

M.Sc. Thesis
Master of Science in Engineering

DTU Compute
Department of Applied Mathematics and Computer Science

CycleGANs for Tissue Segmentation

Hyemee Sohn (s181323)

Kongens Lyngby 2020



DTU Compute

Department of Applied Mathematics and Computer Science

Technical University of Denmark

Matematiktorvet

Building 303B

2800 Kongens Lyngby, Denmark

Phone +45 4525 3031

compute@compute.dtu.dk

www.compute.dtu.dk

Abstract

Manual segmentation of histology images requires evaluation by an expert, meaning it is time-consuming and often of inconsistent quality. In this thesis, we implemented Gadermayr et al.'s unsupervised segmentation approach [1]. The approach requires generating simulated label images, which are theoretical labels of the segmented tissue. The simulated label images are then used as the Y domain in a CycleGAN training with domain X being the original histology images. Using this approach, we successfully segmented fat in liver tissue. When applied to slightly more complex tasks, such as segmenting fat in pancreas tissue, the approach struggled. Instead we segmented the fat in the pancreas using a U-net, a supervised approach [2]. Finally, we attempted to apply Gadermeyr et al.'s approach on a very complex problem, attempting to segment four tissue compartments in the pancreas simultaneously. When following the proposed approach from Gadermeyr' et al.'s paper, it failed to segment the compartments simultaneously. Consequently, we expanded upon their approach by abandoning the previous simulated labels and instead generated more realistic label images with segmentation results from separate U-nets for each tissue compartment. These realistic label images were then used as the simulated labels in domain Y in the CycleGAN training. However, despite having a highly realistic Y domain, the approach failed to segment the pancreas tissue. Further expansion of the model is likely required to allow the method to be applied on more complex tissue segmentation tasks.

Acknowledgements

I would like to thank my thesis supervisor, Vanessa Jurtz from Novo Nordisk. Thank you for all the advices, help, and inspiration for the thesis. Whenever I struggled, Vanessa always steered me in the right direction. It has been astonishing work with Vanessa and I learned a lot from her.

I would also like to thank Professor Ole Winther of the Department of Applied Mathematics and Computer Science at the Technical University of Denmark. He suggested many ways to improve the thesis.

Furthermore, I would also like to thank Novo Nordisk for all the support, including office space, laptop, and access to the GPUs. Without the GPUs, it would have been impossible to train the very large networks on the large images throughout this thesis.

Finally, I would thank all my family and friends who always support me to accomplish my studies.

Sincerely,

Hyemee Sohn

Contents

Abstract	i
Acknowledgements	ii
Contents	iii
1 Introduction	1
2 Theory	4
2.1 Basic Neural Network	4
2.1.1 Activation Functions	7
2.1.2 Convolutional Neural Network	7
2.1.3 Residual block	8
2.2 U-net and Dice Coefficient	9
2.3 Generative Adversarial Network	11
2.4 CycleGAN	13
3 Related Works	16
3.1 Gadermayr's Approach	16
3.2 Revisting CycleGAN for Semi-supervised Segmentation	18
4 Methods	21
4.1 Data Sourced from GTEx	21
4.2 Fat Segmentation in Pancreas and Liver	22
4.2.1 Liver	23
4.2.2 Pancreas	27
4.2.3 Calculating Fat Percentage from Segmentation Results	31
4.3 Segmenting 4 Tissue Compartments in Pancreas	32
4.3.1 Naïvely applying Gadermayr's Approach	32
4.3.2 Train CycleGAN with Label Images from U-net Results	34
5 Results	37
5.1 Fat Segmentation in Pancreas and Liver	37
5.1.1 CycleGAN Successfully Segments Fat in Liver	37
5.1.2 Pancreas	40
5.1.3 Analysis of Fat Percentage in Pancreas and Liver	44

5.2	Segmenting 4 Tissue Compartments in Pancreas	46
5.2.1	Naïve Gadermayr Approach Fails Segmentation Task	47
5.2.2	Gadermayr’s Approach with Labels Generated with U-net Fails Segmentation Task	48
5.2.3	Analysis of Pancreas Segmentation Results by U-net	51
6	Discussion	54
6.1	Future Works	59
7	Conclusion	61
A	An Appendix	62
A.1	Code Availability	62
A.2	Method Extracting Cell Mask	62
A.3	Plots for Relation between Gender and Fat Percentage	63
A.4	Training Curves of U-net for Different Annotations	64
	Bibliography	66

CHAPTER 1

Introduction

In the last 50 years, our understanding of biological systems has advanced greatly. This has allowed for more comprehensive drug discovery processes as we know increasingly more of the molecular mechanisms behind each disease. Histology is the microscopic study of tissue structure in organisms, which has become an essential tool to study diseases and drug treatment effects [3], [4]. It works by staining the tissue with chemicals in order to highlight the particular areas of interest for analysis. In pharmaceutical studies, histology imaging is usually used to examine biopsies or surgical specimens after the specimen has been processed, which can make it difficult to obtain a large number of human histology images. One advantage of analyzing histology images is that they are highly intuitive, one can interpret the results directly by just looking at it.

However, histology images have remained difficult to use in large scale studies. This is due to the assessment of histology images, either qualitative or quantitative, requiring manual assessment by an expert. Naturally, the expert's beliefs and biases will affect the assessment quality. To avoid this inconsistency in quality, an automated quantitative assessment is needed, which requires segmentation of the histology image to begin with. Segmentation of an image is a procedure of subdividing an image into multiple segments to make an analysis of the image more meaningful and easier[5]. However, the size of histology images are huge, so it is time-consuming and cost-expensive to segment them manually.

Naturally, there have been several attempts to automate this manual segmentation task [6], [7]. Many computer vision algorithms achieved somewhat successful results [4], [8]. However, recently new deep learning methods outperform previous state-of-the-art computer vision algorithm in other practical applications [9]. Specifically, we had three goals for this thesis:

1. Explore unsupervised deep learning methods to automate the segmentation of the histology images.
2. Segment fat in the histology images of pancreas and liver tissue
3. Segment multiple tissue compartments simultaneously in the histology images of pancreas tissue.

Our goals aim the automated segmentation process of the histology images, which would provide a pipeline to be used later in future studies using different setups with histology images. The data used in this thesis is from the [GTEx portal](#) (Genotype-Tissue Expression project), a public database with multi-omics data sets and histology images.

All GTEx data stem from deceased people, primarily from traumatic deaths in order to avoid data from people suffering from possible related diseases. Most importantly for this thesis, the GTEx portal contains histology images of various organs with all the metadata.

In the following sections, we will first introduce the reader to the theoretical background of the techniques used in this thesis. We begin by introducing basic knowledge such as what is a neural network, then progress to more advanced applications such as U-net, generative adversarial network (GAN), and CycleGAN. After introducing the theoretical backgrounds of our analysis methods, we will shortly describe related literature. Finally, we will explain our experiments in details and discuss the results.

Biological Background

In addition to segmenting the fat tissue in the pancreas, we chose to segment three additional tissue compartments. We chose to do so, because the pancreas is especially interesting in diabetes research and thus Novo Nordisk A/S. The pancreas is the organ responsible for producing insulin, making it the key organ in diabetes. The tissue compartments we will be segmenting are the pancreatic islets, pancreatic acinar cells, adipose cells (fat), and ductal area, see in figure 1.1. The pancreas and these four tissue compartments will all be introduced in the following paragraph.

The pancreas is an organ that is both parts of the digestive and endocrine system, with the majority of the pancreatic tissue having a role in the digestive system [10]. The white adipose tissue (white fat tissue, see figure 1.1) that is present in the pancreas has no major function in either system. For the convenience of the readers, we refer to the adipose tissue in the pancreas as *fat* throughout this thesis. Meanwhile, the pancreatic acinar cells are the functional units of the digestive exocrine function in the pancreas which synthesizes, stores, and secretes digestive enzymes [10]. The ducts in the pancreas supply and transport pancreatic juice, which contains various enzymes. Another essential role of the pancreas is to regulate blood glucose levels. specifically, it is pancreatic islets that are responsible for this process. When glucose level in the blood becomes low, the alpha cells in the pancreatic islet secrete a hormone called glucagon. Glucagon signals the body to raise glucose levels in the blood by breaking down stored glucose in the body. Conversely, beta cells in pancreatic islet secrete insulin to decrease the blood glucose levels.

Recently, studies have claimed that the fat in pancreas and liver are related to each other [11]–[15]. Accordingly, we attempted to segment the fat in both organs to compare and investigate these claims. To briefly explain to the reader who is not familiar with biology, the liver helps the body metabolize carbohydrate and produces biochemicals for digestion and growth, and synthesizes proteins [16], [17]. The majority of the liver consists of hepatocytes cells, which control various biochemical reactions that happen

Pancreas Tissue Segmentation

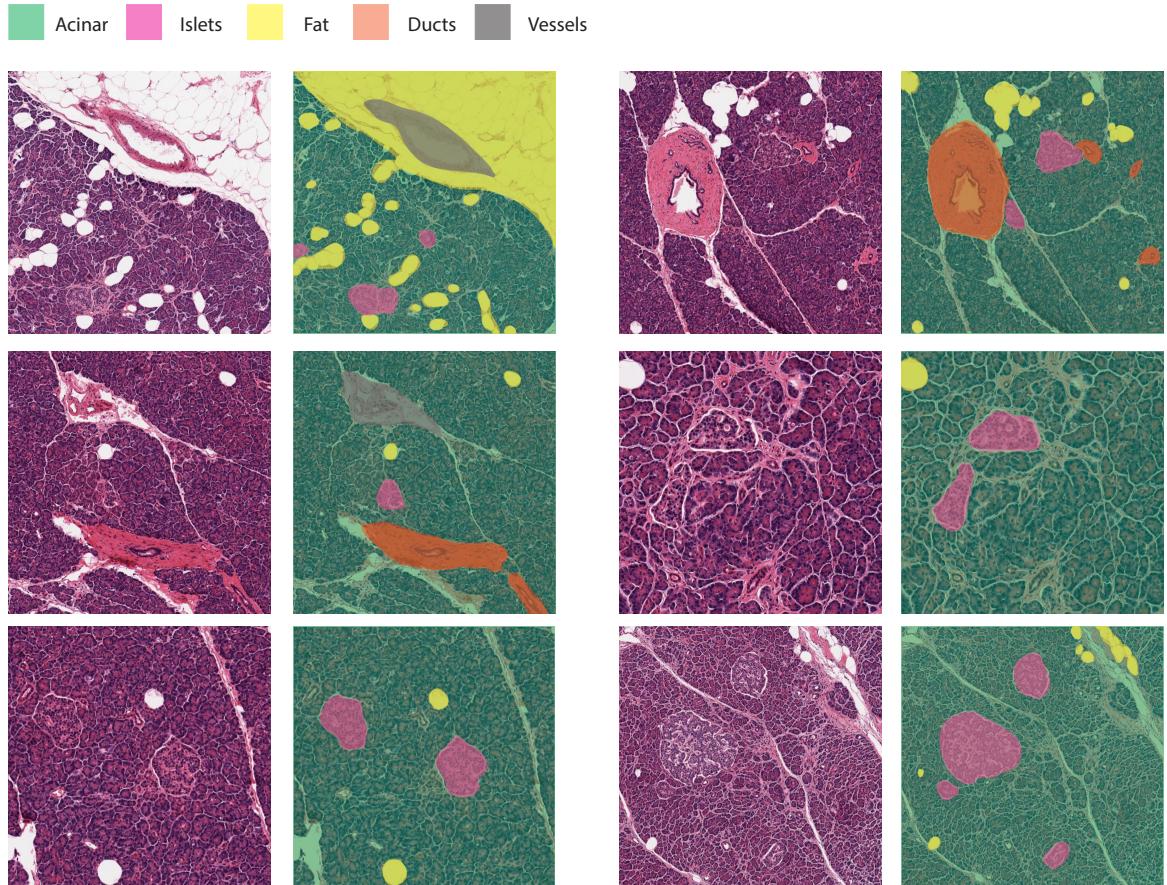


Figure 1.1: Pancreas tissue and results from a manual segmentation. The acinar cells (normal cells of the pancreas), are marked in green. The islets that produce enzymes such as insulin are marked in pink. The white adipose tissues (fat) are marked in yellow. The ducts transporting fluids are shown in orange. Finally, the vessels supplying blood to the tissue are segmented in grey.

in the liver. We will primarily be focusing on the fat tissue in the liver in this thesis.

CHAPTER 2

Theory

In this chapter, we describe the important deep learning theories behind this thesis.

Section 2.1 presents the basic concepts of the neural network including forward and backward propagation, activation functions, convolutional neural network, and residual block. Note that we mainly describe theories related to this thesis. In section 2.2, we describe well-known deep learning technique called U-net proposed by Ronneberger et al.[2]. U-net is a convolutional neural network established to perform supervised segmentation of biomedical images. Section 2.3 introduces a recently developed deep learning framework called Generative Adversarial Network (GAN) proposed by Goodfellow et al.[18]. Lastly, in section 2.4, we describe another deep learning framework called CycleGAN proposed by Zhu et al.[19] used for image-to-image translation using GAN.

2.1 Basic Neural Network

The term neural network originated from the attempts to get mathematical representations of information processing in a biological organism[20]. Training a neural network consists of forward propagation and backward propagation. To understand the basic concept of the neural network, consider a simple fully-connected network architecture with input, output layer and one hidden layer with M hidden units in figure 2.1.

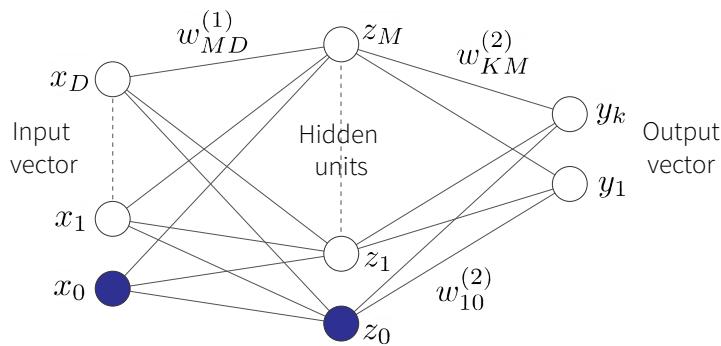


Figure 2.1: An example of a simple two layer neural network described in [20].

Suppose we have input vector $x = (x_1, \dots, x_D)$ with D variables, and consider M linear

combinations of the input variables as:

$$a_j = \sum_{i=1}^D w_{ji}^{(1)} x_i + w_{j0}^{(1)} \quad (2.1)$$

when $j = 1, \dots, M$. The $w^{(1)}$ is a adaptable parameters in the first layer of the network connecting input layer to the hidden layer. More specifically, we will refer $w_{ji}^{(1)}$ as weights and $w_{j0}^{(1)}$ as biases. Each output a_j is then applied with differentiable, non-linear *activation function* $h_1(\cdot)$:

$$z_j = h_1(a_j) \quad (2.2)$$

This non-linear activation function for first layer $h_1(\cdot)$ can be chosen based on the nature of the input data and the ouput target vector. The second layer then takes the ouput from the first layer as input, and computes:

$$a_k = \sum_{j=1}^M w_{kj}^{(2)} z_j + w_{k0}^{(2)} \quad (2.3)$$

when $k = 1, \dots, K$. K is a number of variables in output vector $y = (y_1, \dots, y_K)$. Output activation function h_2 is applied to get output:

$$y_k(\mathbf{x}, \mathbf{w}) = h_2 \left(\sum_{j=1}^M w_{kj}^{(2)} h_1 \left(\sum_{i=1}^D w_{ji}^{(1)} x_i + w_{j0}^{(1)} \right) + w_{k0}^{(2)} \right) \quad (2.4)$$

This process of proceeding input vector to output vector using weights and activation functions is called *forward propagation*.

To train the neural network, the choice of the loss function plays a significant role. One example of the loss function is sum of square loss:

$$E(\mathbf{w}) = \sum_{n=1}^N \|\mathbf{y}(\mathbf{x}_n) - \mathbf{t}_n\|_2^2 \quad (2.5)$$

where $\{\mathbf{x}_n\}$, $n = 1, \dots, N$ is a given set of training samples and $\{\mathbf{t}_n\}$ is a target vector we want to compare with our output vector \mathbf{y} in supervised learning setup. This sum of square error function is appropriate when the target vector is continuous, but not appropriate for classification task when the target vector is discrete. In this case, we can use cross-entropy loss function:

$$E(\mathbf{w}) = - \sum_{n=1}^N \sum_{k=1}^K \{ t_{nk} \ln y_{nk} + (1 - t_{nk}) \ln (1 - y_{nk}) \} \quad (2.6)$$

where $y_{nk} = y_k(\mathbf{x}_n, \mathbf{w})$. These are the two most basic loss functions, but much more complex loss functions are available depending on the training task, model, and input data.

To train the whole network, we should find a weight vector \mathbf{w} that minimizes the chosen loss function $E(\mathbf{w})$. To do this, we utilize stochastic gradient descent to find a minimum of $E(\mathbf{w})$. The smallest value of the loss function $E(\mathbf{w})$ can be found at the point in \mathbf{w} space that makes the derivative of the $E(\mathbf{w})$ as 0. In mathematical expression:

$$\nabla E(\mathbf{w}) = 0 \quad (2.7)$$

Yet, the analytic solution of Equation 2.7 is not easy to compute. Hence we start with choosing initial weights, $\mathbf{w}^{(0)}$, and develop an iterative procedure to update weight vector successively with:

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E(\mathbf{w}^{(\tau)}) \quad (2.8)$$

Here, η is so-called *learning rate* and τ indicates step. At each step, the weight vector moves towards the direction that minimizes the error function.

Now, we can process *backward propagation*, commonly called as backprop. The backward propagation is a simple calculation of gradients of the network architecture based on chain-rule. For example, consider δ_j defined as:

$$\delta_j = \frac{\partial E}{\partial a_j} \quad (2.9)$$

and δ is called *error*. With this δ , the derivative of loss function $\nabla E(\mathbf{w})$ can be written simply as:

$$\frac{\partial E}{\partial w_{ji}} = \delta_j z_i \quad (2.10)$$

The δ of output units of the network in figure 2.1 can be calculated as:

$$\delta_k = y_k - t_k \quad (2.11)$$

The δ of hidden units can be calculated as:

$$\delta_j = h'(a_j) \sum_{k=0}^K w_{kj} \delta_k \quad (2.12)$$

Obtaining this δ is called the backward propagation, since δ is calculated backward through the network. After the backward propagation, we can update the weight \mathbf{w} with formula combined with Equation 2.8 and 2.10:

$$w_{ij} = w_{ij} - \eta \cdot \delta_j \cdot z_i \quad (2.13)$$

The description of this simple neural network can be expanded to larger network architecture, yet the basic concept remains the same.

2.1.1 Activation Functions

Activation functions are functions used in neural networks to compute the weighted sum of input and biases, of which is used to decide if a neuron can be neglected or not[21]. The choice of the activation function depends on the task, input data type, and output type, etc. There are linear and non-linear activation functions and the place of activation function can also differ. We will briefly introduce some of the classic activation functions used in this thesis.

First, *Sigmoid function*. Sigmoid function is usually used at the output layer for the classification problem. The output of Sigmoid function can be considered as probability since the range of Sigmoid is $[0, 1]$. The formula of Sigmoid function is:

$$f(x) = \frac{1}{1 + \exp^{-x}} \quad (2.14)$$

Second, *Hyperbolic Tangent function* which commonly called as *Tanh* activation function. The range of the hyperbolic tangent function is $[-1, 1]$. The Tanh function often results better performances compared to the Sigmoid function [21]. The formula of Tanh function is:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.15)$$

Lastly, *Rectified Linear Unit function* also known as *ReLU* activation function proposed in [22]. ReLU is widely used activation function in various deep learning framework. It is commonly used for the hidden layers. Formula of ReLU activation function is:

$$f(x) = \max(0, x) = \begin{cases} x & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases} \quad (2.16)$$

ReLU activation function rectifies the output value smaller than 0 by make them as 0.

2.1.2 Convolutional Neural Network

Convolutional Neural Network (CNN) is another well-known deep learning approach that is commonly used for various fields and known to have better performances than classical fully-connected neural network [23]. The major reason why CNN is better than the fully-connected neural network is that CNN generalizes the feature of the input data in an improved way. Three factors that make CNN as a powerful tool in deep learning is presented in [23]. First, CNN used a concept of *weight sharing*, and this reduces a huge amount of parameters to be trained. Because of fewer parameters to be trained, CNN trains more smoothly and the chance of overfitting is also reduced much. Second, CNN combines the classification part and feature extracting part. Lastly, CNN is easier to implement a large scale network compared to fully-connected neural networks. General CNN architecture has four elements - convolutional layer, pooling, activation function, and fully connected layer. The simple CNN architecture is depicted in figure 2.2.

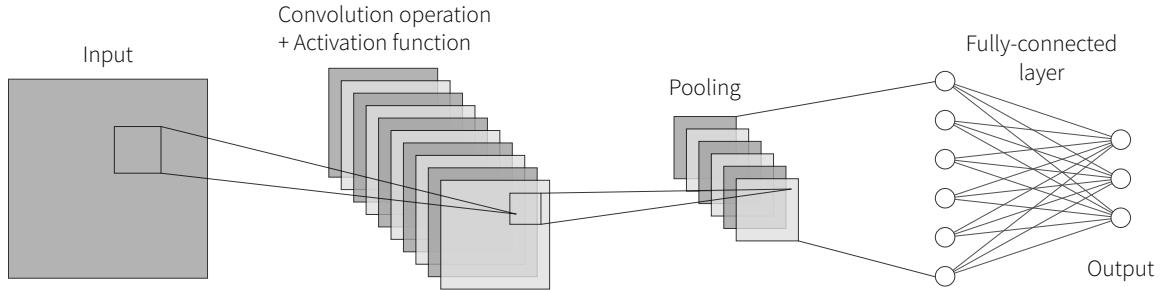


Figure 2.2: Simple architecture of a convolutional neural network with a convolutional layer, pooling layer, and fully connected layer.

The convolutional layer is a layer that uses convolution operation. The weight sharing concept of CNN comes from utilizing this convolution operation. The weight vector w in the fully-connected neural network connects each input unit to each output unit. Yet, in the convolutional layer, the weight vector w is shared throughout the whole input units. This weight vector is commonly called as *filter* or *kernel*, and it slides over the input vector vertically and horizontally to produce a feature map as output vector. This process is called the convolution operation. This convolution operation allows CNN to catch the contextual information such as geometric features, not only pixel-based information. The output from convolution operation is then applied to activation function (most commonly, ReLU activation function).

The output from the convolutional layer is followed by a pooling operation. The dominant advantage of pooling operation is that it reduces the amount of the parameters to be trained and makes the computation of the network easier. Pooling operation slides the window (with a certain size) through the output vector and applies selected pooling method. The most common pooling method is max pooling, which only left the maximum in the sliding window. The last part of the CNN architecture is a fully-connected layer described previously in this section. This fully-connected layer is applied only in the last part of the architecture for classification. The training of the CNN is similar to that of fully-connected networks. It performs stochastic gradient descent and updates the parameters until the network converges.

2.1.3 Residual block

The concept of the residual block is proposed by He et al.[24] in deep residual learning framework. They focused on the vanishing or exploding gradient problem when the network consists of a deep amount of layers. Also, when a network is deep, the degradation of the accuracy problem often happened. To overcome these problems, they proposed the framework commonly called *Residual block* or *ResNet block*. The simple architecture of a single residual block is depicted in figure 2.3.

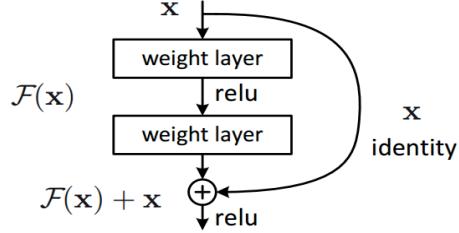


Figure 2.3: An example of a residual network block as presented in [24].

Consider x as an input vector to the residual block, and $\mathcal{F}(x)$ is a residual function approximated from $\mathcal{H}(x) - x$, where $\mathcal{H}(x)$ is an underlying mapping. The term *residual* comes from this approximation using residual difference between x and $\mathcal{H}(x)$. The output vector y is calculated as:

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{\mathbf{w}_i\}) + \mathbf{x} \quad (2.17)$$

where \mathbf{w}_i is a weight vector of i th layer in residual block. The $+$ operator in Equation 2.17 is combination of identity mapping and element-wise addition. The major advantage of the residual block is that it makes the whole network easier to optimize and the network can be deeper without gradient problems and degrading accuracy problem. The residual function $\mathcal{F}(\mathbf{x})$ can use fully-connected layers and convolutional layers in practice.

2.2 U-net and Dice Coefficient

For semantic segmentation, deep convolutional neural networks (CNN) had have outperformed results compared to other computer vision methods. U-net is a fully convolutional network that is a powerful tool for segmenting especially biomedical images. U-net is highly efficient in the sense that it learns without a large set of data and results in more precise segmentations than other fully convolutional networks.

The U-net architecture consists of roughly two big parts. First part is called *contracting path*, which is the left half of the u-shaped network in figure 4.10. Second part is called *expansive path*, which is the right half of the u-shaped network in figure 4.10. In contracting path, downsampling operators, which is repeated application of two 3x3 unpadded convolutions followed by a rectified unit (ReLU) as activation function and 2x2 max pooling with stride 2, are performed repeatedly. With each downsampling operator, the number of feature channels is doubled. In the expansive path, upsampling operators which make the number of features to be half are repeatedly performed. The core idea of this upsampling operator is to use successive 3x3 convolutional layers with ReLU activation function after 2x2 convolution (also known as upconvolution) replaced the usual pooling operators, which leads to the layers can have high-resolution output. In the upsampling part, the feature channels that learned from previous convolutional

layers can be used in higher resolution layers. This can be done by concatenating the cropped feature map from the contracting path after each upconvolution in the upsampling operator. This is why the whole network has a symmetric u-shape. The example of U-net architecture can be seen in figure 4.10. The important feature of U-net is that it has a comprehensive amount of features when the upsampling operator is performed. This makes the network to localize and disseminate the context information to the layer of higher resolution.

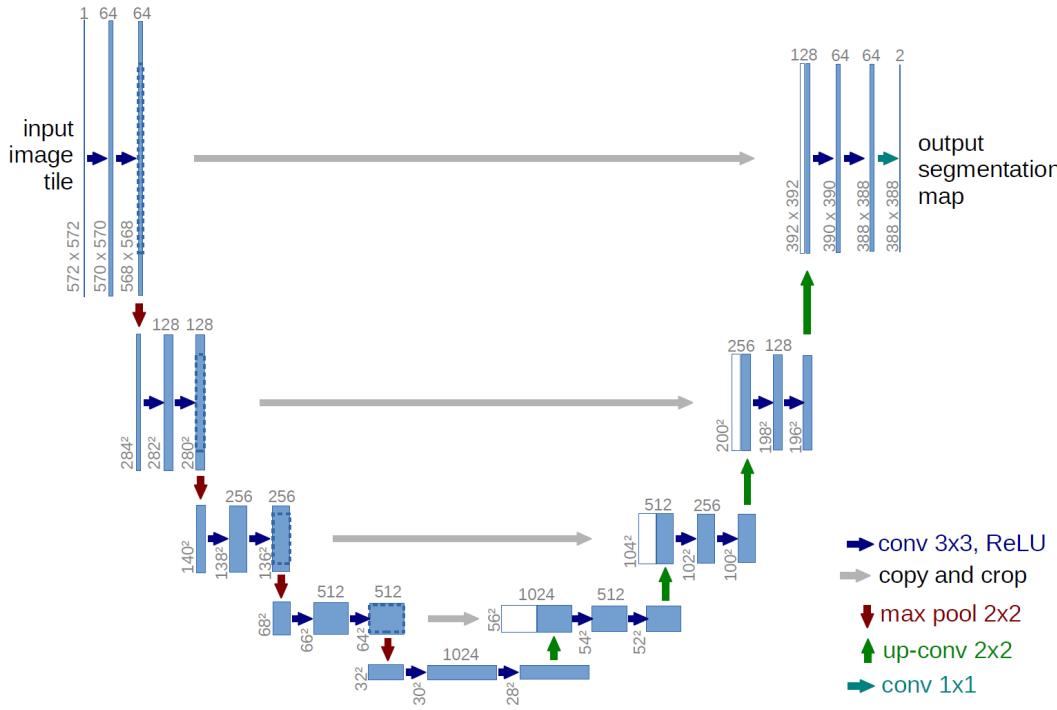


Figure 2.4: The U-net architecture from Ronneberger et al.[2]. The number on the top of each box is the number of channels. The concatenated feature maps from contracting path is denoted as white box in the expansive path. Each colored arrow represent different operations that are explained in the figure legend.

The energy function in the Ronneberger et al.[2] is computed with a combination of a pixel-wise softmax as a final activation function and cross-entropy as a loss function over the final feature map. However, in this thesis, we used sigmoid as a final activation function and dice coefficient loss function instead, for better performances. Dice coefficient loss is a common loss function used for semantic segmentation tasks based on the Sørensen-Dice coefficient. The original formula for Dice coefficient is defined as:

$$DC = \frac{2|X \cap Y|}{|X| + |Y|} \quad (2.18)$$

where X and Y are two given sets and the operator $|\cdot|$ means number of elements of

the set. However, when this formula is applied to binary data, it can be defined as:

$$DC = \frac{2TP}{2TP + FP + FN} \quad (2.19)$$

where TP is a number of true positive, FP is a number of false positive, and FN is a number of false negative. More specifically, TP is the total number of pixels that considered as positive (as 1 in boolean data) in both the ground truth segmentation and the generated segmentation from the network, while FP is the total number of pixels that considered as positive in the generated segmentation from the network, but negative (as 0 in boolean data) in the ground truth segmentation. Likewise, FN is the total number of pixels that considered as negative in the generated segmentation from the network, but positive in the ground truth segmentation. In order to formulate minimizing the loss function, $1 - DC$ is used as a final loss function combined with the sigmoid activation function. For convenience, we will call the segmentation map which is one-channel binary image as *mask image* or *mask*. In mask image, 1 means that the pixel is segmented as object-of-interest we wanted to segment, and 0 means it is not segmented as the object-of-interest.

2.3 Generative Adversarial Network

In order to analyze histology images of liver and pancreas, we will use deep learning methods. Among all the different deep learning methods, we choose extensions of the Generative Adversarial Network (GAN). GANs are one of the most promising deep generative models, which do not have the same limitations as other generative models' such as intractable probabilistic computations from maximum likelihood estimation. The name Generative Adversarial Network(GAN) comes from its adversarial process in training, there are two models that are simultaneously trained together to conflict with each other. One model is called a generative model G , which generates samples from a data distribution and another model is the discriminative model D , which discriminates true samples from samples generated by the generator network, G . From now on, we will refer to the generator network as G and the discriminator network as D .

The generator network generates new samples by transforming a chosen prior distribution $p_z(z)$ on input noise variable z , to learn a distribution p_g of the generator G over data x . The generator, G , is defined as a differentiable function with multilayer perceptron with parameters θ_g , henceforth denoted as a mapping $G(z; \theta_g) : z \rightarrow x'$, where x' is in data space. The discriminator D is also defined as multilayer perceptron with parameters θ_d . It can also be represented as a mapping $D(x; \theta_d) : x \rightarrow p$, which p is a single scalar which represents the probability that x is from the data, not from the distribution p_g .

The two models in the GAN framework have different objectives. The objective of the discriminator D is to discriminate between the sample from the generator G and true

data samples. This can be achieved to maximize the probability of the output of the generator G . Meanwhile, the objective of the generator G is to generate fake samples that can deceive the discriminator D , so D cannot discriminate against the fake sample from G and the true data. This means that G produces high-qualified fake samples that are analogous to true data. This can be achieved to minimize $\log(1 - D(G(z)))$.

The full objective for the GAN is:

$$\min_G \max_D V(D, G) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})} [\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))] \quad (2.20)$$

This is also known as the GAN loss. Optimizing this objective is often stated as a minimax game of the GAN since the generator G tries to minimize the value function $V(D, G)$, while the discriminator D tries to maximize it simultaneously. The GAN is trained by converging to the optimal saddle point of this minimax game. To find this, the optimal discriminator D for any G should be considered. Goodfellow et al. suggest that for any fixed G , the optimal discriminator D is:

$$D_G^*(\mathbf{x}) = \frac{p_{\text{data}}(\mathbf{x})}{p_{\text{data}}(\mathbf{x}) + p_g(\mathbf{x})} \quad (2.21)$$

To prove this, we can rewrite $V(G, D)$ in Eq. 2.20 using definition of expectation:

$$\begin{aligned} V(G, D) &= \int_{\mathbf{x}} p_{\text{data}}(\mathbf{x}) \log(D(\mathbf{x})) dx + \int_{\mathbf{z}} p_{\mathbf{z}}(\mathbf{z}) \log(1 - D(g(\mathbf{z}))) dz \\ &= \int_{\mathbf{x}} p_{\text{data}}(\mathbf{x}) \log(D(\mathbf{x})) + p_g(\mathbf{x}) \log(1 - D(\mathbf{x})) dx \end{aligned} \quad (2.22)$$

For a function $y \rightarrow a \log(y) + b \log(1 - y)$, where $a, b \in \mathbb{R} \setminus \{0\}$, the maximum is at $y = \frac{a}{a+b}$ in $[0, 1]$. If we set $y = D(x)$ and $a = p_{\text{data}}$, $b = p_g(x)$, we got maximum at $\frac{a}{a+b} = \frac{P_{\text{data}}(\mathbf{x})}{p_{\text{data}}(\mathbf{x}) + p_g(\mathbf{x})}$. Since G is fixed, this value will be optimal discriminator $D_G^*(\mathbf{x})$.

Now, we know the optimal discriminator D for any generator G , we can maximize this optimal discriminator D . So, we can rewrite Eq. 2.20 as $C(G)$:

$$\begin{aligned} C(G) &= \max_D V(G, D) \\ &= \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} [\log D_G^*(\mathbf{x})] + \mathbb{E}_{\mathbf{z} \sim p_z} [\log(1 - D_G^*(G(\mathbf{z})))] \\ &= \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} [\log D_G^*(\mathbf{x})] + \mathbb{E}_{\mathbf{x} \sim p_g} [\log(1 - D_G^*(\mathbf{x}))] \\ &= \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} \left[\log \frac{p_{\text{data}}(\mathbf{x})}{P_{\text{data}}(\mathbf{x}) + p_g(\mathbf{x})} \right] + \mathbb{E}_{\mathbf{x} \sim p_g} \left[\log \frac{p_g(\mathbf{x})}{p_{\text{data}}(\mathbf{x}) + p_g(\mathbf{x})} \right] \end{aligned} \quad (2.23)$$

Goodfellow et al. also suggested that the global minimum of $C(G)$ can be achieved if and only if $p_g = p_{\text{data}}$ and then value of $C(G)$ becomes $-\log(4)$. If $p_g = p_{\text{data}}$, then we can conclude $D_G^*(\mathbf{x}) = \frac{1}{2}$ from 2.21. Put this $D_G^*(\mathbf{x})$ in Eq. 2.23, then we can get:

$$\mathbb{E}_{\mathbf{x} \sim p_{\text{tan}}} [-\log 2] + \mathbb{E}_{\mathbf{x} \sim p_g} [-\log 2] = -\log 4 \quad (2.24)$$

Of course this is ideal situation and hard to appear in practical experiments. So, subtract Eq. 2.24 from $C(G) = V(D_G^*, G)$:

$$C(G) = -\log(4) + KL \left(p_{\text{data}} \parallel \frac{p_{\text{data}} + p_g}{2} \right) + KL \left(p_g \parallel \frac{p_{\text{data}} + p_g}{2} \right) \quad (2.25)$$

Here, KL is the Kullback-Leibler divergence. The Kullback-Leibler divergence measures how different two probability distributions are from each other. The Kullback-Leibler divergence is one-directional, thus we need one for the data distribution and one for the generated distribution. The two Kullback-Leibler divergences in Eq. 2.25 can be expressed as the Jensen-Shannon divergence. Jensen-shannon divergence is also a mathematical measurement that measures similarity between two probability distributions, and it is based on Kullback-Leibler divergence. The definition of the Jensen-Shannon divergence is:

$$JSD(P \parallel Q) = \frac{1}{2}KL(P \parallel M) + \frac{1}{2}KL(Q \parallel M) \quad (2.26)$$

where P, Q and M is all probability distributions. By this definition, Eq. 2.25 can be rewrite as:

$$C(G) = -\log(4) + 2 \cdot JSD(p_{\text{data}} \parallel p_g) \quad (2.27)$$

The GAN network is said to be optimizing the Jensen-Shannon divergence that attempts to find the optimial saddle point for the minimax game of the GAN framework.

GANs are widely used nowadays in many real-world applications. The advantage of the GAN is that it does not require any Markov chain, so it can produce nice and clear output unlike other generative models using Markov chains. Furthermore, another advantage is that GANs are trained fully unsupervised. The disadvantage of the generative adversarial network is that it can be somewhat difficult to train. At the beginning of training, the discriminator is often more optimized since the generator network is not very well trained, making it easy to distinguish between real and generated samples. Also, it is hard to find the saddle point. The most significant cons is that there is no certain way to evaluate the performance of GAN.

2.4 CycleGAN

CycleGAN is a framework of image-to-image translation in an unsupervised approach using the GAN framework. Since it is difficult and time-consuming to pair images with labels and sometimes even impossible, unpaired image-to-image translation, CycleGAN, is highly desirable for many real-world implementations [19]. Especially, CycleGAN can be a powerful tool for segmenting images such as medical images or histology images.

CycleGAN translates images between two domains, X and Y . The neural network learns the mapping between two domains, $G : X \rightarrow Y$ which outputs the $\hat{y} = G(x)$, $x \in X$ and $F : Y \rightarrow X$ which outputs the $\hat{x} = F(y)$, $y \in Y$. The mappings are trained to translate one domain to another domain, distributing the output of one domain as identical to

another domain. These mappings are done using the GAN framework, meaning that each mapping consists of two adversarial discriminators D_X and D_Y , which try to distinguish between the image $x \leftrightarrow$ translated image $\hat{x} = F(y)$ and $y \leftrightarrow$ translated image $\hat{y} = G(x)$.

The critical factor that makes the CycleGAN so efficient is the loss function. There are two different losses in the full objective function of CycleGAN. First, classical adversarial loss is used during GAN training. Since we have two GAN frames in CycleGAN, there are two different adversarial losses. For mapping $G : X \rightarrow Y$ and discriminator D_Y , corresponding adversarial loss is:

$$\begin{aligned}\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) &= \mathbb{E}_{y \sim p_{\text{data}}(y)} [\log D_Y(y)] \\ &\quad + \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log (1 - D_Y(G(x)))]\end{aligned}\tag{2.28}$$

Likewise, for mapping $F : Y \rightarrow X$ and discriminator D_X :

$$\begin{aligned}\mathcal{L}_{\text{GAN}}(F, D_X, Y, X) &= \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D_X(x)] \\ &\quad + \mathbb{E}_{y \sim p_{\text{data}}(y)} [\log (1 - D_X(F(y)))]\end{aligned}\tag{2.29}$$

While each mapping attempts to generate the translated images that are similar to images from other domain, each discriminator tries to distinguish the difference between the original image and the translated image. In this double GAN framework, the goal for each mapping is to minimize the adversarial loss so the discriminator thinks the translated images are from its domain. On the contrary, the goal for each discriminator is to maximize the adversarial loss so that it can distinguish that the translated image is the result from the mapping, not from its domain. This contrast makes GAN to learn the mapping while playing minimax game between the mapping and discriminator, i.e. :

$$\min_G \max_{D_Y} \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y)\tag{2.30}$$

$$\min_F \max_{D_X} \mathcal{L}_{\text{GAN}}(F, D_X, Y, X)\tag{2.31}$$

Second part of the total loss function is known as the cycle consistency loss. Since two domains are not aligned with each other, there is infinite possibility of each mapping G and F to translate each domain, and the results are not guaranteed to be meaningful translation. To regularize the mappings to be consistent with each other, the cycle consistency loss is added to two adversarial losses. This is the reason why the second loss is called cycle consistency loss. For each image in its domain, the two mappings should return the original image when it is mapped twice (back to the original domain), i.e. $x \rightarrow G(x) \rightarrow F(G(x)) \approx x$ or $y \rightarrow F(y) \rightarrow G(F(y)) \approx y$. This two can be expressed mathematically:

$$\begin{aligned}\mathcal{L}_{\text{cyc}}(G, F) &= \mathbb{E}_{x \sim p_{\text{data}}(x)} [\|F(G(x)) - x\|_1] \\ &\quad + \mathbb{E}_{y \sim p_{\text{data}}(y)} [\|G(F(y)) - y\|_1]\end{aligned}\tag{2.32}$$

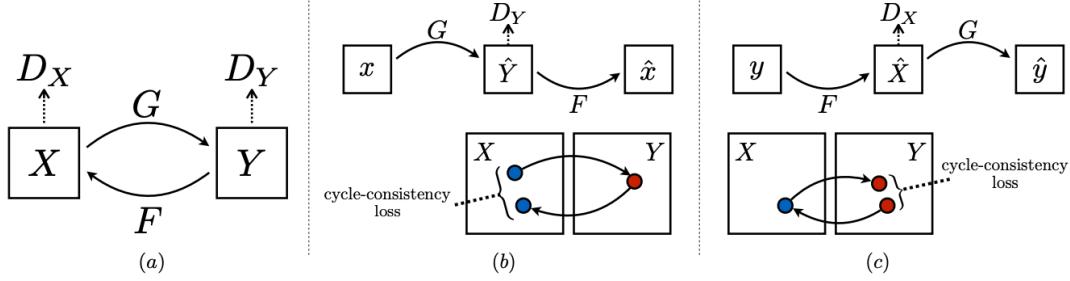


Figure 2.5: A visualization of the cycle consistency loss. In figure 2.5a, we can see two domains (X and Y) and networks that the CycleGAN framework contains, two generators (mapping G and F) and two discriminators (D_X and D_Y). In 2.5b and 2.5c illustrates how the cycle consistency loss forces a 1:1 mapping [19].

It used the L1 norm which means that it compares the pixel-wise difference. Now, with all these losses, the full objective can be defined as:

$$\begin{aligned} \mathcal{L}(G, F, D_X, D_Y) = & \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) \\ & + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) \\ & + \lambda \mathcal{L}_{\text{cyc}}(G, F) \end{aligned} \quad (2.33)$$

The λ in the full objective is a hyperparameter that controls the importance of the cycle consistency loss. If we have a large λ , it means that the mapping will strongly penalized to difference between $x \rightarrow F(G(x))$ and $y \rightarrow G(F(y))$. All these formulations can be simply described in figure 2.5.

CHAPTER 3

Related Works

In this chapter we will summarize the closest related work that has been published previously. In section 3.1, we begin with describing the unsupervised deep learning framework for the segmentation of histology images presented by Gadermayr et al. in their paper [1]. Throughout this thesis, we will refer to this unsupervised framework proposed by Gadermayr et al. as *Gadermayr’s approach*. In section 3.2, we briefly described the paper by Mondal et al.[25] that proposed some supervision to CycleGAN framework.

3.1 Gadermayr’s Approach

The CycleGAN framework proposed by Zhu et al.[19] has been adopted and used for various applications. Gadermayr et al.[1] proposed to use the CycleGAN to segment histology images in a fully unsupervised manner. The motivation of Gadermayr et al. is that an extensive amount of histology images without annotations are not utilized due to time-consuming and cost-intensive manual labeling process.

The whole pipeline of Gadermayr et al.’s approach is described in figure 3.1. Their novel idea was to generate theoretically possible labeled images, meaning generated images that contain objects of similar shape and distribution as expected of the object-of-interest in the histology images. We will refer to these as *simulated label images*. These simulated label images were then considered to be the Y domain in the CycleGAN training, see figure 3.1. In order to generate realistic simulated label images, they developed an annotation model. This annotation model requires approximation or prior knowledge of the underlying distribution and shape of the object-of-interest from domain X which contains original histology images. Knowing these parameters, the model could then generate theoretically possible labeled images, the simulated labels. Since these simulated label images form domain Y , it allows them to utilize CycleGAN to translate between the domains. In their paper, Gadermayr et al. demonstrate their framework by segmenting glomeruli within the kidney tissue, see figure 3.2. Here, they assumed that the number of glomeruli follows a Gaussian distribution and glomeruli has a roundish shape. In this thesis, we will refer to the whole labeled image as *label image* (or *label*) and the label of each object-of-interest in the whole label image as *annotation* i.e fat and cell annotation is part of the total label image.

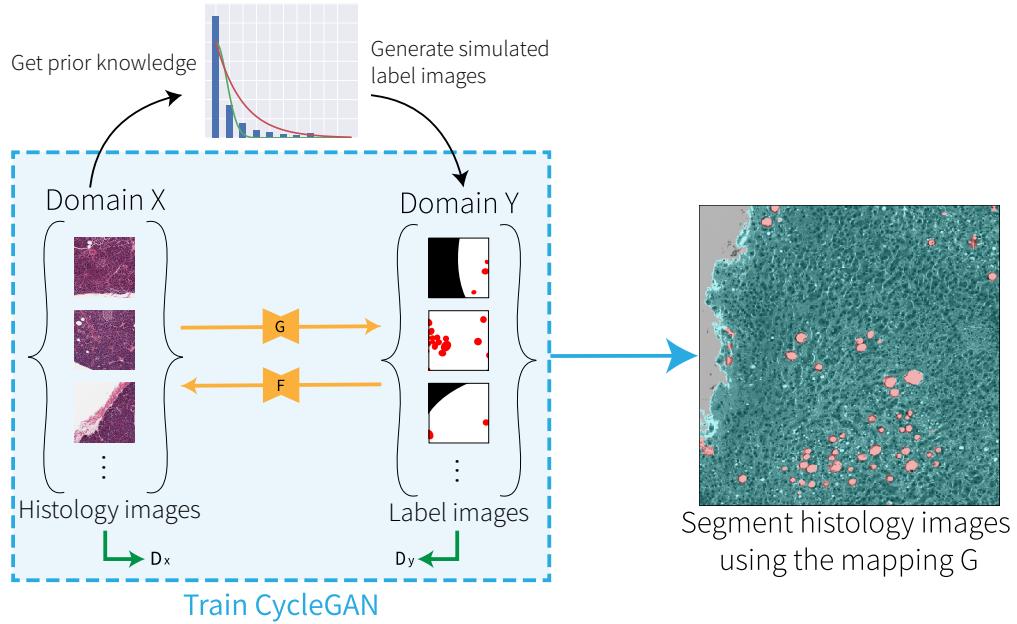


Figure 3.1: Diagram of the Gadermayr's approach. Domain X consists of histology images, and domain Y is generated by using prior knowledge of the tissue to generate theoretical possible labels images, *simulated label images*. A CycleGAN is then trained to translate between the domains, effectively labelling the images.

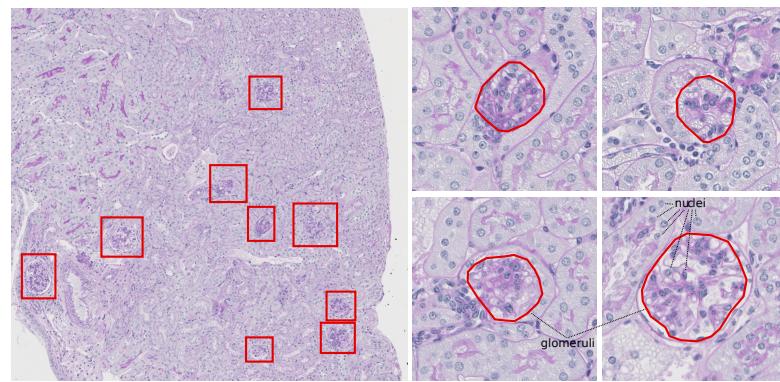


Figure 3.2: An illustration of the data used in the Gadermayr's paper [1]. (left) The whole slide image with glomeruli marked in squares. (right) Images zoomed on glomeruli and an example of a manual annotation of the glomeruli. Note you can see the cell nuclei.

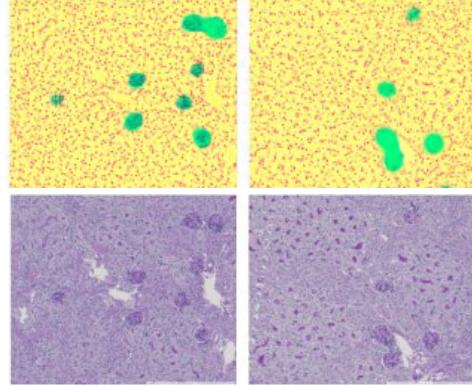


Figure 3.3: The results from the experiment conducted in the Gadermayr’s paper [1]. The upper row shows the generated label images with the generator G , with original histology images in the lower row. The glomeruli are successfully segmented (marked in green color).

They made two annotation models, one generates annotation of only glomeruli, and other generates glomeruli and nuclei. The results shows that experiment with only glomeruli does not successfully segment the histology images. However, experiment with both glomeruli and nuclei successfully labeled their histology images. The results from this experiment can be seen in figure 3.3.

3.2 Revisting CycleGAN for Semi-supervised Segmentation

Mondal et al.[25] proposed a semi-supervised framework for semantic image segmentation using CycleGAN. The main idea of their framework is to compel CycleGAN to perform semantic segmentation with unlabeled data and limited labeled data. In their framework, CycleGAN and the unlabeled data function as regularization of the supervised training with limited labeled data. The lack of labeled data hinders many deep learning methods in being applied more widely. However, if a limited amount of labeled data is available, their method can be applied to perform semantic segmentation. Also, frequently getting partially labeled data is feasible in practice.

In their framework, they used CycleGAN to map from the image domain to the segmentation label domain, and vice versa. They called the two generators G_{IS} and G_{SI} . Generator G_{IS} is the main generator, which translates image to segmentation label, conversely generator G_{SI} translates the segmentation label to the reconstructed image, however, generator G_{SI} is only utilized to improve training. The two discriminators D_I and D_S are discriminators for image domain and segmentation label domain, respectively. We will shortly introduce the loss functions used in their framework. Figure 3.4

described the brief pipeline of their framework.

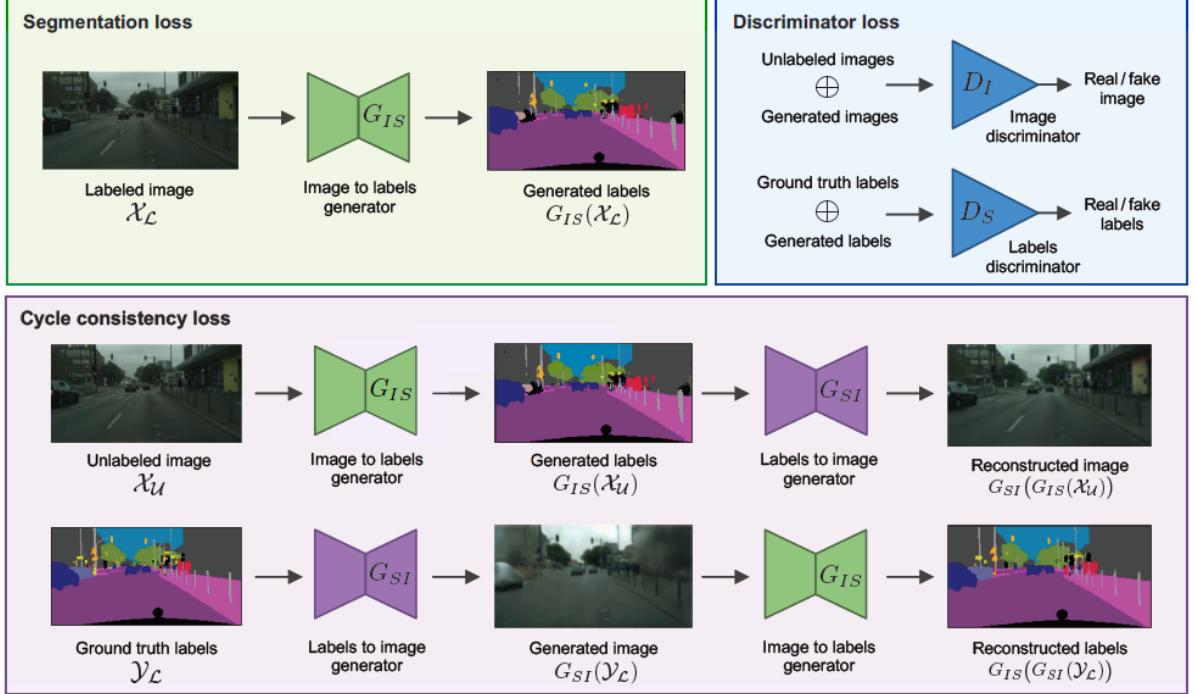


Figure 3.4: Pipeline of proposed framework from paper [25]. It shows different losses used in their paper, including supervised segmentation loss and losses from the CycleGAN framework.

They proposed two additional loss terms in their framework compared to the original CycleGAN. These two loss terms add supervision to the whole framework and will be referred to as the supervised losses. The first loss term is a supervised segmentation loss. It evaluates the difference between the generated labels from labeled images \mathcal{X}_L with ground truth labels \mathcal{Y}_L . Here, we only have a small number of labeled data (\mathcal{X}_L and \mathcal{Y}_L), and far more unlabeled images (\mathcal{X}_U). The loss can be written as:

$$L_{\text{gen}}^S(G_{IS}) = \mathbb{E}_{x,y \sim \mathcal{X}_L, \mathcal{Y}_L} [\mathcal{H}(y, G_{IS}(x))] \quad (3.1)$$

\mathcal{H} is the cross-entropy loss function. The second loss is also considered as a supervised loss. It evaluates difference between labeled images \mathcal{X}_L and reconstructed images of \mathcal{Y}_L with G_{SI} :

$$L_{\text{gen}}^I(G_{SI}) = \mathbb{E}_{x,y \sim \mathcal{X}_L, \mathcal{Y}_L} [\|G_{SI}(y) - x\|_2^2] \quad (3.2)$$

After the supervised losses, we have the adversarial loss and the cycle-consistency loss, both of which are unsupervised losses. Each adversarial loss for each mapping is written as:

$$L_{\text{disc}}^S(G_{IS}, D_S) = \mathbb{E}_{y \sim \mathcal{Y}_L} [(D_S(y) - 1)^2] + \mathbb{E}_{x' \sim \mathcal{X}_U} [(D_S(G_{IS}(x')))^2] \quad (3.3)$$

$$L_{\text{disc}}^I(G_{SI}, D_I) = \mathbb{E}_{x' \sim \mathcal{X}_U} [(D_I(x') - 1)^2] + \mathbb{E}_{y \sim \mathcal{Y}_L} [(D_I(G_{SI}(y)))^2] \quad (3.4)$$

The two cycle-consistency losses are:

$$L_{\text{cycle}}^I(G_{IS}, G_{SI}) = \mathbb{E}_{x' \sim \mathcal{X}_U} [\|G_{SI}(G_{IS}(x')) - x'\|_1] \quad (3.5)$$

$$L_{\text{cycle}}^S(G_{IS}, G_{SI}) = \mathbb{E}_{y \sim \mathcal{Y}_L} [\mathcal{H}(y, G_{IS}(G_{SI}(y)))] \quad (3.6)$$

With these losses, full objective function can be written as:

$$\begin{aligned} L_{\text{total}}(G_{IS}, G_{SI}, D_S, D_I) &= L_{\text{gen}}^S(G_{IS}) + \lambda_1 L_{\text{gen}}^I(G_{SI}) \\ &\quad + \lambda_2 L_{\text{cycle}}^S(G_{IS}, G_{SI}) + \lambda_3 L_{\text{cycle}}^I(G_{IS}, G_{SI}) \\ &\quad - \lambda_4 L_{\text{disc}}^S(G_{IS}, D_S) - \lambda_5 L_{\text{disc}}^I(G_{SI}, D_I) \end{aligned} \quad (3.7)$$

The additional supervised loss gives the whole framework a semi-supervised manner. The novel idea of this framework is to use generator G_{IS} as a segmentation model and introducing the supervised losses. The remaining parts of the model acts as a unsupervised regularizer that helps the generator G_{IS} to learn better segmentation with only a limited number of labeled data. Note that this is not a histology segmentation model, but since it shows promise, it could perhaps be applied to histology segmentation tasks in the future.

CHAPTER 4

Methods

This chapter describes the methods used in this thesis. However, before doing so, we have to define some of the terminologies used in this thesis to prevent confusion. The term *tile* will be used to refer to the small images that are extracted from the whole histology slide. The label of each tissue compartment in the tile is called an *annotation*. The whole label image for one tile or one whole slide image is called *label image* or *label*. For example, we can say annotation of an islet in the label image, as depicted in figure 4.1. Also, we define the term *mask* as a one-channel binary segmentation image containing only one annotation. In figure 4.1, an example of the islet mask can be seen. Also, we defined the term *simulated label* as a generated fake label from the annotation model with the prior knowledge of original histology images.

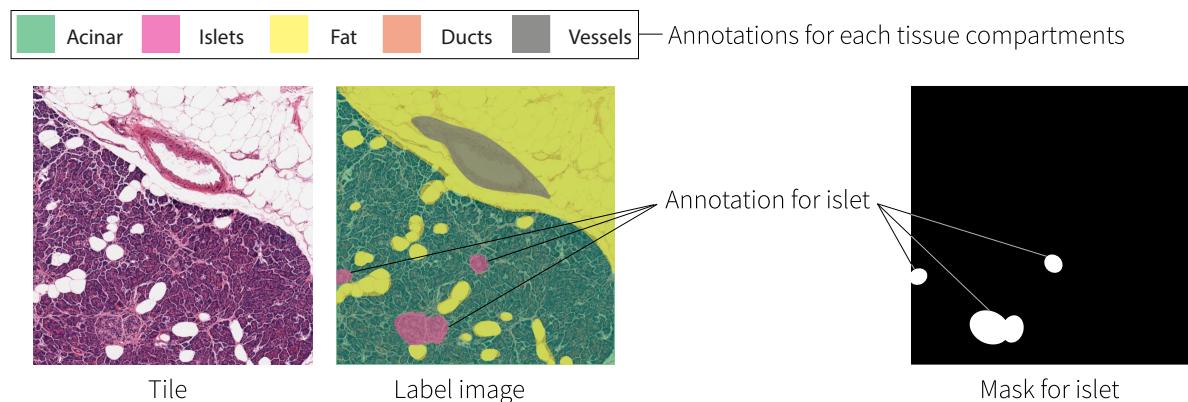


Figure 4.1: The four tissue compartments we will be segmenting and vessels are shown in the label image in their own color. Furthermore, an example of an annotation mask for the islets is shown.

4.1 Data Sourced from GTEx

The histology images we analyzed were sourced from the GTEx portal. The Genotype-Tissue Expression (GTEx) project is a comprehensive public database that contains tissue-specific gene expression and gene regulation data. The purpose of the database is to provide various resources for researchers to investigate human gene expression and

regulation, and their relation with genetic variation. The GTEx Portal provides open access to gene expression data, quantitative trait loci, and most importantly, histology images. The histology images of pancreas and liver from GTEx are both stained using Haematoxylin and Eosin (H&E). H&E staining is designed to highlight the different cellular components, particularly the border between the cell membrane, cytoplasm, and the cell nucleus. A total of 594 images of liver and pancreas from 594 subjects were analyzed in this thesis.

The available histology slide images are very large and have an average size over 1GB. Thus for training, the images were divided into many smaller images, *tiles*, using the *OpenSlide* package from Python. A zoom level of 14 in the *OpenSlide* package was used to create the tiles, which resulted in around 200 to 600 tiles from each GTEx histology slide. Each tile consisted of 512 by 512 pixels in each dimension, with the outer 6 pixels overlapping with adjacent tile. Pure background tiles were removed manually, leaving around 100 to 300 tiles per histology slide. In total, we ended up with 2248 liver tiles from 17 histology slides and 3043 pancreas tiles from 19 histology slides for training. An example of the extracted tiles can be seen in figure 4.2. Notice that the liver tissue (two righter images) is much more homogeneous in nature, thus we expect it to be simpler to analyze.

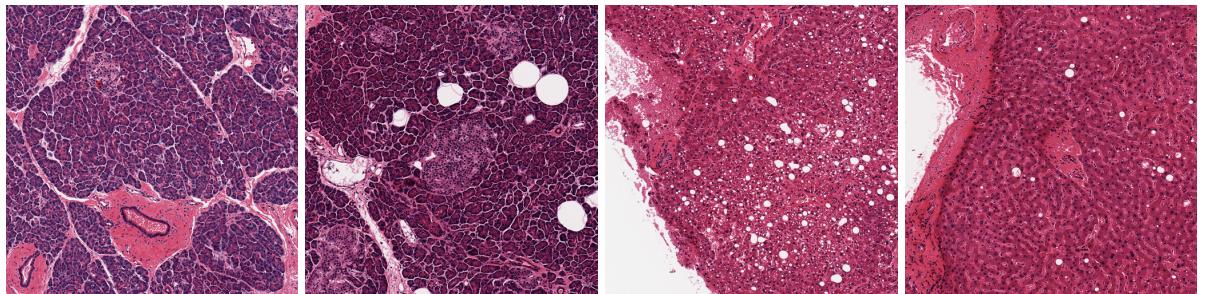


Figure 4.2: Example of extracted tile from histology slide images stained with Haematoxylin and Eosin. The left two images are pancreas tissue images while the two to the right are images of liver tissue.

4.2 Fat Segmentation in Pancreas and Liver

One of our goal was to segment fat in both liver and pancreas histology images in fully unsupervised manner. To achieve this, we implemented Gadermayr's approach[1] explained in section 3.1 to both the liver and the pancreas tissue. Since the complexity of liver tissue is lower than the complexity of pancreas tissue, we first applied Gedermayr's approach to segment fat in the liver. This is described in the following section, 4.2.1. Then we applied Gadermayr's approach to the pancreas as well, described in section 4.2.2.

However, since the results of Gadermayr’s approach applied to the pancreas tissue were not satisfactory, we applied a U-net to segment fat in the pancreas, this process will be described in section 4.2.2.2. Finally, we describe how we calculated the fat percentage from trained model segmentation results in section 4.2.3.

4.2.1 Liver

To get unsupervised segmentation of the liver, we applied Gadermayr’s approach[1]. The main task in the Gadermayr’s approach was to use CycleGAN to translate images from the histology domain into the label domain and vice versa. The first step in this process is to create the simulated label domain. Gadermayr et al.’s annotation model is insufficient and not publicly available [1]. However, they do mention that their annotation model requires a prior assumption of the shape and the occurrence distribution. From this information, we developed our own annotation model to create these simulated labels. The required prior knowledge of the tissue was approximated by analyzing the distribution of the annotations on 100 randomly selected tiles. Using the approximated prior knowledge, we could automate the simulated label image generation for histology tiles.

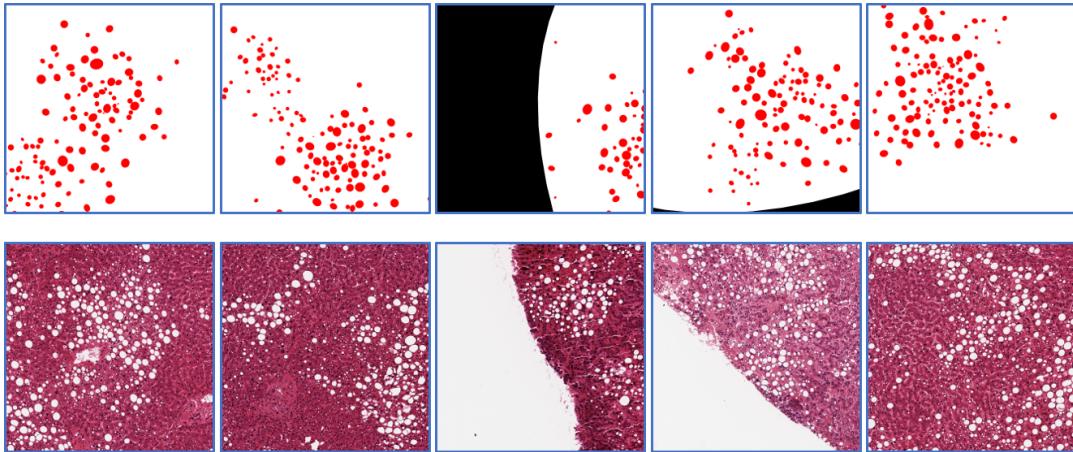


Figure 4.3: The upper row shows the simulated label images generated for the liver tissue. The lower row shows the original tile images of the liver for comparison with the label images in the upper row.

There are notable differences between the fat in the liver and the pancreas. In the pancreas, the fat clusters are dense and adjoined to each other, making it clearly distinguishable from the single fats (those not part of a cluster). From this, we categorized the fat tissue into two types for our simulated labels, *single fats* and *fat clusters or clusters of fat*. However, when it comes to the fat in the liver, we cannot easily separate single fats and fat clusters. Compare the fat in the pancreas tissue from figure 4.1 and the liver fat from figure 4.3. The fat in the liver is much more evenly spread out throughout the

tissue, thus we will refer to the liver fat clusters as *sparse fat clusters*. The characteristic of this sparse fat cluster makes segmentation of fat in the liver much simpler, since the fat cluster and the single fats resemble each other a lot. From the approximated prior distribution, we assumed there is maximum of two sparse fat clusters in each liver tile. The shapes of each fat annotation was assumed to be elliptical. With these prior hyperparameters of the fat in the liver tissue, we built our own annotation model.

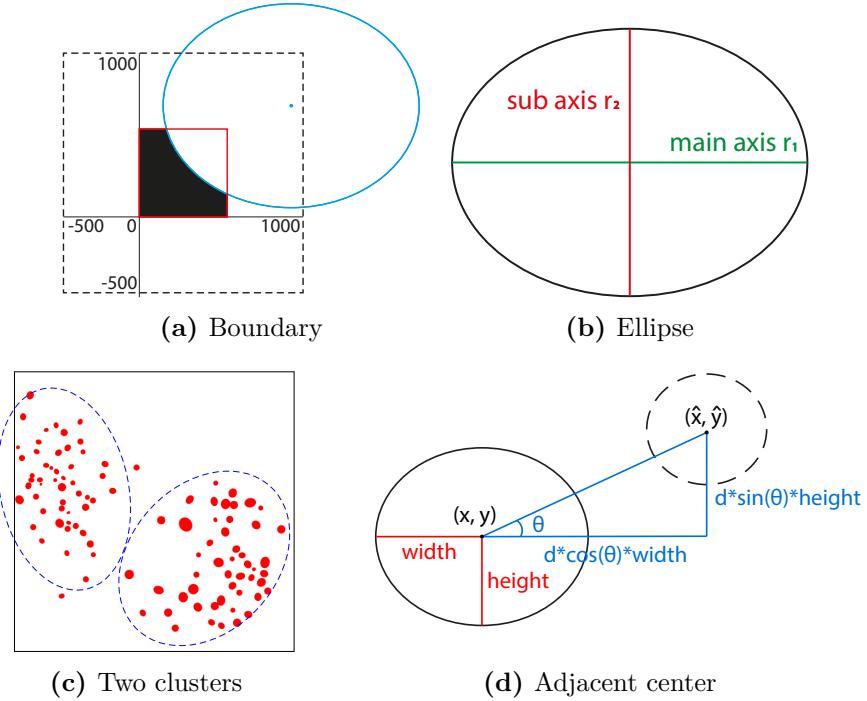


Figure 4.4: Figures to help understanding of how the simulated label images are created. (a) Method for creating boundary (blue ellipse) in the label image (red box). (b) Two axes are depicted within one ellipse, main axis r_1 (green line) and sub axis r_2 (red line). (c) Example of two sparse clusters created within the label image. Two clusters are roughly marked with blue dotted line for intuitive understanding. (d) Method for obtaining adjacent center, (\hat{x}, \hat{y}) , for given ellipse with center (x, y) in the fat cluster.

To generate simulated label images, our annotation model first generated the tissue area in the tile, this is also known as the *boundary*. The boundary is generated by generating an ellipse centered at a point (x, y) , where each x and y are drawn from a uniform distribution between $[-500, 1000]$. This is ± 500 to each side of the histology tile (marked in blue in figure 4.4(a)). This ellipse functions as a mask on the generated annotations, masking out any area not within the ellipse. The masked area takes value 0 (black color), while the non-masked area (within the ellipse) has value 255 (white color). This boundary occurs with 50% probability in our simulated label image. Next, fat annotations are generated within the boundary and the image borders (red box and

blue ellipse figure 4.4(a)). If there is no boundary, annotations are generated anywhere in the image (red box in figure 4.4(a)).

To generate fat annotations, we assumed that each fat takes elliptical shape. Each ellipse has a length for each of the two axes, main axis and sub axis, see figure 4.4(b). We assumed the length of main axis, r_1 , follows a Gaussian distribution $\mathcal{N}(\mu_r, \sigma_r)$. The length of sub axis r_2 is calculated as $r_2 = r_1 \cdot r_\delta$, where r_δ follows another Gaussian distribution $\mathcal{N}(\mu_\delta, \sigma_\delta)$. Finally, the rotation parameter α of the ellipse is drawn from a uniform distribution between $[0, 2\pi]$.

In order to generate the sparse fat clusters of the liver, we had to determine the number of ellipses in each fat cluster. We assumed the number of ellipses in the cluster g_a follows a Gaussian distribution $g_a \sim \mathcal{N}(\mu_{g_a}, \sigma_{g_a})$. In these sparse clusters, there is a small gap between each ellipse. The first ellipse is randomly placed within the boundary, and the center of the next ellipse is determined with a distance parameter d , see figure 4.4(d). The initial ellipse is centered at (x, y) with height $\frac{1}{2}r_1$ and width $\frac{1}{2}r_2$. The center of the next ellipse (\hat{x}, \hat{y}) is then calculated using the formula:

$$(\hat{x}, \hat{y}) = (x + d \cdot \text{width} \cdot \cos \theta, y + d \cdot \text{height} \cdot \sin \theta)$$

where θ is drawn from a uniform distribution between $[0, 2\pi]$. This process is repeated until we end up with the desired number of fat ellipses. Note that the ellipses cannot overlap to each other.

We assumed that there are at most two sparse clusters of fat per tile. The two clusters appeared independently with a probability of 50% and 30% respectively. The first cluster has fewer but larger ellipses that are more spread apart. The number of ellipses in the first cluster follow a Gaussian distribution $g_a \sim \mathcal{N}(\mu_{g_a} = 80, \sigma_{g_a} = 20)$, in which each ellipse follows $r_1 \sim \mathcal{N}(\mu_r = 7, \sigma_r = 2)$ and sub axis $r_2 = r_1 \cdot r_\delta$. The r_δ also follows Gaussian distribution, $r_\delta \sim \mathcal{N}(\mu_\delta = 1.1, \sigma_\delta = 0.1)$. A distance parameter of $d = 7$ was used to space out the individual small ellipses in the cluster. Finally, the rotation parameter, α , of the sparse cluster was sampled from a uniform distribution over $[0, 2\pi]$.

The second cluster contains more, but smaller, ellipses that are placed closer together than in the first cluster. We assumed the number of ellipses in the second cluster follows a Gaussian distribution $g_a \sim \mathcal{N}(\mu_{g_a} = 100, \sigma_{g_a} = 10)$. Each ellipses main axis, r_1 , follows $r_1 \sim \mathcal{N}(\mu_r = 5, \sigma_r = 1)$ and their sub axis, r_2 , is calculated $r_2 = r_1 \cdot r_\delta$. The r_δ again also follows Guassian distribution, $r_\delta \sim \mathcal{N}(\mu_\delta = 1.1, \sigma_\delta = 0.1)$. The rotation parameter, α , of the ellipse is generated identically to the first cluster. The distance between each ellipses is larger at $d = 10$. This is our annotation model that generate the fat annotation in the simulated label image for liver. The color of the fat label is set to red. You can see the final simulated label images with this annotation model in figure 4.3.

4.2.1.1 CycleGAN: Training Details

In accordance with Gadermayr's paper, we trained a CycleGAN to segment the histology tiles. This was done by designating the original histology tiles as domain X , and having the simulated label images as domain Y . This should allow the CycleGAN to translate between the two domains, effectively segmenting the histology tiles. The network architectures for the generator and the discriminator are identical between the mapping F and G .

The generator network architecture is shown in figure 4.5. As shown, the generator network begins with three 2-strided convolutional layers with kernel size 4, followed by six residual blocks with stride 1 and kernel size 4. Finally, three 2-strided transposed convolutional layers with kernel size 4 were applied. Batch normalization and rectified linear units (ReLU) activation function were applied between every layer, except for the last layer, which used Tanh activation function instead and no batch normalization.

The discriminator led off with four 2-strided convolutional layers with kernel size 4, followed by a 1-strided convolutional layer with a kernel size of 4, see figure 4.6. Next, the output tensor was flattened into a 1-dimensional vector and sent into a linear layer for the output. ReLU activation functions were applied between each layer except for the last linear layer, which instead used a Sigmoid activation function. Batch normalization was applied to each convolutional layer except for the first and last ones.

In domain X , we had 2248 liver histology tiles, and to have a similar number for both

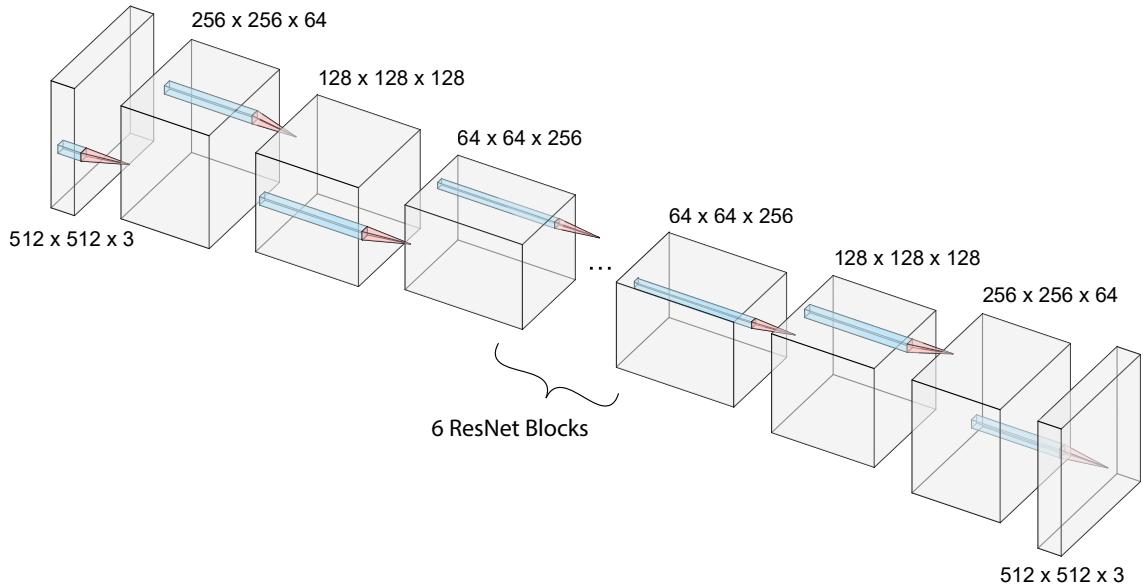


Figure 4.5: An overview of the architecture of the generator network of the CycleGAN. Note that there are four downsampling convolutional layers, followed by six residual blocks and followed by four upsampling layers to generate the samples.

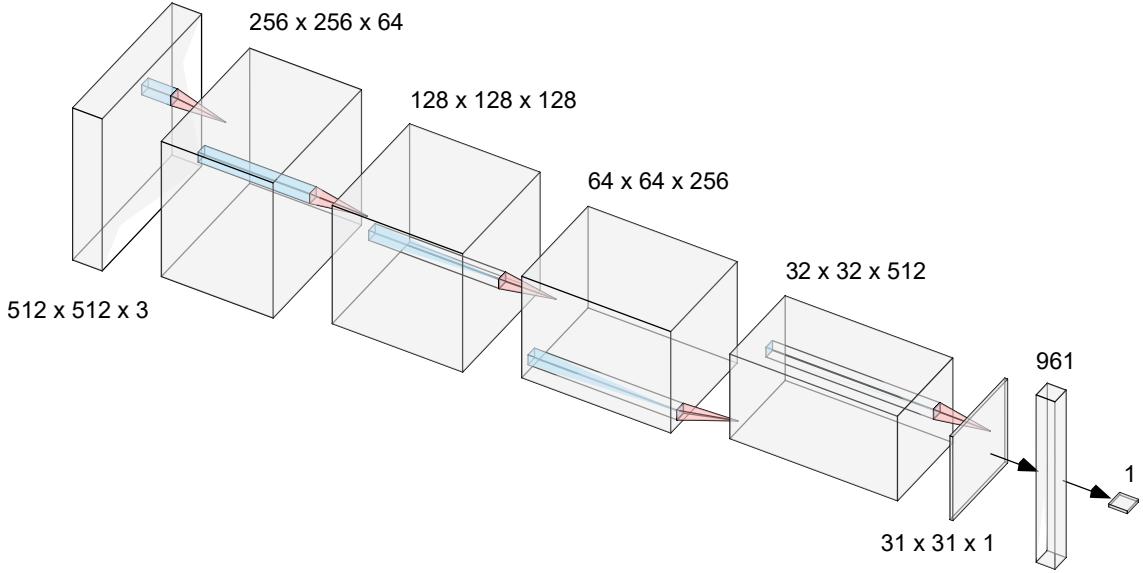


Figure 4.6: An overview of the architecture of the discriminator network of the CycleGAN. Note that there are four downsampling convolutional layers, followed by a single convolutional layer. Finally, a linear layer is applied to output the classification.

domains, we generated 2394 simulated label images for domain Y . Using this data, the CycleGAN was trained for 100 epochs with a batch size of 6. To help the training along, we augmented the data by random horizontally or vertically flipping the images. We used an Adam optimizer with a learning rate of 0.0002 with a decay of 0.1 every 10 epochs. The CycleGAN hyperparameter, λ , was set differently for two domains. For domain X , $\lambda = 10$ was set to penalize difference between $x \leftrightarrow F(G(x))$, and for domain Y , $\lambda = 30$ was set to penalize difference between $y \leftrightarrow G(F(y))$.

4.2.2 Pancreas

4.2.2.1 Gadermayr's Approach

We also applied Gadermayr's approach to segment the fat in the pancreas. The fat annotation generation was done similarly to the fat annotation of the liver tissue. Again, we began by counting 100 randomly selected pancreas tiles to approximate the prior knowledge and generated simulated labeled images. However, in the pancreas, the fat clusters are denser compared to those of the liver. Furthermore, each fat cell in the fat clusters is attached to each other, causing them to often have a squeezed shape. We divided fat annotation into fat cluster annotation and single fat annotation (fats that are not adjacent to other fats). Also, we had two experiments with different annotation models for the pancreas, see figure 4.7. All fat annotations are shown with a red color.

For every experiment, we began by creating the boundary of the tissue. The boundary

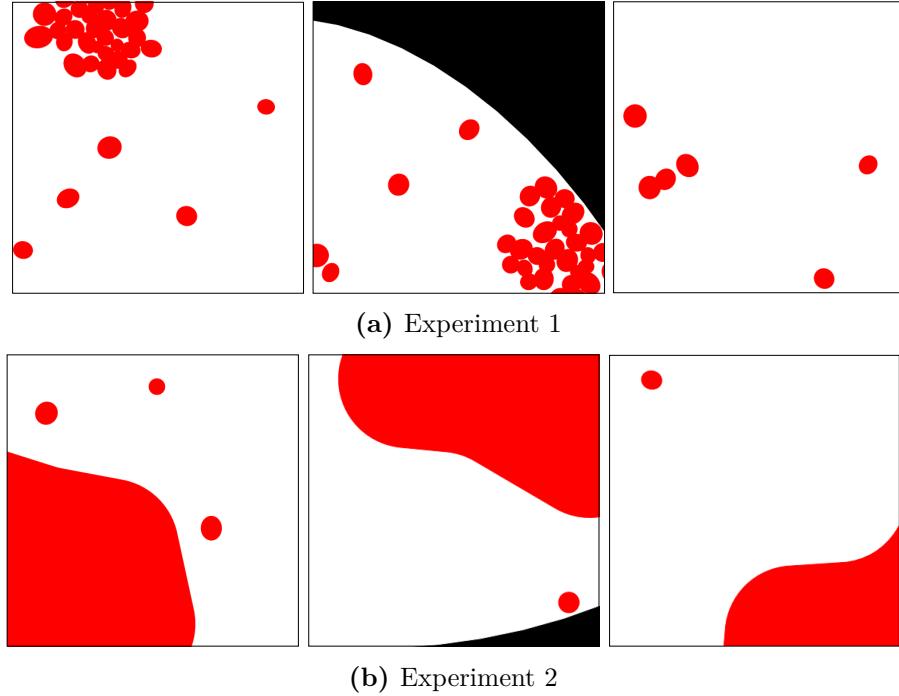


Figure 4.7: Example of the simulated label images with different annotation models. (a) Experiment 1. Example of simulated label images generated with only ellipses as the fat annotation. The fat cluster was generated with a number of ellipses adjacent to each other. (b) Experiment 2. Example of simulated label images generated with a large rounded polygon as the fat cluster, and ellipses as the single fats.

was generated as we did previously in the liver tissue, meaning it is present in 50% of the histology tiles. Then we generated single fat annotations. Note that we used the same annotations for the single fats in both experiments. The approximation from the counting of 100 tiles showed that the number of the single fats in one tile was approximately a Chi-square distribution with 2 degrees of freedom, limited to a maximum of 6. These single fats were generated with elliptical shapes. The process of generating ellipses was identical to the process in the liver in section 4.2.1. The single fats were placed uniformly across the image with a size sampled from a Gaussian distribution. The length of the ellipse's main axis, r_1 , follows $r_1 \sim \mathcal{N}(\mu_r = 10, \sigma_r = 5)$. The length of the sub axis, r_2 , was calculated as $r_2 = r_1 \cdot r_\delta$, where r_δ also follows Gaussian distribution, $r_\delta \sim \mathcal{N}(\mu_\delta = 1.1, \sigma_\delta = 0.1)$. Finally, the rotation parameter α of the ellipse was drawn from a uniform distribution between $[0, 2\pi]$.

After generating the single fats, we generated the fat clusters. The initial experiment (experiment 1) was done by generating fat clusters similar to those in the liver, meaning we generated fat clusters by smaller ellipses inside them with a distance parameter between the small ellipses. However, since fat clusters in pancreas are more dense, we used a shorter distance parameter of $d = 5$ (in the liver we used $d = 7$ or $d = 10$).

Additionally, we allowed the ellipses to overlap each other by 100 pixels to represent the squeezed cell shapes. The number of ellipses in each cluster follows $g_a \sim \mathcal{N}(\mu_{g_a} = 40, \sigma_{g_a} = 30)$. The size of each ellipse in the clusters was sampled identically to the method we used to sample the size of the single fats in the previous section.

However, due to unsatisfactory results, we instead attempted to generate the fat clusters as large rounded polygons to represent them better (experiment 2). Each polygon was randomly generated and placed in the simulated label image. To generate this rounded polygon, we first randomly placed the *center of the polygon*, (x, y) , within the boundary with an average radius, r_{avg} , of 300 pixels, see figure 4.8. The coordinate of each vertex of the polygon is calculated as:

$$(x + r_v \cos \theta, y + r_v \sin \theta) \quad (4.1)$$

where r_v was chosen from a Gaussian distribution $r_v \sim \mathcal{N}(r_{\text{avg}}, \sigma_{r_v})$, and θ was chosen from uniform distribution between $[0, 2\pi]$. Note that we clipped the value of r_v so it can be chosen within range of $[0, 2r_{\text{avg}}]$. σ_{r_v} is a parameter that controls how spiky is the generated polygon. We fixed this spikiness parameter to $\sigma_{r_v} = 0.6$. Once we got coordinates of each vertex, we produced polygon and rounded the vertexes. The polygon was generated and placed uniformly across the images without overlapping other annotations, see figure 4.7(b). In both experiment, the fat clusters in the pancreas were approximated to be appear with 15% probability.

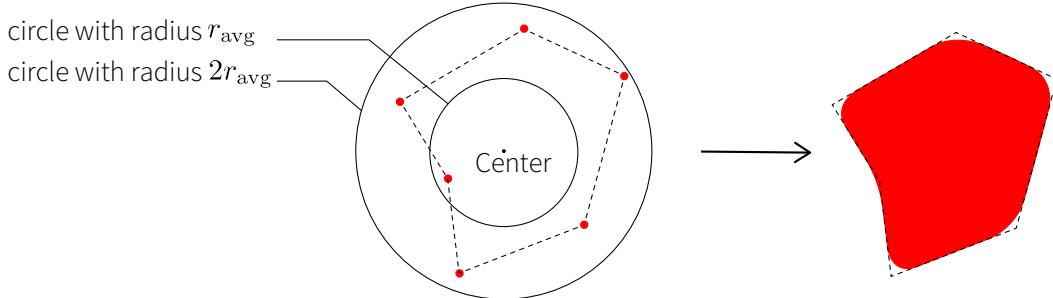


Figure 4.8: Figure illustrating how to randomly generating large rounded polygons. The vertexes around the center of the polygon are shown as red dots (left). These are then rounded to create a rounded polygon (right).

These generated annotations were used as our simulated label images, which we trained a CycleGAN to translate from and to. Domain X consisted of 3043 pancreas histology tiles, and domain Y consisted of 3731 simulated labels. The same CycleGAN architecture described previously in section 4.2.1.1 and shown in figure 4.5 and figure 4.6, was used. All experiments were run for 100 epochs using a batch size of 8 without any use of data augmentation. The network used an Adam optimizer with a learning rate of 0.0002 without a learning rate scheduler. The CycleGAN hyperparameter, λ , was set to 10 for both domains.

4.2.2.2 U-net Segmentation of Pancreas Tissue

Unfortunately, Gadermayr's CycleGAN approach was not successful when applied to the more complex pancreas tissue. Instead, we used a supervised approach, U-net. This required us to manually annotate the histology images since U-net requires aligned data for supervised training. We annotated 186 pancreas histology tiles, two examples are shown in figure 4.9.

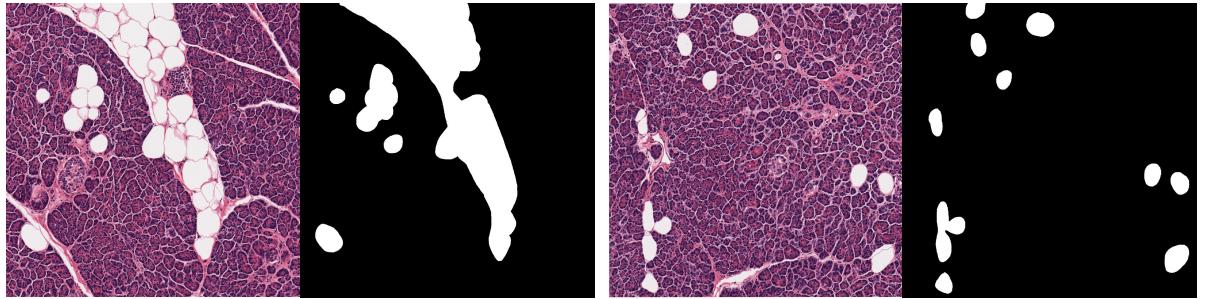


Figure 4.9: Examples of pancreas tile (left in each figure) and aligned manual segmentation (right in each figure).

The U-net architecture consisted of blocks of convolutional layers, see figure 4.10. Each block consists of two convolutional layers with kernel size 3, each followed by batch normalization and ReLU activation function. At the end of the U-net block, either a 2×2 max-pooling operation or a 2×2 up-sampling is applied. The U-net blocks using max-pooling at the end is called downsampling U-net block, and using up-sampling at the end is called upsampling U-net block. The U-net architecture begins with four downsampling U-net blocks and then followed by 4 upsampling U-net blocks. For each downsampling block, the tensor before applying max-pooling is concatenated to the input tensor of the corresponding upsampling block. In the end, two convolutional layers with kernel size 3 followed by batch normalization and ReLU activation function are applied again. Then a single channeled convolutional layer with kernel size 1 and a Sigmoid activation function is applied to output the predicted mask of the fat segmentation, see figure 4.10.

The U-net was trained for 1000 epochs with an Adam optimizer and a learning rate of 0.01, decaying by 0.1 every 20 epochs. Note that since we were only training on the 186 manually annotated images, we had very few data points. Thus, to help facilitate training, we would randomly flip the images either horizontally or vertically and apply random brightness control. Note that random brightness control was only applied to the histology tile, not the mask. This data augmentation would hopefully, create larger diversity in the training data. The model was trained with a evenly weighted loss between dice coefficient loss and binary cross-entropy loss, weighted evenly.

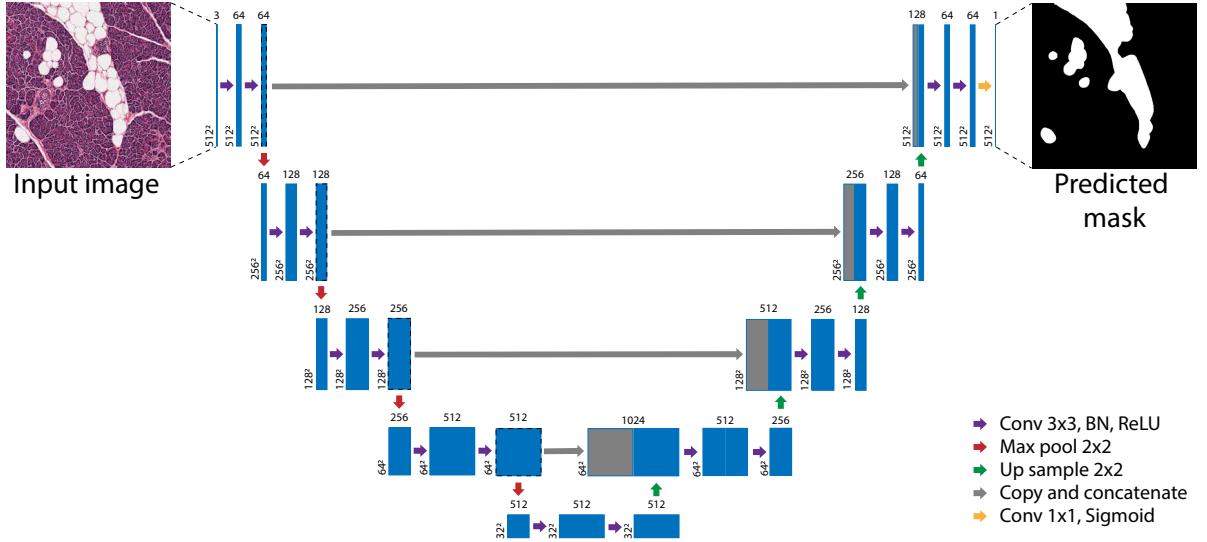


Figure 4.10: Diagram of the specific U-net architecture used in order to segment the pancreas tissue. The U-net consists of 4 downsampling and upsampling steps.

4.2.3 Calculating Fat Percentage from Segmentation Results

The fat percentage in the respective organs was calculated for each subject based on the segmentation results. However, since our models were trained on the histology tiles, we had to sum up the area of the fat and the area of cell from each tile to obtain a full fat percentage for one whole histology slide. This was done to exclude the background area. The cell area in each tile was calculated as the sum of the cell mask, which is a binary image. The cell mask has value as 1 in areas segmented as cell, and 0 in areas segmented as non-cell, meaning the sum of the cell mask indicates the number of pixels that are segmented as cell. The cell mask in each tile can be obtained with a short computer vision algorithm we developed which is described in appendix A.2. The cell mask can be seen in the second image from the left in figure 4.11. The drawback of the algorithm is that it consider the fat at the edge of the tile as background since it used the intensity of the image to get mask image. In the example shown in figure 4.11, we can observe this issue.

After we obtained segmentation results from the trained network, either CycleGAN or U-net, we made a fat mask with the segmentation results. This fat mask can be seen in the third image in figure 4.11. To reduce the incorrect segmentation between fat and background, we removed the non-cell area from the fat mask. An example is shown in figure 4.11, the non-cell area (black) from the cell mask is removed from the fat mask (white area in fat mask of figure 4.11). The pixels segmented as fat in a final fat mask can be seen as a red area in the last image from figure 4.11. Now the area of fat can be

calculated as a sum of the final fat mask.

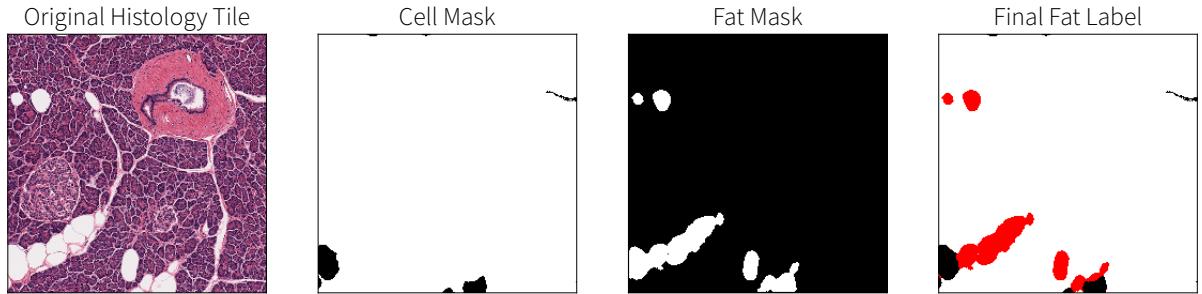


Figure 4.11: Figure illustrating how the final fat percentage is calculated. The cell mask is subtracted from the fat mask to create the final fat label.

Once we have both the area of fat and the cell area, we calculated the total fat percentage of the subject with results from each tiles as:

$$\text{Fat percentage} = \frac{\sum^{\# \text{tiles}} \text{Fat area}}{\sum^{\# \text{tiles}} \text{Cell area}}$$

With this method, we calculated the whole area of fat and cell for one subject without removing empty tiles manually. We then scaled the fat percentage we got from each subjects to have well-distributed results as:

$$\text{Scaled fat percentage} = \frac{\text{Fat percentage}}{\max(\text{Fat percentage}) - \min(\text{Fat percentage})} * 100$$

This scaled fat percentage will henceforth be referred to simply as fat percentage for the rest of the thesis.

4.3 Segmenting 4 Tissue Compartments in Pancreas

4.3.1 Naïvely applying Gadermayr's Approach

Our next goal was to segment all 4 tissue compartments in the pancreas (acinar, islet, fat, and ductal area) simultaneously in an unsupervised manner. An example made by manual annotation of all 4 pancreas tissue compartments can be seen in figure 1.1. Segmenting 4 tissue compartments simultaneously is a much more complex task than what Gadermayr et al.[1] did in their paper. We began by approximating the shape and the

distribution of each annotation by counting them in 100 tiles. Different annotation priors were applied for each tissue compartment depending on their shape and distribution.

Before generating each annotations, we had to generate the boundary of the tissue. The method of generating boundary is described in section 4.2.1. According to the counting, the boundary was generated with 50% probability. All the other annotations were generated within this boundary. Fat annotations were generated as ellipses as described in experiment 1 in section 4.2.2.1.

For islets, we assumed that the number of islets is approximated by a Chi-square distribution with 2 degrees of freedom. We created islets of elliptical shape. Thus, individual islets were generated using the same method as fat annotation, which generates ellipses. However, islets do not form a cluster, so they were placed uniformly across the image. The size of an islet was also approximated by a Gaussian distribution. The length of the main axis of the ellipse, r_1 , follows $r_1 \sim \mathcal{N}(\mu_r = 40, \sigma_r = 15)$. The length of sub axis, r_2 , was calculated as $r_2 = r_1 \cdot r_\delta$. The r_δ also follows Gaussian distribution, $r_\delta \sim \mathcal{N}(\mu_\delta = 1.1, \sigma_\delta = 0.2)$. Finally, the rotation parameter, α , of the ellipse was drawn from a uniform distribution between $[0, 2\pi]$. The islet annotations are depicted in blue. Note that the size of the ellipse for islet annotations are larger than that of fat, however, they are less elongated.

The last remaining annotation is the duct annotation, which is the most complex to generate. Ducts are not always circular in shape and vary greatly in size. Some ducts are smaller than a single fat, and some ducts are so large that they fill up the entire tile. Thus, we classified the ducts into three groups - small, medium, and large ducts. Since small ducts all have an elliptical shape, we generated small ducts similar to islets and fats. The number of small ducts follows the Chi-square distribution with 2 degrees of freedom up to a maximum of 6 per tile. The size of the small ducts is approximated by a Gaussian distribution. The length of the main axis of ellipse, r_1 , follows $r_1 \sim \mathcal{N}(\mu_r = 10, \sigma_r = 5)$. The length of sub axis, r_2 , was calculated as $r_2 = r_1 \cdot r_\delta$. The r_δ also follows Gaussian distribution, $r_\delta \sim \mathcal{N}(\mu_\delta = 1.1, \sigma_\delta = 0.5)$. This means that the small ducts can be more elongated than islets and fats.

Medium ducts and large ducts are not always circular in shape. So we generated polygons with 6 rounded vertexes, similar to how we generated cluster of fat in section 4.2.2, see figure 4.8. The number of the medium ducts was 0 for 70% of the histology tiles, 1 for 20% of the histology tiles, and 2 for the remaining 10% of the histology tiles. The parameter r_{avg} that controls the average size of the polygon, which was set to 100 (pixel) for the medium duct. Large ducts were only present in 20% of the histology tiles. The parameter r_{avg} that controls the size of the polygon was set to 300 (pixel) for large duct. All ducts were also generated and placed uniformly across the images, and the ducts are marked in green in our label images. Full simulated label images generated with the above annotation model can be seen in figure 4.12.

After generating the simulated label images, we trained a CycleGAN to translate between

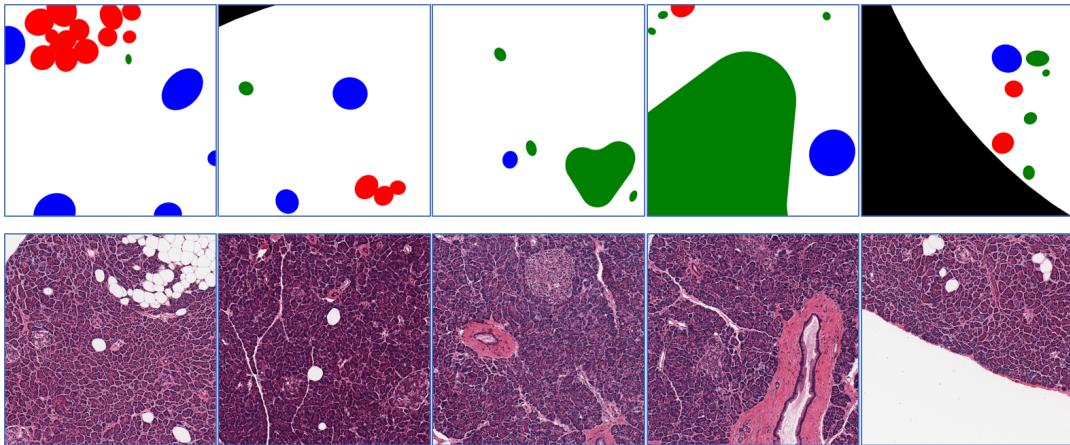


Figure 4.12: Simulated label images generated with all 4 tissue compartments in the pancreas. First row shows the full simulated label images generated and the second row shows original histology images to compare each annotation and actual tissue compartments. In label images, red color indicates fat, blue color indicates islet, green color indicates duct, and white color indicates acinar. The background is colored in black.

the domain X with 3043 tile images and domain Y with 3731 simulated label images. We used the same network architecture as previously specified in section 4.2.1.1. The CycleGAN is trained for 100 epochs with a batch size of 8. An Adam optimizer is used with a learning rate of 0.0002 without a learning rate scheduler. The CycleGAN hyperparameter, λ , is set to 10 for both domains.

4.3.2 Train CycleGAN with Label Images from U-net Results

Since the results from training CycleGAN using simulated label images with our own annotation model were not translating histology images to label images in a satisfactory manner, we decided to change this approach slightly. Since the network often confused the different annotations with each other, we added more details would help the network distinguish each annotation properly. To achieve this, we decided to generate more realistic label images using a U-net. We also added annotations for nuclei and vessel to the label image in addition to the annotations for acinar, islet, fat, and duct.

We trained multiple U-nets to annotate each individual tissue compartment in order to get more precise and more realistic annotations. The results from each U-net were then combined into one label image and used for our CycleGAN training, see figure 4.13. We hoped that more realistic labels would resemble the real histology images and thus allow the CycleGAN to segment the tissue correctly. We trained separate U-nets on 244 manually labeled images to get duct, islet, and vessel annotations. The previous successful U-net fat annotations was used for fat. All U-net architectures and training

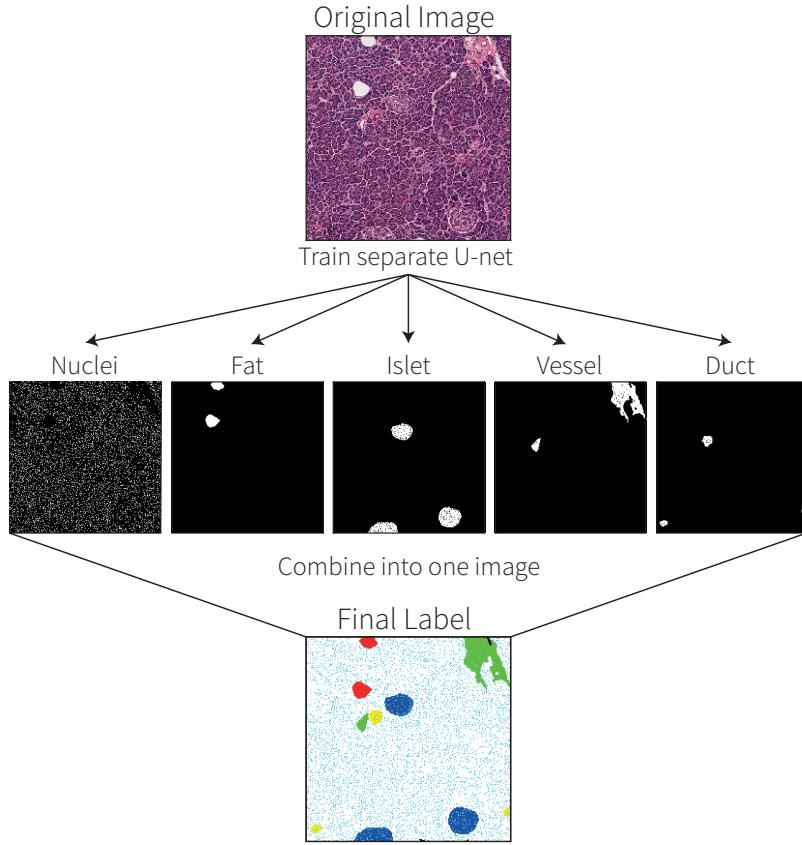


Figure 4.13: How to generate label images for CycleGAN training. For each tissue compartment, a separate U-net was trained to segment the specific tissue compartment. These segmentation results were then overlaid each other into one label image.

details followed the previous description, section 4.2.2.2.

Beyond having more realistic shapes for the different tissue compartments, we chose to add nuclei annotation to the label images. This was done because nuclei annotations allow the network to distinguish between fat and ducts since fat does not have any nuclei in it. Furthermore, the ducts have a unique pattern of nuclei (circular shaped series of nuclei in the middle). It also allows the network to distinguish between islets and acinar cells, since islets have slightly fewer nuclei compared to acinar cells. A U-net was used to add these nuclei annotations. Since U-net is a supervised approach, we had to first annotate images with nuclei annotations. However, unlike other tissue compartments, it is difficult to annotate each cell nucleus manually. So instead, we used a blob detection algorithm to roughly annotate nuclei, see figure 4.14. We used the *difference of Gaussian (DOG)* to detect the blobs [26]. We used 186 pancreas tiles with aligned rough nuclei annotation to train our U-net. The architecture and the training details are again the same as we used for pancreas fat training, section 4.2.2.2.

Once we had all the U-net models for the fat, duct, islet, vessel, and nuclei, we made

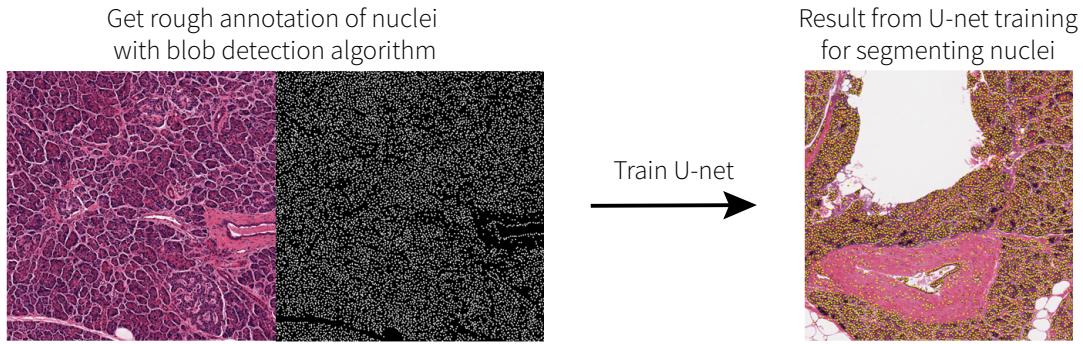


Figure 4.14: Figure displaying results of blob detection algorithm and the resulting segmentation label from U-net segmentation.

new simulated label images with the segmentation results from those U-net models. Full label image can be seen in figure 4.15.

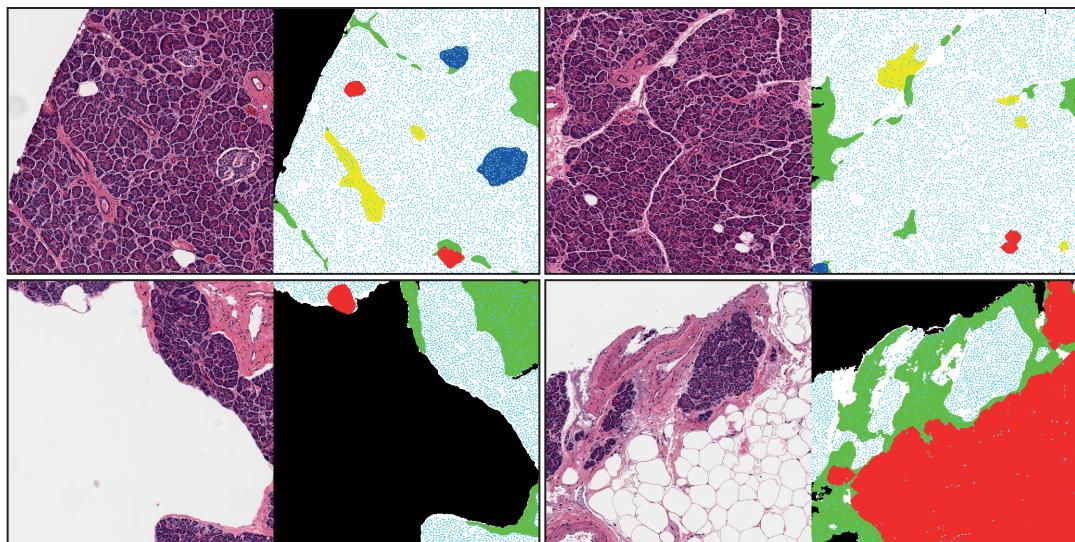


Figure 4.15: Example of original histology image and their corresponding U-net segmentation results. Each of the color in the label image indicates : background-black, acinar-white, fat-red, vessel-green, duct-yellow, islet-blue, and nuclei-cyan .

With these new and more realistic label images from the U-nets as domain Y , we once again trained a CycleGAN. The network architecture and training details followed the liver fat segmentation CycleGAN, described in section 4.2.1.1. The only difference is the number of residual blocks. Here, we used 9 residual blocks instead of 6. We expected this more complex task to require more model flexibility, thus more residual blocks. The number of tiles in domain X and number of label images in domain Y were both 3041. The CycleGAN was trained for 100 epochs with a batch size of 6. The learning rate was set to 0.00002 with no learning rate scheduler. The CycleGAN hyper parameter λ was set to 10 for both domains.

CHAPTER 5

Results

5.1 Fat Segmentation in Pancreas and Liver

5.1.1 CycleGAN Successfully Segments Fat in Liver

To segment the fat in the liver tissue, we applied Gadermayr's approach, which involves generating simulated label images and training a CycleGAN. The learning curves during the training is shown in figure 5.1. The first plot in figure 5.1 depicted two discriminator losses for each domain, and the second plot in figure 5.1 contains sum of two cycle consistency losses and two adversarial losses described in the CycleGAN theory section, 2.4. If discriminator loss is 0, it means that the discriminator can distinguish if the image is from original data, or generated image from the generator. The results from the successfully trained CycleGAN can be seen in figure 5.2. The model we chose to segment the fat is the one from epoch 66. This was based on qualitative screening since the CycleGAN loss curve does not carry significant meaning. The original images and label images shown in figure 5.2 are all from the test set. As we can see, the mapping from image \rightarrow label learned to segment the fat (see figure 5.2(a)). However, the opposite mapping from label \rightarrow image did not learn enough to reconstruct even close to all the details of the original image (see figure 5.2(b)).

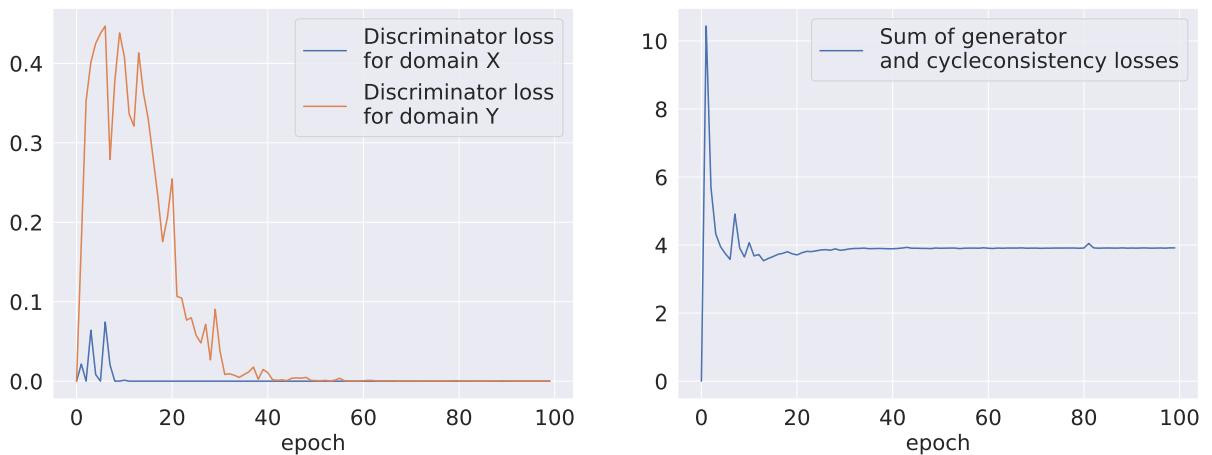


Figure 5.1: The CycleGAN training curves for liver segmentation.

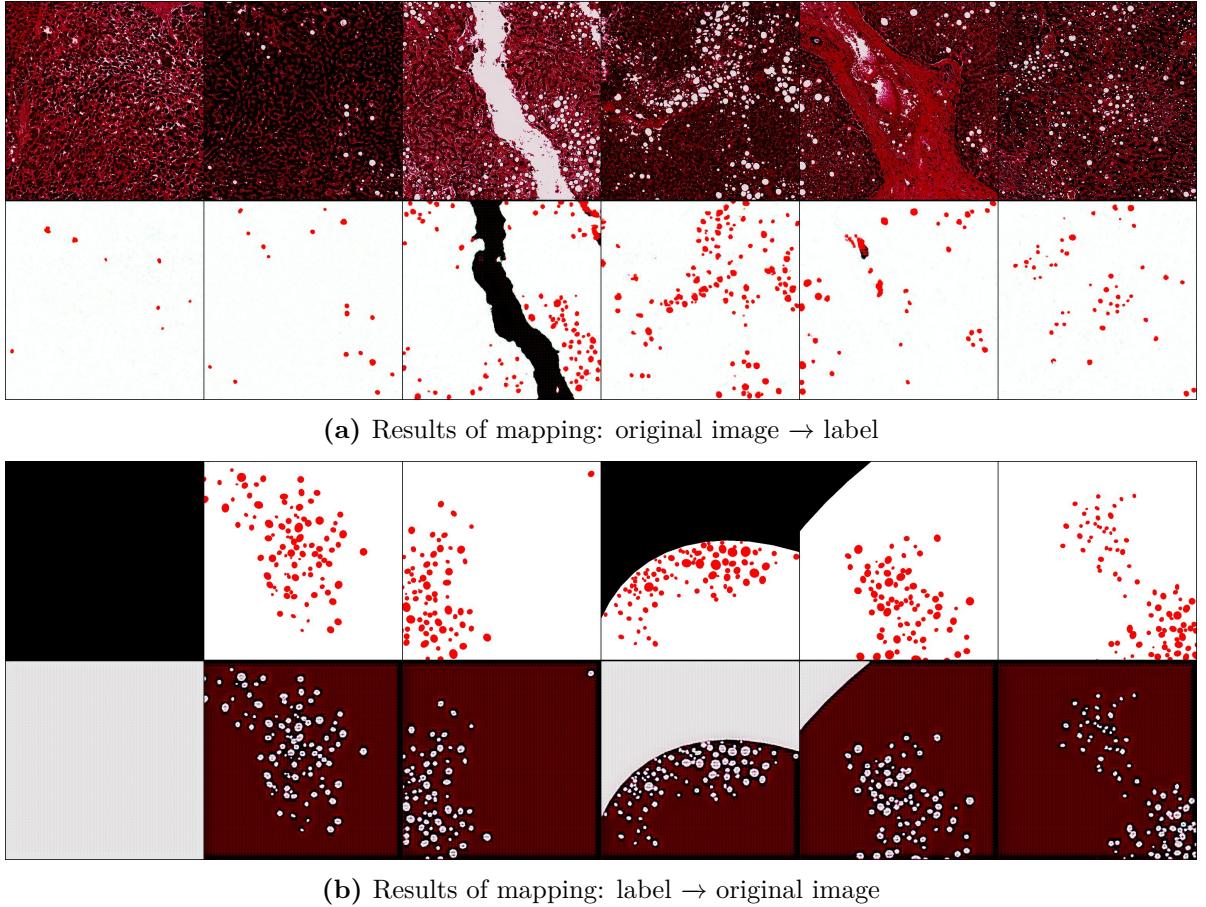


Figure 5.2: Results from CycleGAN fat segmentation of liver tissue. (a) First row is original tile image, and second row is the segmentation results of each tile image. The fat is annotated as red, the cell area is annotated as white, and the background as black. The segmentation is rather successful. (b) First row is automatically generated label image, and second row is the results of reconstruction of original image from label image. The reconstruction results are quite poor.

Using the chosen best performing model, we applied the segmentation across the whole histology slide for all 594 subjects. The segmentation of high-fat, medium-fat, and low-fat subjects were chosen and shown in figure 5.3(a), 5.3(b), and 5.3(c). Cell area is marked in blue and the fat is shown in red (overlaid on the actual whole slide image).

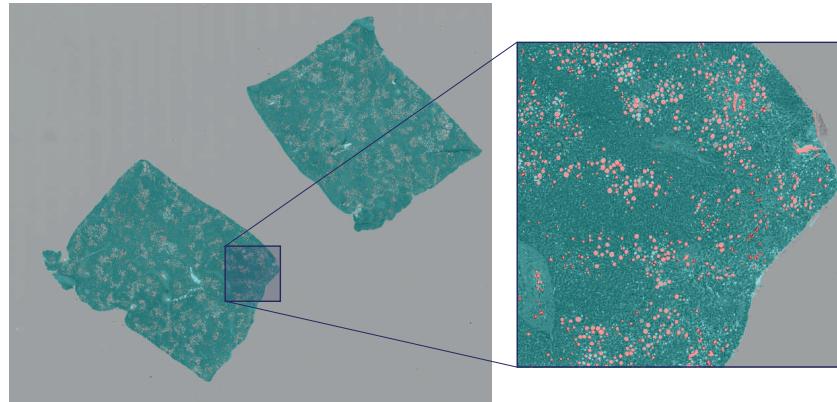
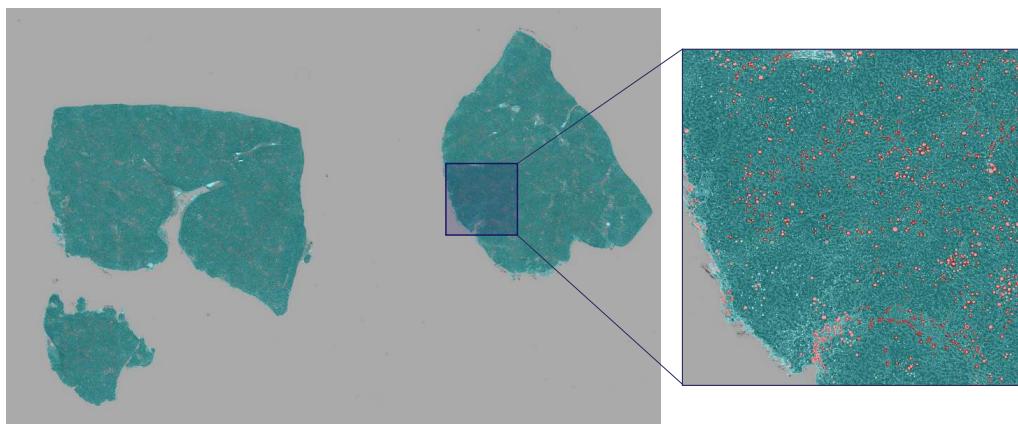
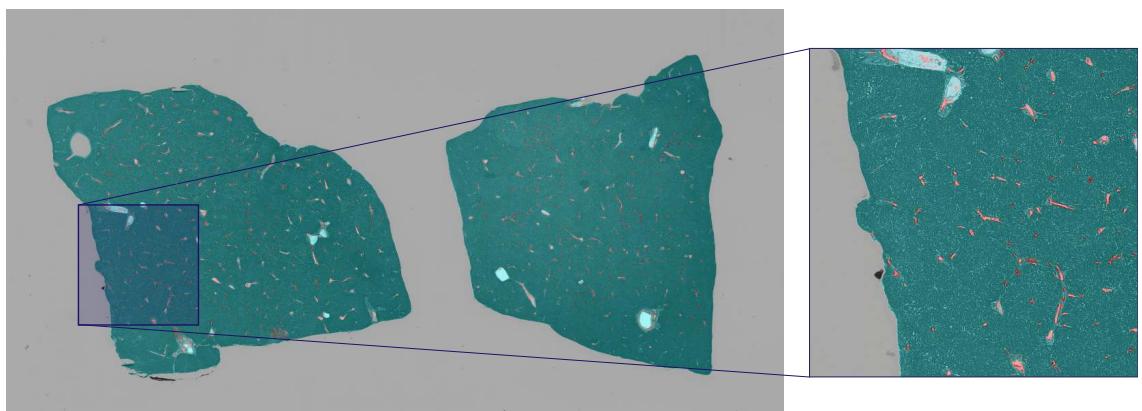
(a) The whole liver histology slide of *high fat* subject(b) The whole liver histology slide of *medium fat* subject(c) The whole liver histology slide of *low fat* subject

Figure 5.3: The whole liver histology slide of each high, medium, and low fat subject overlaid with the CycleGAN segmentation results. The blue area is non-fat and the fat area is shown in red.

5.1.2 Pancreas

5.1.2.1 Gadermayr's Approach Fails to Segment Fat in Pancreas

Next, we applied Gadermayr's approach to a more complex data set, the pancreatic tissue. In this section, we will only focus on the mapping: original image \rightarrow label, the opposite mapping is omitted from the following results. The results from the two pancreas segmentation experiments described in section 4.2.2.1 is presented below.

Experiment 1

The complexity of the pancreas segmentation task primarily stems from having to segment both clusters of fat and single fats. These were similar in the liver, but in the pancreas, they are not very similar. In our initial experiment, we used ellipses to mark both in the simulated labels. Using these label images as the Y domain, the model consistently misinterpreted the pancreatic tissue. In figure 5.4, the fat segmentation is shown in red, background in black, and other cell tissue in white. Figure 5.4(a) highlight an example of a model that only recognizes single fats, and not the clusters of fat. The model is also frequently confusing larger clusters of white fat tissue with the background since they share a white color. In figure 5.4(b), we observe confusion between ducts/vessels and fat clusters. However, the model did show some promises in figure 5.4(c), where it does recognize some fat clusters.

Experiment 2

Looking at figure 5.4(b), we can see that annotations for the fat clusters often do not necessarily take circular shapes. Thus, in the second experiment, we changed the shape of the simulated fat annotations for fat clusters into 6 edged rounded polygons. Note that annotations for single fats remained elliptical. We hoped these more complex shapes would allow the network to distinguish ducts and vessels from fat clusters. The results from the experiment 2 can be seen in figures 5.5(a) and 5.5(b). It appears like the network recognizes the fat cluster as we expected. However, during the training, the single fats are no longer recognized very consistently (figure 5.5(a)) and the fat cluster is still often confused with the duct or vessel (figure 5.5(b)).

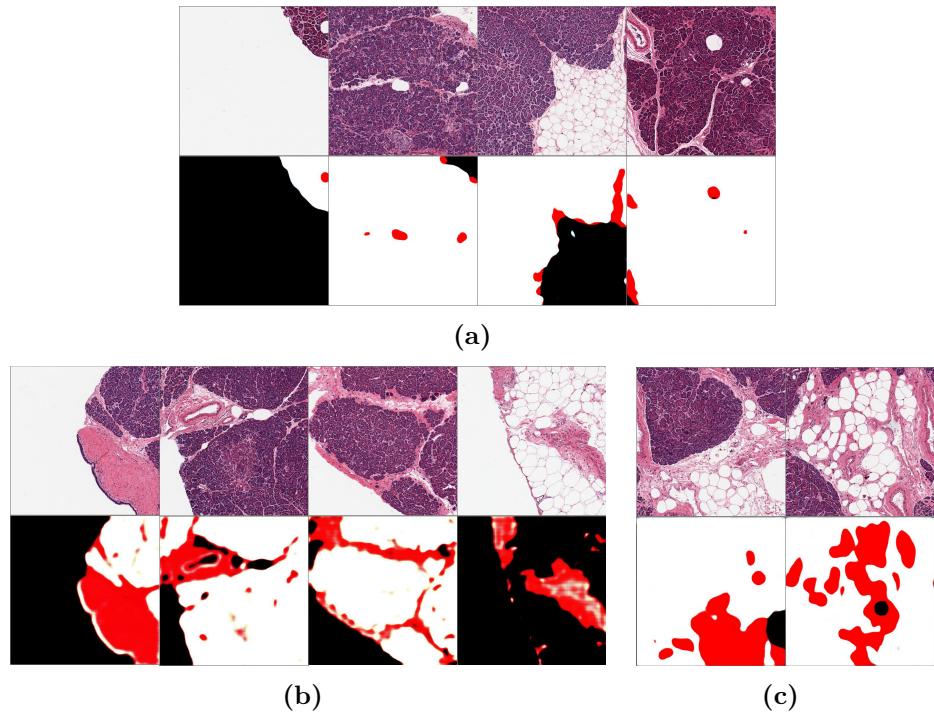


Figure 5.4: Results from Experiment 1, showing the results from the mapping: original image → label. (a) The network only learned single fat, not the fat cluster. It learned fat cluster as background. (b) The network confused duct or vessel as fat cluster. (c) We can see there is possibility for the network to learn fat cluster.

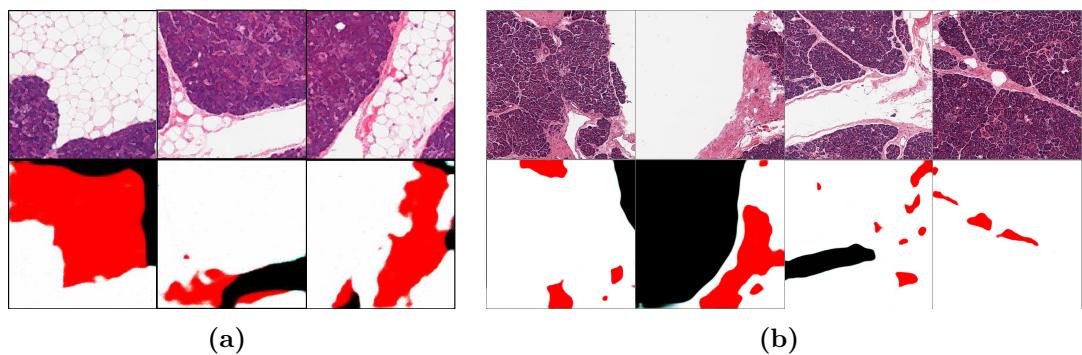


Figure 5.5: Results from Experiment 2, showing the results from the mapping: original image → label. (a) It shows the network only learns cluster of fat, but does not learn single fat. (b) The network confused ductal area or vessel as fat cluster.

5.1.2.2 U-net Successfully Segments Fat in Pancreas

Since the results with Gadermayr's approach were not satisfactory, we trained a U-net to segment the fat in the pancreas. U-net is a powerful deep learning framework for segmenting biomedical images even when only a small amount of data is available. Unlike CycleGAN, U-net has a meaningful loss measure, our losses are shown in figure 5.6. The training loss is consistently decreasing, indicating the model is learning properly. The test loss is often smaller than the training loss during initial training (roughly before the epoch 200). This is because U-net segments the entire histology tile as non-fat (0 in a binary mask) in the early epochs since the majority of the histology tiles are non-fat, making the loss decrease the fastest. This makes the test loss smaller than the training loss in the early epochs, simply due to the test set randomly having more non-fat area. Note that the test error swings widely, due to the very low number of samples in our test set. Labeling the histology tiles was very time consuming, thus only 186 images were labeled, and only 18 of these were used in the test set. The best performing U-net model was chosen based on the combination of test loss and training loss, which is from last epoch 1000. The results from this model can be seen in figure 5.7. Note that all the images in the figure are from the test set. We can observe that the U-net successfully segments both the fat clusters and the single fats of the pancreas.

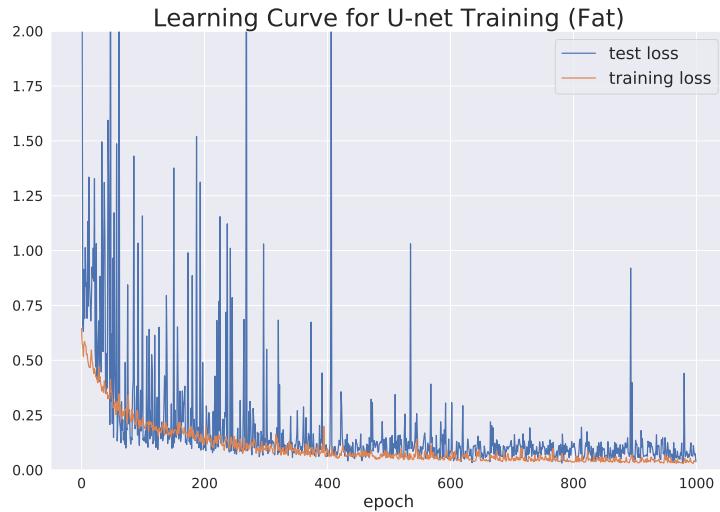


Figure 5.6: Learning curve of U-net.

The results of whole slide segmentation of one subject can be seen in figure 5.8, 5.9, and 5.10. Similar to previously, section 5.1.1, we chose a high-fat, a medium-fat, and a low-fat subject to demonstrate the segmentation. A color overlay has been added on top to highlight the results. The fat is indicated in red and the cell area is indicate in blue. U-net successfully segments all fat in the pancreas tissue.

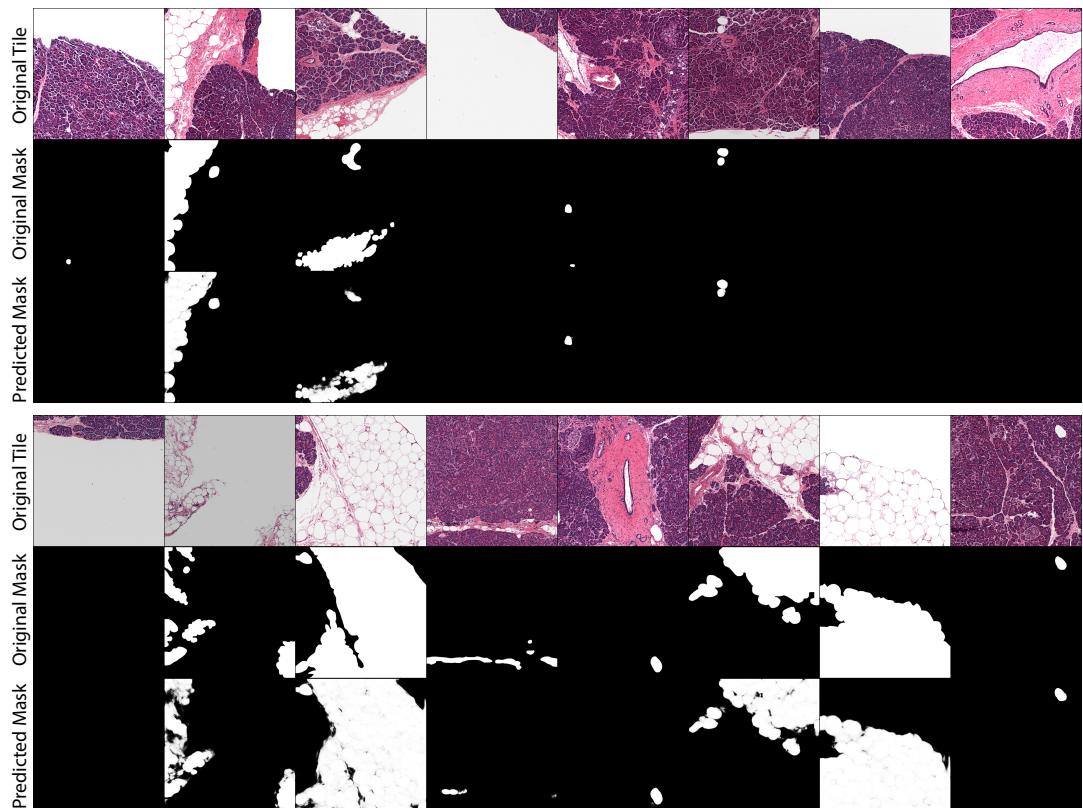


Figure 5.7: Training results from U-net to segment fat in pancreas tissue. The model successfully segments the fat tissue. First row is the original tile images we wanted to segment, second row is the ground truth segmentation, and third row is the predicted segmentation with trained U-net.

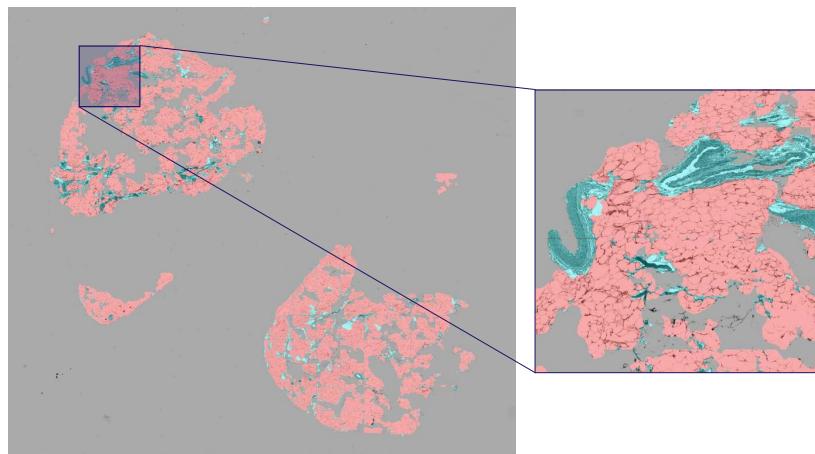


Figure 5.8: The whole pancreas histology slide of *high fat* subject overlaid with the CycleGAN segmentation results. The blue area is non-fat and the fat area is shown in red.

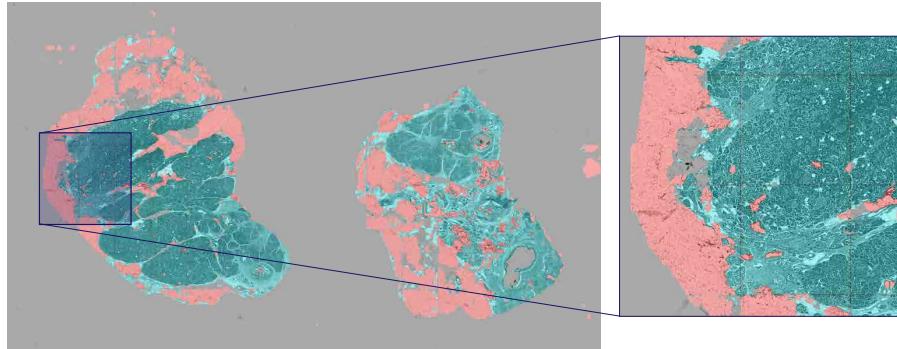


Figure 5.9: The whole pancreas histology slide of *medium fat* subject overlaid with the CycleGAN segmentation results. The blue area is non-fat and the fat area is shown in red.

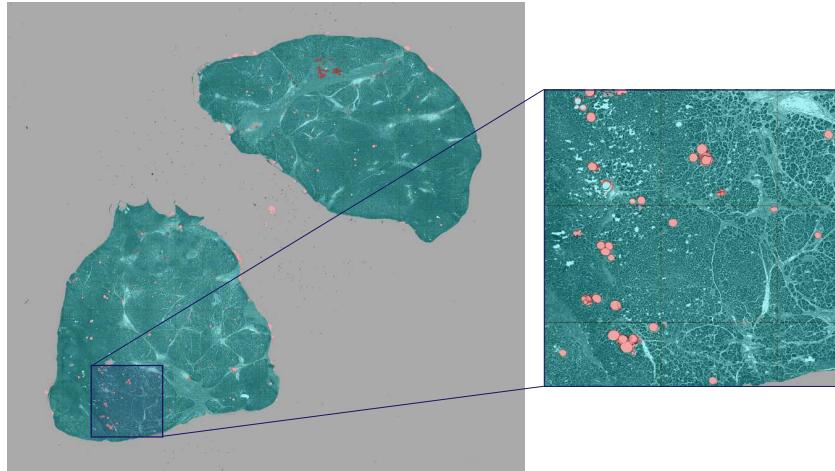


Figure 5.10: The whole pancreas histology slide of *low fat* subject overlaid with the CycleGAN segmentation results. The blue area is non-fat and the fat area is shown in red.

5.1.3 Analysis of Fat Percentage in Pancreas and Liver

From training a CycleGAN on the liver tissue and a U-net on the pancreas tissue, we now have two models that are able to segment the fat in the liver and the pancreas respectively. We applied these two models to all 594 subjects in GTEx with histology images available for both liver and pancreas. From these segmented images, we calculated the percentage of fat within each subject as described in methods section 4.2.3. The scatter plot of the predicted results can be seen in figure 5.11(a), showing that there is a low correlation between fat percentage in the liver and the pancreas (Pearson correlation coefficient = 0.09). Additionally, we divided the subjects into two groups, low-fat group and high-fat group. Subjects in the top 10% fat percentage in their respective organs are considered to be high-fat group and the bottom 90% are considered to be low-fat group, see figure

5.11(b). The high-fat group is colored under the graph in figure 5.11(b), and the dotted line denotes the 90% cutoff. The cutoff was decided since there is a second high bump around the 90% in the distribution of both liver and pancreas.

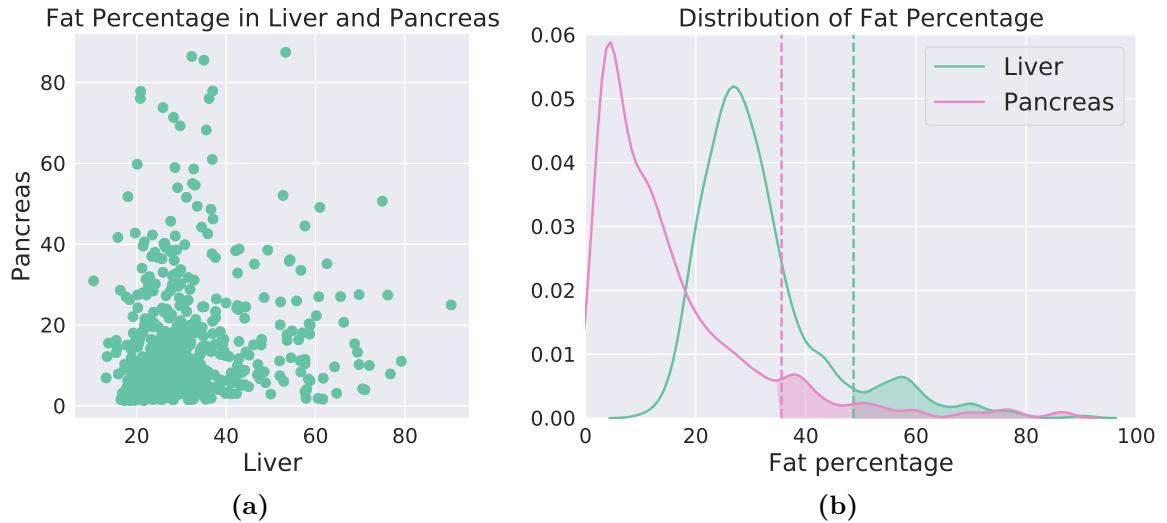


Figure 5.11: (a) Scatter plot of the predicted fat percentage in the liver and the pancreas. The x -axis indicates the fat percentage of the liver and the y -axis indicates the fat percentage of the pancreas. (b) Distribution plot of the predicted fat percentage in the liver and the pancreas, showing each high-fat group and low-fat group.

After dividing the high-fat group and the low-fat group, we investigated the relation between fat percentage and gender. We investigated the gender ratio within the two high-fat groups. The gender distribution was similar to the overall population distribution, see appendix A.3. This indicates that gender does not carry a significant role in pancreas and liver fat percentage.

We also investigated the relation between age and the fat percentage in each organ in figure 5.12. From this, we can conclude that the elderly population tends to have higher fat percentage than the younger population. This was observed in both liver and pancreas and to be expected. However, the 50-59 age group in the liver tends to have the highest liver values. We assumed this is due to randomness and not a special trend.

The GTEx metadata contains all the pathology categories of each subject, which we assumed to be the diseases that each subject is suffering from. In figure 5.13(a), we can compare the number of diseases depending on if they are in the high-fat or low-fat group. There is no apparent difference between high fat and low-fat group in both the liver and the pancreas. However, in figure 5.13(b), we can see the number of diseases does appear to be correlated with age.

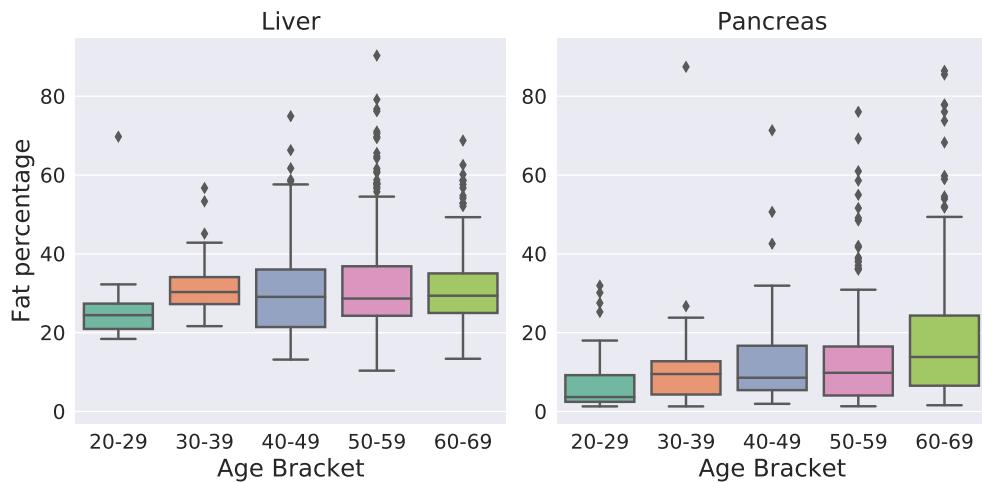


Figure 5.12: Box plot of predicted fat percentage in the liver and the pancreas according to age.

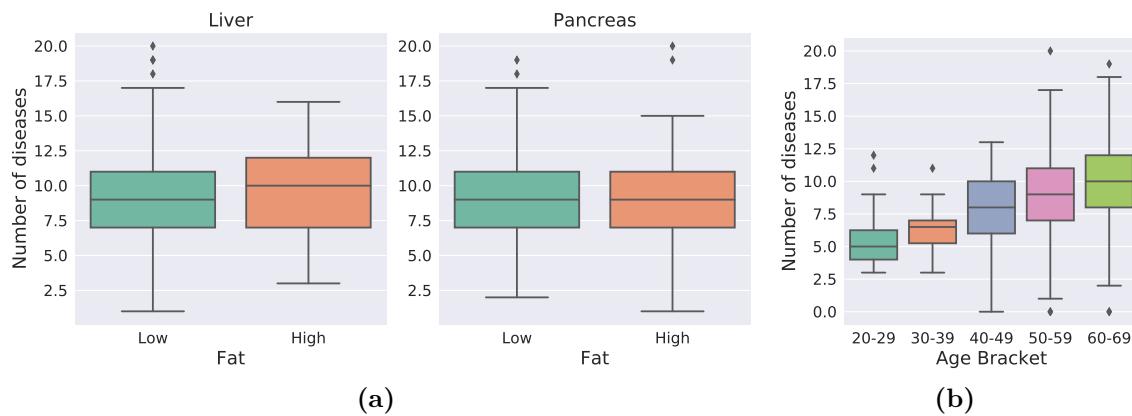


Figure 5.13: (a) Box plots show number of disease according to the high- and low- fat group in the liver and the pancreas. (b) Box plot shows number of diseases according to the age.

5.2 Segmenting 4 Tissue Compartments in Pancreas

Next, we were interested in segmenting multiple tissue areas in the pancreas simultaneously. This time we are segmenting the acinar area (normal pancreas cells), the pancreatic ductal area (duct) that contains pancreatic juice, the islets that secrete hormones such as insulin, and again the fat area.

5.2.1 Naïve Gadermayr Approach Fails Segmentation Task

We initially attempted to apply Gadermayr’s approach once again. We trained a CycleGAN model to translate between the X domain, containing original histology tile images, and the Y domain, which contains the simpler simulated label images described in section 4.3.1. This high complexity segmentation task turns out to be too difficult for the Gadermayr approach, see figures 5.14, 5.15, and 5.16. In these figures, we only showed results from the original image \rightarrow label mapping, and the reverse mapping is omitted. In the label images, the acinar area should be segmented as white color, the fat as red color, the ductal area as green color, the islet as blue color, and the background as black color.

During training, we observed that the network alternately learns acinar, ductal area, fat, and vessel randomly. For instance, in figure 5.14, the ductal area is recognized as islet and the vessel area is recognized as duct. Also in figure 5.15 and figure 5.16, clusters of fat are identified as ductal area. The vessel/ductal area is identified to be islet in figure 5.15, and vessel is identified to be fat in figure 5.16. Moreover, islets were never recognized very well, the network almost always identified the islets as acinar cells. We assumed this is because the color of the islet is similar to that of acinar cells.

Furthermore, single fats and clusters of fat were never recognized at the same time. In figure 5.14, only single fats are recognized and the clusters of fat are determined to be background. The figure 5.15 and figure 5.16 shows example of models that are not able to recognize both single fats and clusters of fat as fat simultaneously.

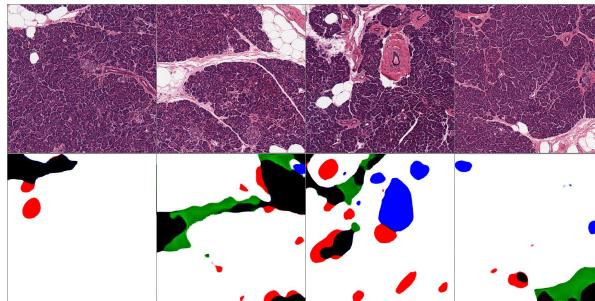


Figure 5.14: Results during training CycleGAN. The vessel is learned as the ductal area (green label) and the ductal area is learned as islet (blue label). We can see that single fat is segmented better than other labels, but still, the cluster is learned as background.

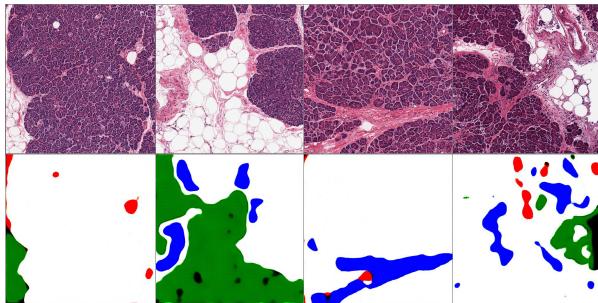


Figure 5.15: Result during training CycleGAN. We can see that the cluster of fat is learned as ductal area (green label). Ductal area and vessel is learned as islet (blue label). This case was the most common case that happened during the training.

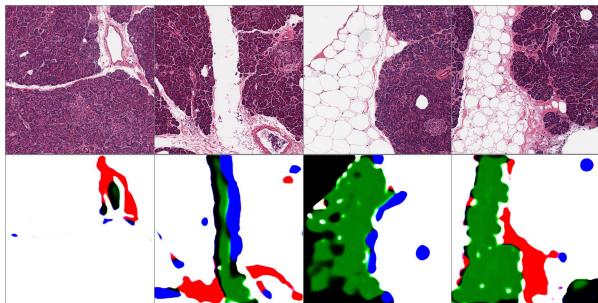


Figure 5.16: Result during training CycleGAN. The fat cluster is learned as ductal area (green label), and the vessel is learned as fat (red label). The single fat and some of the white area is learned as islet (blue label).

5.2.2 Gadermayr’s Approach with Labels Generated with U-net Fails Segmentation Task

We theorized that the previous results were disappointing due to the lack of details in the simulated label images. Thus, we decided to annotate each of the previous four labels using four individual U-nets. These annotations would be used as the simulated label images, providing a lot more realistic shapes for each tissue compartments. Additionally, we decided to add annotations of cell nuclei similar to what Gadermayr et al. did in their paper. The cell nucleus is a membrane-bound organelle that contains most of the cell genome [27]. The motivation behind adding the cell nuclei annotations is to allow the network to distinguish the different tissue compartments. After including nuclei, it is no longer only the slight color difference separating islets and acinar cells, since islets have a lower density of cell nuclei they are more easily separable. Furthermore, ductal area is different from vessel and fat, since it has a unique circular pattern of nuclei in the middle. The process of generating these new realistic simulated label images is described in detail in section 4.3.2.

Unfortunately, this method did not improve our performance significantly. The CycleGAN loss curves are shown in figure 5.17. The first plot shows losses from each discriminator, and the second plot shows sum of the generator losses and two cycle consistency losses from each mapping. The results from mapping G : original image \rightarrow label can be seen in 5.18, 5.19, and 5.20. During the training process, we again observed that the network randomly learned each of the tissue compartments, similar to what we observed without the nuclei annotation previously in section 5.2. In figure 5.18 and 5.19, the ductal area is recognized as fat. In figure 5.19, the fat is recognized as vessel. Also, we observed that the fat cluster is not recognized at all in figure 5.18 and 5.20. And again, the CycleGAN never learned to recognize both the cluster of fat and the single fat as fat simultaneously.

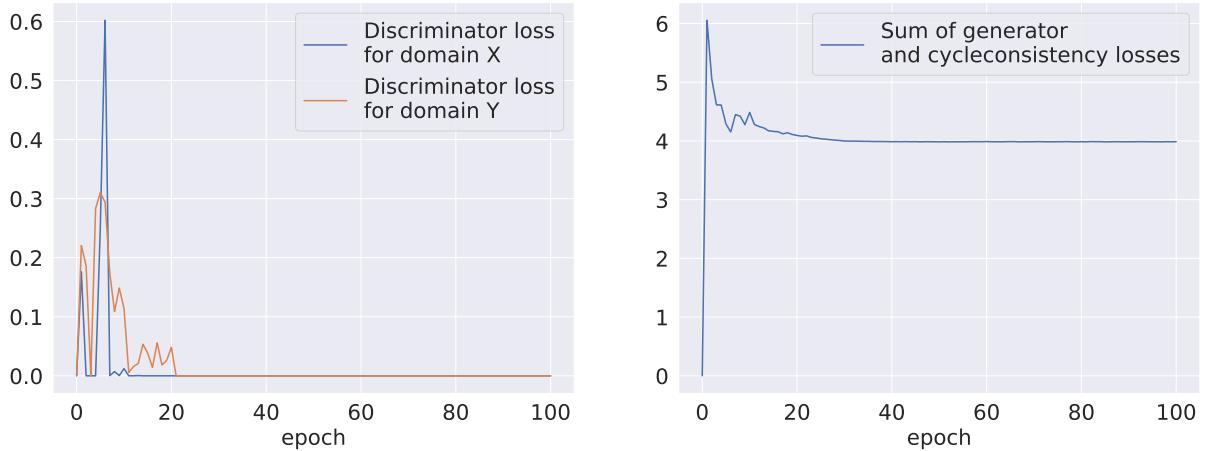


Figure 5.17: The CycleGAN training curves for pancreas segmentation using labels from U-net results.

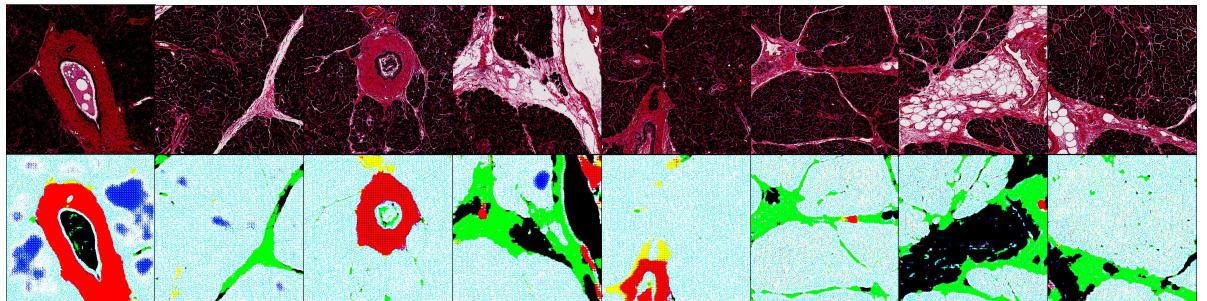


Figure 5.18: Results from mapping G : original image \rightarrow label during training of the CycleGAN with labels from U-net. The ductal area is learned as fat (red label). The vessel (green label) is learned fairly well. Yet, the fat is not learned at all, and just recognized as the background.

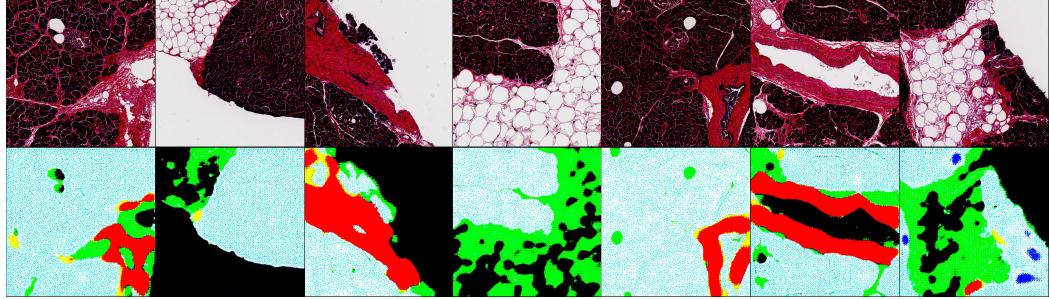


Figure 5.19: Results from mapping $G : \text{original image} \rightarrow \text{label}$ during training of the CycleGAN with labels from U-net. The ductal area is learned as fat (red label) and the fat is learned as vessel (green label).

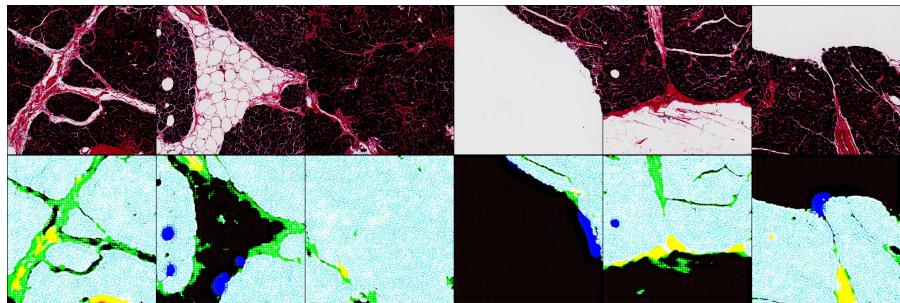


Figure 5.20: Results from mapping $G : \text{original image} \rightarrow \text{label}$ during training of the CycleGAN with labels from U-net. The vessel (green label) and the ductal area (yellow label) are learned fairly good. Single fat is learned as islet (blue label), and fat cluster is recognized as background.

The reconstructed images from mapping $F : \text{label} \rightarrow \text{original image}$ can be seen in 5.21. Compared to the reconstructed images from all the previous results using the CycleGAN, we can observe that the quality is much better. The acinar area now has more details including nuclei, and in the reconstruction of the ductal area (yellow-colored annotations in the label image), we can observe the distinct series of nuclei in the middle of the duct. We can also deduce how the network segmented each of the annotations in the label images with corresponding reconstruction. For example, we can see that the annotation of islet (blue color) in the label images are not segmented properly since the reconstruction of it has white color, which actual islets never have. Additionally, annotation of fat (red color) is not recognized as fat since the reconstruction of it does not have white color of actual fat.

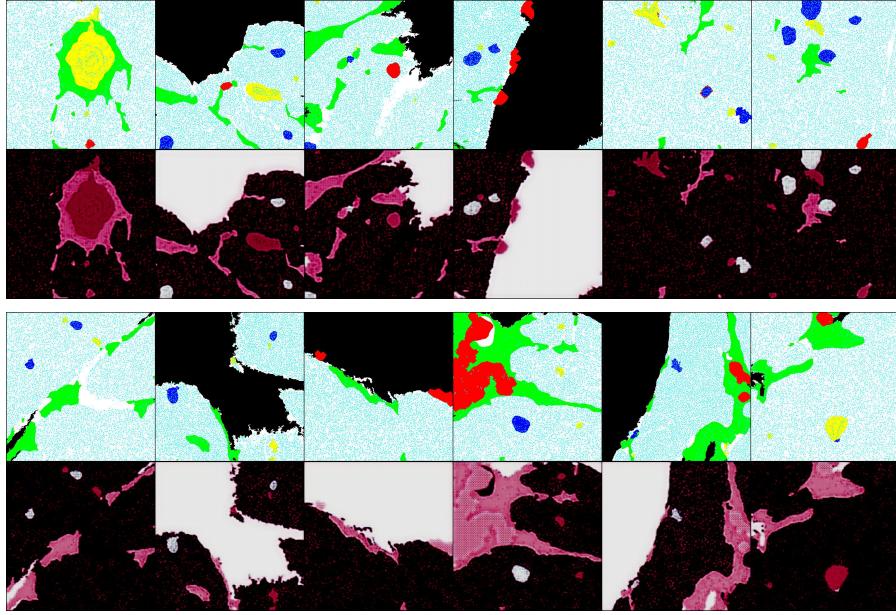


Figure 5.21: Result of reconstructed images from the mapping $F : \text{label} \rightarrow$ original image during training of the CycleGAN with labels from U-net.

5.2.3 Analysis of Pancreas Segmentation Results by U-net

In section 5.2.2, separate U-nets were trained to generate realistic label images used for training a CycleGAN. However, the result of this approach was not successful enough to properly segment the four tissue compartments in the pancreas simultaneously. Instead, we used the U-nets segmentation results and analyzed them together with the metadata from the GTEx portal. All 594 subjects' pancreas histology images were segmented using U-nets. The training curves of the U-nets for islet, duct, and vessel can be found in appendix A.4. The training curve for the nuclei is omitted since the training of the nuclei only took less than 30 epochs. We presented the combined segmentation results of each U-net overlaid on the whole slide histology image with different colors in figure 5.22. The acinar area is indicated in cyan (greenish-blue color), the vessel in green, the islets in blue, the ducts in yellow, and the fat is shown in red. The U-nets for each annotation are selected based on the combination of test loss and training loss. For fat, we used same model chosen in section 5.1.2.2. The U-net model for islet is chosen from epoch 921, for duct is from 526, and the model for the vessel is from epoch 571.

Once we got the segmentation results, we extracted coordinates of the bounding boxes of every islet and duct from each subject and counted the number of islet and duct present. The coordinate of the bounding box means coordinates of the minimum-area rectangular border that encases an object-of-interest. From the bounding box, we calculated the area of each islet and duct. The process of obtaining bounding boxes and the area is done by

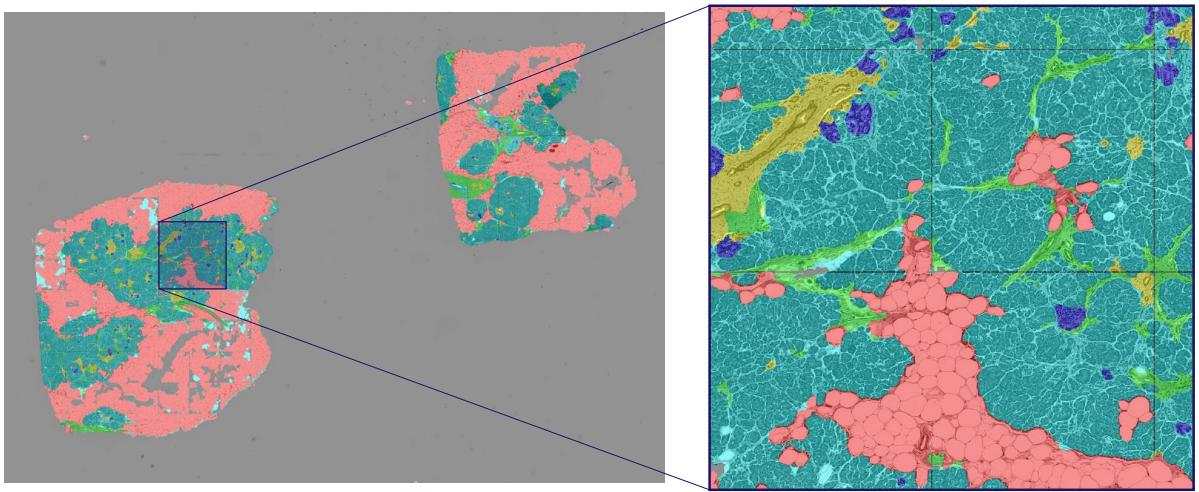
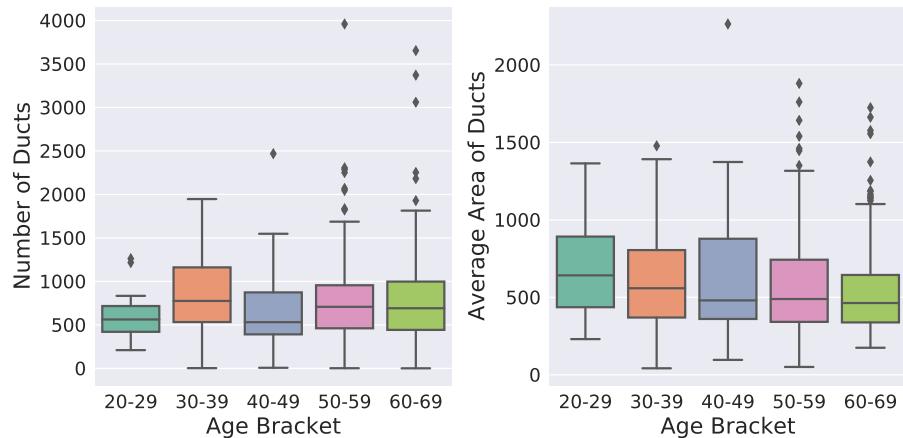


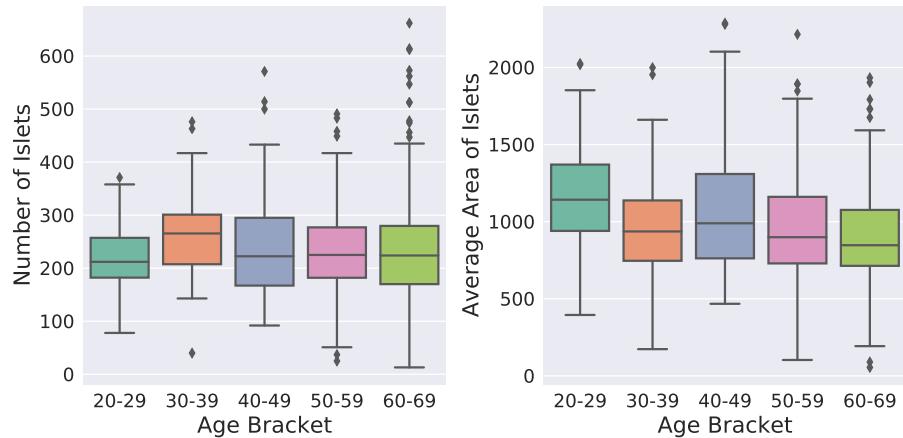
Figure 5.22: Result of whole histology slide segmentation on one subject.

python package *scikit-image*. We then investigated the relationship between the number and size of the islets/ducts and the each subject's age and their pancreas fat percentages.

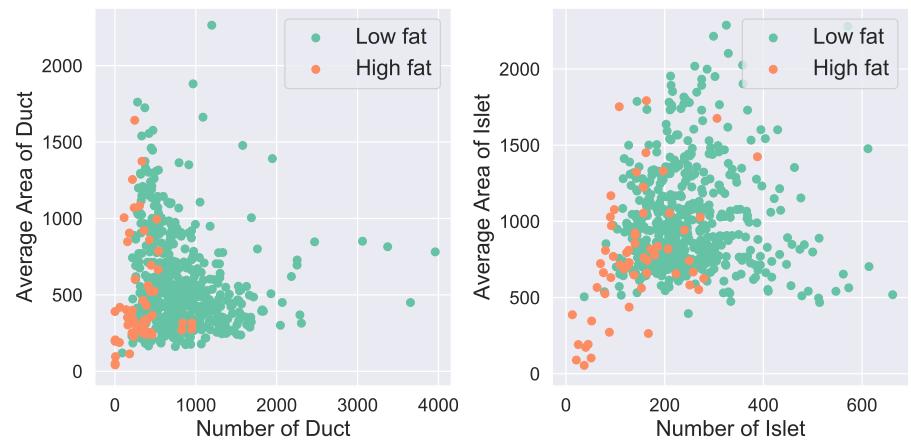
The results from the analysis can be seen in figure 5.23. Figure 5.23(a) and figure 5.23(b) indicate that there is no significant correlation between the number of ducts/islets with age. However, we do observe that the average area of both duct and islet tends to decrease with age. In figure 5.23(c), we have scatter plots between the number and area of each duct and islet, also showing low- and high- fat group.



(a) Box plot of duct with age bracket. Left plots are about number of the duct, and right plot are about average area of the ducts



(b) Box plot of islet with age bracket. Left plots are about number of the islets, and right plot are about average area of the islets



(c) Scatter plot showing relation between number and average area of each duct and islet. Each of them are colored with corresponding high- or low- fat group

Figure 5.23: Analysis of duct and islet in the pancreas associated with age and fat percentage of pancreas.

CHAPTER 6

Discussion

In this thesis, we tried to segment histology images by applying deep learning techniques to avoid the time consuming manual segmentation process. We attempted to automate this process by using the fully unsupervised approach proposed by Gadermayr et al [1]. We had three main objectives. First, explore unsupervised deep learning methods to segment histology images automatically. Second, perform segmentation of fat in pancreas and liver. Third, perform segmentation of four tissue compartments (acinar, fat, islet and ductal area) in pancreas. In the following sections, we will discuss the results in detail. Afterwards, we will highlight possibilities for improvements, and how current issues may be solved in future works, and how our results can be utilized.

The Gadermayr's Approach Successfully Segment Fat in Liver but Not in Pancreas

Gadermayr's approach successfully segmented fat in the liver tissue. From the liver segmentation results, we see that Gadermayr's approach is highly flexible and robust. It is also easily adaptable to other tissue compartments since it is simple to change the parameters of the annotation model. Furthermore, the staining method should not have a large impact on performance, since the algorithm only considers the shape and the distribution of the object-of-interest. This means that if the network can segment images with one staining method, it should also be able to segment images stained with other methods, even without any additional manual labeling. This is assuming that the tissue compartment of interest is stained so that it is distinguishable from the remaining tissue, since different staining methods highlight or hide different tissue compartments.

However, their approach does have some drawbacks when it comes to more complex segmentation tasks. One drawback is that the shape of the object-of-interest has to be very homogeneous throughout the tissue. If it varies a lot, it becomes difficult for the model to recognize all occurrences of the object. Furthermore, it is also difficult to create simulated annotations similar enough for the model to make the connection between the two. We will refer to this as the *homogeneity problem*. Another issue is that the annotation of each object-of-interests in the tissue has to be unique and distinguishable from each other. If each annotation is not unique, then the network fails to find a one-to-one matching between the object-of-interest and the annotations in simulated label images, we will call this the *uniqueness problem*. The homogeneity problem causes

the network to be unable to catch all occurrences of the object-of-interest, while the uniqueness problem causes the network to confuse the object with other objects.

Gadermayr's segmentation approach likely worked well in the liver, due to the fact that the fat in the liver has a very homogeneous shape and color that are unique to fat. The white color and roundish shape are not shared by any other tissue compartment in the liver. This allows the network to easily associate the simulated fat annotations to the fat in the histology images without experiencing either the homogeneity problem nor the uniqueness problem.

However, the network experienced both problems when segmenting fat in the pancreas. The network is only able to recognize either the single fats or the clusters of fat, it is never able to recognize both consistently. Unlike the liver fat, fat in the pancreas does not have a unique color or a unique shape, and this causes the uniqueness problem. In figure 6.1, there are examples of the shape of the fat cluster being very similar to the shape of the duct, vessel, or even with the background in the pancreas. The color of the fat is also not unique since the background shares the same color. These shared properties cause the network to confuse tissue compartments with each other, and thus to suffer from the uniqueness problem.

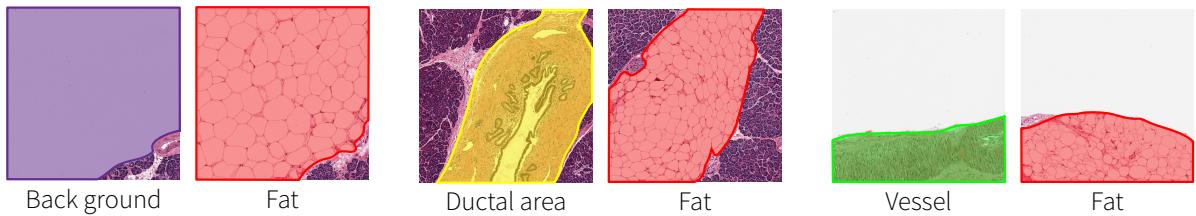


Figure 6.1: Comparison of the fat cluster in the pancreas with other tissue compartments that has similar characteristic in terms of color and shape.

Furthermore, the diverse characteristics of fat clusters in pancreas tissue cause the homogeneity problem. In the liver, the fat clusters are almost single fats placed very closely together, meaning they have a very homogeneous shape even if the fat is in a cluster (see figure 6.2(a)). This allows for the model to easily recognize both clusters of fat and single fats in the liver. However, in the pancreas, fats in the cluster are adjoined to each other, so each of them has a squeezed shape different from the single fats. Also, the annotations of the whole fat clusters vary greatly in shape (see figure 6.2(b)). These characteristics of the pancreas fat cluster lead to the homogeneity problem, which causes the network to be unable to recognize the fat clusters and the single fats consistently.

We attempted to solve these problems of fat in the pancreas with two different approaches (described in methods section 4.2.2.1). However, despite our best efforts to add more homogeneous shapes and uniqueness to allow the network to distinguish the fat from the other tissue, the network failed to do so.

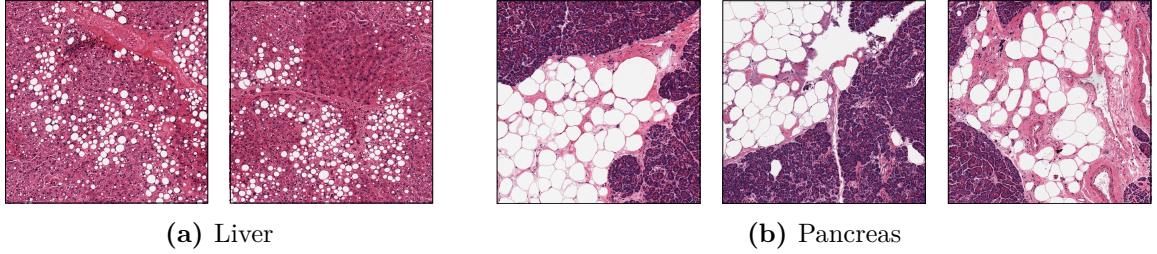


Figure 6.2: Comparison of the fat clusters in the pancreas and the liver. Note how the liver fat is clearly composed of smaller elliptical single fats, while the pancreas fats are not as clear due to being adjoined.

To conclude, Gadermayr’s approach works for segmenting fat in the liver since it does not suffer from either the homogeneity problem or the uniqueness problem. However, fat in the pancreas is more complex and thus the network suffers from both problems, resulting in the CycleGAN being unable to segment the pancreas tissue correctly. From the segmentation results, we observed that the single fats in the pancreas are easily distinguishable from other tissue compartments. We assume this worked since it is similar to the liver fat, highly distinguishable from other tissue compartments, and is homogeneous in shape and color.

Prior Knowledge of Annotations does not have Major Impact in Gadermayr’s Approach

When we applied Gadermayr’s approach to the liver tissue, we experimented with different distributions and different sizes of the fat annotations. However, all these different labels ended up in successful segmentation of the liver fat. Since fat is so unique in the liver, the network performance was not affected greatly by a poorly estimated prior during the generation of the simulated labels. We noticed that the distribution and the shape of the object-of-interest do not have considerable influence on the final segmentation performance, it is rather robust as long as it is uniquely detected as a realistic annotation in the simulated label images.

Uninformative Loss Measurement

We observed that the loss curve of the CycleGAN does not reflect the quality of the segmentation results. Since we were primarily interested in the mapping from histology image to label:

$$G : X(\text{histology images}) \rightarrow Y(\text{simulated labels})$$

we expected that if the mapping G successfully segments the histology images from domain X , then the discriminator loss for domain Y would converge to around 0.5, since this would mean that the discriminator, D_Y , cannot distinguish the difference between

the generated GAN labels and the labels in domain Y . However, in our successful segmentation (segmenting fat in the liver described in section 5.1.1), the discriminator loss did not converge to around 0.5 (see orange line in loss curve in figure 5.1). Instead, the discriminator loss tended to converge towards 0.5 early in training and subsequently dropped to close to 0.

Despite the discriminator loss trending towards 0, we observed improvements to the segmentation during this latter part of the training. We hypothesize that this is due to the network learning to generate more realistic segmentation label than our simple simulated label in domain Y . The resulting labels generated by the generator network from the later epochs are not analogous to labels in domain Y , since it is closer to the actual segmentation of the fat (see figure 6.3). Yet, the discriminator D_Y considered these better segmentation labels as fake labels. Note, that this means that during this part of the training, the discriminator network does not really help the generator network to learn, since the generated labels are always considered fake. To test this hypothesis, we could train a U-net to segment the liver tissue and use the highly realistic U-net segmentation labels as the Y domain. In this test, we should no longer observe the discriminator D_Y loss drops to 0, since the Y domain is now also highly realistic. If the loss still drops to around 0, it would mean that the low loss is not due to more realistic label output from the generator network. Furthermore, note that the late improvements are likely not relying on adversarial training, but likely cycle consistency loss, since we see qualitative improvements in fat annotations during this part of the training. This was not tested since the segmentation was successful, so our priority was to attempt to segment the pancreas instead.

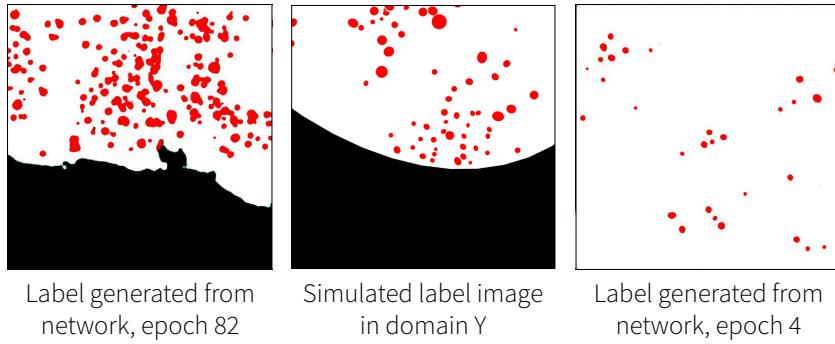


Figure 6.3: Comparison between generated labels from the networks and simulated label image. We can see that the label generated from epoch 4 (right) has elliptical shape like simulated labels in domain Y (middle), yet the label generated from epoch 82 (left) has more realistic shape correspond to actual shape of fat.

We also observed that the discriminator loss for image domain X (blue line in training curve in figure 5.1) quickly converges to 0 and remains there. This means that the discriminator for image domain X always identifies whether the image is from the original

domain or generated image from the mapping:

$$F : Y(\text{simulated labels}) \rightarrow X(\text{histology images})$$

We conclude that the mapping F does not learn much from the beginning and the discriminator for image domain X is much stronger than the generator F . This was to be expected. The simulated label image does not have enough information other than the boundary of cell and fat, so the generator F cannot reconstruct all the small details of the original histology images, see figure 5.2(b) for reconstruction images. The cell in the figure 5.2(b) has a similar color of the actual cell, but does not represent any other small details. This poor reconstruction leads to easy classification for the discriminator, and also to a high cycle consistency loss, making the whole training worse. From this, we conclude that additional information is needed to obtain successful translation between the image domain and the label domain.

To confirm the loss curve of the CycleGAN does not reflect the quality of the segmentation results, we compare the successful liver fat segmentation described above with the unsuccessful segmentation of the four tissue compartments in the pancreas, see section 5.2.2. The loss curve of the unsuccessful training (see figure 5.17) resembles the loss curve of the successful one, except for the loss from the discriminator of image domain X . Since the network had additional information to work within the simulated label image, the generator F learned to generate better reconstruction images to fool the discriminator. Other than that, all the losses converge similarly. We conclude that convergence of the CycleGAN losses does not guarantee good segmentation performance, confirming the non-informative loss functions of GAN networks. Instead, manual qualitative assessments are required.

Additional Information for Simulated Label Images

We applied two different approaches to segment the four tissue compartments in the pancreas. Similar to previously, we initially attempt to segment the tissue fully-unsupervised and used the CycleGAN as proposed by Gadermayr et al. The simulated labels were generated similar to in the liver experiment, a prior of 4 theoretical shapes. However, here we again observe the uniqueness and homogeneity problem, in which mappings are non-unique and tissue compartments get confused with each other.

In order to obtain more unique labels that are more easily associated with the correct tissue compartment, we instead generated the labels using U-net segmentation results of the tissue. Each of the four annotations was generated with separate U-nets. This should allow for close to photo-realistic simulated labels, that hopefully allows for a more unique mapping. In order to allow the network to more easily distinguish the different tissues, we added additional information in forms of nuclei annotation and blood vessel annotation. These were also added by U-net segmentation. This should allow the network to distinguish each tissue compartment from each other. All six annotations

were combined into one image and used as the simulated label images for CycleGAN training, see section 4.3.2 for details.

Unfortunately, even with the more realistic labels and the additional information, the network still did not learn to segment the four compartments properly. From the result, it appears that the uniqueness and homogeneity problem still existed. The performance might have improved compared to naïve Gadermayr’s approach, but it is not a significant improvement to call the results as a success. So, we conclude that additional information with nuclei and vessel annotations was not enough to make the annotations in the pancreas unique.

6.1 Future Works

More Advanced Network Architecture

When building the network architecture for CycleGAN, we used the convolutional neural network combined with 6 or 9 residual blocks as our generator network. As presented throughout chapter 5, this network did not successfully segment the pancreas histology images properly. We considered more advanced network architecture to give the network more capacity to detect more features in the data. Since we observed that the U-net architecture is highly effective in the sense that it preserves the context of information during the training process, we attempted to adopt the U-net architecture as the generator of the CycleGAN. However, this network was very large, and due to a lack of computational power, it was not possible to actually implement this. If there is sufficient computational power, more advanced network architecture could be used for training, which may improve the segmentation result of the pancreas histology images.

More Training Data

Furthermore, we used 3043 pancreas tiles from 19 histology slides for the training process, since preprocessing of the whole slide histology images took a long time. This is roughly 3% of all available whole histology slides from the GTEx portal. It is commonly known that more training data often leads to more accurate training of the network [28]. Hence, utilizing much larger training data might help the network to improved significantly, and hopefully segment the pancreas histology images properly.

Apply Gadermayr’s Approach to Other Segmentation Tasks

We successfully segmented the fat in the liver tissue with Gadermayr’s approach because the fat in the liver tissue is easily distinguishable and homogeneous. Furthermore, poor prior estimation for the annotation model did not affect the results. This clearly shows

the potential of the Gadermayr’s approach. This could be expanded and used on other tissue segmentation as long as they remain rather simple.

Another possible avenue of expansion would be to apply the Gadermayr’s approach to histology images stained differently. Our work so far has been done on H&E staining. Since we successfully segmented the fat in the liver histology images stained with H&E staining, we can apply Gadermayr’s approach to other liver histology images stained differently. Also, one could expand the approach on other tissue and possibly also investigate the CycleGANs ability to generalize. This is all only possible due to the CycleGAN only considering shape and occurrence distribution of the object-of-interest.

Semi-Supervised Approach

The framework proposed by Mondal et al.[25] described in section 3.2 is semi-supervised framework for segmentation task using CycleGAN. It requires partially labeled data set, and uses this labeled data set for additional supervised loss, compelling the CycleGAN to perform better. We can use the segmentation results from this thesis and apply this semi-supervised approach to segment pancreas histology images with the additional supervised losses. Our results from training CycleGAN with realistic label images from U-net results show somewhat promising results, so adding supervised manner may lead the network to successfully segment the histology images. However, since this network has yet to be applied to histology images, we are unsure how well it will perform.

Associate Histology Segmentation with Data Difficult to Interpret

The segmentation results obtained from this thesis can be quantified in terms of organ fat percentage, number/average size/coordinate of the tissue compartments. This data is easily interpretable, however, most of the biological data is not. These obtained data can be associated with more difficult to interpret data to analyze in unison. This could be genetic mutations, RNA sequencing, or proteomics data. All these data sets have a degree of uncertainty associated to them, since much of the biological mechanisms are not completely understood. Associating them with phenotypic data such as histology segmentation results, e.g. organ fat percentage, could lead to novel discoveries.

CHAPTER 7

Conclusion

In this thesis, we attempted to segment histology images by applying deep learning techniques to avoid the time-consuming manual annotation process. We had three main objectives: First, to explore unsupervised deep learning methods to automate the segmentation of the histology images. Second, to perform segmentation of fat in the pancreas and the liver. Third, to perform segmentation of four tissue compartments (acinar, fat, islet and ductal area) in the pancreas using a single model.

We implemented the fully unsupervised approach proposed by Gadermayr et al [1]. and successfully segmented the fat in the liver. However, it failed to segment the fat and the four tissue compartments in the pancreas histology images. Thus, we applied a supervised approach, U-net, to the pancreas histology images and successfully segmented the fat and other tissue compartments using separate models for each of them. These segmentation results were then combined into one label image and used as the Y domain in the CycleGAN training of the Gadermayr approach. We expected these highly realistic labels to result in better segmentation, but unfortunately, the network could still not segment the pancreas histology images correctly. We also conducted analysis between the segmentation results from 594 whole histology slides and metadata from GTEx. We hypothesized that there might be a correlation between the fat in the pancreas and the liver [11]–[15], but we observed no correlation between the two organs.

To conclude, we explored deep unsupervised learning techniques for tissue segmentation, and successfully segmented the fat in the liver tissue fully unsupervised. Furthermore, we segmented the fat and four tissue compartments in the pancreas. However, we had to apply supervised methods to segment the pancreas tissue. Gadermayr's unsupervised approach we applied to the pancreas tissue does show potential, however, we expect a more flexible model architecture is required successfully segment the tissue.

APPENDIX A

An Appendix

A.1 Code Availability

All the scripts that we created for this thesis will be shortly updated after the submission of the thesis in [GitHub repository](#).

A.2 Method Extracting Cell Mask

The cell mask in each tile can be obtained based on the intensity value of the image. Note that in the one-channel gray image, the intensity value of the black color is 0, and the white color is 255. To separate cell area from the white background, we first convert the RGB tile image to the one-channel gray image. Then only remain the pixel with the value under than 230. Then fill the holes to get the full mask. The drawback of this intensity-based method is that the cell with white color at the edge of the tile is considered ad background. For better understanding, see the figure below.

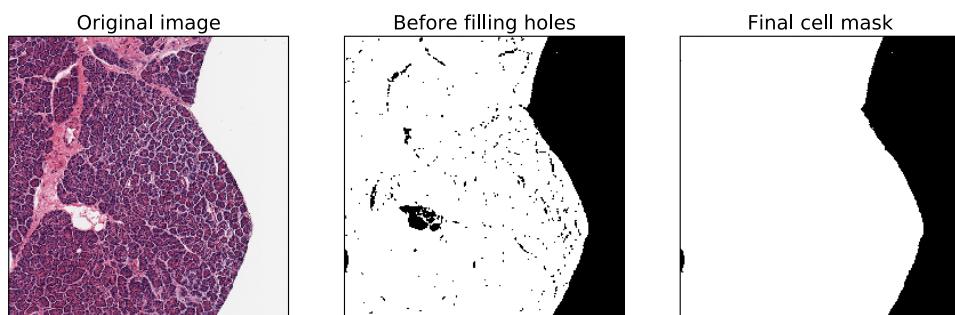


Figure A.1: Illustration of the method extracting cell mask.

A.3 Plots for Relation between Gender and Fat Percentage

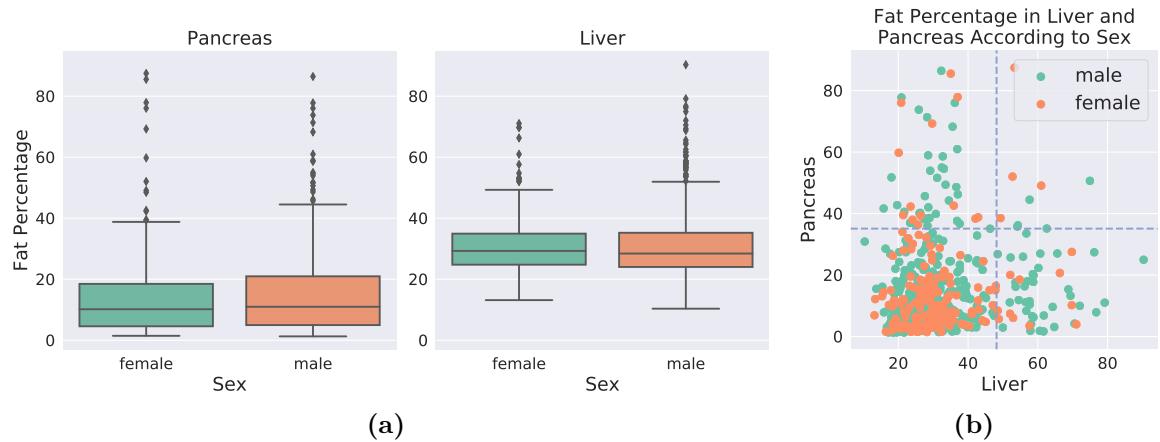


Figure A.2: Plots for relation between gender and fat percentage. (a) Box plot between gender and fat percentage in pancreas and liver. (b) Scatter plot of fat percentage in liver and pancreas colored with gender. x -axis indicates fat percentage in liver, and y -axis indicates fat percentage in pancreas.

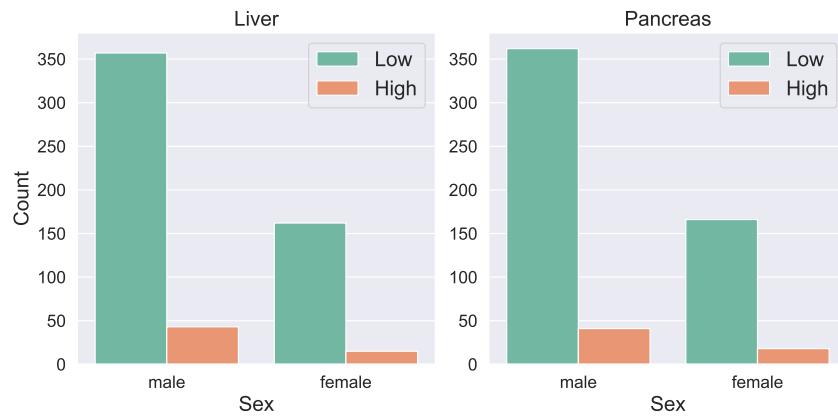


Figure A.3: Count plot of predicted fat percentage in the liver and the pancreas divided with high fat group and low fat group.

A.4 Training Curves of U-net for Different Annotations

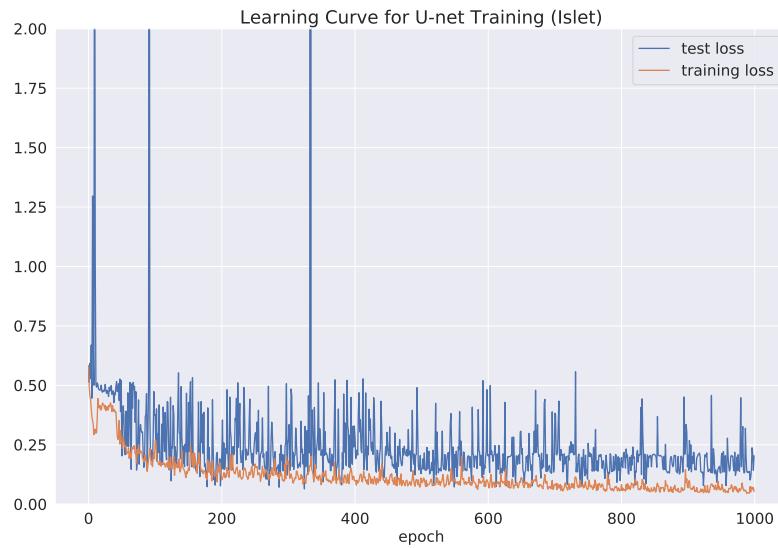


Figure A.4: Training curve of U-net for segmenting islet in the pancreas.

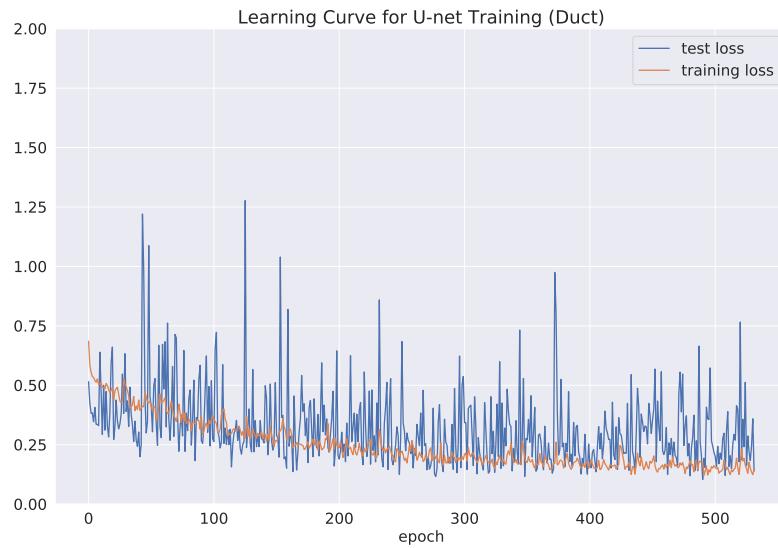


Figure A.5: Training curve of U-net for segmenting duct in the pancreas.

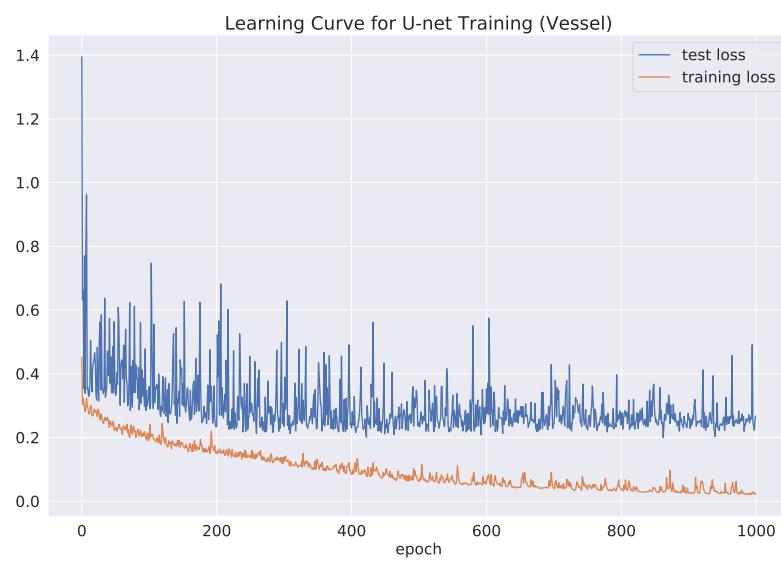


Figure A.6: Training curve of U-net for segmenting vessel in the pancreas.

Bibliography

- [1] M. Gadermayr, L. Gupta, B. M. Klinkhammer, P. Boor, and D. Merhof, “Unsupervisedly Training GANs for Segmenting Digital Pathology with Automatically Generated Annotations,” pages 1–5, 2018. [Online]. Available: <http://arxiv.org/abs/1805.10059>.
- [2] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 9351, pages 234–241, 2015, ISSN: 16113349. DOI: 10.1007/978-3-319-24574-4{_}28.
- [3] V. Tamizhazhagan, “Histological Methods in Life Science,” *International Journal of Biomedical Materials Research*, volume 5, number 6, page 68, 2017, ISSN: 2330-7560. DOI: 10.11648/j.ijbmr.20170506.11.
- [4] H. Irshad, A. Veillard, L. Roux, and D. Racoceanu, “Methods for nuclei detection, segmentation, and classification in digital histopathology: A review-current status and future potential,” *IEEE Reviews in Biomedical Engineering*, volume 7, pages 97–114, 2014, ISSN: 19411189. DOI: 10.1109/RBME.2013.2295804.
- [5] L. Shapiro and G. Stockman, *Computer Vision*. New Jersey: Prentice Hall, 2001, pages 279–325, ISBN: 9780130307965. [Online]. Available: <https://books.google.dk/books?id=FftDAQAAIAAJ>.
- [6] J. Borovc and J. Kybic, “Fully automatic segmentation of stained histological cuts,” *International Student Conference on Electrical Engineering*, pages 1–7, 2013. [Online]. Available: <ftp://cmp.felk.cvut.cz/pub/cmp/articles/borovec/Borovc-Poster2013.pdf>.
- [7] M. Veta, P. J. van Diest, R. Kornegoor, A. Huisman, M. A. Viergever, and J. P. Pluim, “Automatic Nuclei Segmentation in H&E Stained Breast Cancer Histopathology Images,” *PLoS ONE*, volume 8, number 7, pages 1–12, 2013, ISSN: 19326203. DOI: 10.1371/journal.pone.0070221.
- [8] A. M. Khan, N. Rajpoot, D. Treanor, and D. Magee, “A nonlinear mapping approach to stain normalization in digital histopathology images using image-specific color deconvolution,” *IEEE Transactions on Biomedical Engineering*, volume 61, number 6, pages 1729–1738, 2014, ISSN: 15582531. DOI: 10.1109/TBME.2014.2303294.

- [9] M. Lai, “Deep Learning for Medical Image Segmentation,” 2015. [Online]. Available: <http://arxiv.org/abs/1505.02000>.
- [10] B. Young, G. O'Dowd, and P. Woodford, *Wheater's Functional Histology*, 6th Editio. Philadelphia: Elsevier, 2013, page 464, ISBN: 9780702047473.
- [11] D. Wang, X. p. Yu, W. m. Xiao, X. p. Jiao, J. Wu, D. l. Teng, K. y. Wu, M. Zhang, Q. t. Zhu, X. n. Liu, Y. b. Ding, and G. t. Lu, “Prevalence and clinical characteristics of fatty pancreas in Yangzhou, China: A cross-sectional study,” *Pancreatology*, volume 18, number 3, pages 263–268, 2018, ISSN: 14243911. DOI: 10.1016/j.pan.2018.02.004. [Online]. Available: <https://doi.org/10.1016/j.pan.2018.02.004>.
- [12] N. S. Patel, M. R. Peterson, G. Y. Lin, A. Feldstein, B. Schnabl, R. Betten-court, E. Seki, C. B. Sirlin, and R. Loomba, “Insulin resistance increases MRI-estimated pancreatic fat in nonalcoholic fatty liver disease and normal controls,” *Gastroenterology Research and Practice*, volume 2013, 2013, ISSN: 16876121. DOI: 10.1155/2013/498296.
- [13] Z. Dong, Y. Luo, H. Cai, Z. Zhang, Z. Peng, M. Jiang, Y. Li, C. Li, Z. P. Li, and S. T. Feng, “Noninvasive fat quantification of the liver and pancreas may provide potential biomarkers of impaired glucose tolerance and type 2 diabetes,” *Medicine (United States)*, volume 95, number 23, pages 1–7, 2016, ISSN: 15365964. DOI: 10.1097/MD.0000000000003858.
- [14] C. Della Corte, A. Mosca, F. Majo, V. Lucidi, N. Panera, E. Giglioni, L. Monti, L. Stronati, A. Alisi, and V. Nobili, “Nonalcoholic fatty pancreas disease and Nonalcoholic fatty liver disease: More than ectopic fat,” *Clinical Endocrinology*, volume 83, number 5, pages 656–662, 2015, ISSN: 13652265. DOI: 10.1111/cen.12862.
- [15] E.-J. van Geenen, M. M. Smits, M. T. C. M. A. Schreuder, D. L. van der Peet, and C. J. J. Bloemena Elisabeth Mulder, “Nonalcoholic fatty liver disease is related to nonalcoholic fatty pancreas disease,” *Pancreas*, volume 39(8), pages 1185–1190, 2010. DOI: 10.1097/MPA.0b013e3181f6fce2.
- [16] T. R. Martini FH Timmons MJ, *Human Anatomy*, 7th Editio. Pearson, 2012, ISBN: 9780321688156.
- [17] S. R. Abdel-Misih, “Liver Anatomy,” *Surg Clin North Am.*, volume 90(4), pages 643–653, 2010. DOI: 10.1016/j.suc.2010.04.017.
- [18] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” *Advances in Neural Information Processing Systems*, volume 3, number January, pages 2672–2680, 2014, ISSN: 10495258. DOI: 10.3156/jsoft.29.5{_}177{_}2.
- [19] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks,” *Proceedings of the IEEE International Conference on Computer Vision*, volume 2017-Octob, pages 2242–2251, 2017, ISSN: 15505499. DOI: 10.1109/ICCV.2017.244.

-
- [20] C. M. Bishop, *Pattern Recognition And Machine Learning*. Springer, 2006, ISBN: 978-0-387-31073-2.
 - [21] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation Functions: Comparison of trends in Practice and Research for Deep Learning,” pages 1–20, 2018. [Online]. Available: <http://arxiv.org/abs/1811.03378>.
 - [22] V. Nair and G. E. Hinton, “Rectified Linear Units Improve Restricted Boltzmann Machines,” *ICML*, pages 807–814, 2010. [Online]. Available: <https://www.bibsonomy.org/bibtex/2059683ca9b2457d248942520babbe000/dblp>.
 - [23] S. Indolia, A. K. Goswami, S. P. Mishra, and P. Asopa, “Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach,” *Procedia Computer Science*, volume 132, pages 679–688, 2018, ISSN: 18770509. DOI: 10.1016/j.procs.2018.05.069. [Online]. Available: <https://doi.org/10.1016/j.procs.2018.05.069>.
 - [24] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2016-Decem, pages 770–778, 2016, ISSN: 10636919. DOI: 10.1109/CVPR.2016.90.
 - [25] A. K. Mondal, A. Agarwal, J. Dolz, and C. Desrosiers, “Revisiting CycleGAN for semi-supervised segmentation,” 2019. [Online]. Available: <http://arxiv.org/abs/1908.11569>.
 - [26] R. T. Collins, “Mean-shift blob tracking through scale space,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, 2003, ISSN: 10636919. DOI: 10.1109/cvpr.2003.1211475.
 - [27] B. Alberts, A. Johnson, and J. Lewis, *Molecular Biology of the Cell*, 4th editio. New York: Garland Science, 2002, ISBN: 0-8153-3218-1. [Online]. Available: <https://www.ncbi.nlm.nih.gov/books/NBK21054/>.
 - [28] M. Banko and E. Brill, “Scaling to Very Very Large Corpora for Natural Language Disambiguation,” in *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, Toulouse, France: Association for Computational Linguistics, 2001, pages 26–33. DOI: 10.3115/1073012.1073017. [Online]. Available: <https://www.aclweb.org/anthology/P01-1005>.