DEEP LEARNING BASED HISTOPATHOLOGY
IMAGE ANALYSIS FOR CANCER DIAGNOSIS AND TREATMENT

by

HUI QU

A dissertation submitted to the
School of Graduate Studies
Rutgers, The State University of New Jersey
In partial fulfillment of the requirements
For the degree of
Doctor of Philosophy
Graduate Program in Computer Science

Written under the direction of
Dimitris N. Metaxas

and approved by

_________________________

_________________________

_________________________

_________________________

New Brunswick, New Jersey
May, 2021
ABSTRACT OF THE DISSERTATION

Deep Learning based Histopathology Image Analysis for Cancer Diagnosis and Treatment

By HUI QU
Dissertation Director:
Dimitris N. Metaxas

Histopathology plays a vital role in cancer diagnosis, prognosis, and treatment decisions. The whole slide imaging technique that captures the entire slide as a digital image (whole slide image, WSI) allows the pathologists to view the slides digitally as opposed to what was traditionally viewed under a microscope. With the development of computational power and image analysis algorithms, computational methods have been developed for the quantitative and objective analyses of histopathology images, which can reduce the intensive labor and improve the efficiency for pathologists compared with manual examinations. In this dissertation, we focus on deep learning based solutions in histopathology image analysis for cancer diagnosis and treatment, specifically, nuclei segmentation for cancer diagnosis, and gene mutation and pathway activity prediction for cancer treatment.

Nuclei segmentation is a critical step in the automatic analysis of histopathology images. We focus on two directions to tackle the problems in deep learning based nuclei segmentation task. One is the annotation-efficient algorithms. As the fully-supervised learning of deep neural networks requires a large amount of training data, a weakly supervised nuclei segmentation framework based on a portion of nuclear locations is
proposed to alleviate the annotation effort of pathologists. This method achieves comparable performance as the fully-supervised methods and about $60\times$ speed-up (10% points) in the annotation time. The other direction is to improve the instance segmentation performance. The networks and cross entropy loss in current deep learning-based segmentation methods originate from image classification tasks and have drawbacks for segmentation. Therefore, we propose a full resolution convolutional neural network (FullNet) and a variance constrained cross entropy (varCE) loss to improve the fully supervised segmentation performance.

Except for the cell-level heterogeneity that is routinely used for cancer diagnosis, it remains unclear for many cancers that how tissue structures in histopathology slides are related to genomic features like gene alterations and expression patterns. We develop a deep learning model to predict the genetic mutations and biological pathway activities directly from histopathology slides in breast cancer. The weight maps of tumor tiles are visualized to understand the decision-making process of deep learning models. Our results provide new insights into the association between pathological image features, molecular outcomes and targeted therapies for breast cancer patients.
Acknowledgements

Throughout my PhD study I have received a great deal of support and assistance.

First of all, I would like to express my sincere gratitude to my advisor, Prof. Dimitris N. Metaxas, for his support and guidance during my PhD study. He provided excellent research environment and collaborative opportunities to explore challenging tasks in medical image analysis. I am deeply grateful for his insightful talks and encouragement, which have helped me a lot.

Besides, I would like to thank the other members of my PhD committee, Prof. Konstantinos Michmizos, Prof. Hao Wang and Dr. Mei Chen (Microsoft), for their valuable suggestions on my dissertation.

I would also like to thank Dr. Subhajyoti De and Dr. Gregory M. Riedlinger from Rutgers Cancer Institute and Dr. Vinod Rustgi, Dr. Puru Rattan and Dr. Carlos Minacapelli from Robert Wood Johnson University Hospital for their help and guidance during the long-term collaboration. Their expertise in cancer promoted my understanding on related tasks.

I am very grateful to Dr. James K. Min (Weill Cornell Medicine), Dr. Zhenman Yan (SenseBrain), Dr. Mu Zhou (SenseBrain) and Dr. Kevin Brown (Siemens Healthineers) for working as my mentors during my internships. They have offered great help in my research work. My work at Sensetime Research has been continued after I returned to Rutgers and has become part of my dissertation.

My special thanks go to my friends and colleagues, for their insightful discussions and collaborations on projects, and help and support during these years at Rutgers.

Last but not least, I would like to thank the members of my family, especially my wife Qian Zhou, for their love, support and encouragement.
Dedication

To my parents, my parents-in-law, my sisters and my wife

for their endless love and support
# Table of Contents

Abstract .................................................. ii
Acknowledgements ......................................... iv
Dedication ................................................. v
List of Tables ............................................ ix
List of Figures ........................................... xi

1. Introduction ........................................... 1
   1.1. Background ........................................ 1
   1.2. Motivation ......................................... 3
   1.3. Contributions ...................................... 5
   1.4. Dissertation Outline .............................. 7

2. Weakly Supervised Nuclei Segmentation using Partial Points ........ 8
   2.1. Introduction ...................................... 8
   2.2. Related Work ...................................... 10
   2.3. Nuclei Segmentation with Partial Points ............ 12
       2.3.1. Detection with Partial Points ................. 12
       2.3.2. Segmentation with Detected Points .............. 15
   2.4. Nuclei Segmentation with Mixed Points and Selected Masks ....... 20
       2.4.1. Uncertainty Prediction .......................... 20
       2.4.2. Training with Mixed Points and Selected Masks . . . . . . . 21
   2.5. Experiments ...................................... 23
       2.5.1. Datasets and Evaluation Metrics ................. 23
       2.5.2. Detection Results using Partial Points Annotation . . . . 25
3. Improving Nuclei Segmentation by Full Resolution Neural Network and Spatial Constrained Loss

3.1. Introduction

3.2. Methods

3.2.1. FullNet Structure

3.2.2. Variance Constrained Cross Entropy Loss

3.2.3. Post-processing

3.3. Experiments

3.3.1. Datasets and Evaluation Metrics

3.3.2. Implementation Details

3.3.3. Results and Discussion

3.3.4. Ablation Study

3.3.5. Computational Cost

3.4. Application of FullNet on Cancer Imaging Data

3.5. Summary

4. Gene Mutation and Pathway Activity Prediction from Histopathology Images in Breast Cancer

4.1. Introduction

4.2. Methods

4.2.1. Data Processing

4.2.2. Model Structure

4.2.3. Model Training and Finetuning
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Nuclei detection results of different strategies on LC and MO datasets using 10% partial points annotation.</td>
<td>26</td>
</tr>
<tr>
<td>2.2</td>
<td>Nuclei detection results on LC and MO datasets using different ratios of annotation.</td>
<td>26</td>
</tr>
<tr>
<td>2.3</td>
<td>Nuclei segmentation results on LC and MO datasets for our partial points annotation, ground-truth points annotation, points with 5% masks, fully-supervised and state-of-the-arts.</td>
<td>31</td>
</tr>
<tr>
<td>2.4</td>
<td>Nuclei detection results on LC and MO datasets using 10 different sets of 10% initial points.</td>
<td>36</td>
</tr>
<tr>
<td>2.5</td>
<td>Nuclei segmentation results on LC and MO datasets using 10 different sets of 10% initial points.</td>
<td>37</td>
</tr>
<tr>
<td>2.6</td>
<td>Generalization performance of our weakly supervised method using partial points on LC and MO datasets.</td>
<td>37</td>
</tr>
<tr>
<td>3.1</td>
<td>Nuclei segmentation results on MO dataset using different methods.</td>
<td>48</td>
</tr>
<tr>
<td>3.2</td>
<td>Gland segmentation results on GlaS dataset using different methods. Rank of each entry is listed for clear comparison.</td>
<td>48</td>
</tr>
<tr>
<td>3.3</td>
<td>Nuclei segmentation results on MO dataset using FullNet with different layer types (FullNet-B for bottleneck layers) and growth rates ( (k) ). Results are reported on the whole test set.</td>
<td>51</td>
</tr>
<tr>
<td>3.4</td>
<td>Nuclei segmentation results on MO dataset using different weights ( (\alpha) ) of varCE loss. The network is FullNet-B with ( k = 48 ). Results are reported on the whole test set.</td>
<td>52</td>
</tr>
<tr>
<td>4.1</td>
<td>Patient characteristics on the 659 cases from TCGA-BRCA cohort.</td>
<td>59</td>
</tr>
<tr>
<td>4.2</td>
<td>Patient characteristics on the 350 cases from TCGA-LUAD cohort.</td>
<td>60</td>
</tr>
</tbody>
</table>
4.3. Patient characteristics on the 316 cases from TCGA-LIHC cohort. . . 61

4.4. AUC (with 95% CI) achieved by the models trained on the point mutation data of breast cancer. The top six results are reported out of 18 genes. ................................................................. 65

4.5. AUC (with 95% CI) achieved by the models trained on the copy number alteration data of breast cancer. The top six results are reported out of 35 genes. ................................................................. 65

4.6. AUC (with 95% CI) achieved by the models trained on the pathway activity from mRNA expression data (top row) and from CNA data (bottom row). The top three results are reported out of 10 canonical pathways ................................................................. 66
List of Figures

1.1. A H&E stained whole slide image and the nuclei segmentation results within a small area. (a) A whole slide image of 74813 × 53695 in pixels (26mm × 18.6mm in actual size). (b) An image patch of 1024 × 1024 after 20× zoom in. (c) The nuclei segmentation results of the image in (b), cancer, lymphocyte and stroma nuclei are marked with red, green and blue colors, respectively. .................................................. 2

1.2. A typical deep learning-based nuclei segmentation framework. ........ 3

2.1. Overview of the segmentation framework based on partial points. (a) Semi-supervised nuclei detection module: (b1) initial training step in which an initial detector is trained using the partial points annotation, and (b2) self-training step using background propagation. (b) Weakly supervised nuclei segmentation module: (b1) label generation in which the Voronoi label and cluster label are generated using the detected points and original image, (b2) model training using the cross entropy loss, and (b3) model refinement using the CRF loss. ............................................ 12

2.2. Training masks. The orange color indicates ignored pixels, black is background and green is Gaussian masks centered at labeled points. (a) image, (b)-(d) masks used for training in round 1, 2 3 of self-training, respectively. .................................................. 15

2.3. Label generation. (a) image, (b) ground-truth nuclei masks (in green) and Voronoi edges (in red), (c) distance map. ......................... 17
2.4. Overview of the framework with mixed supervision. (a) Uncertainty prediction. The uncertainty map is predicted from the Bayesian CNN trained with two types of proxy labels as in [76]. (b) CNN training using mixed points and selected masks from uncertainty. The annotated masks are combined with the Voronoi and Cluster labels to generate the revised labels for training. .......................................................... 20

2.5. Uncertainty map and masks of selected nuclei. ........................................ 22

2.6. Typical detection results of different strategies from Lung Cancer dataset (first row) and Multi-Organ dataset (second row) using 10% points. (a) Fully supervised training, (b) initial training using simple Gaussian mask, (c) initial training using extended Gaussian mask, (d) self-training using nuclei prediction, (e) self training using background propagation. Yellow dots are the detected nuclei. Green, blue and red circles represent ground-truth with correct detection (TP), ground-truth without correct detection (FN) and false positive detection (FP), respectively. ............. 27

2.7. Typical results using different weights ($\alpha$) of the cluster and Voronoi labels on LC dataset (top row) and MO dataset (bottom row). (a) Image, (b) ground-truth full mask, (c)-(g) are results using different $\alpha$ values. $\alpha = 0$ means using only the cluster label and $\alpha = 1$ means using only the Voronoi label. .......................................................... 29

2.8. The results of using different $\alpha$ values on LC and MO datasets. ........ 29

2.9. The improvements over the baseline using different $\sigma_{pq}$, $\sigma_{rgb}$ and $\beta$ values in the CRF loss for LC (top row) and MO (bottom row) datasets. In each subfigure, the horizontal black dash line indicates the baseline’s performance, and the vertical red dash line indicates the parameter value of the best combination. .......................................................... 30
2.10. The results of all test images using partial points and ground-truth points (100%) in the LC dataset (first row) and MO dataset (second row). *nuclei-bg-diff* is the difference between pixel values of nuclei and background. *nuclei-std* is the standard deviation of the pixel values within nuclei. .......................... 32

2.11. Typical segmentation results using different ratios of points annotation on LC dataset (top row) and MO dataset (bottom row). (a) image, (b) ground-truth full mask, (c)-(f) are results using different ratios of points, (g) the results of using all ground-truth (100%) points without the detection step. Distinct colors represent different nuclei. ............. 34

2.12. Typical bad segmentation results using different ratios of points annotation on LC dataset (top row) and MO dataset (bottom row). (a) image, (b) ground-truth full mask, (c)-(f) are results using different ratios of points, (g) the results of using all ground-truth (100%) points without the detection step. Distinct colors represent different nuclei. ............. 35

2.13. Typical segmentation results in the LC dataset (top row) and MO dataset (bottom row). Distinct colors represent different nuclei. Mixed (rand) and Mixed (un) are the mixed annotation using masks selected randomly and from uncertainty, respectively. ......................... 36

3.1. Network structure of the FullNet. Dense blocks are utilized to reduce the number of feature maps. Each dense block has different dilation factor (the digit under each block) to ensure the receptive field is large enough for recognition. .............................. 41

3.2. Dilation factors in (a) U-net and (b) FullNet. Each red dash line represents a change of dilation factor in FullNet or a down-sampling of feature maps in U-net. ......................... 42

3.3. Gridding artifacts. (a) input image patch (b) probability map with gridding artifacts (c) probability map after adding two more blocks and using hybrid dilation factor strategy. .............................. 43
3.4. Typical results of nuclei segmentation. For each row, from left to right are original image, ground-truth instance label, segmentation results of FCN-pooling, FullNet without varCE and FullNet with varCE loss, respectively. Distinct colors in the images indicate different instances. Rectangles are drawn for clear comparison.

3.5. Typical results of gland segmentation. For each row, from left to right are original image, ground-truth instance label, segmentation results of FCN-pooling, FullNet without varCE and FullNet with varCE loss, respectively. Distinct colors in the images indicate different instances. Rectangles are drawn for clear comparison.

3.6. The effect of varCE loss. (a) input image, (d) ground-truth instance label, (b) and (e) probability map and segmentation result without varCE loss, (c) and (f) probability map and segmentation result with varCE loss.

3.7. Nuclei segmentation using trained FullNet-B ($k = 48$) in four $500 \times 500$ subimages of a same TCGA bladder cancer tissue slide image. The number and average size (pixel number) of segmented nuclei in each image are marked under the segmentation results.

4.1. Illustration of the deep learning workflow for data processing and model evaluation. We processed the WSI data by extracting tiles (1a), identifying tumor titles (1b), and generating small non-overlapping tiles with color normalization. We selected key mutational genes (2) and identified biological pathways from mRNA (or CNA) expressions (3). Model training was based on a pretrained ResNet-101 model with an attention mechanism. After model selection, the trained model was used to test tiles and assess their prediction performances.

4.2. The proposed network structure. Each color normalized tile is fed into a pretrained ResNet-101 to extract a 2048-dimensional feature vector. Feature vectors of all tiles of the same patient are stacked and fed into a MLP with self-attention to predict the mutation probability or pathway activity.
4.3. ROC curves of the top predictions of gene point mutation and copy number alteration from histopathology image slides in breast cancer. 66

4.4. Weight maps of tiles when predicting the point mutation status of TP53 and p53 pathway activity from mRNA expression data in breast cancer. (a) Tumor tiles after data processing. (b) Weight map of tumor tiles in TP53 point mutation prediction. Brighter green tiles have larger weights. (c) Weight map of p53 pathway activity prediction. (d) and (e): Top 20 weighted tiles for TP53 point mutation prediction and p53 pathway prediction, respectively. We marked four tiles that appear in both tasks. These tissues contain poorly differentiated breast carcinoma with small nests, solid sheets and single cells from a pathologist's perspective. 67

4.5. Weight maps of tiles when predicting the point mutation status of RB1 and cell cycle pathway activity from mRNA expression data in breast cancer. (a) Tumor tiles after data processing. (b) Weight map of tumor tiles in RB1 point mutation prediction. Brighter green tiles have larger weights. (c) Weight map of the cell cycle pathway activity prediction. (d) and (e): Top 20 weighted tiles for the RB1 point mutation prediction and cell cycle pathway prediction, respectively. We marked four tiles that appear in both tasks. These tissues contain poorly differentiated breast carcinoma with necrosis or hemorrhage from a pathologist’s perspective. 74

4.6. Weight maps of tiles when predicting the point mutation status of TP53 and p53 pathway activity from mRNA expression data in lung cancer. (a) Tumor tiles after data processing. (b) Weight map of tumor tiles in TP53 point mutation prediction. Brighter green tiles have larger weights. (c) Weight map in p53 pathway activity prediction. (d) and (e): Top 20 weighted tiles for the TP53 point mutation prediction and p53 pathway prediction, respectively. We marked four tiles that appeared in both tasks. These tissues contain moderately differentiated lung carcinoma with papillary growth pattern from a pathologist's perspective. 75
4.7. Weight maps of tiles when predicting the point mutation status of RB1 and cell cycle pathway activity from mRNA expression data in liver cancer. (a) Tumor tiles after data processing. (b) Weight map of tumor tiles in RB1 point mutation prediction. Brighter green tiles have larger weights. (c) Weight map in cell cycle pathway activity prediction. (d) and (e): Top 20 weighted tiles from RB1 point mutation prediction and cell cycle pathway prediction, respectively. We marked four tiles that appeared in both tasks. These tissues contain hepatocellular carcinoma with clear cell change from a pathologist's perspective.
Chapter 1

Introduction

1.1 Background

Cancer is a group of diseases that some cells divide without stopping and spread to other parts of the body [2]. It is a leading cause of death worldwide and almost 10.0 million cancer deaths occurred in 2020 [94]. Cancer is caused by the changes in genes, which eventually affect cell functions, especially cell growth. For example, TP53 is the most commonly mutated gene in all cancers and it produces a protein that suppresses tumors growth [95]. The damage of TP53 gene will likely result in the development of certain cancers. To diagnose cancer, a histopathology test is usually performed by a pathologist. Histopathology refers to the microscopic examination of a biopsy or surgical specimen by a pathologist to study diseases like cancer. Histopathology slides are created from formalin-fixed paraffin-embedded (FFPE) tissue containing both tumor and surrounding normal tissue. These slides are then stained with agents such as hematoxylin and eosin (H&E) and immunohistochemical stains that permit the pathologist to ascertain important features. Pathologists routinely determine the type of cancer, stage of cancer, cancers grade, presence of infiltrating immune cells based on histopathology slides. The correct diagnosis of cancer is essential to select effective treatment options because every cancer type requires a specific treatment regimen such as surgery, radiotherapy, chemotherapy, or targeted therapy.

The development of whole slide imaging technique allows a pathologist to view the histopathology slides digitally as opposed to what was traditionally viewed under a microscope. A digital whole slide image (WSI) usually has a very large resolution (10,000 to > 100,000 in each dimension, e.g., Fig. 1.1(a)) and contains cell-level information about the tissue (Fig. 1.1(b)). It is time-consuming for a pathologist to view all details
Figure 1.1: A H&E stained whole slide image and the nuclei segmentation results within a small area. (a) A whole slide image of $74813 \times 53695$ in pixels ($26mm \times 18.6mm$ in actual size). (b) An image patch of $1024 \times 1024$ after 20x zoom in. (c) The nuclei segmentation results of the image in (b), cancer, lymphocyte and stroma nuclei are marked with red, green and blue colors, respectively.

in a WSI. As a result, some important information could be missed. Besides, the manual examinations are based on pathologists’ experiences and are prone to intra- and inter-observer variability. With improvements in computational power and image analysis algorithms, computational methods [35, 46, 108, 125, 132, 47, 104, 65] have been developed for the quantitative and objective analyses of histopathology images. They can reduce the intensive labor and improve the efficiency for pathologists compared with manual examinations. Nuclei segmentation aims to extracts the pixel-wise mask of each nucleus in an image. It is a critical step in the automatic analysis of histopathology images, because the nuclear features such as average size, density and nucleus-to-cytoplasm ratio are related to the clinical diagnosis and management of cancer. An example of nuclei segmentation is shown in Fig. 1.1(c). For clear illustration, three different types of nuclei, cancer, lymphocyte, stroma, are marked with red, green and blue colors, respectively. With the segmentation results, different types of nuclei can be identified according to their features, and quantitative analyses such as cancer cell counting and tumor purity calculation are also possible.

After cancer being diagnosed, treatment options need to be decided based on the cancer itself, patient’s plan and overall health. Traditional treatments like radiotherapy and chemotherapy have the side effect of killing normal cells when dealing with
cancer cells. Targeted therapy, which used drugs to attack specific targets such as proteins in and on cancer cells, can help treat cancer without affecting normal cells. The development of targeted therapy drugs requires understanding of genetic changes and proteins that drive cancer. For patients, comprehensive genomic analysis such as DNA sequencing test is needed to identify gene mutations and expression in cancer cells. This information is beneficial to the selection of targeted therapy that works best against a particular type of cancer.

1.2 Motivation

In this dissertation, we focus on dealing with challenges in the nuclei segmentation task for cancer diagnosis and gene mutation prediction task for cancer treatment, using deep learning technology.

Deep learning is a type of machine learning and relies on artificial neural networks to learn complex knowledge. It has achieved great success in computer vision tasks such as image classification [54, 36], image segmentation [19], and object detection [83] due to its powerful ability of feature representation. Current state-of-the-art nuclei segmentation methods are almost based on deep learning. They have achieved good segmentation performance but still face some challenges. A typical deep learning-based nuclei segmentation framework is shown in Fig. 1.2. It requires a pair of input image
and ground-truth label to train an encoder-decoder like convolutional neural network (CNN). The network takes the image as input and output a probability map, which is compared with the true label to calculate loss. The loss provides feedback to the network to update its parameters. There are some problems in this framework:

- **Data annotation.** The framework in Fig. 1.2 is supervised training and needs pixel-level annotated labels for each image during training. Considering that the training of CNNs requires a large amount of data, the annotation of labels is laborious, especially in the nuclei segmentation task where the small size and large number of nuclei makes it hard for annotation. What’s worse, it requires domain knowledge from pathologists to annotate nucleus masks. As a result, the annotation of full pixel-level labels has been an obstacle to collecting a large dataset and training a better model in nuclei segmentation.

- **Architecture.** The network in Fig. 1.2 is usually the fully convolution network (FCN) [62] or its variant. The encoder part of the network comes from the image classification network without fully-connected layers. It contains pooling or strided convolution layers, which greatly increase the receptive field to distill more-abstract features and make the network robust against small translations in the image [73]. However, such down-sampling may reduce the localization accuracy of boundaries in segmentation tasks, because the reduced size of feature maps leads to the loss of details like boundaries and tiny objects. To alleviate this problem, post-processing steps are often performed to refine the segmentation results. Other methods, such as the skip connection in U-net [84], can solve this problem to some extent, but it is not good enough.

- **Loss function.** The cross entropy loss is often used for training the network. This loss cares about if the classification of individual pixel is correct or not, but pays no attention to the spatial relationship between pixels. Although the deep neural networks can learn some high-level features like objects prior from ground-truth labels, the segmentation results are still not satisfactory when the objects have various colors, textures and shapes, especially in medical images.
For targeted therapy in cancer treatment, although gene mutations and expression can be identified by genomic analysis technologies like DNA sequencing and whole-transcriptome sequencing, they are not routinely used by all medical centers, because they are time-consuming and present several challenges impairing their adoption into clinical practice. Histopathology slides are now ubiquitously available, and they could be an important tool to predict the mutations and expressions of genes because the phenotypic information present in WSIs is the aggregate effect of molecular alterations on cancer cells. There have been some works utilizing deep learning to predict gene mutations in lung cancer [24] and liver cancer [20], as well as in a pan-cancer setting [48, 30], and to predict gene expression [48, 30, 87]. However, it remain unclear whether biological pathway activities could be associated with H&E stained WSIs. Besides, current methods on mutation and expression predictions assume that the tiles in one slide share the same label as the slide, and obtain the per-slide result by averaging the tiles’ predictions. This assumption is not solid because of the existence of normal and tumor cells and the intra-tumor heterogeneity. Therefore, finding a better way to directly make slide-level predictions is an important task in the research of this area.

1.3 Contributions

In this dissertation, we conduct studies on weakly-supervised nuclei segmentation, designing new network structure and loss function to address the above-mentioned challenges in deep learning-based nuclei segmentation. Besides, we develop a deep learning method to directly produce slide-level predictions for gene mutations and pathway activities in breast cancer. Our contributions can be summarized as follows:

1. We develop a weakly supervised nuclei segmentation framework to obtain comparable segmentation performance as fully supervised methods using partial points annotation. It contains a semi-supervised nuclei detection stage and a weakly supervised nuclei segmentation stage. The detection stage finds all nuclei in the images based on the annotation of a small portion of nuclear locations. It can reduce the false positive detections to a large extent due to a novel back-propagation
strategy in the self-training process. The segmentation stage takes the detected nuclear locations as input and produce instance segmentation results. It combines the advantages of some traditional machine learning methods and deep learning, and achieves very close performance as fully supervised methods. Furthermore, a human-in-loop solution is provided to add mask annotation of 5% representative nuclei to reduce the gap between our weakly supervised method and fully supervised one. Extensive experiments on two datasets demonstrate that the proposed framework achieves great segmentation performance while requiring much less annotation effort.

2. We propose a full resolution neural network architecture and a variance constrained cross entropy loss to improve the nuclei segmentation performance. Unlike the widely used encode-decoder network structure, we avoid pooling layers that are harmful to localization accuracy and keep the original resolution for all feature maps in the network. The details are therefore preserved. Besides, a variance term is added to the cross entropy loss to implicitly place a spatial constraint on pixels of the same instance, thus driving consistent predictions for those pixels. Our method outperforms other state-of-the-art methods in both nuclei segmentation and gland segmentation, although it is not memory efficient.

3. We present a deep learning model with self-attention to make slide-level predictions on gene mutation status and pathway activities from histopathology images in breast cancer. It is the first time to associate histopathology image features with biological pathway activities. Our model can predict the point mutations of six important genes (AUC 0.68 ~ 0.85) and copy number alteration of another six genes (AUC 0.69 ~ 0.79), as well as the activities of three out of ten canonical pathways (AUC 0.65 ~ 0.79). Besides, the self-attention mechanism in the model can automatically learn the weights of each tile during prediction, which is important to find key tiles and help understand the decision-making process of the trained model. Our results provide new insights into the association between pathological image features, molecular outcomes and targeted therapies for breast
cancer patients.

1.4 Dissertation Outline

The remainder of this dissertation is organized as follows. First, we focus on the data annotation problem in nuclei segmentation task and introduce the weakly supervised nuclei segmentation framework in Chapter 2. Then, in Chapter 3 we propose the full resolution neural network and variance constraint cross entropy loss to tackle the problems in architecture and loss function and improve nuclei segmentation performance. In Chapter 4, a deep learning-based model is presented to predict gene mutations and pathway activities in breast cancer, which is potentially beneficial for targeted therapy cancer treatment. Finally, conclusions and discussion on future research directions are presented in Chapter 5.
Chapter 2

Weakly Supervised Nuclei Segmentation using Partial Points

2.1 Introduction

Nuclei segmentation is an important yet challenging task in histopathology image analysis. It aims to extract the pixel-wise mask of each nucleus in the image. The variations in nuclear size, shape and color make it hard to obtain good segmentation performance using traditional algorithms like watershed [118], level sets [74], graph-cut [5], etc. It is also difficult to separate crowded and touching nuclei. Early learning based methods [53, 126] utilize handcrafted feature such as color, texture, and other image-level features to segment nuclear regions. Compared with traditional methods, they can achieve better performance when dealing with the above variations. However, the overall performance is still not good enough.

Modern deep learning based algorithms [107, 55, 67, 64, 81, 39, 57, 78] focus on training deep convolutional neural networks (CNNs) for segmentation, and are more effective than previous methods. However, the fully supervised learning of deep neural networks in these methods requires a large amount of training data, which are pixel-wise annotated by pathologists. It is difficult to collect such datasets because assigning a nucleus/background class label to every pixel in the image is very time-consuming due to the large number and small size of nuclei. Besides, the annotation requires expert domain knowledge. One way to alleviate the annotation burden is weakly supervised learning. It adopts weak labels as supervision and has drawn much attention in both natural and medical image segmentation. In this chapter, we introduce a weakly-supervised nuclei segmentation method [77] using a portion of annotated nuclear locations in the image. It achieves comparable performance as the fully supervised
methods and about $60 \times$ speed-up (10% points) in the annotation time. Furthermore, we propose to add a few newly annotated nuclear masks to points to reduce the gap between the weakly supervised method and fully-supervised one.

Our weakly supervised method consists of two stages: semi-supervised nuclei detection and weakly supervised nuclei segmentation. The goal of the first stage is to train a detector from partial points annotation to predict the locations of all nuclei in training images. A challenge is that there is no clear background information because only part of the nuclei are labeled in an image. To obtain a good initial detector, we first design an extended Gaussian mask to supervise the training with the labeled nuclear locations and ignore most unlabeled areas. Then, we propose a self-training strategy to make use of the unlabeled areas in images, which refines the background information in an iterative fashion and suppresses false positives. The detection stage produces central points of all detected nuclei. However, these detected points cannot be directly used to supervise the training of a segmentation model in the second stage. To address this problem, we take advantage of the original image and the shape prior of the nuclei to derive two types of coarse labels from the nuclei points using the Voronoi diagram and the $k$-means clustering algorithm. The coarse labels are used to train a deep convolutional neural network for the segmentation task. A common problem in most weakly supervised segmentation tasks is inaccurate object boundaries due to missing information. Therefore, post-processing like the dense conditional random field (CRF) [19] or graph search [116] is needed to refine the object boundaries, at the expense of extra processing time. Inspired by Tang et al.’s work [97], we utilize the dense CRF in the loss function to fine-tune the trained model rather than a post-processing step. This efficient inference is more effective in nuclei segmentation from large WSIs.

The overall performance of the above method with points is good, but it is not able to do well in nuclei with non-uniform color even with the dense CRF loss. Therefore, we utilize Bayesian deep learning method [50] to predict the uncertainty in the segmentation results, and pick 5% nuclei with top uncertainty values to annotate the full masks. Those masks are used to update the original Voronoi and cluster labels that are derived from points. The updated two types of labels are then used to train the model based
on a similar framework in the second stage without the dense CRF loss. Experimental results demonstrate that this mixed supervision can achieve evident performance gain with the expense of a little annotation effort for the masks.

2.2 Related Work

Deep CNNs have been applied to nuclei detection and segmentation in recent years. For deep learning-based nuclei detection, Ciresan et al. [23] proposed a mitosis detection method by classifying each pixel using a patch centered on it. Xie et al. [105] developed a structured regression CNN model (SR-CNN) to utilize topological structure information. Sirinukunwattana et al. [89] improved SR-CNN by using a spatial-constrained layer. Xu et al. [110] proposed a stacked sparse autoencoder to learn high-level structure information from unlabeled image patches and trained a classifier using the extracted features of the encoder. These methods depend on patch-based classification or regression, and thus are computationally expensive for large microscopy images. Later on, the fully convolutional neural network (FCN) [62] and its variants were applied in the nuclei detection, which can significantly improve the efficiency in inference by eliminating repeated computations for overlapping patches in the patch-based approaches. A structured regression model for nuclei detection was developed in [106] to produce better distinctive peaks at cell centroids. Zhou et al. [130] presented a Sibling FCN which detects nuclei and classifies them into sub-categories simultaneously. It takes advantage of the mutual information of both tasks to improve performance. Aside from these supervised training methods, Li et al. [58] proposed a semi-supervised learning framework for signet ring cell detection to cope with incomplete annotation and make use of unlabeled images. What is more challenging in our case is that we only have a small portion of nuclei annotated.

In the nuclei segmentation task, current deep learning-based methods can be also roughly divided into patch-based and FCN-based categories. In the patch-based category, Su et al. [93] utilized sparse denoising autoencoder to segment nuclei. Xing et al. [107] obtained initial shape probability maps of nuclei by CNN and then incorporated a top-down shape prior model and a bottom-up deformable model for segmentation.
Kumar et al. [55] formulated the problem as a three-class segmentation task and performed region growing as post-processing based on the initial segmentation results. In the FCN-based category, Naylor et al. [68] solved the problem as a regression task of estimating the nuclei distance map which is beneficial to separate touching or overlapping nuclei. Qu et al. [75] combined the tasks of nuclei segmentation and fine-grained classification into one framework. To solve the problem of insufficient training data, Mahmood et al. [64] synthesized additional training images using CycleGAN [131]. And Hou et al. [39] generated images of different tissue types and adopted an importance sampling loss during segmentation according to the quality of synthesized images. We also use the FCN-based framework, but with weak labels (central points).

Compared to fully supervised methods, weakly supervised approaches have the advantage of reducing manual annotation effort. In natural image segmentation, Papanдреou et al. [71] proposed Expectation-Maximization (EM) method for training with image-level or bounding-box annotation. Pathak et al. [72] added a set of linear constraints on the output space in loss function to exploit the information from image-level labels. Compared to image-level annotation, points annotation has better location information for each object. Bearman et al. [9] incorporated an objectness prior in the loss to guide the training of a CNN, which helps separate objects from background. Scribbles annotation, which requires at least one scribble for every object, is a more informative type of weak label. Lin et al. [60] adopted scribbles annotation to train a graphical model that propagates the information from the scribbles to the unmarked pixels. The most widely used weak annotation is the bounding box, both in natural images [27, 80] and in medical images [116, 127]. Kervadec et al. [51] used a small fraction of full labels and imposed a size constraint in their loss function, which achieved good performance but is not applicable for multiple objects of the same class.

Although existing weakly supervised methods have achieved good performance in natural and medical image segmentation, most weak annotations are not suitable for nuclei segmentation task. Image-level annotation cannot be used in medical image segmentation where object classes in images are usually fixed (e.g., nuclei and background in our task). Scribbles annotation is also not suitable for our task due to the small
size and large number of nuclei. It is difficult and time-consuming to label an image using bounding boxes for hundreds of nuclei, especially when the density is high. Points annotation is a good choice in terms of preserving information and saving annotation effort, but the objectiveness prior in the points supervision work [9] is not working here since nuclei are small and thus the prior is inaccurate. Different from existing weakly supervised methods, we propose to employ partial points annotation for nuclei segmentation.

2.3 Nuclei Segmentation with Partial Points

The overview of our weakly supervised method based on partial points is shown in Fig. 2.1. The semi-supervised detection stage aims to detect all nuclear locations from the annotated points and the weakly supervised segmentation stage extract nuclei masks from the detected points.

2.3.1 Detection with Partial Points

As shown in Fig. 2.1(a), the detection stage consists of two steps: initial training with extended Gaussian masks and self-training with background propagation.
Initial training with extended Gaussian masks

The first step of our detection method aims to train an initial detector using the labeled nuclei in each image. However, the points indicating nuclear locations cannot be directly applied for training. They are often used to generate binary masks for pixel classification [130], or structured proximity masks for regression [106]. In our case, it is not possible to follow these methods because most areas in an image are unlabeled. In order to tackle this issue, we define an extended Gaussian mask $M$ according to the labeled points:

$$
M_i = \begin{cases} 
\exp\left(-\frac{D_i^2}{2\sigma^2}\right) & \text{if } D_i < r_1, \\
0 & \text{if } r_1 < D_i < r_2, \\
-1 & \text{otherwise}, 
\end{cases}
$$

(2.1)

where $D_i$ is the distance from pixel $i$ to the closest labeled point, $r_1$ is average radius of the nuclei and can be calculated using the validation set, $r_2$ is a parameter to control the range of background area and is set to $r_2 = 2r_1$. $\sigma$ is the Gaussian bandwidth. In $M$, 0 means the background pixel and -1 means unlabeled pixel which will be ignored during training. The underlying assumption is that pixels in the annular area $r_1 < D_i < r_2$ belong to the background, which is reasonable because most nuclei are surrounded by background pixels.

With the extended Gaussian masks, we are able to train a regression model for nuclei detection. We replace the encoder part of U-net [84] with the convolution layers of ResNet-34 [36] (shown in Fig. 2.1), which is more powerful in representation ability and can be initialized with pretrained parameters. The network is trained with a mean squared loss $L_{mse}$ with respect to the corresponding extended Gaussian mask:

$$
L_{mse} = \frac{1}{|\Omega|} \sum_{i \in \Omega} w_i (p_i - M_i)^2,
$$

(2.2)

where $\Omega$ is the set consisting of non-ignored pixels, $p_i$ is the predicted probability of being nucleus by the network, and $w_i$ is the weight of pixel $i$. Considering the imbalance between the labeled points and background pixels, we set $w_i = 10$ for pixels with mask value greater than 0 and $w_i = 1$ for background pixels. The detection results are
obtained by thresholding the probability map and finding the centroids of connected components.

**Self-training with background propagation**

The detection performance of the initial model is not good enough because of the small number of labeled nuclei and large ignored areas. The unlabeled regions in images can be utilized to improve the performance by semi-supervised learning methods such as self-training. Intuitively, the initial model could predict the nuclei locations on the unlabeled regions. The predicted nuclei are then used to supervise the model training along with the originally labeled nuclei, like what the authors did on cell detection in [58]. However, we find that there are too many false positives among the newly detected nuclei because the trained model with the small number of labeled nuclei is not good. The false positives mislead the training during iterations, resulting in worse detection performance. Therefore, we propose an iterative learning strategy to refine the background map during self-training. Because the background tissue and blank areas are easier to be distinguished, producing less false positives.

The process of self-training is shown in Fig. 2.1(a). In each round of self-training, the background map is firstly obtained from the trained model of the previous round (or the initial model for the first round). Then the background map is combined with the original labeled points to generate a new mask for training. In background map generation, we select background pixels in the probability map if $p_i < 0.1$ or $p_i > 0.7$. The first criterion is straightforward since $p_i$ is the probability of being nuclei. The second one is considered because many background pixels get predicted values close to 1, especially in the first stage. This behavior is expected because most background pixels are ignored during training. If the initialized model predicts them as nuclei pixels, the predictions will remain unchanged. In order to prevent from adding true positive into background, for $p_i > 0.7$ case we only take into account the large connected components of areas greater than the average nuclei area, i.e., $\pi r^2_1$. The updated mask $\tilde{M}$ is finally generated by adding the new background information to the original extended Gaussian
Figure 2.2: Training masks. The orange color indicates ignored pixels, black is background and green is Gaussian masks centered at labeled points. (a) image, (b)-(d) masks used for training in round 1, 2, 3 of self-training, respectively.

\[
\tilde{M}_i = \begin{cases} 
  M_i & \text{if } D_i < r_2, \\
  0 & \text{if } D_i > r_2 \text{ and } (p_i < 0.1 \text{ or } p_i > 0.7) \\
  -1 & \text{otherwise}, 
\end{cases}
\]  

(2.3)

An example is shown in Fig. 2.2 to illustrate the masks in different rounds of self-training. The foreground nuclei annotation (green pixels) is kept unchanged during the iterations while the background area (pixels in black) grows up gradually. In the third round, the background has high accuracy and the ignored pixels (orange) are almost all nuclei.

### 2.3.2 Segmentation with Detected Points

After obtaining a good detection model, we can predict the nuclei locations on all training images. Although the detected points are not 100% accurate, we have much more information for segmentation compared to the initial partial points. In this subsection, we describe our weakly supervised method to segment nuclei from the detected points. In particular, our point-level supervision for training a nuclei segmentation model consists of three steps: (1) coarse pixel-level labels generation using the detected points from the nuclei detection stage; (2) segmentation network training with the generated coarse labels; (3) model refinement using the dense CRF loss.
From point-level to pixel-level labels

The point-level labels (detected points) cannot be used directly for the training of a CNN with the cross entropy loss due to the lack of (negative) background labels since all annotated points belong to the (positive) nuclei category. To solve this issue, the first step is to exploit the information we have to generate useful pixel-level labels for both classes. We have the following observations:

- Each point is expected to be located or close to the center of a nucleus, and the shapes of most nuclei are nearly ellipses, i.e., they are convex.

- The colors of nuclei pixels are often different from the surrounding background pixels.

Based on these observations, we propose to utilize the Voronoi diagram and $k$-means clustering methods to produce two types of pixel-level labels.

Voronoi labels. A Voronoi diagram is a partitioning of a plane into convex polygons (Voronoi cells) according to the distance to a set of points in the plane. There is exactly one point (seed point) in each cell and all points in a cell are closer to its seed point than other seed points. In our task, the detected points in an image can be treated as seed points to calculate the Voronoi diagram as shown in Fig. 2.1(b). For each Voronoi cell, assuming that the corresponding nucleus is located within the cell, then the Voronoi edges separate all nuclei well and the edge pixels belong to the background. This assumption holds for most of the nuclei because the detected points are around the centers and nuclear shapes are nearly convex (Fig. 2.3(b)).

Assigning the Voronoi edges as background pixels and the detected points (dilated with a disk kernel of radius 2) as nuclei pixels, we obtain the Voronoi point-edge label (Fig. 2.3). All other pixels are ignored during training. Note that although the pixels on the Voronoi edge between two touching nuclei may not necessarily be background, the edges are still helpful in guiding the network to separate the nuclei. The Voronoi labels aim to segment the central parts of nuclei and are not able to extract the full masks, because they lack the information of nuclear boundaries and shapes. To overcome
this weakness, we generate another kind of labels that contain this complementary information.

**Cluster labels.** Considering the color difference between nuclei and background pixels, it is feasible to perform a rough segmentation using clustering methods. We choose \( k \)-means clustering to extract both nuclei and background pixels from the image, and generate the cluster labels. Given an image \( x \) with \( N \) pixels \((x_1, x_2, \cdots, x_N)\), \( k \)-means clustering aims to partition the \( N \) pixels into \( k \) clusters \( S = (S_1, S_2, \cdots, S_k) \) according to the feature vector \( f_{x_i} \) of each pixel \( x_i \), such that the sum of within-cluster variances is minimized:

\[
\arg\min_S \sum_{i=1}^k \sum_{x \in S_i} \|f_x - c_i\|^2.
\]  

(2.4)

We use \( k \)-means to divide all pixels into \( k = 3 \) clusters: nuclei, background and ignored. The cluster that has maximum overlap with points label is considered as nuclei, and the cluster that has minimum overlap with the dilated points label is considered as background. The remaining one is the ignored class. The pixels of the ignored class are often located around the nuclear boundaries, which are hard for a clustering method to assign correct labels.

For the feature vector \( f \), color is a straightforward choice. However, clustering with color will result in wrong assignments for pixels inside some nuclei that have non-uniform colors. To cope with this issue, we add a distance feature. In a distance map (Fig. 2.3(c)), each value indicates the distance of that pixel to the closest nuclear point and therefore incorporates the spatial information. In particular, the pixels that belong
to nuclei should be close enough to points in the label while background pixels are relatively far from those points. The distance map can be calculated by the distance transform of the complement image of detected points. Combining the distance value $d_i$ with the RGB color values $(r_i, g_i, b_i)$ as the feature vector $f_{x_i} = (\hat{d}_i, \hat{r}_i, \hat{g}_i, \hat{b}_i)$ in $k$-means clustering, we obtain the initial cluster labels. $\hat{d}_i$ is the clipped value by truncating large values to 20 and $\hat{r}_i, \hat{g}_i, \hat{b}_i$ are normalized values such that every feature has similar value range.

The cluster label (Fig. 2.1(b)) is generated by refining the clustering result with morphological dilation and erosion, which are done separately in each Voronoi cell to avoid connecting close nuclei. The cluster labels have more shape information about the nuclei compared with the Voronoi labels, at the expense of more errors and uncertainties. We argue that these two types of labels are complementary to each other and will jointly lead to better results.

**Training deep neural networks with pixel-level labels**

Once we have the two types of pixel-level labels, we are able to train a deep convolutional neural network for nuclei segmentation. The network structure is the same as that in nuclei detection. It outputs two probability maps of background and nuclei, which are used to calculate two cross entropy losses with respect to the cluster label $L_{\text{cluster}}$ and Voronoi label $L_{\text{vor}}$:

$$L_{\text{cluster/vor}}(y, t) = -\frac{1}{|\Omega|} \sum_{i \in \Omega} [t_i \log y_i + (1 - t_i) \log(1 - y_i)],$$

(2.5)

where $y$ is the probability map, $t$ is the cluster label or Voronoi label, and $\Omega$ is the set consisting of non-ignored pixels. The final loss is

$$L_{\text{ce}} = \alpha L_{\text{vor}} + (1 - \alpha) L_{\text{cluster}},$$

(2.6)

where $\alpha$ is a balancing parameter.

**Model refinement using dense CRF loss**

The model trained using the two types of labels is able to predict the mask of individual nucleus with high accuracy. To further improve the performance, we refine the
nuclear boundaries with the dense CRF loss. Previously, post-processing such as region growing [55], graph search [116] or dense CRF [19] is often utilized to refine the segmentation results. These algorithms introduce more computational complexity, making them unsuitable for the processing of high resolution Whole Slide Images. To solve this problem, similar to [97], we embed the dense CRF into the loss function during training to improve the accuracy. The loss function is not calculated during inference, and therefore will not introduce additional computational cost after training.

Let $\tilde{y} = (\tilde{y}_1, \tilde{y}_2, \cdots, \tilde{y}_N)$ denote the predicted label (0 for background and 1 for nuclei) from probability maps $y$ and $t$ be the label. The dense CRF is to minimize the energy function:

$$E(\tilde{y}, t) = \sum_i \phi(\tilde{y}_i, t_i) + \sum_{i,j} \psi(\tilde{y}_i, \tilde{y}_j),$$

(2.7)

where $\phi$ is the unary potential that measures how likely a pixel belongs to a certain class, and $\psi$ is the pairwise potential that measures how different a pixel’s label is from all other pixels’ in the image. The unary term is replaced with the cross entropy loss $\mathcal{L}_{ce}$. The pairwise potential usually has the form:

$$\psi(\tilde{y}_i, \tilde{y}_j) = \mu(\tilde{y}_i, \tilde{y}_j)W_{ij} = \mu(\tilde{y}_i, \tilde{y}_j) \sum_{m=1}^{K} w_m k_m(\tilde{f}_i, \tilde{f}_j),$$

(2.8)

where $\mu$ is a label compatibility function, $W_{ij}$ is the affinity between pixels $i, j$ and is often calculated by the sum of Gaussian kernels $k_m$. Here we choose $\mu$ as the Potts model, i.e., $\mu(\tilde{y}_i, \tilde{y}_j) = [\tilde{y}_i \neq \tilde{y}_j]$, and bilateral feature vector $\tilde{f}_i = \left(\frac{p_i}{\sigma_{pq}}, \frac{q_i}{\sigma_{pq}}, \frac{r_i}{\sigma_{rgb}}, \frac{g_i}{\sigma_{rgb}}, \frac{b_i}{\sigma_{rgb}}\right)$ that contains both location and color information. $\sigma_{pq}$ and $\sigma_{rgb}$ are Gaussian bandwidth.

To adapt the energy function to a differentiable loss function, we relax the pairwise potential as [97]:

$$\psi(\tilde{y}_i, \tilde{y}_j) = \tilde{y}_i (1 - \tilde{y}_j) W_{ij}.$$  

(2.9)

Therefore, the dense CRF loss can be expressed as:

$$\mathcal{L}_{crf}(y, t_{cluster}, t_{vor}) = \mathcal{L}_{ce}(y, t_{cluster}, t_{vor}) + \beta \mathcal{L}_{pair}(y),$$

(2.10)

where $\mathcal{L}_{pair}(y) = \sum_{i,j} y_i (1 - y_j) W_{ij}$ is the pairwise potential loss and $\beta$ is the weighting factor. The CRF loss is used to fine-tune the trained model. Due to the large number of pixels in an image, the cost of directly computing the affinity matrix $W = [W_{ij}]$ is
prohibitive. For instance, there are \( N^2 = 1.6 \times 10^9 \) elements in \( W \) for an image of size 200 \( \times \) 200 that has \( N = 40000 \) pixels. We adopt fast mean-field inference based on high-dimensional filtering [3] to compute the pairwise potential term.

### 2.4 Nuclei Segmentation with Mixed Points and Selected Masks

The framework with mixed supervision is shown in Fig. 2.4, which consists of two steps: (1) uncertainty prediction using Bayesian CNN, (2) model training using mixed points and selectedly annotated masks from uncertainty.

![Diagram of the framework with mixed supervision](image)

**Figure 2.4**: Overview of the framework with mixed supervision. (a) Uncertainty prediction. The uncertainty map is predicted from the Bayesian CNN trained with two types of proxy labels as in [76]. (b) CNN training using mixed points and selected masks from uncertainty. The annotated masks are combined with the Voronoi and Cluster labels to generate the revised labels for training.

#### 2.4.1 Uncertainty Prediction

There are two major types of uncertainty: aleatoric uncertainty and epistemic uncertainty [50]. The former captures the noise in the input data, which is also called data uncertainty. The latter captures the ignorance about which model generated the data and is referred as model uncertainty [50]. We focus on the data uncertainty because we want to find representative nuclei for mask annotation.

The uncertainty prediction part (Fig. 2.4(a)) is based on the architecture in the segmentation stage of Section 2.3. Like [50], we change the CNN into a Bayesian CNN by placing a Gaussian distribution over the vector \( f_i^W \) before the last softmax layer for
each pixel $i$:
\[
\hat{x}_i | W \sim \mathcal{N}(f^W_i, (\sigma^W_i)^2),
\]  
\(2.11\)

where $W$ is the parameter matrix of the network. $(\sigma^W_i)^2$ is the Gaussian noise variance (a diagonal matrix with one element for each logit value) and treated as the data uncertainty. The corrupted vector $\hat{x}_i$ is squashed with the softmax function to obtain the probability vector $p_i$ for pixel $i$. The loss function is an expectation over the Gaussian distribution. Monte Carlo sampling is used to approximate the expected loss because we have no idea about the true distribution. Specifically, we draw samples from the logits of $f^W_i$, and rewrite Eqn. (2.11) as
\[
\hat{x}_{i,t} = f^W_i + \sigma^W_i \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, I),
\]  
\(2.12\)

where $t \in \{1, 2, \ldots, T\}$ and $T$ is the sampling number. The final output for pixel $i$ is $y_i = \frac{1}{T} \sum_t \text{Softmax}(\hat{x}_{i,t})$.

The original points label is not sufficient to supervise the training. Therefore, two types of pixel-wise proxy labels are derived using $k$-means clustering and Voronoi diagram partition algorithms as in Section 2.3.2, namely the cluster label and Voronoi label. In the labels, the pixels with red and green colors belong to background and nuclei, respectively. Pixels in black are ignored during training.

The loss function of the Bayesian model is the sum of cross entropy loss for each type of labels:
\[
\mathcal{L}_{\text{cluster/vor}} = -\frac{1}{|\Omega|} \sum_{i \in \Omega} [q_i \log y_i + (1 - q_i) \log(1 - y_i)],
\]  
\(2.13\)

\[
\mathcal{L}_{\text{ce}} = \mathcal{L}_{\text{vor}} + \mathcal{L}_{\text{cluster}},
\]

where $q_i$ is the true label for pixel $i$, $\Omega$ is the set consisting of non-ignored pixels.

### 2.4.2 Training with Mixed Points and Selected Masks

**Nuclei selection for annotation.** With the predicted uncertainty map (see Fig. 2.5(b)), the average uncertainty value within an area of radius 10 centered at each nuclear point is calculated and treated as the uncertainty of that nucleus. Since we model the data uncertainty, the nuclei with high uncertainty values have large noise in the Voronoi
label or cluster label. They often have non-uniform colors or similar color as the background, for example, some nuclei in the top right area of Fig. 2.5(a). They are hard to be accurately segmented using only points label. Therefore we sort the uncertainty values of all nuclei and select the top 5% representative nuclei for mask annotation (see Fig. 2.5(c)).

**Revising labels with masks.** For the selected nuclei, the masks are more accurate than their points annotation. Therefore, we integrate their masks into the cluster and Voronoi labels. In the cluster label, we copy the nuclei masks to the corresponding areas and add a small dilated area of 2-pixel width around each mask as background. In the Voronoi label, we replace the points with corresponding masks and mark the other pixels in those Voronoi cells as background, as shown in Fig. 2.4. These newly introduced background pixels may be not always correct, but they are beneficial to the training combined with the masks.

**Training with revised labels.** The revised labels combining the information of mixed points and masks annotation are used to train a regular CNN (see Fig. 2.4(b)). The loss function is the same as step 1 in Eqn. (2.13). We don’t train the model with dense CRF loss as in Section 2.3 because the selected masks already contain important shape information, and it will take an effort to find suitable parameters for the CRF loss.
2.5 Experiments

2.5.1 Datasets and Evaluation Metrics

To validate the proposed methods, we conduct experiments on two datasets of H&E stained histopathology images.

**Lung Cancer (LC) dataset.** We generated this dataset by extracting 40 images of size $900 \times 900$ from 8 lung adenocarcinoma or lung squamous cell carcinoma cases, i.e., H&E stained WSIs with 20x magnification. They are split into the training, validation and test sets, consisting of 24, 8 and 8 images, respectively. 24401 nuclei are annotated with masks.

**Multi-Organ (MO) dataset.** It is a public dataset released by Kumar et al. [55], and consists of 30 images of size $1000 \times 1000$ which are taken from multiple hospitals including a diversity of nuclear appearances from seven organs [55]. The variability in this dataset is large because of the heterogeneity between organs and cancer types. There are 12, 4 and 14 images in training, validation and test sets.

Both datasets have full mask annotation. We use the bounding box centers of the nuclear masks as ground-truth for the detection. To generate the partial points annotation in the training set, we randomly sample a certain ratio of points.

**Evaluation Metric for Nuclei Detection.** We adopt the common metrics for detection tasks: precision (P), recall (R) and F1 score. They are defined as

\[
P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}, \quad F1 = \frac{2TP}{2TP + FP + FN} \tag{2.14}
\]

where $TP, FP, FN$ are the number of true positives, false positives and false negatives, respectively. A detected nucleus is a true positive if it locates in a circle centered at a nuclear centroid with $r$-pixel radius, otherwise it is a false positive. The ground-truth points which have no corresponding detection are false negatives. If there are multiple detected points for the same ground-truth point, only the closest one is considered as a true positive. $r$ is the rough average nuclear radius computed using the validation set. We set $r = 8$ for the LC dataset and $r = 11$ for the MO dataset. We also adopt the mean ($\mu_d$) and standard deviation ($\sigma_d$) of the detection distance error to evaluate the
localization accuracy. They are defined as

$$
\mu_d = \frac{1}{N_{TP}} \sum_{i=1}^{N_{TP}} d_i, \quad \sigma_d = \sqrt{\frac{1}{N_{TP}} \sum_{i=1}^{N_{TP}} (d_i - \mu_d)^2}
$$

(2.15)

where $N_{TP}$ is the total number of true positive detected nuclei in all test images, $d_i$ is the Euclidean distance between the $i$-th groundtruth point and the true positive detection.

**Evaluation Metric for Nuclei Segmentation.** Four metrics are used to evaluate the segmentation performance. At pixel-level, we use pixel accuracy and pixel-level F1 score. Because nuclei segmentation is an instance segmentation task, two object-level metrics are also used: object-level Dice coefficient [91] ($Dice_{obj}$) and the Aggregated Jaccard Index (AJI) [55]. The Dice coefficient [28] measures how much the segmented object $S_j$ overlaps with its groundtruth object $G_i$. It is defined as

$$
Dice(G_i, S_j) = \frac{2|G_i \cap S_j|}{|G_i| + |S_j|}
$$

(2.16)

The object-level Dice coefficient takes into account each object individually, and measure how well each segmented object overlaps with the ground truth objects, as well as how well each ground truth object overlaps the segmented objects [91]. $Dice_{obj}$ is defined as

$$
Dice_{obj}(G, S) = \frac{1}{2} \sum_{i=1}^{n_G} \gamma_i Dice(G_i, S^*(G_i)) + \frac{1}{2} \sum_{j=1}^{n_S} \eta_j Dice(G^*(S_j), S_j)
$$

(2.17)

where $\gamma_i$, $\eta_j$ are the weights related to object areas, $G$, $S$ are the set of ground-truth objects and segmented objects, $S^*(G_i)$, $G^*(S_i)$ are the segmented object that has maximum overlapping area with $G_i$ and ground-truth object that has maximum overlapping area with $S_i$, respectively. The correspondence is established if the overlapping area of two objects are more than 50%.

The AJI takes both detection and segmentation performance into consideration. It computes an aggregated intersection cardinality numerator, and an aggregated union cardinality denominator for all groundtruth and segmented nuclei [55]:

$$
AJI = \frac{\sum_{i=1}^{n_G} |G_i \cap S(G_i)|}{\sum_{i=1}^{n_G} |G_i \cup S(G_i)| + \sum_{k \in K} |S_k|}
$$

(2.18)
where $S(G_i)$ is the segmented object that has maximum overlap with $G_i$ based on the Jaccard index, $K$ is the set containing segmentation objects that have not been assigned to any ground-truth object.

### 2.5.2 Detection Results using Partial Points Annotation

The aim of this experiment is to detect all nuclei in an image using the model trained with partial points annotation.

**Implementation details.** Color normalization [82] is applied to all images to remove color variations caused by staining. We extract 16 image patches of size $250 \times 250$ from each training image, and randomly crop $224 \times 224$ patches as network inputs. Other data augmentations are conducted including random crop, scale, rotation, flipping, and affine transformations. The encoder part of the network is initialized with the pre-trained parameters. The model is trained using the Adam optimizer [52] for 80 epochs in initial training and each round of self-training. The batch size is 16. The learning rate is $1e^{-4}$ for LC dataset and $1e^{-3}$ for MO dataset. The parameters in the extended Gaussian mask in Eqn. (2.1) are $r_1 = 8, r_2 = 16, \sigma = 2$ for LC dataset and $r_1 = 11, r_2 = 22, \sigma = 2.75$ for MO dataset. During inference, models from round 3 of self-training are used to evaluate on the test data, since it is already converged.

### Results and discussion

We compare different strategies mentioned in Section 2.3.1:

- **Full**: fully-supervised training using all annotated nuclei.
- **GM**: initial training using simple Gaussian masks from partial points annotation, i.e., no ignored pixels.
- **ext-GM**: initial training using our proposed extended Gaussian masks from partial points annotation.
- **ST-nu**: updating the label by adding detected nuclei in the self-training step.
Table 2.1: Nuclei detection results of different strategies on LC and MO datasets using 10% partial points annotation.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>P</th>
<th>R</th>
<th>F1</th>
<th>$\mu_d$</th>
<th>$\sigma_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>LC</td>
<td>Full</td>
<td>0.8767</td>
<td>0.9141</td>
<td>0.8950</td>
<td>1.14</td>
<td>1.03</td>
</tr>
<tr>
<td></td>
<td>GM</td>
<td>0.6322</td>
<td>0.6337</td>
<td>0.6329</td>
<td>2.34</td>
<td>2.00</td>
</tr>
<tr>
<td></td>
<td>ext-GM</td>
<td>0.7483</td>
<td>0.9306</td>
<td>0.8296</td>
<td>1.74</td>
<td>1.45</td>
</tr>
<tr>
<td></td>
<td>ST-nu</td>
<td>0.7505</td>
<td>0.9016</td>
<td>0.8192</td>
<td>1.70</td>
<td>1.49</td>
</tr>
<tr>
<td></td>
<td>ST-bg</td>
<td>0.8605</td>
<td>0.9171</td>
<td>0.8879</td>
<td>1.42</td>
<td>1.22</td>
</tr>
<tr>
<td>MO</td>
<td>Full</td>
<td>0.8420</td>
<td>0.8665</td>
<td>0.8541</td>
<td>2.68</td>
<td>2.00</td>
</tr>
<tr>
<td></td>
<td>GM</td>
<td>0.5574</td>
<td>0.7650</td>
<td>0.6449</td>
<td>3.85</td>
<td>2.68</td>
</tr>
<tr>
<td></td>
<td>ext-GM</td>
<td>0.7932</td>
<td>0.8471</td>
<td>0.8193</td>
<td>2.87</td>
<td>2.18</td>
</tr>
<tr>
<td></td>
<td>ST-nu</td>
<td>0.7754</td>
<td>0.8138</td>
<td>0.7941</td>
<td>2.95</td>
<td>2.17</td>
</tr>
<tr>
<td></td>
<td>ST-bg</td>
<td>0.8238</td>
<td>0.8328</td>
<td>0.8282</td>
<td>2.90</td>
<td>2.07</td>
</tr>
</tbody>
</table>

Table 2.2: Nuclei detection results on LC and MO datasets using different ratios of annotation.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>P</th>
<th>R</th>
<th>F1</th>
<th>$\mu_d$</th>
<th>$\sigma_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>LC</td>
<td>Full</td>
<td>0.8767</td>
<td>0.9141</td>
<td>0.8950</td>
<td>1.14</td>
<td>1.03</td>
</tr>
<tr>
<td></td>
<td>5%</td>
<td>0.8564</td>
<td>0.9171</td>
<td>0.8857</td>
<td>1.53</td>
<td>1.30</td>
</tr>
<tr>
<td></td>
<td>10%</td>
<td>0.8605</td>
<td>0.9171</td>
<td>0.8879</td>
<td>1.42</td>
<td>1.22</td>
</tr>
<tr>
<td></td>
<td>25%</td>
<td>0.8517</td>
<td>0.9399</td>
<td>0.8936</td>
<td>1.35</td>
<td>1.17</td>
</tr>
<tr>
<td></td>
<td>50%</td>
<td>0.8502</td>
<td>0.9414</td>
<td>0.8935</td>
<td>1.30</td>
<td>1.12</td>
</tr>
<tr>
<td>MO</td>
<td>Full</td>
<td>0.8420</td>
<td>0.8665</td>
<td>0.8541</td>
<td>2.68</td>
<td>2.00</td>
</tr>
<tr>
<td></td>
<td>5%</td>
<td>0.8021</td>
<td>0.8441</td>
<td>0.8226</td>
<td>3.04</td>
<td>2.13</td>
</tr>
<tr>
<td></td>
<td>10%</td>
<td>0.8238</td>
<td>0.8328</td>
<td>0.8282</td>
<td>2.90</td>
<td>2.07</td>
</tr>
<tr>
<td></td>
<td>25%</td>
<td>0.8259</td>
<td>0.8440</td>
<td>0.8349</td>
<td>2.97</td>
<td>2.05</td>
</tr>
<tr>
<td></td>
<td>50%</td>
<td>0.8237</td>
<td>0.8821</td>
<td>0.8519</td>
<td>2.76</td>
<td>2.03</td>
</tr>
</tbody>
</table>

- **ST-bg**: updating the label by propagating background pixels in the self-training step.

For both ST-nu and ST-bg, ext-GM is used in the initial training. The detection results using 10% points in each training image are reported in Table 2.1.

**Initial training strategies.** In the two initial training strategies, the ext-GM achieves better performance on both datasets. For GM, all the unlabeled nuclei are treated as background, which biases the training and guides the network to predict pixels as background more aggressively. Our extended Gaussian masks force the model to focus on the areas around the labeled points. As a result, the trained model is able to make correct predictions in similar unlabeled regions.
Figure 2.6: Typical detection results of different strategies from Lung Cancer dataset (first row) and Multi-Organ dataset (second row) using 10% points. (a) Fully supervised training, (b) initial training using simple Gaussian mask, (c) initial training using extended Gaussian mask, (d) self-training using nuclei prediction, (e) self training using background propagation. Yellow dots are the detected nuclei. Green, blue and red circles represent ground-truth with correct detection (TP), ground-truth without correct detection (FN) and false positive detection (FP), respectively.

Self-training strategies. In the self-training stage, compared with the results of the first stage (ext-GM), updating the nuclei (ST-nu) decreases the performance. The reason is that the number of false positives in the newly added nuclei is comparable to that of the labeled nuclei, resulting in a negative effect on training. In contrast, our background propagation strategy (ST-bg) keeps the labeled nuclei unchanged and gradually increases the background area (as shown in Fig. 2.2), which doesn’t introduce false positives during training. Therefore, it can improve both recall and precision, resulting in a much higher F1 score and localization accuracy.

Comparison to fully-supervised case. Compared with the results of full annotation (Full), our method (ST-bg) can achieve comparable performance while using much fewer annotation data. On the LC dataset, the precision, recall and F1 are 98.2%, 100.3%, 99.2% of the fully-supervised results, respectively. And these numbers are 97.8%, 96.1%, 97.0% respectively on the MO dataset. Besides, the localization error $\mu_d \pm \sigma_d$ is also very close to that using full annotation on both datasets.

The typical qualitative results of different training strategies from both datasets using 10% points are shown in Fig. 2.6.
Different ratios of annotation. To explore how the proposed nuclei detection method behaves when the ratio of points changes, we trained with 5%, 10%, 25% and 50% partial points annotation. The results are shown in Table 2.2. The detection accuracy (F1-score) increases and localization error ($\mu_d \pm \sigma_d$) decreases as more annotations are available. With 50% points annotated, the performance is nearly the same as that using full annotation. Even with only 5% annotation, the F1 score of our method can reach 99.0% of that using full annotation on LC dataset and 96.3% on the MO dataset, which substantiates the effectiveness of our algorithm. The results of MO dataset are slightly worse compared with those on the LC dataset because the MO dataset is more challenging due to the diversity in nuclear size and appearance.

2.5.3 Segmentation Results using Ground-truth Points

In this subsection, we present the experimental results of our segmentation method based on all ground-truth points. We first try to obtain the optimal value of $\alpha$ in Eqn. (2.6) and discuss the effects of two types of labels, then show the effects of the dense CRF loss, and finally compare our results with fully supervised ones.

Implementation details. In weakly supervised settings we train a model for 100 epochs with a learning rate of 1e-4, and fine-tune the model using the dense CRF loss for 20 epochs with a learning rate of 1e-5. In fully supervised settings, we train 200 epochs using binary masks with a learning rate of 1e-4. The validation set is used to select the best model for testing.

Results and discussion

The effects of two types of labels. In order to explore how the two types of generated labels work on the model training, we change the values of $\alpha$ in Eqn. (2.6). As $\alpha$ changes from 0 to 1, all four metrics increase in the beginning and then decrease (shown in Fig. 2.8). Compared to the results using only the cluster labels ($\alpha = 0$), those with Voronoi labels ($\alpha = 1$) are better in the object-level metrics, but worse in pixel-level metrics. This is because the model trained with Voronoi labels predicts the central parts of nuclei, resulting in small separated instances (Fig. 2.7(g)). While
Figure 2.7: Typical results using different weights ($\alpha$) of the cluster and Voronoi labels on LC dataset (top row) and MO dataset (bottom row). (a) image, (b) ground-truth full mask, (c)-(g) are results using different $\alpha$ values. $\alpha = 0$ means using only the cluster label and $\alpha = 1$ means using only the Voronoi label.

lacking the Voronoi edge information, the model using cluster labels is not able to separate close nuclei (Fig. 2.7(c)). In contrast, segmentation results using both labels (Fig. 2.7(d)-(f)) are better than those with either label alone, because they have both the shape information from the cluster label and the nuclei/background information from the Voronoi label. The best performance is achieved when $\alpha$ is around 0.5, thus we set $\alpha = 0.5$ for all subsequent experiments.

The effects of dense CRF loss. In the dense CRF loss, the Gaussian bandwidth parameters $\sigma_{pq}$ and $\sigma_{rgb}$ control the affinity between pixel pairs, thus having an impact on the effect of the loss along with the weight $\beta$. We perform an ablation study on the three parameters to show how their values affect the segmentation performance. The ranges are $\sigma_{pq} \in \{3, 6, 9, 12, 15\}$, $\sigma_{rgb} \in \{0.05, 0.1, 0.2, 0.3\}$ and
Figure 2.9: The improvements over the baseline using different $\sigma_{pq}$, $\sigma_{rgb}$ and $\beta$ values in the CRF loss for LC (top row) and MO (bottom row) datasets. In each subfigure, the horizontal black dash line indicates the baseline’s performance, and the vertical red dash line indicates the parameter value of the best combination.

$\beta \in \{0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001\}$. Among different value combinations, the best for LC dataset is $\sigma_{pq} = 9$, $\sigma_{rgb} = 0.2$, $\beta = 0.001$, and that of MO dataset is $\sigma_{pq} = 9$, $\sigma_{rgb} = 0.1$, $\beta = 0.005$.

The results of varying the value of one parameter based on the best combination are shown in Fig. 2.9. For simplicity, we plot the improvements of the finetuned models over the baseline. When $\sigma_{pq}$ is small, only pixel pairs that are close enough have a large affinity. As a result, the current pixel’s prediction is affected by local pixels, which leads to good results for small objects. On the contrary, a large $\sigma_{pq}$ is good for large objects. Taking all nuclei into account, $\sigma_{pq}$ should not be too small or too large, which is revealed by the results in Fig. 2.9. The rule is the same for $\sigma_{rgb}$. The weight $\beta$ adjusts the importance of the unary and pairwise potentials in the loss (Eqn. (2.10)). A large $\beta$ emphasizes the pairwise relationship obtained from the image during the fine-tuning process, biasing the baseline model trained on the labels. Therefore, the performance degrades a lot, especially on the LC dataset. While a small $\beta$ makes the fine-tuning less effective.
Table 2.3: Nuclei segmentation results on LC and MO datasets for our partial points annotation, ground-truth points annotation, points with 5% masks, fully-supervised and state-of-the-arts.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Supervision</th>
<th>Method</th>
<th>Pixel-level</th>
<th>Object-level</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Acc</td>
<td>F1</td>
</tr>
<tr>
<td>LC</td>
<td>Full masks</td>
<td>Qu et al. [75]</td>
<td>-</td>
<td>0.8860</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fully-sup</td>
<td>0.9615</td>
<td>0.8771</td>
</tr>
<tr>
<td></td>
<td>GT points</td>
<td>without CRF</td>
<td>0.9413</td>
<td>0.8028</td>
</tr>
<tr>
<td></td>
<td></td>
<td>with CRF</td>
<td>0.9433</td>
<td>0.8120</td>
</tr>
<tr>
<td></td>
<td>Partial points</td>
<td>5%</td>
<td>0.9262</td>
<td>0.7612</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10%</td>
<td>0.9312</td>
<td>0.7700</td>
</tr>
<tr>
<td></td>
<td></td>
<td>25%</td>
<td>0.9331</td>
<td>0.7768</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50%</td>
<td>0.9332</td>
<td>0.7819</td>
</tr>
<tr>
<td></td>
<td>Points &amp; Masks</td>
<td>random</td>
<td>0.9485</td>
<td>0.8283</td>
</tr>
<tr>
<td></td>
<td></td>
<td>uncertainty</td>
<td><strong>0.9501</strong></td>
<td><strong>0.8366</strong></td>
</tr>
<tr>
<td>MO</td>
<td>Full masks</td>
<td>CNN3 [55]</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DIST [68]</td>
<td>-</td>
<td>0.7623</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CIA-Net [129]</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fully-sup</td>
<td>0.9194</td>
<td>0.8100</td>
</tr>
<tr>
<td></td>
<td>GT points</td>
<td>without CRF</td>
<td>0.9052</td>
<td>0.7745</td>
</tr>
<tr>
<td></td>
<td></td>
<td>with CRF</td>
<td>0.9071</td>
<td><strong>0.7776</strong></td>
</tr>
<tr>
<td></td>
<td>Partial points</td>
<td>5%</td>
<td>0.8951</td>
<td>0.7540</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10%</td>
<td>0.8997</td>
<td>0.7490</td>
</tr>
<tr>
<td></td>
<td></td>
<td>25%</td>
<td>0.8966</td>
<td>0.7511</td>
</tr>
<tr>
<td></td>
<td></td>
<td>50%</td>
<td>0.8999</td>
<td>0.7566</td>
</tr>
<tr>
<td></td>
<td>Points &amp; Masks</td>
<td>random</td>
<td>0.9111</td>
<td>0.7753</td>
</tr>
<tr>
<td></td>
<td></td>
<td>uncertainty</td>
<td><strong>0.9114</strong></td>
<td><strong>0.7748</strong></td>
</tr>
</tbody>
</table>

Results using ground-truth points. With the above parameter settings, the results using all ground-truth points are shown in Table 2.3. The segmentation performance of our weakly supervised method using all ground-truth points is close to that of the fully supervised models with the same network structure. On the Lung Cancer dataset, the gaps for accuracy, F1 score, Dice and AJI are 2.0%, 7.2%, 5.9%, 6.9%, respectively. On the MultiOrgan dataset, the gaps for accuracy and F1 score are 1.1% and 4.7%, respectively. However, the fully supervised model has very low Dice and AJI, since for fair comparison we didn’t perform post-processing to separate the touching nuclei for any of the methods. The weakly supervised model is able to separate most of them due
Figure 2.10: The results of all test images using partial points and ground-truth points (100%) in the LC dataset (first row) and MO dataset (second row). \textit{nuclei-bg-diff} is the difference between pixel values of nuclei and background. \textit{nuclei-std} is the standard deviation of the pixel values within nuclei.

to the Voronoi labels while the fully supervised model failed to achieve this. Compared to the CNN3 method in [55], our method achieved a similar accuracy in terms of the AJI value. Compared to the state-of-the-art DIST method [68], our approach has a higher pixel-level F1 score, but still has room for improvement on the nuclear shapes, as indicated by the AJI values.

2.5.4 Segmentation Results using Detected Points

The settings, including parameters in the loss function and training details, are the same as those using ground-truth points. The only difference is that compared to the ground-truth points there are errors in the detected points, i.e., false positives, false negatives and localization errors. As a result, the errors in the generated Voronoi labels and cluster labels using detected points increase, which will degrade the performance.

We report the results using detected points from different ratios of initial points
annotation in Table 2.3. Even with only 5% annotated points, the proposed framework
can achieve satisfactory segmentation performance compared to the fully-supervised
ones. As the annotated points increase from 5% to 50%, the overall segmentation per-
formance becomes better. This is quite reasonable because the performance is affected
by the detection results and the detection error decreases for higher annotated points
ratio, as shown in Table 2.2.

To explore the underlying factors that affect the performance, we compute two
statistical metrics of the datasets. One is the average difference between the pixel
values of nuclei and its surrounding background \( \text{nuclei-bg-diff} \) and defined as:

\[
\text{nuclei-bg-diff} = \sum_{i=1}^{N} \frac{A_i}{A_{\text{total}}} (\mu_{i}^{\text{nucleus}} - \mu_{i}^{\text{bg}}),
\]

where \( N \) is the number of nuclei in the image, \( A_i \) and \( A_{\text{total}} \) are the areas of the \( i \)-th
nucleus and all nuclei, \( \mu_{i}^{\text{nucleus}} \) and \( \mu_{i}^{\text{bg}} \) are the average pixel values of the \( i \)-th nucleus
and its surrounding background. We treat the annular area with radius 3 around
the nucleus as its background area. The larger the \( \text{nuclei-bg-diff} \) is, the better the
algorithm recognizes each nucleus. The other metric is the standard deviation of pixel
values within each nucleus \( \text{nuclei-std} \), and defined as:

\[
\text{nuclei-std} = \sum_{i=1}^{N} \frac{A_i}{A_{\text{total}}} \sigma_{i}^{\text{nucleus}},
\]

where \( \sigma_{i} \) is the standard deviation of pixel values within the \( i \)-th nucleus. The smaller
the \( \text{nuclei-std} \) is, the more uniform color in each nucleus, which improves segmentation
of the entire nucleus. We show the segmentation results of all test images of LC and MO
datasets in Fig. 2.10, respectively. It can be observed that images with large \( \text{nuclei-bg-
diff} \) and small \( \text{nuclei-std} \) have much better segmentation performance, e.g., KZ-5 in LC
dataset and Kidney2 in MO dataset. Besides, for those with small \( \text{nuclei-bg-diff} \) and
large \( \text{nuclei-std} \), the performance gap between partial points and ground-truth points
are larger than other images, e.g., MV-5 in LC dataset and Prostate1 in MO dataset.
Because the nuclei in these images have similar appearance as background pixels and
large color variance, thus are hard to be accurately extracted and more sensitive to
the errors when using partial points. Typical segmentation results of both datasets are
shown in Fig. 2.11 and Fig. 2.12.
2.5.5 Segmentation Results using Mixed Points and Masks

Implementation details. The preprocessing on images and data augmentations are the same as those in the weakly supervised segmentation. Adam optimizer is used to train both Bayesian and regular CNNs. The learning rate and batch size are 1e-4 and 8, respectively. In the uncertainty prediction step, the model is trained 100 epochs. The number of Monte Carlo sampling is set to $T = 20$. In the training with mixed labels step, the model is trained 150 epochs.

Results and discussion. The quantitative and typical qualitative results on the Lung Cancer and Multi-Organ datasets are shown in Table 2.3, and Fig. 2.13. We compare our method with the models trained with ground-truth points, the state-of-the-art methods on each dataset [68, 75, 129], and the same network trained with full mask labels.

Compared with the weakly supervised method using ground-truth points in Section 2.5.3, this method using mixed annotation with uncertainty achieves much better performance. The gain attributes to two aspects. One is that the masks introduce extra shape information about some nuclei. That’s why the results using the masks of randomly selected nuclei also outperform those using only points. The other aspect is the uncertainty helps to pick the representative nuclei, resulting in better accuracy.
Figure 2.12: Typical bad segmentation results using different ratios of points annotation on LC dataset (top row) and MO dataset (bottom row). (a) image, (b) ground-truth full mask, (c)-(f) are results using different ratios of points, (g) the results of using all ground-truth (100%) points without the detection step. Distinct colors represent different nuclei.

than random selection, especially on nuclei with non-uniform colors (see Fig. 2.13). Besides, our method doesn’t rely on the CRF loss because the shape information of newly added nuclei can help training. It is more convenient compared to searching suitable parameters in the CRF loss.

The addition of a few full masks narrows the gap between weakly supervised training and the fully supervised training using the same network and full masks. On the Lung Cancer dataset, the differences on pixel accuracy, F1, Dice and AJI are 1.2%, 4.6%, 3.2%, 2.8%, respectively. On the Multi-Organ dataset, the fully supervised model has low Dice and AJI because it cannot separate the crowded nuclei well.

Compared to the state-of-the-art fully-supervised methods [68, 75, 129], our method obtains close overall performance, but still needs to be improved with regard to the object-level accuracy. The introduction of 5% masks is not enough to cover the shape information of all nuclei, leading to sub-optimal solutions in extremely hard cases. For example, in the cases of Fig. 2.13, methods using weak supervision cannot accurately capture the shape of those nuclei without seeing sufficient masks of similar nuclei in the training set. It remains open on how to handle this issue effectively in the weakly supervised setting.
Table 2.4: Nuclei detection results on LC and MO datasets using 10 different sets of 10% initial points.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>P</th>
<th>R</th>
<th>F1</th>
<th>$\mu_d$</th>
<th>$\sigma_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>LC</td>
<td>0.8544</td>
<td>0.9225</td>
<td>0.8868</td>
<td>1.43</td>
<td>1.23</td>
</tr>
<tr>
<td>MO</td>
<td>0.8227</td>
<td>0.8453</td>
<td>0.8330</td>
<td>2.92</td>
<td>2.11</td>
</tr>
</tbody>
</table>

2.5.6 Sensitivity and Generalization Analyses

Two more experiments are conducted to analyze the sensitivity of our weakly supervised method using partial points and the generalization performance of the trained models.

**Sensitivity.** To explore how the initial selected points will affect the final performance of our method, we randomly select ten different sets of 10% initial points, and perform detection and segmentation. The results are reported in Table 2.4 and Table 2.5. The small variances in the metrics indicate that our method is not sensitive to the choice of initial points.

**Generalization.** Generalization performance is an important aspect when applying a method to other datasets. We use the best segmentation model trained on one dataset (LC/MO) to test its performance on the other dataset (MO/LC). The results are shown...
Table 2.5: Nuclei segmentation results on LC and MO datasets using 10 different sets of 10% initial points.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Pixel-level</th>
<th>Object-level</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Acc</td>
<td>F1</td>
<td>Diceobj</td>
</tr>
<tr>
<td>LC</td>
<td>mean</td>
<td>0.9278</td>
<td>0.7695</td>
<td>0.7571</td>
<td>0.5880</td>
</tr>
<tr>
<td></td>
<td>std</td>
<td>0.00264</td>
<td>0.00381</td>
<td>0.00443</td>
<td>0.00782</td>
</tr>
<tr>
<td>MO</td>
<td>mean</td>
<td>0.8982</td>
<td>0.7484</td>
<td>0.7089</td>
<td>0.5158</td>
</tr>
<tr>
<td></td>
<td>std</td>
<td>0.00472</td>
<td>0.00753</td>
<td>0.00666</td>
<td>0.00413</td>
</tr>
</tbody>
</table>

Table 2.6: Generalization performance of our weakly supervised method using partial points on LC and MO datasets.

<table>
<thead>
<tr>
<th>Train → Test</th>
<th>Ratio</th>
<th>Pixel-level</th>
<th>Object-level</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Acc</td>
<td>F1</td>
<td>Diceobj</td>
</tr>
<tr>
<td>MO → LC</td>
<td>5%</td>
<td>0.9271</td>
<td>0.7589</td>
<td>0.7418</td>
<td>0.5608</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10%</td>
<td>0.9213</td>
<td>0.7518</td>
<td>0.7297</td>
<td>0.5555</td>
<td></td>
</tr>
<tr>
<td></td>
<td>25%</td>
<td>0.9222</td>
<td>0.7551</td>
<td>0.7320</td>
<td>0.5588</td>
<td></td>
</tr>
<tr>
<td></td>
<td>50%</td>
<td>0.9226</td>
<td>0.7559</td>
<td>0.7336</td>
<td>0.5608</td>
<td></td>
</tr>
<tr>
<td>LC → MO</td>
<td>5%</td>
<td>0.9004</td>
<td>0.7419</td>
<td>0.7028</td>
<td>0.4884</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10%</td>
<td>0.8964</td>
<td>0.7338</td>
<td>0.6913</td>
<td>0.4971</td>
<td></td>
</tr>
<tr>
<td></td>
<td>25%</td>
<td>0.8974</td>
<td>0.7234</td>
<td>0.6886</td>
<td>0.4870</td>
<td></td>
</tr>
<tr>
<td></td>
<td>50%</td>
<td>0.8970</td>
<td>0.7232</td>
<td>0.6986</td>
<td>0.5030</td>
<td></td>
</tr>
</tbody>
</table>

in Table 2.6. When applying the MO models to LC test set (MO → LC), they can achieve 92% to 99% performance compared to the models trained on the LC dataset. On the MO test set, the models trained on the LC dataset can achieve 95% to 99% performance compared to the models trained on the MO dataset. The results illustrate the good generalization performance of our model on different nuclei datasets.

2.5.7 Annotation Time

Dr. Riedlinger, a board-certified pathologist annotated eight images (one per case) in the LC dataset using points, bounding boxes and full masks, respectively. The average time spent on each image (about 600 nuclei on average) for full masks is 115 minutes, while for bounding boxes 67 minutes. However, it only takes about 14 minutes for all points annotation and less than 2 minutes for 10% points annotation. Adding 5% masks introduces extra 6 minutes based on points annotation, which is not a big deal.
compared to full masks.

2.6 Summary

In this chapter, we first present a new weakly supervised nuclei segmentation method using only a small portion of nuclei locations. In the first stage, a semi-supervised nuclei detection algorithm is proposed to obtain the locations of all nuclei from the partial annotation. In the second stage, we perform nuclei segmentation using the detected points as weak labels. We generate the Voronoi label and cluster label from the detected points and take advantage of the dense CRF loss to refine the trained model. Our method achieves comparable performance as fully supervised methods while requiring much less annotation effort which in turn allows us to analyze large amounts of data.

In order to further improve the segmentation performance on nuclei with non-uniform colors, we propose to annotate 5% full masks of representative nuclei. Uncertainty map is estimated from a Bayesian CNN trained with points annotation. Then the 5% representative hard nuclei are selected according to the uncertainty map to obtain the full masks. The masks are combined with the two types of labels derived from points annotation to supervise the training of a segmentation model. The addition of a few full masks reduces the gap between segmentation in the weakly supervised setting and fully supervised setting while requiring a little extra annotation effort.
Chapter 3
Improving Nuclei Segmentation by Full Resolution Neural Network and Spatial Constrained Loss

3.1 Introduction

In the last chapter, we discuss how to address the data annotation problem in nuclei segmentation using weak labels. In this chapter, we focus on the problems of network structure and loss function in current deep learning-based nuclei segmentation methods.

Deep CNNs were originally developed for image classification. They have similar structures, consisting of successive convolutional and pooling layers, followed by several fully-connected layers. This type of structure can be directly applied in image segmentation by predicting each pixel into different classes using a patch around the pixel as input, which is done by most of early deep learning-based nuclei segmentation algorithms [107, 46, 55]. There are two main drawbacks for such methods as described in [84]. One is the high computational cost because each forward pass can only predict the class of one pixel. Besides, there is a trade-off between localization accuracy and the use of context [84]. Smaller patches contain little context while large patches result in lower localization accuracy due to more pooling layers. Another type of structure for image segmentation is the fully convolutional networks (FCNs), proposed by Long et al. [62]. It replaces the fully-connected layers of previous architecture for classification with up-sampling and convolutional layers, enabling the output of full resolution probability maps. Based on Long’s work, Ronneberger et al. [84] proposed U-net that has more up-sampling layers and skip connections between layers in down-sampling and up-sampling parts. It takes advantage of the detailed features in down-sampling part for localization. FCNs are more efficient and accurate for segmentation tasks. In medical image segmentation, variations of FCNs have been used to solve different problems,
such as neuronal structures segmentation [84], gland segmentation [18, 33, 90, 115], liver segmentation [22] and nuclei segmentation [34, 67, 68, 129, 75, 76, 77, 81].

Although FCNs for segmentation have achieved promising results, there are limitations. The encoder part of FCNs is inspired by structures originally designed for image classification and contains pooling or strided convolution operations. These operations can greatly increase the receptive field to distill more-abstract features and make the network robust against small translations in the image [73]. However, such down-sampling may reduce the localization accuracy of boundaries in segmentation tasks, because the reduced size of feature maps leads to the loss of details like boundaries and tiny objects. As a result, post-processing steps are often needed to refine the segmentation results. Other methods, such as the skip connection in U-net [84], can solve this problem to some extent, but it is not good enough. Another issue in these methods is the cross entropy loss used for training. The loss only cares about if the classification of individual pixel is correct or not, and pays no attention to the spatial relationship between pixels. Although the deep neural networks can learn some high-level features like objects prior from ground-truth labels, the segmentation results are still not satisfactory when the objects have various colors, textures and shapes, especially in medical images.

To solve these issues, we propose a full resolution neural network (FullNet) to improve the localization accuracy and a variance constrained cross entropy (varCE) loss to enhance the learning of pixels’ spatial relationship. The FullNet consists of densely connected layers [42] and doesn’t contain any pooling or up-sampling layers. All feature maps and output have the same full resolution as the input image, thus keeping the information (e.g., edges) as much as possible to improve segmentation performance. For each convolution layer, dilated convolution [120] is utilized to increase the receptive field, similar to the effect of pooling operations in classification networks. The varCE loss combines the regular cross entropy term with a variance term. The cross entropy term performs classification for each pixel. The variance term aims to reduce the variance of pixels’ probabilities within each instance. It works as a spatial constraint on pixels belonging to the same instance, and helps the network to better understand
the shapes of objects. The combination of FullNet and varCE loss is able to achieve state-of-the-art performance on nuclei segmentation and gland segmentation.

3.2 Methods

In this section, we will first introduce the structure of FullNet and the variance constrained cross entropy loss. Then the self-supervised learning is adopted to obtain the pretrained weights for FullNet.

3.2.1 FullNet Structure

The structure of our FullNet is shown in Fig. 3.1. It consists of a $3 \times 3$ convolutional layer, seven blocks of densely connected layers with different dilation factors, a $1 \times 1$ convolutional layer after each block, and a final $3 \times 3$ convolutional layer.

There are no pooling layers in the network, thus the feature maps retain the same resolution as the input to avoid the loss of details by pooling. The potential memory problem caused by the large number and size of feature maps is solved by the dense block structure. As in [42], each layer in a dense block takes all preceding feature maps as input, and outputs a fixed number of feature maps. This can enhance the feature reuse and strengthen feature propagation, thus achieving good performance with fewer feature maps. Different from the original dense blocks, we adopt Conv-LeakyReLU-BN instead of BN-ReLU-Conv structure for each basic dense layer. Putting BN after convolution in a dense layer can significantly reduce the memory requirements of intermediate feature maps, because for each dense layer the number of input feature maps is often much
Figure 3.2: Dilation factors in (a) U-net and (b) FullNet. Each red dash line represents a change of dilation factor in FullNet or a down-sampling of feature maps in U-net.

larger than that of outputs. LeakyReLU is preferred due to its benefit of solving “dying ReLU” problem [63]. We also adopt $1 \times 1$ convolution before each basic dense layer to reduce the number of input feature maps as in [42]. This bottleneck dense layer has the structure $Conv(1 \times 1)$-LeakyReLU-BN-$Conv(3 \times 3)$-LeakyReLU-BN and the $1 \times 1$ convolution has $4k$ output feature maps, where $k$ is the growth rate, i.e., the number of output features of each dense layer. The $1 \times 1$ convolution layer right after each dense block can further compress the number of features by half.

In order to ensure the large receptive field for recognition, we employ dilated convolution [119] in dense blocks. Dilated convolution of a function $F$ and a filter $f$ is defined as

$$ (F *_d f)(p) = \sum_{s + dt = p} F(s)f(t) $$

(3.1)

where $*_d$ is the dilated convolution operator with dilation factor $d$. When $d = 1$, it becomes ordinary convolution. When $d$ is larger than 1, the value at position $p$ is generated by discontinuous values in a larger range, which means a larger receptive field. The dilation factors in dense blocks are designed by taking the U-net [84] as reference. In the U-net, each pooling layer reduces the size of feature maps by half, which doubles the effective receptive field since the filter size keeps the same. In the FullNet, to compensate for the shrinkage of receptive field after removing pooling layers, we use ordinary convolution in the first block and double the dilation factor for each subsequent block until block 5. In other words, the dilation factors are $d = (1, 2, 4, 8, 16)$ for the first five blocks of FullNet, shown in Fig. 3.2. In the figure, each red dash line indicates a change in dilation factor in FullNet. Each black rectangle represents a convolution.
Figure 3.3: Gridding artifacts. (a) input image patch (b) probability map with gridding artifacts (c) probability map after adding two more blocks and using hybrid dilation factor strategy.

layer with batch normalization and nonlinear activation, or a pooling/up-convolution layer in U-net. Filter size and the number of output feature maps are also written in the black rectangles. In FullNet, \( /2 \) means compressing the number of feature maps by half. No expansive path is needed like U-net to recover the resolution of feature maps because it doesn’t change. Although dilated convolutions can increase receptive fields, it has a side effect. From Eqn. (3.1), we know the pixel value at position \( p \) is generated by values in discontinuous locations of previous layer’s output. There is a gap of \( d – 1 \) between adjacent values, resulting in the gridding artifacts in feature maps and outputs (see Fig. 3.3(b)), especially when the dilation factor is large. Two strategies are utilized to alleviate this problem. Firstly, we use the similar solution in [120], i.e. adding two more blocks in the end of the network with progressively lower dilations. The dilation factors of each block in this final network become \( d = (1, 2, 4, 8, 16, 4, 1) \) and the number of blocks is 7. Besides, we use the hybrid dilation factors strategy proposed by Wang et al. [102] in all blocks except the first and last ones to reduce the artifacts further.

The network outputs three-class probability maps: inside object areas, edges and background areas. The predicted edges are beneficial to separate the crowded and touching nuclei/glands.
3.2.2 Variance Constrained Cross Entropy Loss

In segmentation tasks, cross entropy loss is most widely used to classify each pixel into the correct class. The cross entropy loss for $M$ classes is defined as

$$L_{CE}(y, t, w) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{m=1}^{M} w_i t_i^{(m)} \log y_i^{(m)}$$ (3.2)

where $N$ is the number of all pixels, $y_i^{(m)}$ is the probability of pixel $i$ belonging to class $m$, $t_i^{(m)} \in \{0, 1\}$ is the corresponding groundtruth label of class $m$, $w_i$ is the optional weight for pixel $i$.

The cross entropy loss doesn’t consider the spatial relationship between pixels. When an object in the image has non-uniform color or texture, the network often fails to segment the whole object using cross entropy loss. To address the problem, we add a variance term to the cross entropy loss:

$$L_{var}(y, t) = \frac{1}{C} \sum_{c=1}^{C} \frac{1}{|S_c|} \sum_{i=1}^{|S_c|} (\mu_c - \hat{y}_i)^2$$ (3.3)

where $C$ is the number of instances, $S_c$ is the set of pixels that belong to instance $c$, $|S_c|$ is the number of pixels in set $S_c$, $\hat{y}_i$ is the probability of the correct class for pixel $i$, and $\mu_c$ is the mean value of pixels’ probabilities $\hat{y}_i$ in set $S_c$:

$$\mu_c = \frac{1}{|S_c|} \sum_{i=1}^{|S_c|} \hat{y}_i$$ (3.4)

Note that the variance is computed within each instance, thus placing a local constraint for pixels belonging to the same instance. Ideally, if the variance is zero for a certain instance, pixels in the range of this instance have the same probabilities $\hat{y}_i$, resulting in consistent predictions for those pixels.

The final variance constrained cross entropy (varCE) loss can be written as

$$L_{varCE} = L_{CE} + \alpha L_{var}$$ (3.5)

where $\alpha$ is a parameter that adjusts the weight of the variance term. In the varCE loss, the variance term pulls the pixels in each instance towards the same class and meanwhile the cross entropy term drives these pixels to the correct class. As a result, the segmentation performance will be enhanced.
For datasets with highly skewed frequencies of classes, the weight $w_i$ in Eqn. (3.2) is often set according to the class of pixel $i$. Here we don’t use class weights. Instead, we calculate the weight of each pixel according to its distances to the two nearest objects. Pixels between close or touching objects are assigned much larger weights because those pixels are more important to split objects. The weight of pixel $x_i$ is calculated using a similar method in [84]:

$$w(x_i) = 1 + w_0 \cdot \exp \left( -\frac{(d_1(x_i) + d_2(x_i))^2}{2\sigma^2} \right)$$

(3.6)

where $d_1, d_2$ are the distances to the nearest and the second nearest object, respectively. In the experiments, we set $\sigma = 5$ pixels for nuclei segmentation, $\sigma = 20$ pixels for gland segmentation, and $w_0 = 10$ for both tasks.

### 3.2.3 Post-processing

With the trained model, we get a three-class segmentation map for a test image. The map corresponding to inside object areas is selected as the initial segmentation map. Morphological operations including connected component labeling, small areas removal and dilation with a disk filter are performed to obtain the final segmentation results. Because the predicted edges are often thicker than true edges, which help to separate touching objects, the initial prediction of the inside is not the exact mask of objects. In order to include the periphery areas, the dilation operation is necessary.

### 3.3 Experiments

#### 3.3.1 Datasets and Evaluation Metrics

**Datasets.** As in the last chapter, we evaluate our method on nuclei segmentation task using the public Multi-Organ dataset [55]. In order to better illustrate the effects of the varCE loss, we conduct experiments on the gland segmentation task using the MICCAI 2015 gland segmentation challenge dataset (GlaS) [90]. It consists of 165 images that come from 16 H&E stained histological sections of colorectal adenocarcinoma. They are separated into train, testA and testB parts, containing 80, 65 and 20 images, respectively. Train and testA parts have both benign and malignant cases, while testB
set are mostly malignant cases.

**Evaluation Metrics.** For nuclei segmentation, we use the same metrics in [55]: F1-score (Eqn. (2.14)), average Dice coefficient (Eqn. (2.16)), average Hausdorff distance [44], and AJI (Eqn. (2.18)). The F1-score used here is a little different from that in the detection task due to the criterion for deciding the TP, FP, FN objects. A segmented object $S_j$ is considered as a true positive if it overlaps with at least 50% of a groundtruth object $G_i$. Hausdorff distance measures the shape similarity between segmented and groundtruth objects, and the Hausdorff distance between groundtruth object $G_i$ and segmented object $S_j$ is

$$H(G_i, S_j) = \max \left\{ \sup_{x \in G_i} \inf_{y \in S_j} d(x, y), \sup_{y \in S_j} \inf_{x \in G_i} d(x, y) \right\} \quad (3.7)$$

where $d(x, y)$ is the distance between pixels $x$ and $y$. For gland segmentation, three official metrics in the MICCAI 2015 Challenge are used: F1-score, object-level Dice coefficient and object-level Hausdorff distance. The object-level Dice coefficient was explained in Section 2.5 Eqn. (2.17). The object-level Hausdorff distance was introduced in [90] to calculate the overall segmentation accuracy between a pair of corresponding groundtruth and segmented objects. Similar to the object-level Dice coefficient in Eqn. (2.17), it is defined as

$$H_{obj}(G, S) = \frac{1}{2} \sum_{i=1}^{n_G} \gamma_i H(G_i, S^*(G_i)) + \frac{1}{2} \sum_{j=1}^{n_S} \eta_j H(G^*(S_j), S_j) \quad (3.8)$$

where $\gamma_i, \eta_j$ are the weights related to object areas, $G, S$ are the set of ground-truth objects and segmented objects, $S^*(G_i), G^*(S_i)$ are the segmented object that has maximum overlapping area with $G_i$ and ground-truth object that has maximum overlapping area with $S_i$, respectively. The correspondence is established if the overlapping area of two objects are more than 50%. When a groundtruth object $G_i$ does not have a corresponding segmented object, the Hausdorff distance is calculated between $G_i$ and the nearest (in Hausdorff distance) segmented object to $G_i$ in that image. The same rule is applied for segmented objects.
3.3.2 Implementation Details

In consideration of the memory usage and the model effectiveness, we set the number of dense layers in each block as 6. The compression ratio of layers between dense blocks is 0.5. Besides, a small dropout rate (0.1 in our experiments) is used in each convolution layer to avoid overfitting. During training, we randomly crop images of 208 × 208 pixels with the batch size of 8 as input. The FullNet is trained using Adam optimizer. The weight for variance term in the loss function is set to $\alpha = 0.5$ for both datasets. The learning rate and training epoch are 0.001 and 300 for nuclei segmentation, and 0.0005 and 1000 for gland segmentation. Since both datasets are not large enough, we perform data augmentations like random scale, flip, rotation, affine and elastic transformation. Test time augmentations are utilized for more accurate predictions.

3.3.3 Results and Discussion

The qualitative results on the nuclei segmentation and gland segmentation are shown in Table 3.1 and Table 3.2. Typical qualitative results are shown in Fig. 3.4 and Fig. 3.5. FullNet-B is the bottleneck version with a growth rate $k = 48$. The FCN-pooling is the same structure as FullNet-B except that there are pooling layers (2×) after the first four dense blocks and up-sampling layers (4×) before the last two blocks, and no dilated convolution. It is used to illustrate the effect of full resolution feature maps in the FullNet.

For nuclei segmentation, we compare the FullNet with two methods on the Multi-Organ dataset. One is the patch-based CNN3 method [55], which uses a small patch to predict the central pixel’s class and utilizes region growing as post-processing to refine the segmentation results. The other one is the U-net, the reference structure of our FullNet. From the results in Table 3.1, it is obvious that FullNet outperforms CNN3 on both test sets. Compared to U-net, the FullNet also achieves better performance on both test sets. With the varCE loss, the segmentation accuracies are improved for nearly all metrics.

For gland segmentation, we compare our method with state-of-the-arts [18, 33, 111,
Table 3.1: Nuclei segmentation results on MO dataset using different methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Same organ test set</th>
<th>Different organ test set</th>
<th>All test set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F1  Dice  H  AJI</td>
<td>F1  Dice  H  AJI</td>
<td>F1  Dice  H  AJI</td>
</tr>
<tr>
<td>CNN3 [55]</td>
<td>0.8222 0.7301 7.39 0.5154</td>
<td>0.8327 0.8051 8.03 0.4989</td>
<td>0.8267 0.7622 7.66 0.5083</td>
</tr>
<tr>
<td>U-net [84]</td>
<td>0.8510 0.7962 6.89 0.5815</td>
<td>0.8401 0.7732 10.57 0.5481</td>
<td>0.8463 0.7863 8.47 0.5672</td>
</tr>
<tr>
<td>FCN-pooling</td>
<td>0.8569 0.7032 8.80 0.4835</td>
<td>0.8311 0.7022 10.09 0.4889</td>
<td>0.8458 0.7028 9.35 0.4858</td>
</tr>
<tr>
<td>FullNet-B w.o. varCE</td>
<td><strong>0.8725</strong> 0.8043 6.44 0.5944</td>
<td><strong>0.8720</strong> 0.8177 7.37 0.6270</td>
<td><strong>0.8723</strong> 0.8100 6.84 0.6084</td>
</tr>
<tr>
<td>FullNet-B w/ varCE</td>
<td>0.8718 <strong>0.8080</strong> 6.30 <strong>0.6043</strong></td>
<td>0.8566 <strong>0.8192</strong> 7.35 <strong>0.6285</strong></td>
<td>0.8653 <strong>0.8128</strong> 6.75 <strong>0.6147</strong></td>
</tr>
</tbody>
</table>

Table 3.2: Gland segmentation results on GlaS dataset using different methods. Rank of each entry is listed for clear comparison.

<table>
<thead>
<tr>
<th>Method</th>
<th>testA</th>
<th>testB</th>
<th>rank sum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F1  rank</td>
<td>Dice_{obj} rank</td>
<td>H_{obj} rank</td>
</tr>
<tr>
<td>CUMedVision [18]</td>
<td>0.912 5</td>
<td>0.897 7</td>
<td>45.42 6</td>
</tr>
<tr>
<td>Xu et al.(a) [111]</td>
<td>0.858 9</td>
<td>0.888 8</td>
<td>54.20 8</td>
</tr>
<tr>
<td>Xu et al.(b) [112]</td>
<td>0.893 8</td>
<td>0.908 3</td>
<td>44.13 4</td>
</tr>
<tr>
<td>Yan et al. [114]</td>
<td>0.924 2</td>
<td>0.902 6</td>
<td>49.88 7</td>
</tr>
<tr>
<td>MILD-Net [33]</td>
<td>0.914 4</td>
<td>0.913 2</td>
<td>41.54 3</td>
</tr>
<tr>
<td>Yang et al. [115]</td>
<td>0.921 3</td>
<td>0.904 4</td>
<td>44.74 5</td>
</tr>
<tr>
<td>FCN-pooling</td>
<td>0.896 7</td>
<td>0.875 9</td>
<td>55.54 9</td>
</tr>
<tr>
<td>FullNet-B w.o. varCE</td>
<td>0.906 6</td>
<td>0.904 4</td>
<td><strong>37.49</strong> 1</td>
</tr>
<tr>
<td>FullNet-B w/ varCE</td>
<td><strong>0.931</strong> 1</td>
<td><strong>0.922</strong> 1</td>
<td>41.27 2</td>
</tr>
</tbody>
</table>


In the results of Table 3.2, the proposed FullNet with varCE loss obtains the best performance on four columns, and has very similar performance on the other two. Our method can separate close/touching glands very well, which is beneficial for the improvements on the object-level metrics. The rank sum of all evaluation metrics illustrates that our approach achieves the best overall performance.

The effect of full resolution feature maps. For both tasks, the FullNet has much better performance than FCN-pooling, because FCN-pooling cannot produce accurate shapes and edges while the full resolution maps in FullNet are able to, as shown in Fig. 3.4 and Fig. 3.5. The accurate localization of edges is beneficial for separating touching nuclei and glands.

The effect of varCE loss. Both quantitative and qualitative results have proven that
the variance term in the loss function can enhance the segmentation performance. We take an example in Fig. 3.6 to illustrate why it works. The gland in the left part of the image is hard for the network to recognize due to its large lumen and incomplete contour. Without the variance term, the pixels in the lumen part have pretty low probabilities, thus are classified as background pixels. With the variance term, it correlates the pixels belonging to the lumen part to pixels of the boundary part by minimizing the variance of probabilities inside the gland. As a result, the pixels’ probabilities in the lumen part increase and most pixels of the gland are classified correctly.

3.3.4 Ablation Study

To explore the importance of the layer type and the growth rate $k$ in FullNet and the value of $\alpha$ for the varCE loss, we perform ablation studies using the nuclei segmentation MO dataset.

**Basic VS. bottleneck layers.** The results of FullNet with basic layer and bottleneck layer are shown in Table 3.3. For the same growth rate, bottleneck layer has nearly
Figure 3.6: The effect of varCE loss. (a) input image, (d) ground-truth instance label, (b) and (e) probability map and segmentation result without varCE loss, (c) and (f) probability map and segmentation result with varCE loss.

Table 3.3: Nuclei segmentation results on MO dataset using FullNet with different layer types (FullNet-B for bottleneck layers) and growth rates ($k$). Results are reported on the whole test set.

<table>
<thead>
<tr>
<th>Method</th>
<th>Params</th>
<th>F1</th>
<th>Dice</th>
<th>H</th>
<th>AJI</th>
</tr>
</thead>
<tbody>
<tr>
<td>FullNet ($k = 12$)</td>
<td>0.465 M</td>
<td>0.8438</td>
<td>0.7919</td>
<td>7.31</td>
<td>0.5793</td>
</tr>
<tr>
<td>FullNet ($k = 24$)</td>
<td>1.780 M</td>
<td>0.8684</td>
<td>0.7983</td>
<td>7.65</td>
<td>0.5862</td>
</tr>
<tr>
<td>FullNet ($k = 48$)</td>
<td>6.959 M</td>
<td>0.8632</td>
<td>0.8086</td>
<td>6.94</td>
<td>0.6040</td>
</tr>
<tr>
<td>FullNet-B ($k = 12$)</td>
<td>0.464 M</td>
<td>0.8629</td>
<td>0.8018</td>
<td>7.44</td>
<td>0.5925</td>
</tr>
<tr>
<td>FullNet-B ($k = 24$)</td>
<td>1.803 M</td>
<td>0.8706</td>
<td>0.8056</td>
<td>7.01</td>
<td>0.6095</td>
</tr>
<tr>
<td>FullNet-B ($k = 48$)</td>
<td>7.103 M</td>
<td>0.8723</td>
<td>0.8100</td>
<td>6.84</td>
<td>0.6084</td>
</tr>
</tbody>
</table>

the same number of parameters as basic layer, but outperforms basic layer in almost all four metrics, which indicates that bottleneck layer is more effective than basic layer with regard to the performance. However, bottleneck layer does not reduce the computational cost as expected in our experiments. The reason is that the number of dense layers in each block is 6, which is relatively small to reveal the computational efficiency of bottleneck layer. In our experiments, we use bottleneck layer because of its better performance.

**Growth rate $k$.** The results with different growth rates are also presented in Table 3.3. For both basic and bottleneck layers, there is a general trend that FullNets
Table 3.4: Nuclei segmentation results on MO dataset using different weights ($\alpha$) of varCE loss. The network is FullNet-B with $k = 48$. Results are reported on the whole test set.

<table>
<thead>
<tr>
<th>Method</th>
<th>F1</th>
<th>Dice</th>
<th>H</th>
<th>AJI</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha = 0$</td>
<td>0.8723</td>
<td>0.8100</td>
<td>6.84</td>
<td>0.6084</td>
</tr>
<tr>
<td>$\alpha = 0.05$</td>
<td>0.8704</td>
<td>0.8075</td>
<td>6.92</td>
<td>0.6114</td>
</tr>
<tr>
<td>$\alpha = 0.1$</td>
<td>0.8715</td>
<td>0.8086</td>
<td>6.84</td>
<td>0.6139</td>
</tr>
<tr>
<td>$\alpha = 0.5$</td>
<td>0.8653</td>
<td>0.8128</td>
<td>6.75</td>
<td>0.6147</td>
</tr>
<tr>
<td>$\alpha = 1$</td>
<td>0.8618</td>
<td>0.8122</td>
<td>6.86</td>
<td>0.6082</td>
</tr>
<tr>
<td>$\alpha = 5$</td>
<td>0.8312</td>
<td>0.8038</td>
<td>7.48</td>
<td>0.5787</td>
</tr>
</tbody>
</table>

perform better as $k$ increase, as the model capacity grows accordingly. The number of parameters also increases when $k$ becomes larger, but is still much smaller compared with the reference model U-net (about 31 million parameters). It attributes to the parameter-efficient dense block structure.

The weight $\alpha$ in the varCE loss. The nuclei segmentation results using different weights $\alpha$ in the varCE loss are shown in Table 3.4. FullNet-B with $k = 48$ is used for all $\alpha$ values. It can be observed that the overall performance increases as $\alpha$ becomes larger and then decrease. The best performance is achieved around $\alpha = 0.5$. When $\alpha$ is too large (e.g., $\alpha = 5$), the variance term in Eqn. (3.5) dominates the loss function and the cross entropy term cannot pull some instances to correct classes.

3.3.5 Computational Cost

It takes 0.140 seconds for a forward pass of our proposed FullNet-B ($k = 48$) on images of size $256 \times 256$ on an NVIDIA Quadro RTX6000 GPU. For comparison, the time of the FCN-pooling and U-net is 0.039 seconds and 0.008 seconds, respectively. The measurements were averaged over 100 individual forward passes. The computational cost of FullNets is much higher than the encoder-decoder-based networks (e.g., FCN-pooling and U-net), because the full resolution feature maps require a large amount of computation. This is also revealed in the training process. For example, it takes about 55GB GPU memory to train the FullNet-B ($k = 48$) with a batch size of 8 in the nuclei segmentation experiments. This limitation of high computational cost impedes the applications of FullNets in time sensitive or memory sensitive tasks. Besides, due
Figure 3.7: Nuclei segmentation using trained FullNet-B \((k = 48)\) in four \(500 \times 500\) subimages of a same TCGA bladder cancer tissue slide image. The number and average size (pixel number) of segmented nuclei in each image are marked under the segmentation results.

to this limitation we cannot test the performance of deeper FullNets with more dense blocks or a larger growth rate as in DenseNet, especially when the size of the input image is very large. This problem may be alleviated with the rapid development of GPU technology in the near future.

### 3.4 Application of FullNet on Cancer Imaging Data

We apply the proposed FullNet-B to analyze H&E slides for TCGA bladder cancer cohort [38]. As an example, we select four subimages of size \(500 \times 500\) from one whole slide image of bladder tumor, shown in the first row of Fig. 3.7. The leftmost subimage comes from a tumor-adjacent normal region, and the other three images come from three different tumor regions. Nuclei segmentation is performed using the trained FullNet-B \((k = 48)\) model with the varCE loss. The segmentation results are good, as shown in the second row of Fig. 3.7. The computational nuclei segmentation results of 15 representative \(500 \times 500\) image patches were evaluated by an expert pathologist, and the average segmentation accuracy measured is about 92%.
After segmentation, nuclei features like nuclei number and average size (pixel number) are calculated based on the segmentation results, marked under each column in Fig. 3.7. We observe that the tumor regions, on average have higher nuclei-density compared to normal regions, and there is high variation in nuclei number and size distribution among different regions in the tumor, which may be indicative of intra-tumor phenotypic heterogeneity.

3.5 Summary

In this chapter, we propose a full resolution convolutional network (FullNet) and a variance constrained cross entropy (varCE) loss for nuclei and gland segmentation in histopathology images. The FullNet removes all pooling layers in the fully convolutional network and takes advantages of densely connected layers and dilated convolutions to resolve memory and small receptive field problems. The varCE loss places a spatial constraint on pixels that assists the network to learn the shape of objects. The proposed framework achieves state-of-the-art segmentation results of both nuclei and glands in H&E stained histopathology images. In future work, we consider to explore the performance of deeper FullNets, and the possibility of reducing the computational cost while retaining the performance.
Chapter 4

Gene Mutation and Pathway Activity Prediction from Histopathology Images in Breast Cancer

4.1 Introduction

Breast carcinoma is the most common cancer among women worldwide that consists of a heterogeneous group of diseases with different histological, prognostic and clinical outcomes [14]. Approximately 50% of all women diagnosed with breast cancer can develop metastatic diseases such as liver and lung cancers [8]. In the past decades, substantial efforts have been made to deepen our understanding of breast cancer risk factors, molecular pathogenesis, and treatment development. Especially, high-throughput molecular profiling reveals that multiple genetic mutations and biological signaling pathways could have a great influence on tumor progression and overall survival [29].

Comprehensive genomic analysis has identified key driver genetic mutations that are responsible for therapeutic implication and outcome prediction of breast cancer. The tumor suppressor gene TP53 is found altered in breast carcinoma in approximated 30% of all cases with prognostic implication [12]. Overexpression of ERBB2 is also an adverse prognostic indicator correlated with decreased survival in breast cancer [10]. Given certain types of mutations, targeted therapies for patient subgroups have been developed. For example, the PI3K inhibitor is designed to be responsive for patients with the PIK3CA mutation, which is a key driver gene associated with oncogenesis and hyperactivity of the PI3K pathway. The identification of driver mutations is essential for targeted therapy and clinical diagnosis of breast malignancies.

However, comprehensive genomic analysis is difficult to implement at scale, but H&E stained whole slide images are ubiquitously available. As we have shown in previous chapters, WSIs can offer a computationally effective and efficient means to
quantitatively characterize cell-level heterogeneity of cancer specimens. Pathologists routinely use WSIs to identify nuclei features, diagnose cancer status and measure the histopathological grade of cancer tissues. However, it remains unclear for many cancers that how the genetic features (e.g., gene mutation, biological pathway activity) affect cancer and whether there are direct relationships between genetic features and image features. By uncovering correlations between image and genetic features, it can promote the understanding of some biologic mechanisms and pathways of gene expression and find more biomarkers that are predictive to clinical outcomes, which are beneficial for cancer treatment. Specifically, there is a lack of research linking WSI with gene mutations for advancing clinical assessment in breast cancer. Preliminary evidence suggests that it is possible to apply deep-learning approaches to automatically predict cancer subtypes in multiple cancers [24, 40, 26], to predict mutations in lung [24] and liver cancers [20], to classify mesothelioma [25], and to predict pan-cancer prognosis for patients [16]. Besides, DNA methylation patterns can also be predicted from whole slide images [128]. However, developing deep-learning classifiers for mutation(s) prediction and the associated biological pathway activities from whole slide images remains to be elucidated in breast cancer.

In this chapter, we develop WSI-based deep learning classifiers for predicting key mutation outcomes and important biological pathway activities in breast cancer. We collected 659 patients with breast invasive carcinoma from The Cancer Genome Atlas (TCGA) [38]. Each patient contains an H&E stained histopathology whole slide image, mutation data with point mutation status of 18 genes and copy number alteration (CNA) of 35 genes, and omics data with the mRNA expression data and CNA data of all genes. We formulate the prediction tasks as binary classification problems and train attention-based models to predict the mutation status and pathway activity. Since there are not direct pathway activity labels, we derive the labels from mRNA expression data or CNA data for each patient. Our model can predict the point mutations of six important genes (AUC 0.68 ~ 0.85) and copy number alteration of another six genes (AUC 0.69 ~ 0.79). Additionally, the trained models can predict the activities of three out of ten canonical pathways (AUC 0.65 ~ 0.79). The attention technique in the
model allows us to visualize the weight maps of tumor tiles in WSI to understand the
decision-making process of deep learning models. We also highlight WSI visual interac-
tions between mutation and its related pathway, enabling a head-to-head comparison to
reinforce our major findings. Furthermore, we validate our analysis in a pan-cancer set-
ting on liver and lung cancer cohorts to gain additional insights of mutation prediction
across cancers based on the metastatic associations derived from breast cancer. Our
results provide new insights into the association between pathological image features,
molecular outcomes and targeted therapies for breast cancer patients.

4.2 Methods

The workflow of our method is shown in Fig. 4.1. In this section, we first describe
the data processing steps, including the patients selection from the original data co-
hort of TCGA, the processing of histopathology images and the generation of pathway
activity labels. Secondly, the model structure and training details are presented. The
visualization of weight maps is introduced finally.

4.2.1 Data Processing

Patient selection. The original TCGA-BRCA cohort consists of 1,098 patients with
H&E stained whole slide images, genomic data and additional clinical information.
We analyzed the 1,133 Formalin-Fixed Paraffin-Embedded (FFPE) slides that were
generated by fixing a specimen in formaldehyde and then embedding it in a paraffin
wax block for cutting. We further filtered out low quality FFPE slides according to
the following criteria: (1) There is no diagnostic time information in a slide with low
visual quality. (2) A slide has extensive blurred areas or is abnormally stained with little
informative tissue areas. After slide selection preprocessing, we collected 659 slides (659
patients) along with the corresponding omics data (mRNA expression and copy number
alteration). The 659 cases in the TCGA-BRCA dataset were randomly partitioned into
training, validation and test sets based on 70%, 15%, and 15% ratios respectively.
The characteristics of the breast cancer patients in each set are shown in Table 4.1.
Figure 4.1: Illustration of the deep learning workflow for data processing and model evaluation. We processed the WSI data by extracting tiles (1a), identifying tumor titles (1b), and generating small non-overlapping tiles with color normalization. We selected key mutational genes (2) and identified biological pathways from mRNA (or CNA) expressions (3). Model training was based on a pretrained ResNet-101 model with an attention mechanism. After model selection, the trained model was used to test tiles and assess their prediction performances.

In addition, we collected 350 patients with lung adenocarcinoma from TCGA-LUAD cohort and 316 patients with liver hepatocellular carcinoma from TCGA-LIHC cohort to validate our method following the same selection process. Each dataset is split into training and testing sets with 20% and 80% ratios, where the training set is utilized to fine-tune the developed models trained from the breast cancer data. The patients’ characteristics of the lung and liver datasets are shown in Table 4.2 and Table 4.3, respectively.

**Histopathology data pre-processing.** For each slide, we extracted non-overlapping tiles of $512 \times 512$ at $20\times$ magnification and removed background tiles (Fig. 4.1(a)). A
Table 4.1: Patient characteristics on the 659 cases from TCGA-BRCA cohort.

<table>
<thead>
<tr>
<th>Dataset (n)</th>
<th>Train (462)</th>
<th>Val (99)</th>
<th>Test (99)</th>
<th>Overall (659)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age (year)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>56.6</td>
<td>57.6</td>
<td>57.4</td>
<td>56.9</td>
</tr>
<tr>
<td>Range</td>
<td>27-90</td>
<td>26-90</td>
<td>34-90</td>
<td>26-90</td>
</tr>
<tr>
<td>Sex, n (%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Male</td>
<td>7 (1.5)</td>
<td>3 (3.0)</td>
<td>0 (0.0)</td>
<td>10 (1.5)</td>
</tr>
<tr>
<td>Female</td>
<td>454 (98.5)</td>
<td>96 (97.0)</td>
<td>99 (100.0)</td>
<td>649 (98.5)</td>
</tr>
<tr>
<td>Stages, n (%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I/IA/IB</td>
<td>78 (16.9)</td>
<td>14 (14.1)</td>
<td>18 (18.2)</td>
<td>110 (16.7)</td>
</tr>
<tr>
<td>II/IIA/IIB</td>
<td>273 (59.2)</td>
<td>56 (56.6)</td>
<td>56 (56.6)</td>
<td>385 (58.4)</td>
</tr>
<tr>
<td>III/IIIA/IIB/IIIC</td>
<td>95 (20.6)</td>
<td>25 (25.3)</td>
<td>24 (24.2)</td>
<td>144 (21.9)</td>
</tr>
<tr>
<td>IV</td>
<td>8 (1.7)</td>
<td>2 (2.0)</td>
<td>0 (0.0)</td>
<td>10 (1.5)</td>
</tr>
<tr>
<td>X</td>
<td>3 (0.7)</td>
<td>2 (2.0)</td>
<td>1 (1.0)</td>
<td>6 (0.9)</td>
</tr>
<tr>
<td>N/A</td>
<td>4 (0.9)</td>
<td>0 (0.0)</td>
<td>0 (0.0)</td>
<td>4 (0.6)</td>
</tr>
<tr>
<td>Subtypes, n (%)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Luminal A</td>
<td>186 (40.3)</td>
<td>47 (47.5)</td>
<td>44 (44.4)</td>
<td>277 (42.0)</td>
</tr>
<tr>
<td>Luminal B</td>
<td>105 (22.8)</td>
<td>17 (17.2)</td>
<td>20 (20.2)</td>
<td>142 (21.5)</td>
</tr>
<tr>
<td>Her2</td>
<td>34 (7.4)</td>
<td>6 (6.1)</td>
<td>5 (5.1)</td>
<td>45 (6.8)</td>
</tr>
<tr>
<td>Basal</td>
<td>91 (19.7)</td>
<td>13 (13.1)</td>
<td>20 (20.2)</td>
<td>124 (18.8)</td>
</tr>
<tr>
<td>Normal</td>
<td>14 (3.0)</td>
<td>5 (5.1)</td>
<td>3 (3.0)</td>
<td>22 (3.3)</td>
</tr>
<tr>
<td>N/A</td>
<td>31 (6.7)</td>
<td>11 (11.1)</td>
<td>7 (7.1)</td>
<td>49 (7.4)</td>
</tr>
</tbody>
</table>

background tile was determined if its mean pixel value is higher than 220. We focus on tumor areas in the whole slide images therefore we adopted a semi-automatic labeling method to identify tumor tiles. The labeling process was implemented by the initial clustering and manual refinement. In the first step, k-means clustering was performed on all tiles for each slide. Specifically, each $512 \times 512$ tile was downsampled to $128 \times 128$, which was then flattened into a 49152-length feature vector. These feature vectors were then clustered into two groups (i.e., tumor and non-tumor regions). In the second step, a pathologist (20 years of clinical experience) additionally verified the segmentation quality of the slides and revised inaccurate results of slides to ensure the tumor labeling results were reasonable. For example, the tiles with marks in the annotated slides were remove manually if they were left after the clustering step. We then performed color normalization using the method [82] to eliminate the color variations in different slides. The tiles of the same slide were processed by using the same slide-level pixel mean and standard deviation during the normalization. After WSI tile extraction and tumor tile
Table 4.2: Patient characteristics on the 350 cases from TCGA-LUAD cohort.

<table>
<thead>
<tr>
<th>Dataset (n)</th>
<th>Train (70)</th>
<th>Test (280)</th>
<th>Overall (350)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Age (year)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>64.6</td>
<td>64.9</td>
<td>64.9</td>
</tr>
<tr>
<td>Range</td>
<td>40-85</td>
<td>38-88</td>
<td>38-88</td>
</tr>
<tr>
<td><strong>Sex, n (%)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Male</td>
<td>37 (52.9)</td>
<td>119 (42.5)</td>
<td>156 (44.6)</td>
</tr>
<tr>
<td>Female</td>
<td>33 (47.1)</td>
<td>161 (57.5)</td>
<td>194 (55.4)</td>
</tr>
<tr>
<td><strong>Stages, n (%)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I/IA/IB</td>
<td>40 (57.1)</td>
<td>154 (55.0)</td>
<td>194 (55.4)</td>
</tr>
<tr>
<td>II/IIA/IIB</td>
<td>14 (20.0)</td>
<td>76 (27.1)</td>
<td>90 (25.7)</td>
</tr>
<tr>
<td>III/IIIA/IIB/IIB/IIIC</td>
<td>8 (11.4)</td>
<td>37 (13.2)</td>
<td>45 (12.9)</td>
</tr>
<tr>
<td>IV</td>
<td>7 (10.0)</td>
<td>12 (4.3)</td>
<td>19 (5.4)</td>
</tr>
<tr>
<td>X</td>
<td>0 (0.0)</td>
<td>0 (0.0)</td>
<td>0 (0.0)</td>
</tr>
<tr>
<td>N/A</td>
<td>1 (1.4)</td>
<td>1 (0.4)</td>
<td>2 (0.6)</td>
</tr>
</tbody>
</table>

selection, there are 703,804, 140,981 and 167,530 tiles used in the training, validation and testing sets of breast cancer, 130,659 and 465,925 tiles in the training and testing sets of lung cancer, and 116,635 and 570,073 tiles in the training and testing sets of liver cancer.

**Mutated genes and pathway activity identification.** To ensure a sufficient amount of training WSIs for mutated genes, for point mutation we selected 18 important genes in breast cancer which were mutated at least 3% in the 659 patients, and for CNA we selected 35 genes with mutation percentage greater than 5%. In the validation tasks of lung and liver cancers, we used the same criterion to select gene profiles, resulting in 9 point mutation genes and 14 CNA genes in the lung cancer and 7 point mutation genes and 25 CNA genes in the liver cancer for analysis.

In the pathway activity prediction task, we identified ten canonical signaling pathways with frequent genetic alterations. The pathway activity in each patient was obtained by a weighted sum of the genes’ expression data or CNA data in the pathway. Then the activity was binarized as activated if it is greater than zero and inactivated otherwise. For each pathway, we generated two types of activity labels from mRNA
Table 4.3: Patient characteristics on the 316 cases from TCGA-LIHC cohort.

<table>
<thead>
<tr>
<th>Dataset (n)</th>
<th>Train (63)</th>
<th>Test (253)</th>
<th>Overall (316)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Age (year)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>59.7</td>
<td>59.3</td>
<td>59.4</td>
</tr>
<tr>
<td>Range</td>
<td>17-84</td>
<td>16-90</td>
<td>16-90</td>
</tr>
<tr>
<td><strong>Sex, n (%)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Male</td>
<td>38 (60.3)</td>
<td>174 (68.8)</td>
<td>212 (67.1)</td>
</tr>
<tr>
<td>Female</td>
<td>25 (39.7)</td>
<td>79 (31.2)</td>
<td>104 (32.9)</td>
</tr>
<tr>
<td><strong>Stages, n (%)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I/IA/IB</td>
<td>30 (47.6)</td>
<td>120 (47.4)</td>
<td>150 (47.5)</td>
</tr>
<tr>
<td>II/IIA/IIB</td>
<td>12 (19.0)</td>
<td>63 (24.9)</td>
<td>75 (23.7)</td>
</tr>
<tr>
<td>III/IIIA/IIIB/IIIC</td>
<td>13 (20.6)</td>
<td>54 (21.3)</td>
<td>67 (21.2)</td>
</tr>
<tr>
<td>IV</td>
<td>0 (0.0)</td>
<td>3 (1.2)</td>
<td>3 (0.9)</td>
</tr>
<tr>
<td>X</td>
<td>0 (0.0)</td>
<td>0 (0.0)</td>
<td>0 (0.0)</td>
</tr>
<tr>
<td>N/A</td>
<td>8 (12.7)</td>
<td>11 (4.3)</td>
<td>19 (6.0)</td>
</tr>
</tbody>
</table>

expression data or CNA data for each patient as follow:

\[ v^{s,i} = \frac{1}{N_{\text{gene}}^i} \sum_{n=1}^{N_{\text{gene}}^i} w_{n}^{s,i} u_{n}^{s,i}, \]

\[ l^{s,i} = \begin{cases} 1, & \text{if } v^{s,i} > 0, \\ 0, & \text{otherwise}. \end{cases} \]  \hspace{1cm} (4.1)

where \( i = 1, 2, \ldots, 10, \ s = 1, 2, \ldots, 659, \ v^{s,i} \) is the activity level of pathway \( i \) in patient \( s, \ l^{s,i} \) is the binary activity label of pathway \( i \) in patient \( s, \ N_{\text{gene}}^i \) is the number of important genes that are involved in the pathway \( i \) according to Sanchez-Vega et al’s work [86], \( u_{n}^{s,i} \) is the expression level or CNA level of gene \( n \) in pathway \( i \) and patient \( s, \ w_{n}^{s,i} \) is the corresponding weight, which takes value 1 if the gene is an oncogene and -1 if it is a tumor suppressor. The CNN aims to predict the binary label \( l^{s,i} \), i.e., whether a pathway is activated \( (l^{s,i} = 1) \) or inactivated \( (l^{s,i} = 0) \) in a patient.

4.2.2 Model Structure

Our model architecture is shown in Fig. 4.2. It consists of two main modules: feature extractor and multi-layer perceptron (MLP) predictor with self-attention. The feature extractor aims to obtain a feature vector representing the input tile. We use the convolutional layers of ResNet-101 [36] as the feature extractor, which is widely
used in image classification tasks and has shown powerful feature representation ability in various applications. This sub-network is pre-trained on the ImageNet dataset [85] and kept unchanged during training and testing. Through the feature extraction, each input tile is represented by a 2,048-dimensional feature vector, resulting in a feature matrix of $N \times 2048$ for slide of patient $s$, where $N$ is the number of tumor tiles in the slide and varies from slide to slide.

The MLP predictor sub-network follows the feature extractor to output the final prediction. It consists of three fully connected layers and one self-attention layer (Fig. 4.2). The first two fully connected layers have 512 and 128 neurons, respectively, reducing the size of feature matrix to $N \times 128$. The self-attention layer is used to compute the importance weight of each tile’s feature vector and guide the network to pay more attention to the crucial tiles. Self-attention has been used successfully in a variety of tasks in natural language processing [61, 21, 101] and computer vision [122] to model relationships between widely separated spatial regions. In this paper, we make slight modifications based on the method in Zhang et al. [122]:

\[
\begin{align*}
  f(x) &= W_f x, \quad g(x) = W_g x \\
  \alpha_{i,j} &= \text{softmax} \left( f(x_i)^T g(x_j) \right) \\
  o_j &= \sum_{i=1}^{N} \alpha_{j,i} x_i, \quad y = x + \gamma o
\end{align*}
\]

where $x$ is the input feature matrix, $W_f$ and $W_g$ are $1 \times 1$ convolution filters, $\alpha_{j,i}$ indicates how much attention the model pays to the $i$-th tiles features when computing the $j$-th
tiles activation $o_j$, $\gamma$ is a trainable parameter controlling the scale of the attention. $y$ is the output of the self-attention layer after an average pooling, which is the global feature vector representing all tumor tiles of a slide. The final fully connected layer transforms the global feature to a prediction.

Because the gene mutation status prediction and pathway activity prediction are formulated as classification tasks, the cross entropy loss is used to train the models.

4.2.3 Model Training and Finetuning

Model training and evaluation

The feature extractor sub-network (ResNet-101 without fully-connected (FC) layer) is pre-trained and fixed during training for all prediction tasks. Therefore, we extract the feature vectors of tumor tiles for all patients beforehand and save them to the disk. Training the prediction module from the saved feature vectors can greatly accelerate the training speed. During feature extraction, each $512 \times 512$ tile is resized to $224 \times 224$ image and normalized by the mean and standard deviation of the ImageNet dataset [85] before feeding to the pretrained ResNet-101. The prediction sub-network is trained with the Adam optimizer for 30 epochs. The initial value of $\gamma$ in the self attention layer is 1. The learning rate of $\gamma$ is set to 0.001 and all other parameters have a learning rate of 0.0001. The best model is saved when achieving the best performance on the validation set. For different tasks (e.g., point mutation, pathway activity), the models for breast cancer are all trained from scratch. It took approximately 6 minutes to train the MLP with self-attention (30 epochs, batch size 8) on a NVIDIA TITAN Xp GPU. Our training is efficient because our method can directly provide slide-level prediction instead of tile-level predictions as done in Fu et al. [30]

To evaluate the model’s performance on the validation set (for model selection) and test set, we use the area-under-the-curve (AUC) in both mutation prediction and pathway activity prediction tasks. The AUC is the area under the ROC curve, which is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. AUC informs the capability of a model in distinguishing
between classes. The 95% confidence interval (CI) of each AUC score is calculated by 1,000 bootstrapping to estimate the uncertainty of AUC.

Model fine-tuning on the lung and liver cancer data

During model fine-tuning, we fix the parameters of the first two fully connected (fc) layers of the prediction sub-network and fine-tune the self-attention layer and the last fc layer. We assume that image features learned from breast cancer data could be also useful for in a pan-cancer setting. This fine-tuning strategy could help to investigate if there is any underlying relationship between the data of breast cancer and lung or liver cancer. We did not fine-tune models for all possible gene profiles from TCGA, because the point mutation and CNA percentage of some genes are extremely low in lung or liver cancers. Only genes with >3% point mutation or >5% CNA were fine-tuned and tested in our study.

4.2.4 Visualization

The self-attention layer in our model can produce the importance weights of tiles in a slide in the prediction tasks, which is helpful for us to explore the biological interpretation value of deep-learning classifiers. We compute the log value of the weight of each tile and project it to the original location in the whole slide image, resulting in the weight map (Fig. 4). The weight $\beta_i$ is computed according to Eqn. (4.2), (4.3), (4.4):

$$\beta_i = 1 + \gamma \sum_{j=1}^{N} \alpha_{i,j} \quad (4.5)$$

Besides, we select tiles with top twenty largest weights in a slide to show the appearance of important tiles.

4.3 Experiments

4.3.1 Prediction of Gene Mutation Status

We trained our models on pathology images to predict significant mutations profiles in breast cancer. More specifically, our deep learning model extracted mutation-specific
Table 4.4: AUC (with 95% CI) achieved by the models trained on the point mutation data of breast cancer. The top six results are reported out of 18 genes.

<table>
<thead>
<tr>
<th>Gene</th>
<th>RB1</th>
<th>CDH1</th>
<th>NF1</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUC (95% CI)</td>
<td>0.852 (0.740~0.969)</td>
<td>0.776 (0.625~0.914)</td>
<td>0.768 (0.449~0.949)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Gene</th>
<th>NOTCH2</th>
<th>TP53</th>
<th>MAP3K1</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUC (95% CI)</td>
<td>0.740 (0.515~0.917)</td>
<td>0.729 (0.621~0.828)</td>
<td>0.682 (0.419~0.949)</td>
</tr>
</tbody>
</table>

Table 4.5: AUC (with 95% CI) achieved by the models trained on the copy number alteration data of breast cancer. The top six results are reported out of 35 genes.

<table>
<thead>
<tr>
<th>Gene</th>
<th>FGFR1</th>
<th>EIF4EBP1</th>
<th>KAT6A</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUC (95% CI)</td>
<td>0.794 (0.677~0.894)</td>
<td>0.742 (0.595~0.871)</td>
<td>0.732 (0.523~0.941)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Gene</th>
<th>HEY1</th>
<th>ZNF217</th>
<th>RAB25</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUC (95% CI)</td>
<td>0.715 (0.510~0.894)</td>
<td>0.693 (0.498~0.870)</td>
<td>0.686 (0.528~0.826)</td>
</tr>
</tbody>
</table>

feature vectors from tumor tiles and predicted the gene mutation probability of the corresponding patient (see Fig. 4.2). We seek to predict two types of gene mutations including point mutation and CNA. Our model demonstrated high-level performance on predicting the point mutation status of multiple important genes. The AUC scores for top results of point mutation and CNA are shown in Table 4.4 and Table 4.5, respectively, and the corresponding ROC curves are shown in Fig. 4.3. For example, we found that our model is highly predictive on TP53 (AUC = 0.729), which is the most frequently mutated gene in breast cancer with prognostic implication. Our models also showed good results on predicting mutations of RB1 (AUC 0.852), CDH1 (AUC 0.776), NF1 (AUC 0.768), NOTCH2 (AUC 0.740) in breast cancer.

Additionally, we found that our deep learning classifier performed well (AUC>0.65) on predicting the CNA status in breast cancer, including six genes of FGFR1, EIF4EBP1, KAT6A, HEY1, ZNF217, and RAB25. Importantly, our model is explainable due to the implemented self-attention mechanism, where we were able to identify key tiles in the process of model prediction (Fig. 4.4). For example, we computed each tile’s weight that contributes to the final global feature vector and presented the weight map of a patient about TP53 in Fig. 4.4(b) and RB1 in Fig. 4.5(b). The corresponding top 20 weighted tiles are also shown in Fig. 4.4(d) and Fig. 4.5(d), respectively.
Table 4.6: AUC (with 95% CI) achieved by the models trained on the pathway activity from mRNA expression data (top row) and from CNA data (bottom row). The top three results are reported out of 10 canonical pathways

<table>
<thead>
<tr>
<th>Pathway</th>
<th>p53</th>
<th>PI3K</th>
<th>Cell cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>p53</td>
<td>0.798 (0.696~0.890)</td>
<td>0.666 (0.544~0.777)</td>
<td>0.654 (0.543~0.760)</td>
</tr>
<tr>
<td>Myc</td>
<td>0.795 (0.671~0.893)</td>
<td>0.668 (0.536~0.795)</td>
<td>0.640 (0.344~0.939)</td>
</tr>
</tbody>
</table>

4.3.2 Prediction of Biological Pathway Activities

We developed deep learning models to predict the activities of ten canonical biological pathways [86] identified in breast cancer for each patient. The pathway activity levels were derived from either mRNA expression data or the CNA data (see Section 4.2.1) to supervise the model training. The model structure and training method were kept the same as in the mutation prediction task.

When using the mRNA expression data to represent pathway activity, we found that the p53, PI3K and cell cycle pathways are predictable (AUC>0.65, Table 4.6). When using the CNA data for pathway activity, the Myc pathway achieves the highest AUC (0.795). The notch pathway (AUC 0.668) and p53 pathway (AUC 0.640) also have significant performance.
Figure 4.4: Weight maps of tiles when predicting the point mutation status of TP53 and p53 pathway activity from mRNA expression data in breast cancer. (a) Tumor tiles after data processing. (b) Weight map of tumor tiles in TP53 point mutation prediction. Brighter green tiles have larger weights. (c) Weight map of p53 pathway activity prediction. (d) and (e): Top 20 weighted tiles for TP53 point mutation prediction and p53 pathway prediction, respectively. We marked four tiles that appear in both tasks. These tissues contain poorly differentiated breast carcinoma with small nests, solid sheets and single cells from a pathologist’s perspective.

We also explored the visual interpretation of biological pathway activity and driver mutations as shown in histopathology images. To allow a joint analysis, we chose the biological pathway [86] that is associated with the available driver gene mutations. For example, the visualization of the weight map on the p53 pathway is shown in Fig. 4.4(c) and the top 20 weighted tiles are also offered in Fig. 4.4(e). Meanwhile, we used the same patient as the result in gene mutation prediction (Fig. 4.4), which shows the highlighted areas in TP53 mutation prediction (Fig. 4.4(b)) and the top 20 weighted tiles (Fig. 4.4(d)). We found that those tiles are highly correlated to
both TP53 mutation and p53 pathway activity. This finding increases the confidence in our prediction because TP53 is a key gene in the p53 pathway, and one would expect a relationship between the mutation status and pathway activity. Similarly, the same observation can be found between RB1 mutation prediction result and cell cycle pathway prediction result in Fig. 4.5. Overall, we have seen a shared similarity among highlighted tiles despite the complexity of biological pathway activities.

4.3.3 Validation on Lung and Liver Cancers

Next, we validated our modeling approach in two different cancers namely lung adenocarcinoma and hepatocellular carcinoma. In the lung adenocarcinoma (TCGA-LUAD) cohort, 9 genes for point mutation and 14 genes for CNA were used for model testing. Our fine-tuned models developed from the breast cancer cohort can predict the point mutation of TP53 (AUC 0.705) and Notch2 (AUC 0.656), the copy number alteration of FGFR1 (AUC 0.676), the p53 pathway activity (AUC 0.602) from mRNA expression data, and the activities of Myc pathway (AUC 0.658) and PI3K pathway (AUC 0.601) from CNA data. Overall, the responses on the pathway prediction are not as good as those on mutation prediction. Notably, the mutations of TP53 gene occur in about 50% of non-small cell lung cancer (NSCLC) and TP53 mutation is associated with worse prognosis with treatment resistance [66], therefore the prediction of TP53 mutation is also helpful for the diagnosis of lung cancer.

In the liver hepatocellular carcinoma (TCGA-LIHC) cohort, the numbers of tested genes are 7 and 25 for point mutation and CNA, respectively. Our fine-tuned models can predict the point mutation of RB1 (AUC 0.795), the copy number alteration of TGF2 (AUC 0.718), the pathway activity of cell cycle from mRNA expression data (AUC 0.614), and Myc pathway activity from CNA data (AUC 0.602). In particular, RB1 is a key inhibitor of cell cycle progression in HCC patients [4, 43, 88], and RB1 mutations are significantly associated with reduced cancer-specific and recurrence-free survival after resection in HCC patients [4, 43, 88]. Therefore, the prediction of RB1 mutation has potential prognosis value for those patients.
We also visualized the weight maps of TP53 mutation and p53 pathway of a representative patient in lung cancer in Fig. 4.6, and those of RB1 mutation and cell cycle pathway in the liver cancer in Fig. 4.7. In both examples, we can observe similar morphological patterns identified from a pathologist’s perspective in the two weight maps and tile appearances in the top 20 weighted tiles, which are similar to our observations for breast cancer.

4.3.4 Discussion

As cancers are caused by gene mutations, the prediction of key gene mutations based on whole slide images could promote the targeted treatment of cancer patients [12, 69, 100, 99, 92, 7, 13, 98]. For example, our models can predict TP53 point mutation (AUC 0.729) and FGFR1 copy number alteration (AUC 0.794) with high accuracies. TP53 is a tumor suppressor gene that plays a key role in many cellular pathways controlling cell proliferation, cell survival, and genomic integrity [69]. It is mutated frequently in breast cancer [12, 69] and has been associated with poor prognosis [12, 69, 100]. The FGFR1 gene is a member of the fibroblast growth factor receptor (FGFR) family that regulates important biological processes including cell proliferation and differentiation during development and tissue repair [99]. In breast cancer, FGFR1 amplification is the most frequent genomic aberration [92], and may lead to dysregulated FGF receptors and promote cancer growth and metastasis. Extensive works [92, 7, 13, 98] have shown that FGFR1 could be a therapy target in breast cancer (e.g., the anti-FGFR1 dovitinib (TKI1258) therapy [7]). With the prediction of TP53 mutation and FGFR1 alteration, our models offered novel insights into selecting patient subgroups for the targeted therapy from digitalized WSI scans.

We extended our study to analyze biological pathway prediction based on whole slide images that has seldom been addressed previously. Biological pathways are the interactions among molecules in a cell that result in certain products or changes in cancer [1]. Several important signaling pathways have been identified as frequently and genetically altered in cancer [86]. We showed that deep learning can predict pathology activity levels, providing valuable information for prognosis and therapeutic planning.
For example, the p53 pathway activity (predicted with 0.798 AUC in our method) is associated with more aggressive disease and worse overall survival in breast cancer [31]. The Myc pathway (predicted with 0.795 AUC) acts as a key regulator of cell growth and proliferation which has been linked to the basal-like breast cancer [109, 70], and can serve as a target for this aggressive subtype in breast cancer.

To overcome the interpretability challenges of AI-powered models, we employed a self-attention mechanism that is able to visualize the region of interest that contributed to outcomes prediction. In other words, we can display the weight map of each tumor tile to understand the decision making of the classifiers, highlighting the regions that contribute most to the final prediction. An example of the visualized weights map when predicting TP53 point mutation and p53 pathway activity is shown in Fig. 4.4(b) and Fig. 4.4(c), respectively. Tiles with brighter green colors have larger weights, indicating that those tiles are most important in the decision-making process. Interestingly, the highlighted regions in both tasks are approximately located in the same part of the whole slide image, and the top 20 weighted tiles in the two tasks shown in Fig. 4.4(d) and Fig. 4.4(e), are also similar. The possible reason could be that TP53 is a crucial gene in the p53 pathway thus the predictions depend on similar image features in this example. This type of methodology and visualization has the potential to enable the improved exploration of the relationship between the image morphological features and molecular outcomes, as well as the relationship between genes and biological pathways, which can lead to new discoveries in breast cancer development.

To validate our method, we further extended the trained deep learning models on lung and liver cancers with transfer learning. We hypothesize that the models trained using breast cancer data can also predict important gene mutations and pathway activities in lung and liver cancers since they are two common sites for the breast cancer metastatic spread [103]. Out of the well predicted genes in breast cancer, the point mutations of TP53 and Notch2, and the copy number alteration of FGFR1 can also be predicted in lung cancer (LUAD). In the liver cancer (LIHC), the well predicted genes are RB1 and TGFβ2. These genes are indeed highly related to the diagnosis of lung cancer [66, 17, 37] and liver cancer [4, 49]. The different results on liver and
lung cancers may be caused by the tissue differences of the two cancers. The pathway prediction results in the two cancers are not as good as gene mutation predictions, probably because pathways are more complicated than gene mutations thus are more challenging to predict. However, the well-predicted pathways in breast cancer still get the highest AUC scores in lung (p53) and liver (cell cycle, Myc) cohorts.

Overall, our study highlights deep characterization of breast cancer, its mutation outcome, and biological pathway activity. We present unique insights into WSI visual interactions between mutation and its pathway, enabling a head-to-head comparison to reinforce our major findings. Our approach can be a useful computational tool for gene mutation pre-screening, prior to the costly gene mutation analysis such as next-generation sequencing. Our evaluation strategy differs from pan-cancer studies [48, 30, 87] which evaluate performance on each cancer individually. We measured the performance across cancer types by training on breast cancer and validating on liver and lung cancers, which is more challenging due to inherent differences of cancer tissues [56]. In terms of model development, we directly provide slide-level predictions without assuming that each tile or super-tile shares the same label as the whole slide. The self-attention mechanism in our network further enables us to visualize the importance of tiles during the decision-making process, instead of the probabilities of mutations or expression for tiles. This finding can be used to better understand which image morphological features are related to certain gene mutations or pathway activities. Finally, our approach is a data-driven workflow that does not require nuclei detection [6] as a prerequisite for specific prediction tasks.

While building associations between histopathology and molecular profiles is promising, the identified genotype-phenotype relationships here are not intended to replace standard transcriptomic tests. Given the confirmation from our collaborative pathologist that there is a lack of consensus on molecularly defined patterns seen from histopathological scans, we expect our detectable findings could complement pathologists routine workflow. One limitation of the study is that workflow is based on
formalin-fixed, paraffin-embedded (FFPE) slides given their quality of preserving microscopic characteristics of tissues, while frozen tissues could also be considered for extended analysis. Our computational analysis has a dependence on the feature extractor pretrained on natural images (ImageNet dataset [85]). There is a domain gap between natural images and pathology images. Therefore, the exploration of appropriate feature representations of pathology tiles and their parameters will be crucial to assess the validity and reproducibility of algorithms. To maximize the power of deep learning approaches, it is also necessary to address data scarcity in histopathology-related tasks. There are often a significant portion of data samples that are insufficient and under-represented for certain mutation prediction tasks (e.g., only $<5\%$ mutant samples). High-quality, large-scale pathological data with precise molecular annotations will be needed to boost model development. Alternatively, transfer learning has proven to be useful in computer vision tasks when training samples are less available [96, 121]. Therefore, a pre-trained classifier built from diverse pathological datasets may provide superior results compared with our cancer type-specific model.

4.4 Summary

In this chapter, we demonstrated that key gene mutation outcomes and biological pathway activity of breast cancer can be predicted by deep-learning classifiers from whole slide images. We further validated the deep-learning model to infer mutation status on liver and lung cancers, respectively. Our WSI-based deep learning models can identify the point mutation status of 6 genes (RB1, CDH1, NF1, NOTCH2, TP53 and MAP3K1) and the copy number alteration of another 6 genes (FGFR1, EIF4EBP1, KAT6A, HEY1, ZNF217 and RAB25) in breast cancer. To deepen our understanding of cancer biology, we explored the predictive power of deep learning to predict underlying biological pathway activity, a challenging task involving complex biological relations among gene expressions. From the activity levels of 10 canonical signaling pathways derived from the mRNA expression data and copy number alteration inputs, we found that 3 important pathways (p53, pi3k and cell cycle) measured by mRNA expression and 2 pathways (Myc and Notch) measured by copy number alteration can be well
predicted from our analysis.

In conclusion, we demonstrate that deep neural networks can be used to predict molecular outcomes in breast cancer including gene mutations and biological pathway activities from histopathology whole-slide images. Our extensive results highlighted new findings among genotype-phenotype associations, offering insights into the identification of targeted therapies for breast cancer treatment.
Figure 4.5: Weight maps of tiles when predicting the point mutation status of RB1 and cell cycle pathway activity from mRNA expression data in breast cancer. (a) Tumor tiles after data processing. (b) Weight map of tumor tiles in RB1 point mutation prediction. Brighter green tiles have larger weights. (c) Weight map of the cell cycle pathway activity prediction. (d) and (e): Top 20 weighted tiles for the RB1 point mutation prediction and cell cycle pathway prediction, respectively. We marked four tiles that appear in both tasks. These tissues contain poorly differentiated breast carcinoma with necrosis or hemorrhage from a pathologist’s perspective.
Figure 4.6: Weight maps of tiles when predicting the point mutation status of TP53 and p53 pathway activity from mRNA expression data in lung cancer. (a) Tumor tiles after data processing. (b) Weight map of tumor tiles in TP53 point mutation prediction. Brighter green tiles have larger weights. (c) Weight map in p53 pathway activity prediction. (d) and (e): Top 20 weighted tiles for the TP53 point mutation prediction and p53 pathway prediction, respectively. We marked four tiles that appeared in both tasks. These tissues contain moderately differentiated lung carcinoma with papillary growth pattern from a pathologists perspective.
Figure 4.7: Weight maps of tiles when predicting the point mutation status of RB1 and cell cycle pathway activity from mRNA expression data in liver cancer. (a) Tumor tiles after data processing. (b) Weight map of tumor tiles in RB1 point mutation prediction. Brighter green tiles have larger weights. (c) Weight map in cell cycle pathway activity prediction. (d) and (e): Top 20 weighted tiles from RB1 point mutation prediction and cell cycle pathway prediction, respectively. We marked four tiles that appeared in both tasks. These tissues contain hepatocellular carcinoma with clear cell change from a pathologist’s perspective.
In this dissertation, we focused on several challenges in current deep learning-based nuclei segmentation task for cancer diagnosis and explored the possibility of predicting gene mutations and biological pathway activities from histopathology images for cancer treatment. The weakly supervised nuclei segmentation framework presented in Chapter 2 alleviates the data annotation burden of pathologists. With only 10% nuclear locations, it obtains comparable segmentation performance as the state-of-the-art methods using full pixel-level annotation while achieving a 60× speed-up in the annotation time. It is also not sensitive to the selection of initial locations of nuclei, which are favorable to its application in real-world scenarios. To address the issues in architecture and loss function of nuclei segmentation, we proposed the full resolution neural network (FullNet) to keep details in features maps and the variance constrained cross entropy loss to place a implicit spatial constraint on pixels belonging to a same instance. This solution improves the segmentation performance and achieves state-of-the-art results in both nuclei and gland segmentation tasks. Apart from nuclei segmentation, we explored the relationship between gene mutation status, pathway activity levels and the histopathology images via a deep learning model. We found in breast cancer that the point mutation of six important genes and copy number alteration of another six genes are predictable. And three canonical pathways’ activities are also related to image features. Our method was further validated on lung and liver cancers. These results could provide new insights to discover biomarkers for targeted therapy in breast cancer treatment.

There are several potential future directions related to improving the efficiency and performance of nuclei segmentation and better discovering the genetic features from
histopathology images. First, there is another avenue to address the data annotation problem in nuclei segmentation, i.e., realistic image synthesis by the generative adversarial networks (GANs) [32]. GANs have shown tremendous progress in image synthesis tasks, such as image-to-image translation [45, 131] and text-to-image synthesis [123]. We can random generate nuclei masks and then synthesize realistic histopathology images from those masks to augment the training data. It has been shown in [39] that the synthetic images can achieve comparable segmentation performance as real images. However, the quality and diversity of synthetic images may not be good enough to surpass the real annotated images. Considering that there are a lot of unannotated real histopathology images, it could be a better direction to combine GAN with the weakly or semi-supervised learning to generate more diverse and realistic images.

Besides reducing the annotation burden, the issue of lacking annotated data could be alleviated by making use of data from multiple sources via federated learning [117, 59, 11]. Federated learning aims to train a machine learning model using data from distributed data centers (e.g., hospitals, research institutes) without exchanging them. The privacy-preserving feature of federated learning makes it a promising direction in medical applications since medical data usually cannot be shared outside local centers due to privacy issues. Our preliminary results [15, 79] have shown that it is feasible to combine the advantages of GAN and federated learning to learn the overall distribution of distributed data, therefore producing high quality and diverse synthetic images for nuclei segmentation task.

Moreover, the proposed FullNet can be improved in three aspects. One is to reduce its computational and memory cost before it can be widely used in other segmentation tasks. We plan to explore techniques in light-weighted networks [41, 124] like group convolution, depth wise separable convolution and channel shuffle. Another is adopting self-supervised learning to pretrain FullNet on a large unannotated images and then fine-tune it on small medical datasets. The performance is expected to improve like applying ImageNet pretrained weights on classification tasks. As for the grid artifacts of dilated convolution, the current solution of using hybrid dilation factors is not perfect. It will be much better if we can find another strategy to increase the receptive field by
avoiding the use of large dilation factors.

In the gene mutation and pathway activity prediction tasks, there is much room to improve our current method on data processing and model structure.

**More accurate data processing steps.** The tumor tiles selected by our semi-automatic labeling strategy is not accurate enough. If we can train a high quality tumor/normal tile classifier beforehand to select tumor tiles, the prediction results may be improved. Besides, our method to identify pathway activity levels from mRNA expression or copy number alteration data is preliminary. More sophisticated label generation algorithms should be designed with the cooperation of experts in this area.

**Better feature extractor in the model.** In the current model we use the pre-trained ResNet-101 as the feature extractor. However, it was pretrained on natural images thus the extracted features may be not good enough for histopathology images because of the large domain shift. A better feature extractor can be trained on histopathology images by variational autoencoder like in [113].
References


[49] Katz, L.H., Likhter, M., Jogunoori, W., Belkin, M., Ohshiro, K., Mishra, L.: Tgf-
β signaling in liver and gastrointestinal cancers. Cancer letters 379(2), 166–172
(2016)


ysis 54, 88–99 (2019)


[53] Kong, H., Gurcan, M., Belkacem-Boussaid, K.: Partitioning histopathological
images: an integrated framework for supervised color-texture segmentation and
cell splitting. IEEE Transactions on Medical Imaging 30(9), 1661–1677 (2011)

25, 1097–1105 (2012)

dataset and a technique for generalized nuclear segmentation for computational
pathology. IEEE transactions on medical imaging 36(7), 1550–1560 (2017)

[56] Levy-Jurgenson, A., Tekpli, X., Kristensen, V.N., Yakhini, Z.: Spatial transcrip-
tomics inferred from pathology whole-slide images links tumor heterogeneity to

[57] Li, J., Hu, Z., Yang, S.: Accurate nuclear segmentation with center vector encod-
ing. In: International Conference on Information Processing in Medical Imaging.

[58] Li, J., Yang, S., Huang, X., Da, Q., Yang, X., Hu, et al., Z.: Signet ring cell de-
tection with a semi-supervised learning framework. In: International Conference

[59] Li, T., Sahu, A.K., Talwalkar, A., Smith, V.: Federated learning: Challenges,
(2020)

[60] Lin, D., Dai, J., Jia, J., He, K., Sun, J.: Scribblesup: Scribble-supervised convo-
lutional networks for semantic segmentation. In: Proceedings of the IEEE Con-

(2017)

segmentation. In: Proceedings of the IEEE conference on computer vision and


[75] Qu, H., Riedlinger, G., Wu, P., Huang, Q., Yi, J., De, S., Metaxas, D.: Joint segmentation and fine-grained classification of nuclei in histopathology images. In:


