fold, 195 1024 × 1024 RGB images were used as the input and binary masks with the same size as the label, or ground-truth, in which pixel value 1 (positive) meant tissue and pixel value 0 (negative) meant background. Input images and their corresponding masks were augmented by three transformations: (i) Random rotation within the range of -180 and 180 degrees, (ii) random horizontal flipping, and (iii) random vertical flipping.

The validation dataset contained 49 images for each fold. Considering the changes in the validation loss for two networks, ResNext101 with around 51 million parameters and EfficientNet-B3 with less than 18 million parameters, through 50 epochs, Figure 3.7, it seems that both have the same pattern; 20 epochs appeared to be enough for proper network training. This would take around 20 minutes for a medium-size network and 40 minutes for a large network which is a negligible cost considering the benefits of using networks.

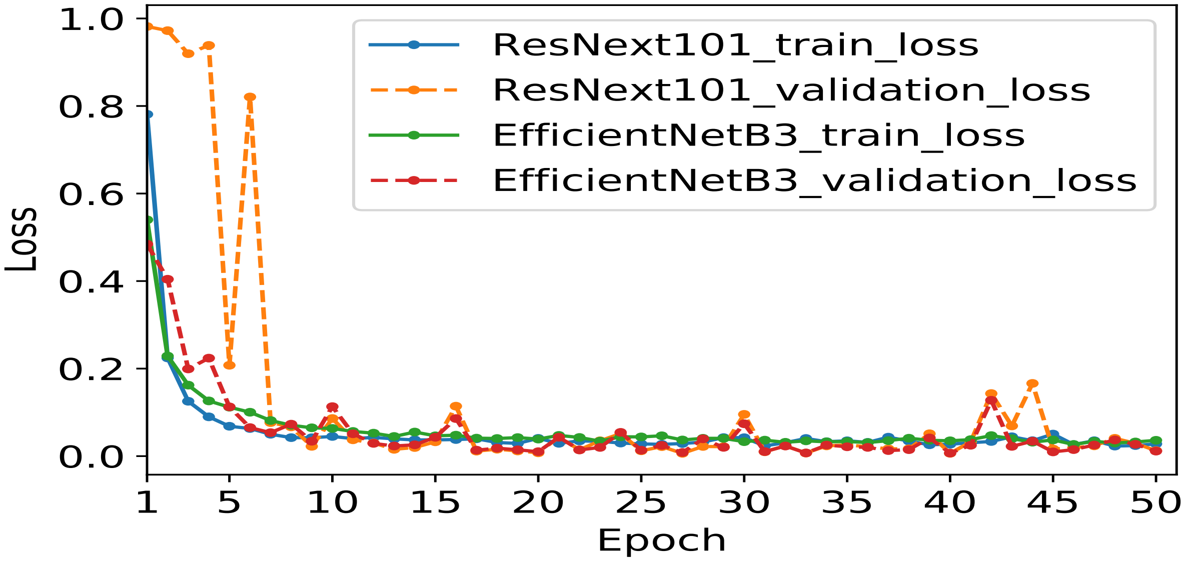


Figure 3.7: Training and validation loss for ResNext-101 and EfficientNet-B3

[3] Implemented by Abtin Riasatian

3.3.2 Methods Chosen for Comparison

To compare our results against other methods, we used the same input images fed to our networks as their input and calculated their performance against the ground-truth masks. All methods were checked to be able to work with the given inputs.

We compared our results against four traditional computer vision methods[4]:
(i) FESI algorithm[5] is improved by changing the color space of the input image from BGR to LAB and the value of the first two channels, lightness and red/green value, are changed to maximum intensity value[5]. The color space of the resulting image is changed to gray-scale and binarized using the mean value of the image as threshold. This binary image is passed to the Gaussian filter instead of using the absolute value of the Laplacian of the gray-scale image as done in the original paper. (ii) We used *locate_tissue_cnts* function available in the open-source Python package, Tissuloc [7], as a recently developed method for comparative purposes. We modified the function in a way that it uses the thumbnail image as input. Also, all of the input parameters of the function are set to default values except *min_tissue_size* which is set to 50 to make sure the algorithm would detect all tissue parts. (iii) HistomicsTK Python library as one of the most popular libraries in the histopathology domain. *saliency.tissue_detection.get_tissue_mask* function was used as the tissue segmentation method.

We set the input parameters *deconvolve_first* to False, *n_thresholding_steps* to 1 and *min_size* threshold to 50.
(iv) Otsu binarization method as one of the well-known algorithms to classify pixels into foreground and background. The RGB thumbnail images are first converted to gray-scale and then the Otsu method is applied.

3.3.3 Performance Evaluation

In the test phase, we evaluated all methods using ground-truth masks via 5-fold cross-validation. In addition to the processing time, four different metrics including Jaccard index [2], Dice coefficient [1], sensitivity, and specificity [10], were measured. The definition of these metrics are presented below:

[4] Implemented by Abtin Riasatian

[5] Taken from https://github.com/alexander-rakhlin/he_stained_fg_extraction

$$\text{Jaccard} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}, \tag{3.1}$$

$$\text{Dice} = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}}, \tag{3.2}$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{3.3}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}, \tag{3.4}$$

where TP, TN, FP, and FN denote the number of true positives, true negatives, false positives, and false negatives, respectively. The segmented pixels are considered as positive where they are labeled as tissue and vice versa.

3.3.4 Results and Analysis

In addition to Improved FESI and TissueLoc methods, we chose HistomicsTK tissue segmentation and Otsu algorithm to compare our networks' results against well-known methods for histopathology image analysis.

Table 3.1 shows that all networks, except VGG16, outperform all four handcrafted methods considering their overall performance. The most important advantage of networks

Method	Time (s)	Jaccard Index	Dice Coeff.	Sensitivity	Specificity
MobileNet	0.11	0.95	0.97	0.99	0.99
EfficientNet-B3	0.18	0.95	0.97	0.99	0.98
ResNet50	0.16	0.94	0.97	0.99	0.98
DenseNet121	0.16	0.93	0.96	0.99	0.98
ResNext101	0.50	0.93	0.96	0.99	0.98
VGG16	0.11	0.75	0.82	0.99	0.81
Improved FESI [5]	0.11	0.86	0.92	0.91	0.97
TissueLoc [7]	0.26	0.81	0.88	0.88	0.97
Otsu algorithm	0.02	0.81	0.89	0.82	0.99
Histomics-TK	0.13	0.78	0.87	0.79	0.99

Table 3.1: Summary of results: Comparing our networks with image-processing methods (gray rows).

over the handcrafted methods is their high sensitivity ($\approx 99\%$) which almost guarantees tissue preservation in the generated masks. It is also worth mentioning that the networks have managed to generate very specific results while preserving the sensitivity, all of the networks, except VGG-16 having greater than or equal to 98% specificity. In addition, the speed of mask generation in the networks is as fast as handcrafted methods such as Improved FESI and TissueLoc while achieving considerably higher Jaccard Index and Dice Coefficients.

It can be seen that Jaccard Index for the networks with the best performances, namely MobileNet and EfficientNet-B3, is 9% higher than the best handcrafted method, namely Improved FESI. Also the box plot in Figure 3.8 indicates the stability of the networks' results compared to the handcrafted methods.

To compare network backbones, it can be seen that MobileNet has shown the best performance. Also, EfficientNet-B3 has been very competitive. The poor performance of VGG-16 could be due to several reasons. First of all, this network has a large number of parameters (more than 23 million) while it only has 66 layers compared to other networks such as MobileNet with more than 8 million parameters and 128 layers and EfficientNet-B3 with around 18 million parameters and 418 layers. Also, the use of batch normalization and ReLU activation layers in the convolution blocks in other architectures have the benefits of avoiding internal covariate shift, which results in faster convergence, and keeping the network sparse, causing the generalization error to decrease, respectively. Since VGG-16 lacks these layers in its architecture, it converges with difficulty.

Figure 3.9 depicts a visual overview of the proposed network results against the common image-processing methods. Four different slides are shown in this figure, each having a challenge. From left to right:

The first slide has its glass slide margin in the scanned slide which could be mistakenly segmented as the tissue like in Otsu and HistomicsTK's results.

The second example contains fat tissue, which is hard to segment due to its similarity to the background color. A black marker trace can also be seen at the bottom left side of the slide. Both EfficientNet-B3 and MobileNet networks have done a good job at this example ignoring the marker trace and taking the fat as the tissue, which is desired. VGG-16 has taken the fat part but has also labeled some dirt and the marker trace as the tissue. However, all of the handcrafted methods have missed the fat and only the Improved FESI algorithm has managed to ignore the marker trace but it is at the cost of losing more tissue parts than other handcrafted methods, which is not a desirable behavior.

The challenge of the third slide is the bubble in the bottom right, formed during the slide preparation. Also, there are some very small dirt and tissue parts on the slides which are good indicators of the algorithms' sensitivity and specificity. MobileNet and EfficientNet-B3 have labeled the bubble as background and also have correctly labeled small tissue and dirt areas. It can be observed that the EffiecientNet-B3 network has been more conservative about the tissue parts and have labeled more parts as tissue than the MobileNet. This difference might be due to the higher specificity of the MobileNet. VGG-16 has performed very badly, segmenting almost all of the slide as tissue. The improved FESI and TissueLoc algorithms have performed well removing the artifact and dirt but this is done by the price of losing some tissue parts, which is not negligible in this problem. The HistomicsTK and Otsu algorithms have generated almost the same mask, failing to remove the bubble in addition to losing some tissue parts.

The fourth slide is a lung tissue and contains air sacs which is another challenge for the tissue segmentation algorithms. MobileNet and EfficientNet-B3 have a good overall performance for this slide but seem to have lost a part of a small tissue at the right side of the slide. VGG-16 has also generated a pretty good segmentation mask, losing just some small tissue parts. The Improved FESI algorithm has missed a lot of tissue parts which means low sensitivity while the TissueLoc has segmented a lot of background as the tissue which means low specificity. The Otsu and HistomicsTK methods have done well with their only problem not being conservative enough which results in minor tissue loss in their results.

3.4 Conclusion

Tissue background segmentation is crucial for patch selection which is required for training deep networks. In this chapter, we have compared the performance of U-Net with various custom topologies (backbones) for the identification of tissue regions in whole slide images. Using different networks combines the strength of current state-of-the-art CNNs with the custom architecture of the U-Net model for image segmentation. Whereas U-Net topologies can generate masks with 99% sensitivity, handcrafted methods struggled to approach high 80%. MobileNet and EfficientNet-B3 appeared to be the best backbone topologies for the U-Net.

The main contributions of this part are creating a manually labeled dataset consisting of 244 thumbnails, chosen from the publicly available TCGA dataset. As well, this chapter compared the performance of different encoders as the backbone of U-Net topology for the tissue segmentation task and proposing a tailored algorithm for this problem which can be used as a replacement for the tissue segmentation functions in the KimiaNet pipeline.

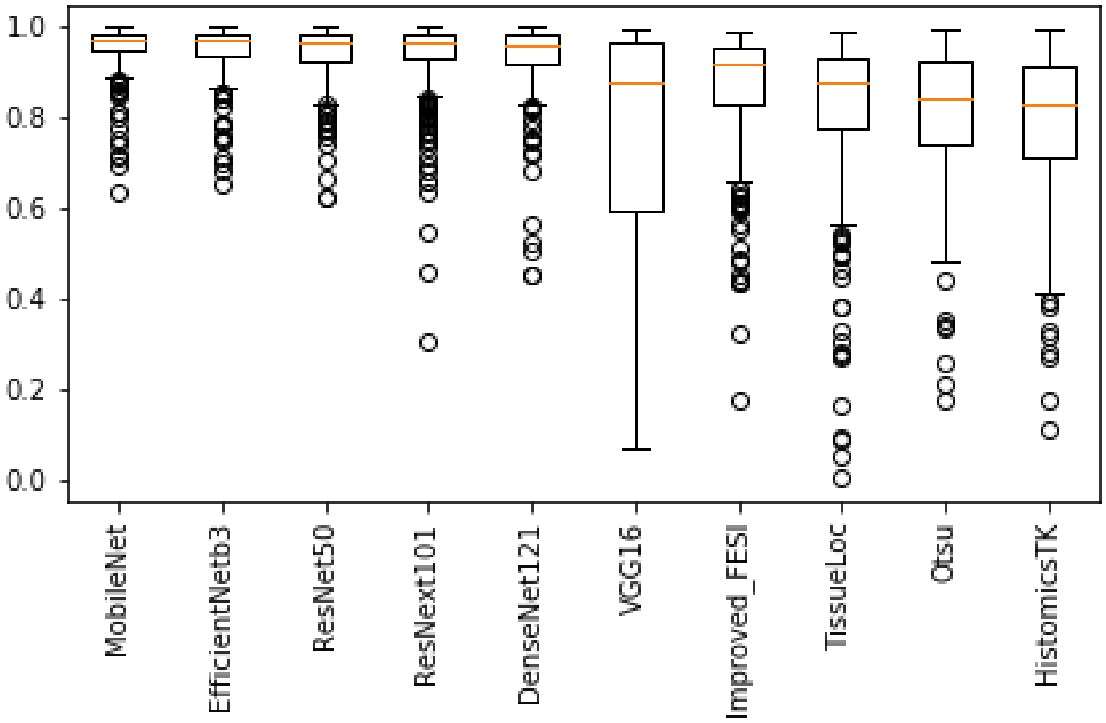


Figure 3.8: Jaccard Index Boxplot for different methods

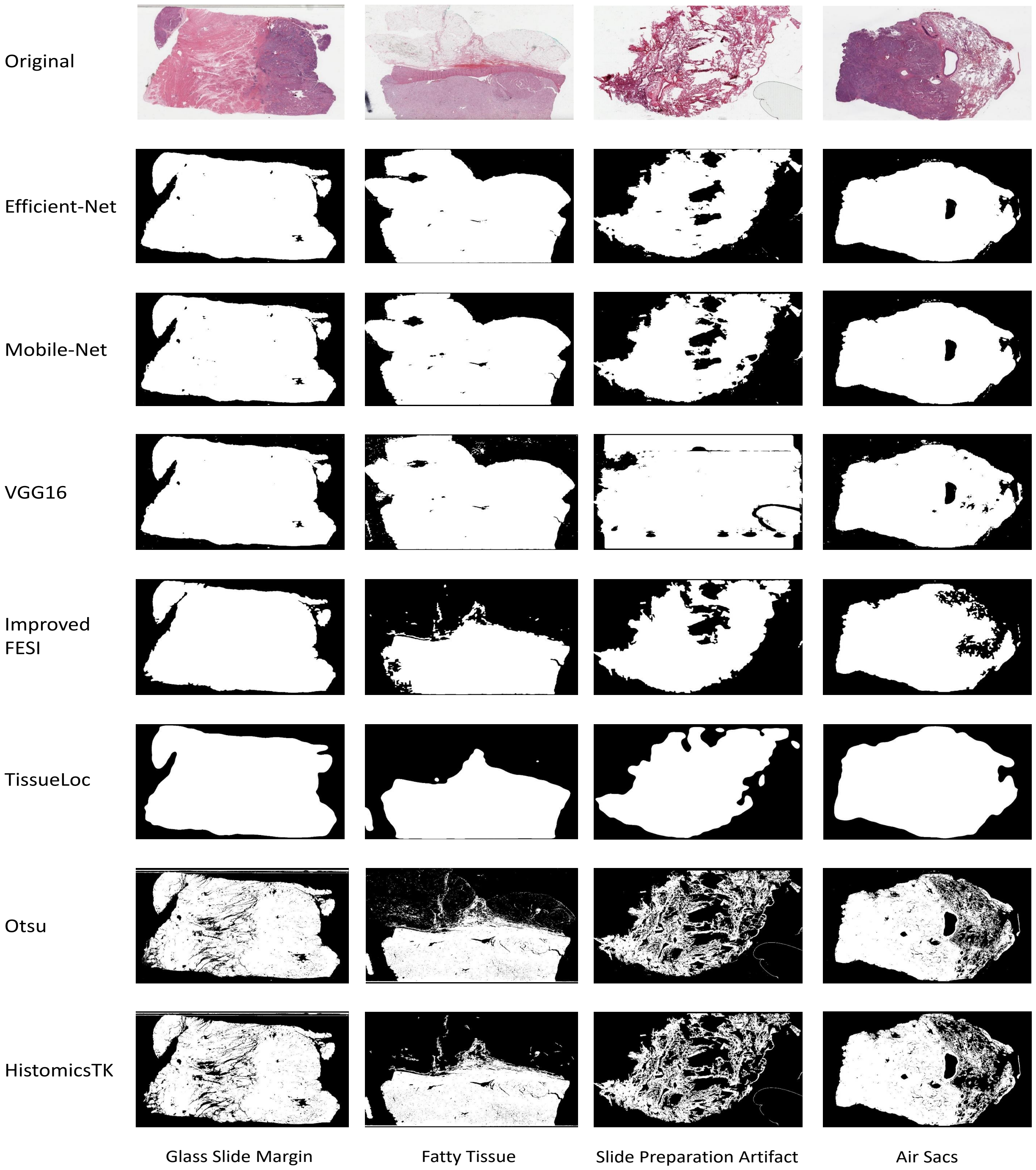


Figure 3.9: Example results of different methods in tissue segmentation task for challenging cases.

4. Data Preparation and Training

4.1 Public Image Datasets

Using public datasets has the benefit of reproducibility of research results in addition to the possibility of evaluating improvements or new methods against the current ones. Public datasets also help conduct high-quality research due to having a more realistic distribution of data [30, 35]. Two examples of publicly available databases in the area of histopathology, which could be used to create smaller datasets, are TCGA, containing 30,072 WSIs of different cancer types, and CAMELYON17 with 1000 WSIs of breast cancer (lymph node metastasis)[23].

TCGA repository (i.e., Genomic Data Commons, GDC) with 11,007 cases containing 30,072 WSIs overall is a publicly available repository [15, 41, 9]. The WSIs are depicting 25 organs (primary sites) with 32 cancer subtypes which can be seen in Table 4.1. Each case is associated with a lot of information such as "Morphology", "Primary Diagnosis", "Tissue or Organ of origin", "Patient Age at the time of Diagnosis", "Tumor Stage", "Gender", "Race" and "Ethnicity".

4.2 Creating a Reusable Dataset

The objective of this part was to create a general dataset for different research purposes rather than a dataset only limited to this research. So we had to impose some additional constraints. The first step was filtering out the frozen section biopsy WSIs due to their low quality which could result in negative effects on the learning process of the network. This means that only permanent section biopsy WSIs were used in this dataset and it is done by selecting the "Diagnostic Slides" option under the "Experimental Strategy" bar of the “Files” tab on GDC repository web site. Since the only option for the “Program” bar under the “Cases” tab is “TCGA”, we do not need to specify our program.

We tried to divide the data into the most detailed classes so it could be used for different research purposes by simply merging some classes with respect to one aspect or characteristic of the classes. Therefore, each class is made unique by a combination of three characteristics of the cases which are “morphology”, “primary diagnosis” and “tissue or organ of origin”. For example, a class of the dataset had the label of *['8520/3', 'Lobular carcinoma, NOS', 'Breast, NOS']* which means that its morphology code is 8520/3, its primary diagnosis is Lobular carcinoma and the tissue is taken from the Breast.

To clean the data, the cases having at least one of the three characteristics as 'Not Reported' or missing were removed. Next, classes with sample count of less than 20 WSIs were removed. The reason behind this was to have at least 2 samples for the test dataset for each class, assuming the test dataset ratio at 20%.

4.3 Dataset for KimiaNet

4.3.1 Dividing the Whole Slide Images

We used the general dataset explained in the last section with some changes and considerations for creating the train, validation and test datasets for this study. We chose the validation and test samples from the cases with only one WSI for the sake of simplicity of performance calculation. For each class of the general dataset, cases with a single WSI were separated and shuffled. Then two groups of samples with the size of ten percent of that class were added to validation and test datasets. Finally, the WSIs with missing magnification information or a magnification less than 20× were removed. This results in a test dataset with 744 slides accounting for around 10% of the data. The validation set had 741 slides, also almost 10% of the data. The rest of the data, containing 7,126 WSIs and about 80% of the whole dataset of permanent cases of TCGA after cleaning was assigned to the training set. While the chosen ratio for train, validation and test datasets is common in machine learning but the validation and test datasets could even be smaller due to the large size of the whole dataset. The reason behind this statement is that it is expected that a set randomly chosen from the a large dataset is more likely to capture its distribution.

We also used a different WSI labeling approach in the dataset which is based on the tumor types. Each case in TCGA dataset is assigned to one of the 32 tumor types shown in Table 4.1. Each tumor type is divided into a number of tumor subtypes, for example, Endocrine tumor type has the subtypes of Thyroid carcinoma (THCA), Adrenocortical carcinoma (ACC) and Pheochromocytoma and Paraganglioma (PCPG). Using tumor types is a common way of labeling the data in the recent studies [9] and it lets us do two types of search, (i) searching among all of the tumor types, i.e., horizontal search, and (ii) searching among tumor subtypes of a specific tumor type, i.e., vertical search. Methods of searching and evaluating the proposed algorithms will be discussed in detail in Chapter 5 (Experiments). It is worth mentioning that two of the classes of the 32 tumor types were removed during the process of data cleaning mentioned earlier in this subsection and also in section 4.2. The removed classes were UCEC (Uterine Corpus Endometrial Carcinoma) due to the missing morphology information, and DLBC (Lymphoid Neoplasm Diffuse Large B-cell Lymphoma) class for not having any sub-classes, with respect to the labeling scheme mentioned in section 4.2, with at least 20 cases[1].

4.3.2 Patch Extraction

Since the WSIs are too large to be fed to the neural networks, small size images within WSIs, called patches, should be extracted. We chose the size of the patches to be 1000 × 1000 at 20× magnification which is the largest size that we could feed to a neural network considering available computational resources. We chose to work at 20× magnification because 10× magnification could not provide enough details of tissue, e.g., nuclei details, while 40× magnification would result in a large number of patches and very long training time as a result. Figure 4.1 shows sample images for TCGA dataset.

For patch extraction from the test WSIs, first the 1× magnification of each WSI is extracted for preprocessing which results in locating the patches to be extracted[2]. The preprocessing at 1× magnification is done by first removing the red, blue and green markers from the thumbnail using a public library[3]. After that, the background pixels are removed by doing some process on the standard deviation of the three channel values of each pixel and performing thresholding on the output of moving an average kernel on the thumbnail in HSD color space. Finally, the non-overlapping patches are extracted by moving through the tissue background mask and checking whether 90% or more of the patch contains tissue,

[1] Implementation of the process explained in this subsection were done by Abtin Riasatian, with Morteza Babaie, Prof.Tizhoosh and Hany Kashani helping in discussions

[2] We cannot divide the entire WSI to patches since the number of patches would be too large prohibitively prolonging the learning process. Besides, o not all the patches are informative enough

[3] Python WSI Preprocessing

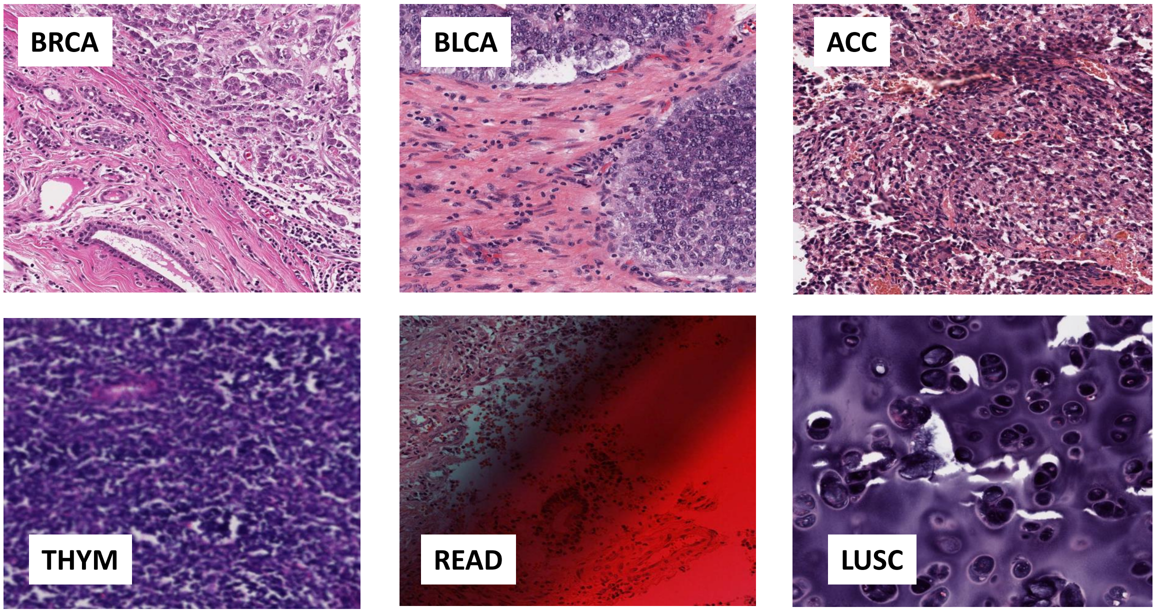


Figure 4.1: Sample patches from TCGA dataset (see Table 4.1 for abbreviations).

i.e., less than 10% background, and it is not low contrast. This resulted in 116,088 patches for the test dataset. This process was parallelized using the Dask python library[4].

Due to the large number of WSIs in the training and validation sets, we had to use a different method for patch extraction. We applied the mosaic generation of the Yottixel search engine [21]. This algorithm first partitions the 5× magnification of each WSI into 9 different regions based on their color decomposition using K-means algorithm (the value 9 is chosen empirically based on the maximum number of visually different tissue types from the perspective of a pathologist). After that, 15% of the patches of each partition is randomly extracted with the constraint of spatial diversity which was implemented by another K-means algorithm. These patches can be seen as a way to represent the WSI with less amount of data. However, they may not be suitable for training a reliable cancer feature extractor since not all of the patches are from cancerous areas in WSI (TCGA data

[4] The process explained in this subsection were implemented by Abtin Riasatian with the help of Kimia Lab members: (i) The code was parallelized by Shivam Kalra. (ii) Amir Safarpoor and Sobhan Shafiei's Tissue Segmentation and Contrast Checking codes were reused

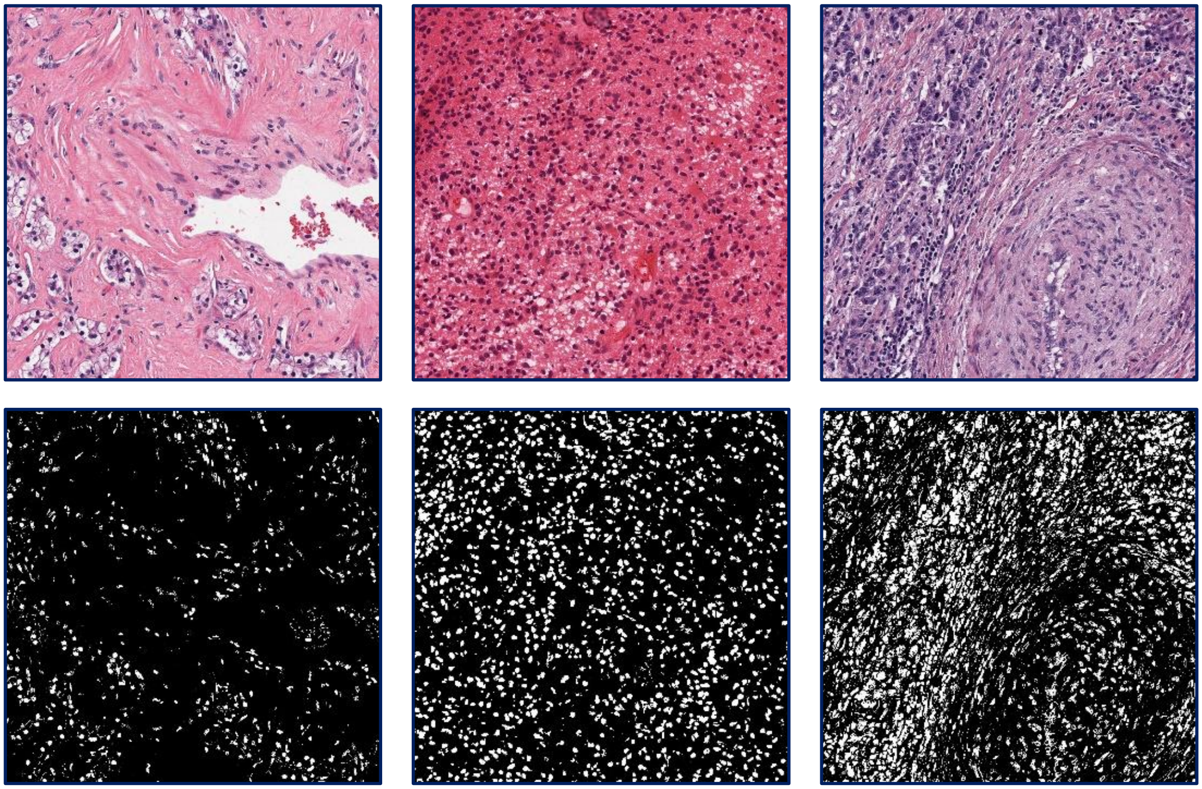
Algorithm 2 Modified Yottixel Algorithm

```
$m_I \leftarrow 20\times$                                   ▷ Magnification for indexing
$m_C \leftarrow 5\times$                                    ▷ Magnification for clustering
$l \leftarrow 1000$                                         ▷ Patch size $l \times l$ at $m_I$
$n_C \leftarrow 9$                                          ▷ Number of clusters at $m_C$
$p \leftarrow 15\%$                                         ▷ Mosaic percentage
$T_{\text{Cell}} \leftarrow 20\%$                           ▷ Top cases among sorted cellularity
$\mathbf{A} \leftarrow \text{readWSI}(fileName)$            ▷ Read an image
procedure YottixelIndex($\mathbf{A}, m_I, m_C, l, n_C, p, T_{\text{Cell}}$)
    $\mathbf{S} \leftarrow \text{Segment}(\mathbf{A}, m_C)$                 ▷ Separate tissue/background
    $P \leftarrow \text{Patching}(\mathbf{A}, \mathbf{S}, m_I, l)$          ▷ Get all patches
    $C \leftarrow \text{KMeansCluster}(P)$                                  ▷ Cluster patches
    $M \leftarrow \text{getMosaic}(C, p, \mathbf{A})$                       ▷ Select a mosaic
    $M' \leftarrow \text{cellMosaic}(M, T_{\text{Cell}})$                   ▷ Keep cell patches
    $F \leftarrow \text{Network}(M')$                                       ▷ Get features
    return $F$                                                              ▷ Set of features for $\mathbf{A}$
```

is not labeled). As most TCGA images depict high-grade carcinomas, a nuclei segmentation function was implemented to use only the top 20% of the patches with respect to their cell nuclei amount (assigning high cellularity to high-grade carcinoma). To measure the cell nuclei ratio of each patch, first color deconvolution is applied to convert the color space from RGB to hematoxylin and eosin using the HistomicsTK library. Then the hematoxylin channel was binarized using an empirical threshold[5] which results in the nuclei mask of the patch (Figure 4.2). The cell nuclei ratio of each patch is calculated by the number of "positive" pixels in the nuclei mas divided by the area of the patch. In the end, patches are sorted according to their cell nuclei ratio and the top 20% of them are chosen for the final dataset if their file size is larger than 110 KB to remove background patches (see Algorithm 2). This resulted in the final training and validation datasets with 242,202 and 24,646 patches, respectively[6]. Figure 4.3 depicts an example of patches extracted by the Yottixel algorithm and filtered with respect to their nuclei ratio in the next step.

[5] A possible improvement is to use Otsu thresholding

[6] The coordinates of these patches were provided by Shivam Kalra, the patch extraction and nuclei segmentation codes were implemented by Abtin Riasatian



KIRC
Nuclei Ratio: 13%

GBM
Nuclei Ratio: 41%

STAD
Nuclei Ratio: 61%

Figure 4.2: Examples of cell nuclei segmentation

4.3.3 Nuclei Ratio, a Heuristic for Patch Labelling

In the last subsection, it is mentioned that patches in training and validation datasets are chosen based on their nuclei ratio. Since these patches, which are expected to be the cancerous regions, will be used for training the network and are labeled using the tumor type information of the WSI that they are extracted from. This approach is a solution for the problem of labeling the patches and is used as a replacement for manually annotating the WSI which can only be done by a pathologist and is an expensive and time-consuming process. But what is the rationale behind this approach?

As uncontrolled cell growth is a common sign for carcinomas, mainly seen in areas with unusually high presence of cell nuclei (e.g., small cell carcinoma is extremely hypercellular),

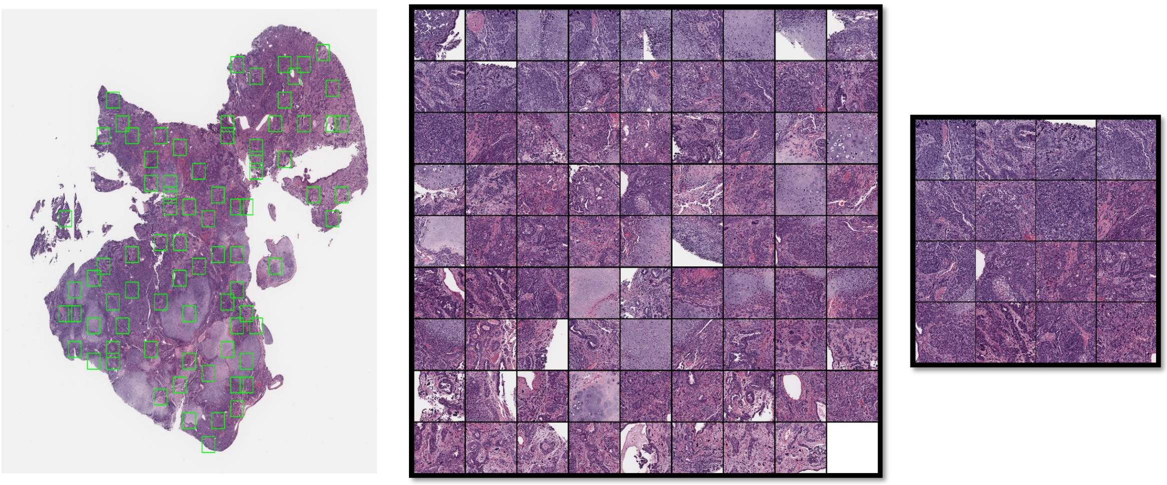


Figure 4.3: A WSI and its selected mosaic patches (left), Yottixel mosaic with 80 patches (middle), modified cellMosaic with 16 patches (right).

nuclei ratio can be used to filter out most of the benign/healthy patches [42]. We chose nuclei ratio to automate patch selection, an indirect or soft way of patch labeling, as this is one of the common features of cancer that spans most neoplasms, especially high-grade carcinomas. This means that we can use nuclei ratio to select patches with a higher probability of malignant, i.e., cancerous. However, high nuclei ratio can also be observed in inflammatory tissue and non-neoplastic cell types in some cases [13] but we believe that the advantages would outweigh the drawbacks of this approach. Since a WSI contains several thousand cells with many of them being cancerous in the TCGA dataset, it is very unlikely to completely miss tumors by patching a WSI.

4.4 Training

We trained/fine-tuned the DenseNet-121 architecture with four different settings to be able to investigate the effect of fine-tuning a network with a dataset tailored for histopathology on its performance compared to the available general-purpose networks. So we fine-tuned the last dense block, the last two dense blocks and the last three dense blocks of DenseNet-121 architecture and named them KimiaNet-I, KimiaNet-II, KimiaNet-III, respectively. We also trained the whole DenseNet-121 which we call KimiaNet-IV (all weights are re-

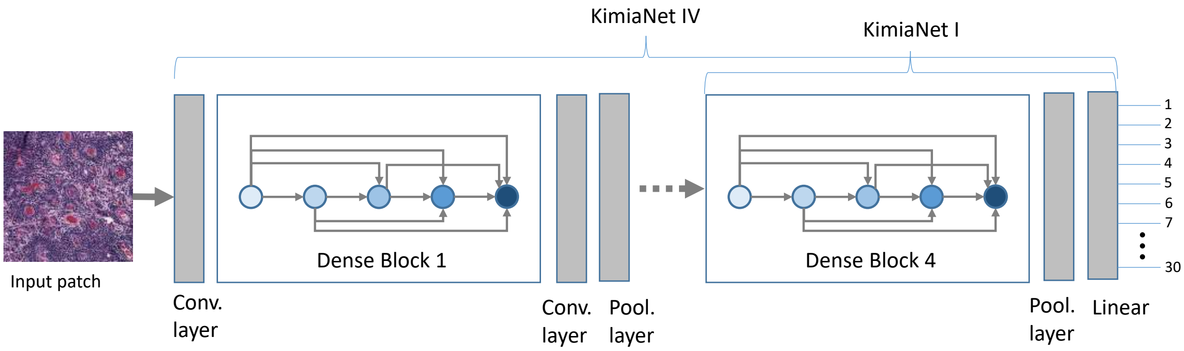


Figure 4.4: DenseNet architecture of KimiaNet.

learned). The general architecture of the KimiaNets can be seen in Figure 4.4

The Pytorch framework was used to implement the training and testing of the networks. Each model was trained/fine-tuned on 4 Tesla V100 GPUs with 32GB memory for each GPU. The size of batches were set to 256, 128, 128 and 64 for KimiaNet-I, KimiaNet-II, KimiaNet-III and KimiaNet-IV, respectively. Each network was trained for about 20 epochs with the early stopping conditioned on three consecutive decreases in the validation accuracy. The epoch time for models I to IV was around 60, 75, 90 and 110 minutes, respectively. Adam optimizer was used for all models with initialized the learning rate of 0.0001 and scheduled to decrease the learning rate by a factor of ten ($\gamma = 0.1$) every 5 epoch. Each model was initialized with ImageNet pre-trained weights. The input of the network was batches of 1000×1000 patches of $20\times$ magnification and one of the 30 classes of tumor types was assigned to each patch as its label (Table 4.1). The loss function was set to cross-entropy for training these classifiers[7].

4.5 Conclusion

In this chapter, the creation of a dataset for training KimiaNet was discussed. This dataset consists of 242,202, 24,646 and 116088 patches of size 1000×1000 pixels for training, validation and test datasets, respectively, extracted at $20\times$ magnification from 7,126, 741 and 744 WSIs. These WSIs were extracted from the publicly available TCGA repository. The

[7] Implemented by Kimia Lab member Danial Maleki

label of each WSI is employed as a soft label for its patches using a proposed clustering-based mosaic approach with a constraint on high cellularity to facilitate the training. To investigate the effect of fine-tuning a network using histopathology images on the performance of histopathology image retrieval, the DenseNet-121 topology was trained in four configurations, namely KimiaNet I, II, III and IV, in which the last dense block, the last two dense blocks, the last three dense blocks and the whole network was trained, respectively.

Code	Primary Diagnosis	#Patients
ACC	Adrenocortical Carcinoma	86
BLCA	Bladder Urothelial Carcinoma	410
BRCA	Breast Invasive Carcinoma	1097
CESC	Cervical Squamous Cell Carcinoma and Endocervical Adenoc.	304
CHOL	Cholangiocarcinoma	51
COAD	Colon Adenocarcinoma	459
DLBC	Lymphoid Neoplasm Diffuse Large B-cell Lymphoma	48
ESCA	Esophageal Carcinoma	185
GBM	Glioblastoma Multiforme	604
HNSC	Head and Neck Squamous Cell Carcinoma	473
KICH	Kidney Chromophobe	112
KIRC	Kidney Renal Clear Cell Carcinoma	537
KIRP	Kidney Renal Papillary Cell Carcinoma	290
LGG	Brain Lower Grade Glioma	513
LIHC	Liver Hepatocellular Carcinoma	376
LUAD	Lung Adenocarcinoma	522
LUSC	Lung Squamous Cell Carcinoma	504
MESO	Mesothelioma	86
OV	Ovarian Serous Cystadenocarcinoma	590
PAAD	Pancreatic Adenocarcinoma	185
PCPG	Pheochromocytoma and Paraganglioma	179
PRAD	Prostate Adenocarcinoma	499
READ	Rectum Adenocarcinoma	170
SARC	Sarcoma	261
SKCM	Skin Cutaneous Melanoma	469
STAD	Stomach Adenocarcinoma	442
TGCT	Testicular Germ Cell Tumors	150
THCA	Thyroid Carcinoma	507
THYM	Thymoma	124
UCEC	Uterine Corpus Endometrial Carcinoma	558
UCS	Uterine Carcinosarcoma	57
UVM	Uveal Melanoma	80

Table 4.1: The TCGA codes (in alphabetical order) of all 32 primary diagnoses and corresponding number of evidently diagnosed patients in the dataset

5. Experiments

After fine-tuning/training KimiaNet models I to IV, we designed a number of experiments to assess the performance of these models as a feature extractor, i.e., how representative these features are. In all of the following experiments, the features are extracted from the last pooling layer of the network.

To evaluate the quality of the features extracted by the models trained, two types of experiments have been done on the TCGA dataset, i.e., horizontal and vertical search, which will be discussed in the following subsections[1].

5.1 Horizontal Search and Analysis of the Results

In horizontal search, we measure the accuracy of the algorithm in finding the WSI with a similar tumor type to the query WSI among all WSIs in the dataset.

In this experiment, the following tasks were performed for networks KimiaNet I to IV and DenseNet-121 with ImageNet pre-trained weights. First, features are extracted for each patch in the TCGA test dataset (116,088 patches). Then these features are barcoded. Barcoding is the process of binarization of the features based on moving a 1-dimensional window with the size of 2 on the features and outputting value one if the value on the left was smaller or equal to the value on the right and zero otherwise [39, 21]. An example of barcoding is illustrated in diagram 5.1.

After that we used the "leave-one-out" approach by repeatedly iterating over all WSIs, taking one as the query WSI and the rest as the database. The distance between the query WSI and the rest of the WSIs were calculated based on the "median-of-min" approach.

[1] A code implemented by Shivam Kalra was reused for these experiments. The adaptation of the code to the project, some debugging and implementing additional features were done by Abtin Riasatian with Morteza Babaei's help in relevant discussions. The codes were mostly executed by Morteza Babaei.

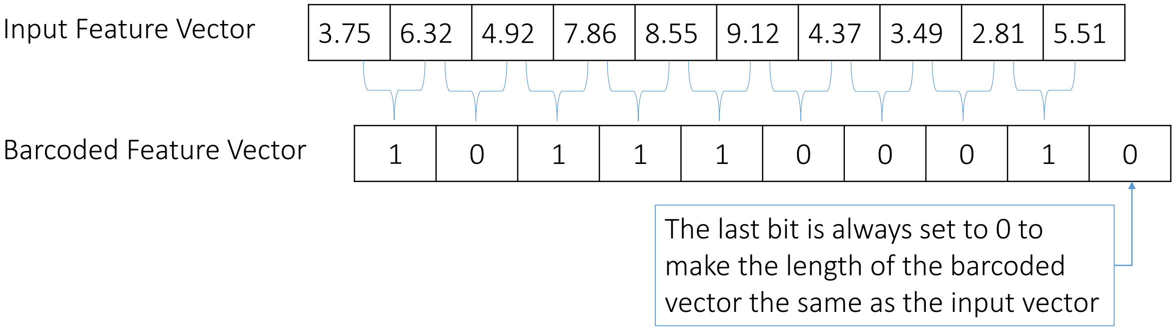

Input Feature Vector
3.75
6.32
4.92
7.86
8.55
9.12
4.37
3.49
2.81
5.51
Barcoded Feature Vector
1
0
1
1
1
0
0
0
1
0
The last bit is always set to 0 to make the length of the barcoded vector the same as the input vector

Figure 5.1: An example of barcoding a feature vector with the length of 10

In this method, the minimum Hamming distance between the barcoded features of each patch of the query WSI and the barcoded features of all patches of each of the WSIs in the database is calculated and then the medium of all minimum distances is taken as the distance between the query WSI and another WSI. Then the 3-Nearest Neighbours of the query WSI are considered as the suggestions of the algorithm for WSIs similar to the query. The label of the majority determines the label for the query WSI. The results of this experiment are reported in Table 5.1 and Diagram 5.2

As Table 5.1 demonstrates, KimiaNet I, in which only the last dense block was fine-tuned, has a considerable increase in performance (with accuracy 73.6 ± 18.0) compared to DenseNet-121 initialized with ImageNet weights (with accuracy 44.8 ± 19.1). As more blocks are fine-tuned the average accuracy increases and their standard deviation decreases, with performances 76.2 ± 16.6, 81.8 ± 13.3 and 85.4 ± 11.1 for KimiaNets II to IV. To compare the model with the best results, KimiaNet IV with maximum accuracy for all tumor types except one, against pre-trained DenseNet, as a commonly used image feature extractor and classifier, we can see that the performance is, on average, improved by 40.7 ± 12.9. The maximum amount of improvement was for melanocytic tumor type, 68%, while brain that had the lowest level of improvement, 27%, reached 99% in KimiaNet IV. Figure 5.3 compares results of DenseNet and KimiaNet (IV) for an example query WSI.

Tumor Type	Patient #	DN	I	II	III	IV	diff
Brain	74	72	96	97	99	99	+27
Breast	91	53	86	87	91	91	+38
Endocrine	72	65	86	89	93	92	+28
Gastro.	88	53	74	81	80	84	+31
Gynaec.	30	13	43	40	47	57	+44
Head/neck	32	25	75	69	81	88	+63
Liver	51	43	67	69	80	88	+45
Melanocytic	28	18	57	54	75	86	+68
Mesenchymal	13	23	38	62	69	69	+46
Prostate/testis	53	57	89	91	94	96	+39
Pulmonary	86	56	83	86	85	86	+30
Urinary tract	123	59	89	89	88	89	+30

Table 5.1: 3-Nearest Neighbors accuracy (%) for the horizontal search among 744 WSIs for differently fine-tuned/trained KimiaNet. The best results are highlighted. The last column shows the improvement of accuracy (%) through KimiaNet compared to DenseNet.

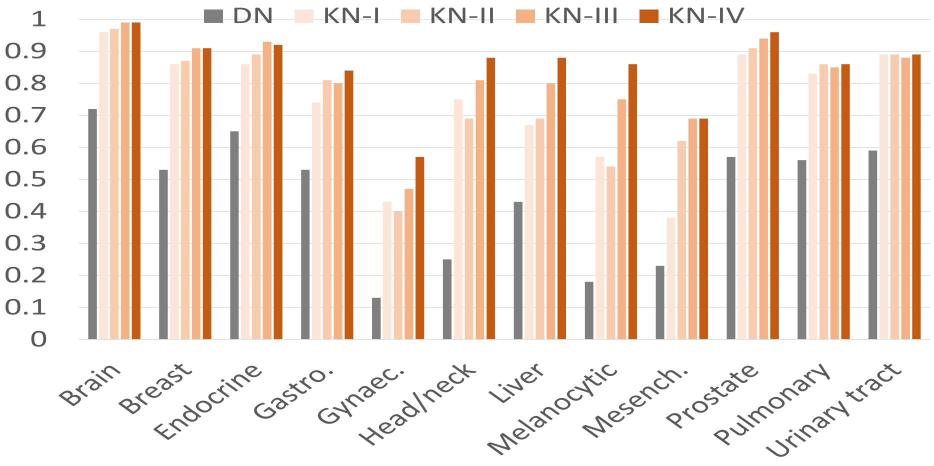


Figure 5.2: Horizontal search results (accuracy, in percentage) for TCGA data.

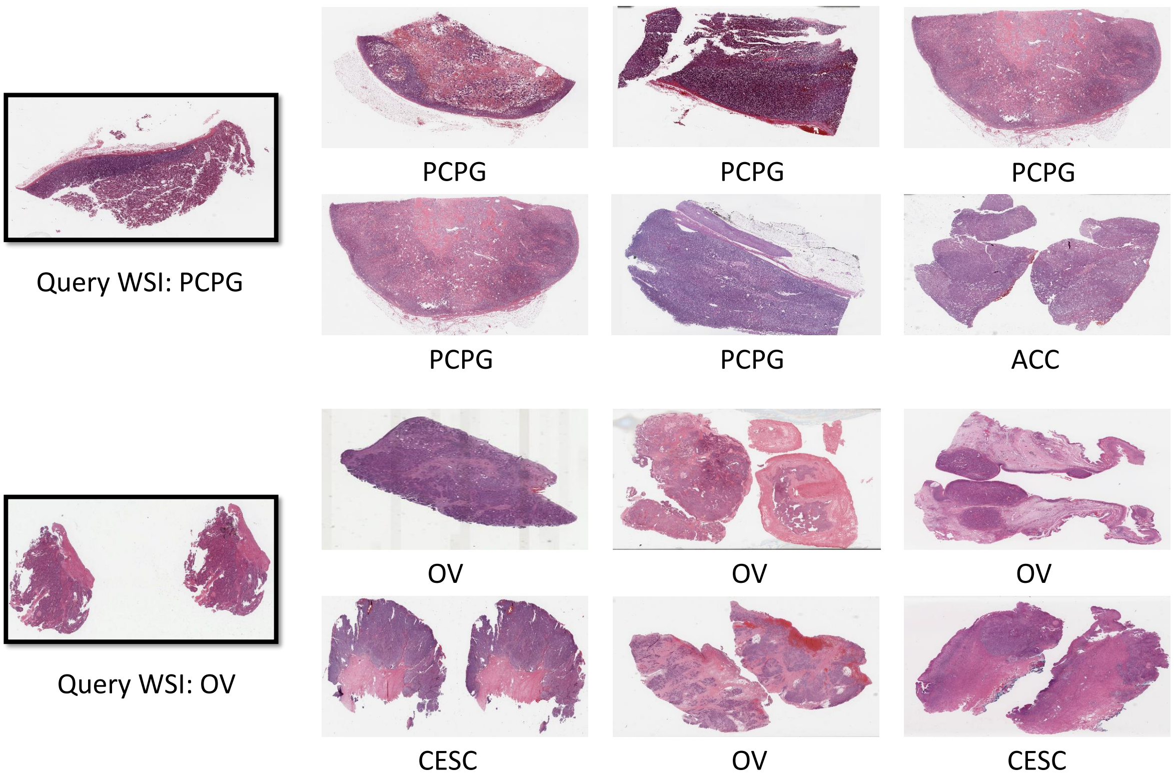


Figure 5.3: Results for two sample query WSIs (left): Corresponding search results based on KimiaNet IV features (top row for each query WSI) and DenseNet features (bottom row for each query WSI) and their assigned TCGA primary diagnosis. For TCGA project IDs see Table 4.1

5.2 Vertical Search and Analysis of the Results

In vertical search, we measure the performance of the algorithm in finding the WSI with the similar tumor subtype to the query WSI among all WSIs of the same tumor site in the dataset. For example, a query WSI is given with the subtype label "Low Grade Glioma" (LGG) which is a brain tumor. So the algorithm is supposed to search among all WSIs in the same tumor site, brain in this case, and suggest WSIs with the same tumor subtype, namely LGG. In vertical search, the discriminative power of the feature extractor for distinguishing the subtypes of tumors in the same site is evaluated, in this example distinguishing LGG WSIs from the other brain tumor, namely "Glioblastoma Multiforme"

(GBM).

This experiment was performed similar to the experiment for the horizontal search (leave-one-out validation) except the search space for each query WSI was restricted to the same tumor site as the query. Also, we realized that accuracy may not be a good metric to evaluate the performance of this type of search; if there is a biased algorithm which classifies all WSIs of the brain site as GBM, the accuracy for GBM would be 100% while the accuracy for LGG is 0%. These results are not easy to translate and cannot show how well the algorithm is performing (balance between sensitivity and specificity). For this reason, we chose to use the F1-measure as the evaluation metric for this type of search where for each subtype, the query subtype is taken as the True label and all other subtypes are considered as the False label. The F1-measure is a harmonic mean of sensitivity and specificity [8].

The results of each model for the vertical search are shown in Table 5.2. As the results suggest, KimiaNet models, especially III and IV, have considerably improved the performance of the pre-trained DenseNet-121. Although the model with the most number of maximum values was KimiaNet IV, with 15 maximum values, KimiaNet III had also a good performance with 13 maximum values. Also, the performance improvements, from the pre-trained DenseNet-121, have been greater for the subtypes with few slides. For example, the F1-score for Uveal Melanoma (UVM), with only 4 slides, was remarkably increased from 0% to 86% in KimiaNet III and 67% in KimiaNet IV. Another example is Mesothelioma (MESO), having only 5 slides, rose from 0% to 33% and 75% in KimiaNets III and IV, respectively. This shows that the KimiaNet models can perform well even on the low data regimes while getting better results than the available feature extractors. For example, F1-scores for Stomach Adenocarcinoma (STAD) and Lung Squamous Cell Carcinoma (LUSC), having 30 and 43 slides, have been significantly enhanced from 63% and 69% in pre-trained DenseNet-121 to 86% and 84% in KimiaNet IV, respectively.

To compare the discrimination power of the features extracted by KimiaNet-IV against the pre-trained DenseNet-121, a fixed number of patches were randomly chosen from each tumor type and fed to both networks. To be able to visualize the features in 2-d, their dimensionalities were reduced using the t-SNE method. Figures 5.4 and 5.5 demonstrate the huge affect of training DenseNet-121 with histopathology data. As can be observed, the features of different tumor types generated by KimiaNet-IV are clearly separated from each other while almost none of the tumor types have a specific boundary for the features generated by DensetNet-121.

Site	Subtype	n_{slides}	DN	I	II	III	IV
Brain	LGG	39	71	75	82	85	81
Brain	GBM	35	77	73	80	83	81
Endocrine	THCA	51	94	98	98	99	100
Endocrine	ACC	6	25	25	20	55	44
Endocrine	PCPG	15	57	75	73	80	85
Gastro.	ESCA	14	50	73	50	83	78
Gastro.	COAD	32	65	76	75	75	76
Gastro.	STAD	30	63	77	73	84	86
Gastro.	READ	12	22	30	26	29	30
Gynaeco.	UCS	30	75	86	60	75	86
Gynaeco.	CESC	17	88	97	84	97	94
Gynaeco.	OV	10	67	89	74	95	95
Liver, panc.	CHOL	4	29	40	31	40	40
Liver, panc.	LIHC	35	86	94	87	97	96
Liver, panc.	PAAD	12	70	73	56	82	76
Melanocytic	SKCM	24	92	94	94	98	94
Melanocytic	UVM	4	0	40	40	86	67
Prostate/testis	PRAD	40	99	100	99	100	100
Prostate/testis	TGCT	13	96	100	96	100	100
Pulmonary	LUAD	38	65	73	72	69	78
Pulmonary	LUSC	43	69	74	74	75	84
Pulmonary	MESO	5	0	0	0	33	75
Urinary tract	BLCA	34	90	96	93	93	93
Urinary tract	KIRC	50	83	95	99	97	97
Urinary tract	KIRP	28	77	91	91	91	91
Urinary tract	KICH	11	48	86	78	84	86

Table 5.2: *k*-NN results, k=3, for the vertical search among 744 WSIs. The best results are highlighted. F1-measure has been reported here instead of simple classification accuracy. For TCGA codes see Table 4.1 in Appendix.

5.3 Conclusion

In this chapter, two types of search, namely horizontal and vertical search, were conducted in order to have a measurement of the effect of fine-tuning the DenseNet-121 network with histopathology images on the histopathology image search task. In the horizontal search experiment, KimiaNet-IV achieved the highest accuracy for all of the tumor types (except for "Endocrine") with 92% having only 1% gap with the highest accuracy reached by KimiaNet-III. Also, the results of the vertical search experiment illustrated that the features generated by KimiaNet-IV are more informative than the other KimiaNet versions and pre-trained DenseNet-121. Additionally, the t-SNE plots showed that the deep features generated by KimiaNet form visually distinguishable clusters where each cluster is associated with a cancer sub-type. This is while the DenseNet-121's deep features of different classes were mostly mixed together. Moreover, KimiaNet-IV was evaluated using two external datasets, "Endometrium" [37] and "Colorectal" [22], on the classification task in which it achieved considerable results.

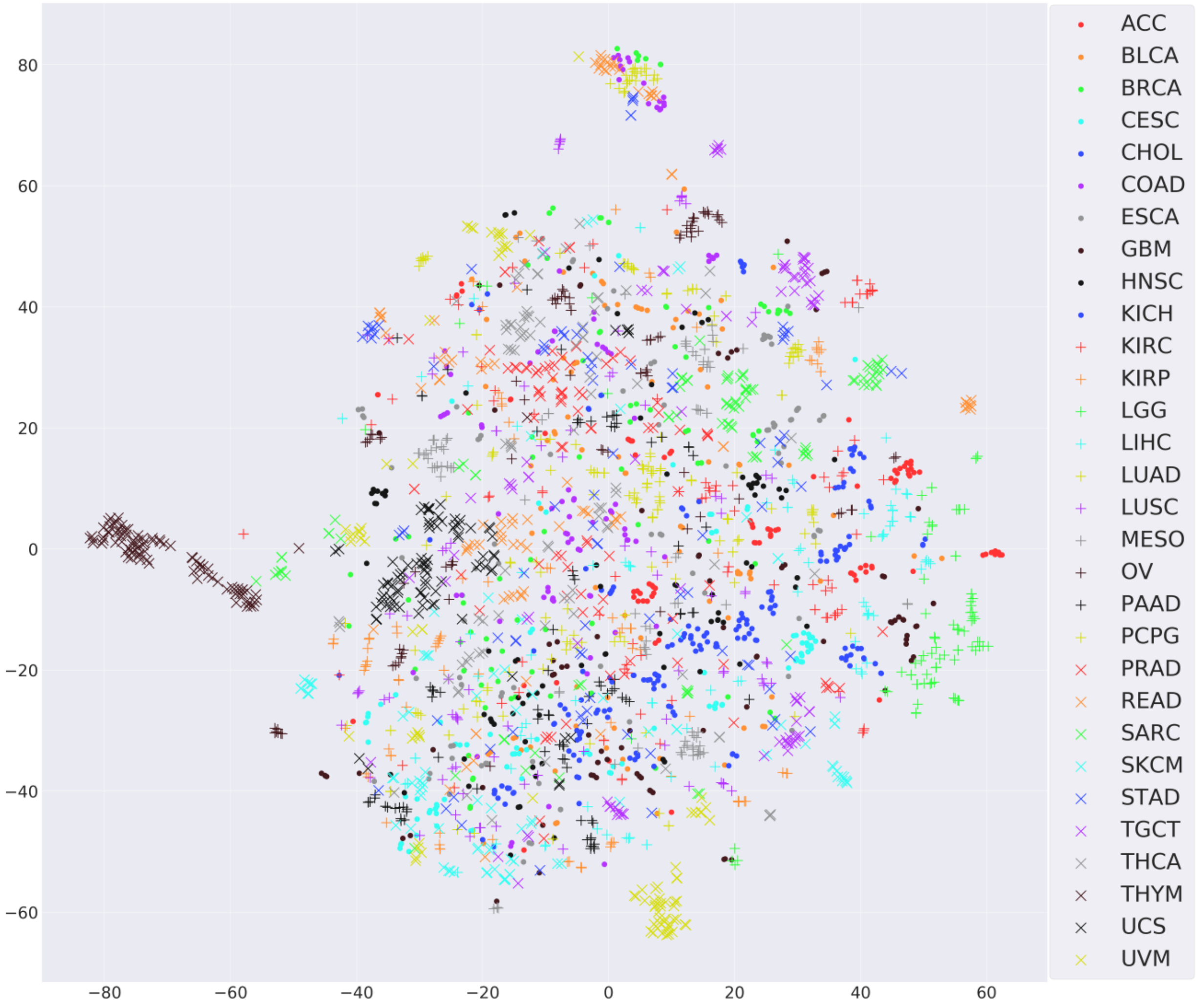


Figure 5.4: t-SNE visualization of randomly selected test patches for DenseNet: no trend of distinct clusters of cancer subtypes is visible.

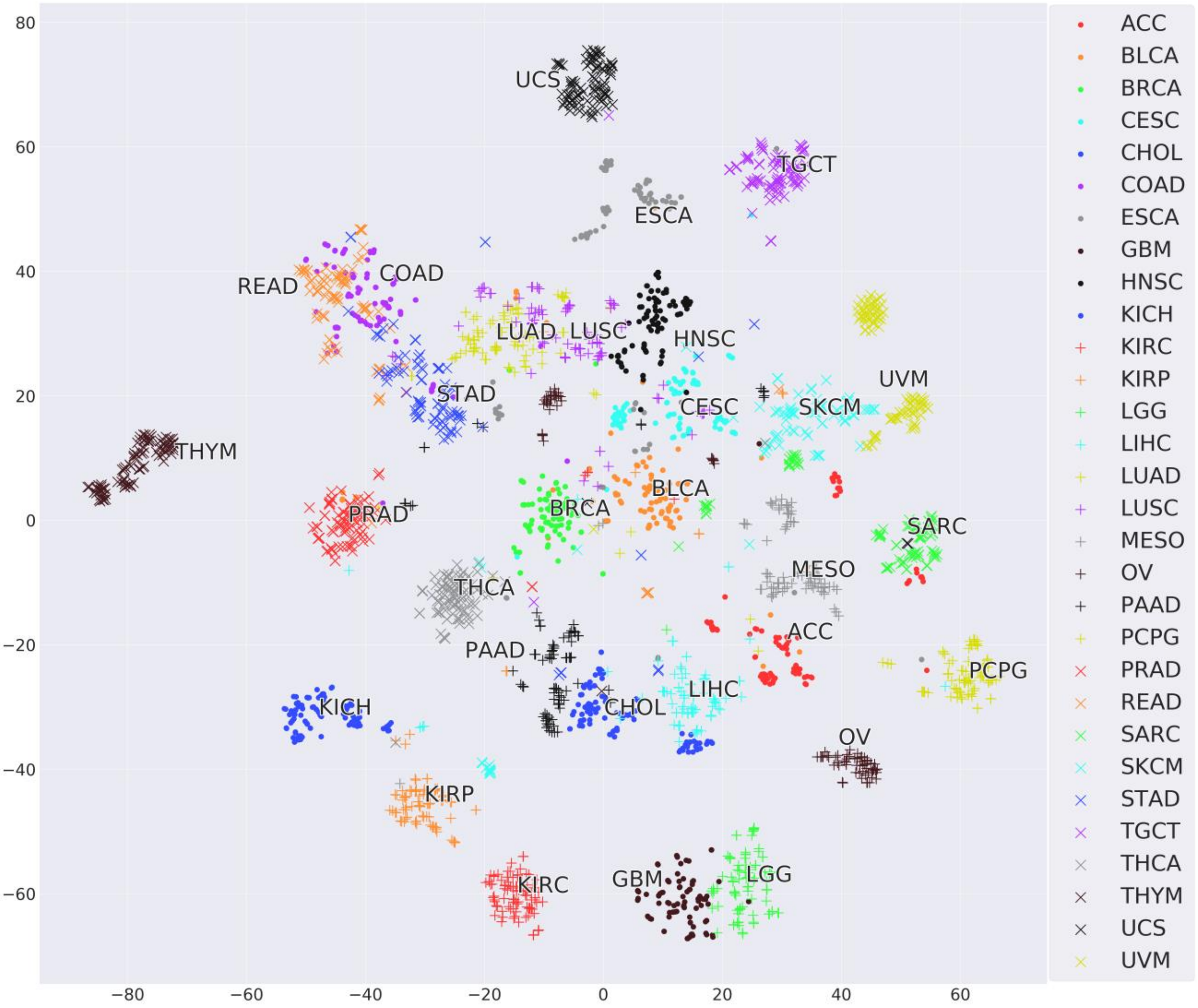


Figure 5.5: t-SNE visualization of randomly selected test patches for KimiaNet: a clear trend of distinct and separable clusters of cancer subtypes is visible.

6. CBR Networks: DenseNet versus Occam's Razor

6.1 Motivation

Occam's Razor, as one of the basic scientific principles, postulates that smaller/simpler solutions tend to be the right solution. A recently published paper by Raghu et al. [31] suggests that transfer learning improves the performance of the algorithm negligibly and also small networks with simple designs can compete with the commonly used architectures, mentioned as ImageNet architectures. Particularly, in medical imaging tasks such as classification of retinal fundus and chest x-ray images, simpler networks may perform better which is due to the fundamental differences of these tasks with the classification of natural images (see Figure 6.1). These differences are: (i) In medical imaging tasks the image has mostly a similar pattern and decisions are made based on local variations while in the natural images, for example, image of a cat, the boundaries are clear, and (ii) The number of classes in medical imaging tasks are much smaller than the number of classes in natural image classification. For example, in experiments mentioned in 5.1 and 5.2, the number of classes were 30 while the ImageNet dataset has 1000 classes. This means that probably the last layers of the ImageNet architectures are "*overparameterized*" for medical imaging tasks. (iii) In addition, medical image datasets mostly have smaller sizes compared to natural image datasets which could be caused by the high expense of creating these datasets.

In this chapter, a family of simple architectures is proposed which are made from the repetitions of convolution, batch normalization and ReLU layers (short CBR networks). Reported experiments in literature test the performance of four variations of the CBR networks against common networks such as ResNet-50, with 23.5 Million weights, and Inception-V3, with 23 Million weights [31]. The results show that a network with a simple

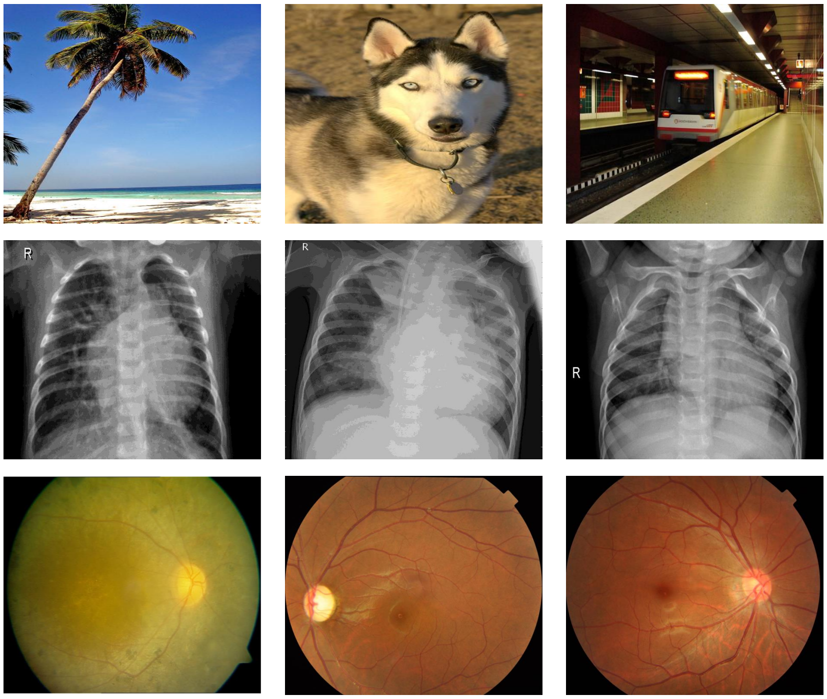


Figure 6.1: Examples of natural images, taken from [12], chest x-rays, taken from [27] and retinal fundus images, taken from [25]

design called Small CBR with only 2 Million weights could compete with the mentioned large networks in the classification tasks for retinal fundus and x-ray images. However, the DenseNet-121 topology with 7 Million parameters, which is smaller than two of the CBR networks with more than 8 Million parameters, was not investigated in the literature.

6.2 Experiments

We first implemented the two CBR architectures called Tiny and Small exactly as stated in the literature [31] [1]. We tried adding a dense layer before the last fully connected layer which resulted in the improvement of the performance of the CBR networks. So the experiments were done adding this change to the CBR architectures Small, LargeT,

[1] Implemented by Abtin Riasatian

LargeW. We also did experiments with a modified version of the Small network which had an additional (fifth) CBR block in its topology[2]. The number of parameters for the CBR networks can be found in Table 6.1. The networks were initialized with random weights and trained from scratch for around 40 epochs using the same TCGA dataset as used for KimiaNet models. All settings were the same as for the training of the KimiaNet models except that the learning rate was initialized with 0.003 and the batch size was 32 (see Subsection 4.4). After training the networks, both horizontal and vertical search methods were performed on the TCGA test dataset, same as KimiaNet models, using these networks as feature extractors. The results will be compared against pre-trained DenseNet-121 and KimiaNet-IV in the next section.

Network	Number of Parameters
CBR Small	2 Millions
CBR Modified Small	5.5 Millions
CBR LargeT	8.5 Millions
CBR LargeW	8.5 Millions
DenseNet-121	7 Millions
KimiaNet-IV	7 Millions

Table 6.1: A comparison between the number of parameters of CBR networks, DenseNet-121 and KimiaNet-IV

6.3 Analysis of the Results

For the horizontal search, as it can be observed in Table 6.2, KimiaNet-IV has achieved the maximum accuracy for all of the tumor types. The performance of CBR Small and CBR Modified Small networks slightly better than the pre-trained DenseNet-121 with accuracy mean and standard deviations of 48 ± 20 and 46 ± 22 against 45 ± 20, respectively. Considering the large number of parameters of both CBR Large networks, around 8.5 Million, compared to DenseNet-121 and KimiaNet-IV, both with 7 Million weights, their results are not competitive enough, having a large gap in performance with KimiaNet-IV. This shows that not only the KimiaNet architecture is not overparameterized, but its design, for example densely connected layers, has helped it to make better use of its parameters compared to the simple design of CBR networks.

[2] Implemented by Danial Maleki

Tumor Type	Patien #	Small	Modified Small	LargeT	LargeW	DN	KN-IV
Brain	74	64	66	72	73	72	99
Breast	91	59	65	74	68	53	91
Endocrine	72	65	57	76	65	65	92
Gastro.	88	58	66	67	63	53	84
Gynaec.	30	30	27	37	27	13	57
Head/neck	32	47	59	63	56	25	88
Liver	51	31	29	39	31	43	88
Melanocytic	28	32	18	32	36	18	86
Mesenchymal	13	0	0	15	0	23	69
Prostate/testis	53	62	58	68	62	57	96
Pulmonary	86	56	49	56	51	56	86
Urinary tract	123	66	60	66	63	59	89

Table 6.2: Horizontal Resutls for CBR

Almost the same pattern can be seen in Table 6.3 for the vertical search. KimiaNet-IV has achieved the maximum F1-score for 23 subtypes out of 26 while the CBR Large networks could only achieve 2 or 3 maximum values. Also, the mean F1-Score and the standard deviation for KimiaNet-IV is 84±16 while this number is 74±21 and 71±23 for the LargeT and LargeW networks, respectively. This means that KimiaNet-IV accuracy is both higher and more stable than the CBR results while having around 1.5 Million parameters more. An interesting point is the close performance of the CBR Small network, 70 ± 23, to the CBR LargeT, 71 ± 23 which shows that the simple design of the CBR networks can not leverage the full power of its parameters.

6.4 Conclusion

In this chapter, the idea of replacing the DenseNet-121 architecture with smaller/simpler networks with negligible loss in performance was investigated. To this end, four simple networks, the so-called CBR networks, were implemented and tested against both DenseNet-121 and KimiaNet. The results showed that KimiaNet features are better than the CBR networks with a considerable gap. These simple networks performed only slightly better than the pre-trained DenseNet. It can be concluded that KimiaNet appears to be in compliance with the Occam's Razor within the spectrum of tested topologies; it is as small as necessary to be reliably accurate.

Site	Subtype	n_{slides}	Small	Modified Small	LargeT	LargeW	DN	KN-IV
Brain	LGG	39	76	77	79	80	71	81
Brain	GBM	35	69	72	75	76	77	81
Endocrine	THCA	51	97	96	97	98	94	100
Endocrine	ACC	6	0	0	0	0	25	44
Endocrine	PCPG	15	71	69	71	71	57	85
Gastro.	ESCA	14	64	50	70	64	50	78
Gastro.	COAD	32	70	73	78	74	65	76
Gastro.	STAD	30	67	74	73	74	63	86
Gastro.	READ	12	42	45	48	32	22	30
Gynaeco.	UCS	30	75	100	75	67	75	86
Gynaeco.	CESC	17	91	92	82	94	88	94
Gynaeco.	OV	10	82	82	67	82	67	95
Liver, panc.	CHOL	4	22	22	22	40	29	40
Liver, panc.	LIHC	35	89	92	90	90	86	96
Liver, panc.	PAAD	12	57	67	64	67	70	76
Melanocytic	SKCM	24	92	90	94	92	92	94
Melanocytic	UVM	4	50	0	40	33	0	67
Prostate/testis	PRAD	40	96	96	96	99	99	100
Prostate/testis	TGCT	13	88	88	88	96	96	100
Pulmonary	LUAD	38	46	51	57	57	65	78
Pulmonary	LUSC	43	51	59	55	60	69	84
Pulmonary	MESO	5	57	29	50	57	0	75
Urinary tract	BLCA	34	93	92	90	92	90	93
Urinary tract	KIRC	50	88	91	86	90	83	97
Urinary tract	KIRP	28	82	77	81	82	77	91
Urinary tract	KICH	11	73	80	76	76	48	86

Table 6.3: Vertical Resutls for CBR

7. Summary and Conclusions

Deep learning has achieved considerable progress in recent years. Computer vision, in particular medical image analysis has also benefited from this advancement. Using pre-trained deep networks is established as an off-the-shelf solution for image representation. Natural images in *ImageNet*, a large repository of natural images, have been frequently used to train such networks with different architectures such as VGG, Inception, ResNet and EfficientNet. Whereas it is common knowledge that fine-tuning or training deep networks are expected to yield better results, employing pre-trained networks is quite ubiquitous. This is largely due to the lack of labeled data, design challenges, and computational costs of training deep networks.

In this thesis, the DenseNet-121 architecture, a commonly used pre-trained deep network, was fine-tuned/trained with different configurations to study the effect of fine-tuning a network with domain-specific images on the expressiveness of the generated deep features. Consequently, four different models have been trained, starting from only fine-tuning the last dense block of the DenseNet-121 and continue to unfreeze the parameters of more dense blocks, i.e., the last two dense blocks, the last three dense blocks and the whole network. These networks were named KimiaNet-I to KimiaNet-IV, respectively.

One of the main challenges of training a network with histopathology images is their very large size, a digitized slide could be as large as 100,000 by 100,000 pixels. This means histopathology whole slide images, WSIs, cannot be processed by the networks as they are. To deal with this problem, these large WSIs are divided into smaller sub-images called *patches* with appropriate size for training. However, this arises two problems: (i) The number of patches may still be too large for many WSIs; the training process would take a prohibitively long time. (ii) Due to the high cost of manual delineation of the slides or patches (to highlight regions of interest, e.g., cancerous regions) expecting labeled image is not practical in the large archives hospitals. In this study, these two problems are resolved by proposing a clustering-based mosaic approach for WSI representation with a constraint on high cellularity to employ the WSI label as a soft label for patches to facilitate the training.

The source of data for training *KimiaNet*, the DenseNet topology for histopathology, is the publicly available TCGA repository which contains a 11,766 diagnostic digital slides. These slides were first divided into patches which are grouped into different clusters based on their staining (color) features. A total of 15% of the patches of each cluster were randomly chosen. To filter out the patches with low cellularity, a sequence of color deconvolution and thresholding was performed. These patches form a representative set of their source WSI and can be labeled with the same cancer type. Equating the patch label with the WSI label based on high cellularity is based on the idea that high-grade carcinomas are generally associated with uncontrolled cell growth which can be spotted by a high number of cell nuclei in a region/patch. This was proposed as a way of *soft labeling* histopathology images in a raw repository where images have not been processed after acquisition.

The different versions of KimiaNet were fine-tuned/trained with around 240,000 patches extracted at 20× magnification spanning 30 different tumor sub-types. Several experiments were designed and performed to investigate the change in the quality and representativeness of the deep features generated by KimiaNet versions compared to the pre-trained DenseNet-121.

For comparative experiments, two types of search modes were conducted. The first type called "horizontal search", which is defined as measuring how accurate the algorithm performs for searching among all of the WSIs in the database and suggesting WSIs with the same *tumor type* as the query slide. In this experiment, KimiaNet-IV achieved the maximum accuracy for all of the tumor types except "Endocrine" with 92% having only 1% gap with the maximum accuracy reached by KimiaNet-III. The second search mode is "vertical search" in which a query WSI from a specific tissue region is taken as input and the algorithm searches through the WSIs of the same tissue region and tries to suggest WSIs with the same *cancer sub-type* (i.e., the same primary diagnosis) as the query slide. The results of this experiment showed that the features generated by KimiaNet-IV are more descriptive than its other versions and pre-trained DenseNet-121.

Another way for understanding the effect of training on the expressiveness of the deep features is visualization. To do this, a fixed number of patches were selected from each cancer type and fed to the KimiaNet-IV as well as the DenseNet-121 to extract their deep features. The dimensionality of these two sets of features were reduced using the t-SNE method so they could be visualized in a two-dimensional space (reduced from 1024 dimensions). Results showed that the features generated by KimiaNet form visually distinguishable clusters where each cluster is associated with a cancer sub-type. However, the DenseNet-121's deep features for different classes were too close to each other with large overlaps such that no obvious discrimination could be observed.

Some recent works state that medical imaging tasks could be performed with simpler/smaller network architectures maintaining the same performance as the commonly used but much larger topologies. Hence, four networks containing blocks of convolutional, batch-normalization and ReLU layers (the so-called *CBR networks*) were implemented and tested against DenseNet-121 and KimiaNet. The results demonstrated that KimiaNet features are by far better than the CBR networks. These simple networks performed only slightly better than the pre-trained DenseNet. This validates the DenseNet-121 as a good candidate for KimiaNet's architecture in compliance with the Occam's Razor principle.

In addition, and as a required pre-processing step before training, the performance of different networks for the U-Net topology were compared to the handcrafted methods in the task of tissue extraction in histopathology slides. Around 240 WSIs were acquired from TCGA and segmented manually at a low magnification to train the networks. The results illustrated the superiority of the deep networks. The proposed tissue segmentation exhibited 99% sensitivity and specificity for the test data.

References

[1] Raja S Alomari, Ron Allen, Bikash Sabata, and Vipin Chaudhary. Localization of tissues in high-resolution digital anatomic pathology images. In *Medical Imaging 2009: Computer-Aided Diagnosis*, volume 7260, page 726016. International Society for Optics and Photonics, 2009.

[2] Péter Bándi, Rob van de Loo, Milad Intezar, Daan Geijs, Francesco Ciompi, Bram van Ginneken, Jeroen van der Laak, and Geert Litjens. Comparison of different methods for tissue segmentation in histopathological whole-slide images. In *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, pages 591–595. IEEE, 2017.

[3] Aïcha BenTaieb and Ghassan Hamarneh. Deep learning models for digital pathology. *arXiv preprint arXiv:1910.12329*, 2019.

[4] Erin Brender, Alison Burke, and Richard M. Glass. Frozen Section Biopsy. *JAMA*, 294(24):3200–3200, 12 2005.

[5] Daniel Bug, Friedrich Feuerhake, and Dorit Merhof. Foreground extraction for histopathological whole slide imaging. In *Bildverarbeitung für die Medizin 2015*, pages 419–424. Springer, 2015.

[6] Hye Yoon Chang, Chan Kwon Jung, Junwoo Isaac Woo, Sanghun Lee, Joonyoung Cho, Sun Woo Kim, and Tae-Yeong Kwak. Artificial intelligence in pathology. *Journal of pathology and translational medicine*, 53(1):1, 2019.

[7] Pingjun Chen and Lin Yang. tissueloc: Whole slide digital pathology image tissue localization. *J. Open Source Software*, 4(33):1148, 2019.

[8] Nancy Chinchor and Beth M Sundheim. Muc-5 evaluation metrics. In *Fifth Message Understanding Conference (MUC-5): Proceedings of a Conference Held in Baltimore, Maryland, August 25-27, 1993*, 1993.

[9] Lee Ad Cooper, Elizabeth G Demicco, Joel H Saltz, Reid T Powell, Arvind Rao, and Alexander J Lazar. Pancancer insights from the cancer genome atlas: the pathologist's perspective. *The Journal of pathology*, 244(5):512–524, 2018.

[10] Nilanjan Dey, Venkatesan Rajinikanth, Amira S Ashour, and João Manuel RS Tavares. Social group optimization supported segmentation and evaluation of skin melanoma images. *Symmetry*, 10(2):51, 2018.

[11] Kevin Faust, Sudarshan Bala, Randy van Ommeren, Alessia Portante, Raniah Al Qawahmed, Ugljesa Djuric, and Phedias Diamandis. Intelligent feature engineering and ontological mapping of brain tumour histomorphologies by deep learning. *Nature Machine Intelligence*, 1(7):316–321, 2019.

[12] Fei-Fei, Deng, Russakovsky, Berg, and Li. Imagenet dataset. http://www.image-net.org/. Accessed: 2020-07-09.

[13] Yu Fu, Alexander W Jung, Ramon Viñas Torne, Santiago Gonzalez, Harald Vohringer, Mercedes Jimenez-Linan, Luiza Moore, and Moritz Gerstung. Pan-cancer computational histopathology reveals mutations, tumor composition and prognosis. *bioRxiv*, page 813543, 2019.

[14] Metin N Gurcan, Laura Boucheron, Ali Can, Anant Madabhushi, Nasir Rajpoot, and Bulent Yener. Histopathological image analysis: A review. *IEEE reviews in biomedical engineering*, 2:147, 2009.

[15] David A Gutman, Jake Cobb, Dhananjaya Somanna, Yuna Park, Fusheng Wang, Tahsin Kurc, Joel H Saltz, Daniel J Brat, Lee AD Cooper, and Jun Kong. Cancer digital slide archive: an informatics resource to support integrated in silico analysis of tcga pathology data. *Journal of the American Medical Informatics Association*, 20(6):1091–1098, 2013.

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

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

[18] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

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

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

[21] S. Kalra, C. Choi, S. Shah, L. Pantanowitz, and H. R. Tizhoosh. Yottixel – an image search engine for large archives of histopathology whole slide images, 2019.

[22] Jakob Nikolas Kather, Cleo-Aron Weis, Francesco Bianconi, Susanne M Melchers, Lothar R Schad, Timo Gaiser, Alexander Marx, and Frank Gerrit Zöllner. Multi-class texture analysis in colorectal cancer histology. *Scientific reports*, 6:27988, 2016.

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

[24] Richard Levenson. Histopathology is ripe for automation, 2017.

[25] Linchundan. Retinal fundus dataset. https://www.kaggle.com/linchundan/fundusimage1000. Accessed: 2020-07-09.

[26] Yun Liu, Krishna Gadepalli, Mohammad Norouzi, George E Dahl, Timo Kohlberger, Aleksey Boyko, Subhashini Venugopalan, Aleksei Timofeev, Philip Q Nelson, Greg S Corrado, et al. Detecting cancer metastases on gigapixel pathology images. *arXiv preprint arXiv:1703.02442*, 2017.

[27] Mooney. Chest x-ray pneumonia dataset. https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia. Accessed: 2020-07-09.

[28] Romain Mormont, Pierre Geurts, and Raphaël Marée. Comparison of deep transfer learning strategies for digital pathology. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2262–2271, 2018.

[29] Liron Pantanowitz, Paul N Valenstein, Andrew J Evans, Keith J Kaplan, John D Pfeifer, David C Wilbur, Laura C Collins, and Terence J Colgan. Review of the current state of whole slide imaging in pathology. *Journal of pathology informatics*, 2, 2011.

[30] Heather A Piwowar and Wendy W Chapman. Public sharing of research datasets: a pilot study of associations. *Journal of informetrics*, 4(2):148–156, 2010.

[31] Maithra Raghu, Chiyuan Zhang, Jon Kleinberg, and Samy Bengio. Transfusion: Understanding transfer learning for medical imaging. In *Advances in neural information processing systems*, pages 3347–3357, 2019.

[32] Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, et al. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv:1711.05225*, 2017.

[33] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

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

[35] Tomas Skripcak, Claus Belka, Walter Bosch, Carsten Brink, Thomas Brunner, Volker Budach, Daniel Büttner, Jürgen Debus, Andre Dekker, Cai Grau, et al. Creating a data exchange strategy for radiotherapy research: towards federated databases and anonymised public datasets. *Radiotherapy and Oncology*, 113(3):303–309, 2014.

[36] Fabio A Spanhol, Luiz S Oliveira, Paulo R Cavalin, Caroline Petitjean, and Laurent Heutte. Deep features for breast cancer histopathological image classification. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1868–1873. IEEE, 2017.

[37] Hao Sun, Xianxu Zeng, Tao Xu, Gang Peng, and Yutao Ma. Computer-aided diagnosis in histopathological images of the endometrium using a convolutional neural network and attention mechanisms. *IEEE Journal of Biomedical and Health Informatics*, 2019.

[38] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019.

[39] H. R. Tizhoosh, Shujin Zhu, Hanson Lo, Varun Chaudhari, and Tahmid Mehdi. Minmax radon barcodes for medical image retrieval, 2016.

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

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

[42] William D Travis. Pathology and diagnosis of neuroendocrine tumors: lung neuroendocrine. *Thoracic surgery clinics*, 24(3):257–266, 2014.

[43] Jason W Wei, Laura J Tafe, Yevgeniy A Linnik, Louis J Vaickus, Naofumi Tomita, and Saeed Hassanpour. Pathologist-level classification of histologic patterns on resected lung adenocarcinoma slides with deep neural networks. *Scientific reports*, 9(1):1–8, 2019.

[44] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[45] Jianming Zhang, Chaoquan Lu, Xudong Li, Hye-Jin Kim, and Jin Wang. A full convolutional network based on densenet for remote sensing scene classification. *Math. Biosci. Eng*, 16(5):3345–3367, 2019.

[46] Ke Zhang, Yurong Guo, Xinsheng Wang, Jinsha Yuan, and Qiaolin Ding. Multiple feature reweight densenet for image classification. *IEEE Access*, 7:9872–9880, 2019.

Nomenclature

biopsy Biopsies are small samples of tissue taken from a mass or tumor that are examined under a microscope to make a diagnosis. Biopsies are used most often to determine whether cancer cells are present, although certain infections and other diseases can be diagnosed as well [4]. xiv, 1, 57

frozen section biopsy A specific type of biopsy procedure called the frozen section was developed in order to make a rapid diagnosis of a mass during surgery. During the frozen section procedure, the surgeon removes a portion of the tissue mass. This biopsy is then given to a pathologist who freezes the tissue, cuts it, and then stains it with various dyes so that it can be examined under the microscope. The procedure usually takes only minutes [4]. xiv, 25, 57

model In this thesis, the word "model" has the same meaning as the "trained neural network". iv, 2, 5–7, 17, 22, 32, 35, 36, 39, 46, 49

permanent section biopsy In this procedure the tissue is placed in a fixative solution, embedded in wax, thinly cut, and then stained. Although this takes longer than a frozen section biopsy, the permanent section leads to better-quality microscope slides [4]. 25