DEEP STRUCTURED LEARNING IN MEDICAL IMAGE ANALYSIS

by

Bin Kong

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Computing and Information Systems

Charlotte

2020

Approved by:

Dr. Shaoting Zhang

Dr. Min C. Shin

Dr. Srinivas Akella

Dr. Weichao Wang

ABSTRACT

BIN KONG. Deep Structured Learning in Medical Image Analysis. (Under the direction of DR. SHAOTING ZHANG)

Deep learning-based techniques have been widely employed in solving various medical image analytical problems. Currently, most of these methods directly employ deep architectures from natural image scenarios without considering the specific structures in the input/output variables, resulting in a suboptimal solution. In this dissertation, we systematically discuss deep structured learning in medical image analysis (MIA). Particularly, this dissertation is organized by answering the following questions: 1) how to model complex dependencies among the input/output variables with deep neural networks, 2) how to enforce prior structural knowledge in deep structured learning, and 3) how to model certain special structures in MIA problems. More specifically, we first introduce a formal formulation of structured learning in MIA and present a general deep structured learning framework to address this problem. Second, we enforce the prior structural knowledge in the loss function to further improve the analytical performance. Third, as an example of special structures in medical imaging, we introduce how to model the tree structures in coronary arteries with tree-structured convolutional long short-term memory. Finally, we further introduce a special structured learning problem in medical imaging which involves sequential decision making. Accordingly, a deep reinforcement learning-based solution is proposed. To put our discussion in the context of MIA, we evaluated our approaches on several MIA tasks, i.e., cardiac recognizing from MRI sequences, metastasis detection in whole-slide images (WSIs), coronary artery segmentation from 3D computed tomography angiography (CTA) volumes, and axon tracing. The superior performance demonstrates the effectiveness.

ACKNOWLEDGEMENTS

This dissertation would not have been possible without the support of many people. First and foremost, I would like to express my special thanks to my advisor Dr. Shaoting Zhang. You provided me with the perfect balance of guidance and freedom. Under your guidance, I found that I really fell in love with deep learning. I also want to thank you for your perspective and for helping me pursue and define projects with more impact. As my teacher and mentor, you have taught me more than I could ever give you credit for here. You has shown me, by your example, what a good scientist and person should be. I am also thankful to Dr. Min C. Shin, Dr. Weichao Wang, Dr. Srinivas Akella for serving on my committee and for their helpful comments.

Nobody has been more important to me in this long and arduous journey than the members of my family. I would like to thank my parents, brother, and wife, whose love and guidance are with me in whatever I pursue. In particular, I must express my special gratitude to Congcong Jin, my wife, for your continued support and encouragement. You have stood by me through all my travails, my absences, my fits of pique and impatience. You are the best companion I can have in life.

I also want to thank my labmates in VIA lab, especially to Zhongyu Li, Lance Rice, and Junjie Shan. It is due to the friendly and supportive environment in VIA lab that I was lucky enough to find so many great people to work with. I really enjoyed my collaborations with you. Many thanks to Dr. Min C. Shin for organizing VIA lab so well that I feel in the heart that I am a family member here.

For over three years I was supported by CuraCloud corporation for which I want to sincerely thank the financial support and the amazing people I have collaborated with. A particular thank you goes to Dr. Shanhui Sun and Dr. Xin Wang. I was amazed that they could give so much helpful and technical feedback, not only in long conversations during my internship but also after I go back to UNC Charlotte.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

aFD Average frame difference

CAD Computer-aided diagnosis

CCTA Coronary computed tomography angiography

CNN Covolutional neural network

ConvGRU Convolutional gated recurrent unit

ConvLSTM Convolutional long short-term memory

CT Computed tomography

DBM Deep Boltzmann machine

DNN Deep neural network

DRL Deep reinforcement learning

ED End-diastole

ES End-systole

FCNN Fully convolutional neural network

GRU Gated recurrent unit

LSTM Long short-term memory

MAP Maximum a posteriori

MIA Medical image analysis

MLE Maximum likelihood estimation

MLP Multilayer perceptron

MR Magnetic resonance

PET Positron emission tomography

RL Reinforcement learning

RNN Recurrent neural network

WSI Whole slide image

CHAPTER 1: INTRODUCTION

Deep learning is emerging as a powerful tool with hierarchical architectures for modeling high-level abstractions of the data. It consists of layers of non-linear transformations. To date, numerous variants of deep learning frameworks [13, 14, 15, 16, 17, 18, 19] have been proposed to address problems in various fields of study such as computer vision [20, 21, 22, 23], natural language processing [24, 25, 26, 27], and speech recognition [28, 29]. Recent progresses in deep learning have enabled a lot of breakthroughs in medical image analysis (MIA) tasks, ranging from classification [30, 31, 32, 8, 7, 33], detection [34, 35, 36, 37], segmentation [38, 39, 40, 41, 42], registration [43, 44, 45], image synthesis [46, 47], to diagnostic report generation [48, 49].

While deep learning is driving the rapid growth of modern MIA, the common practice is to directly use the well-established deep learning models [14, 50, 51] from the computer vision community for MIA problems, without considering specific structures, or dependencies among the input/output variables, resulting in a suboptimal solution.

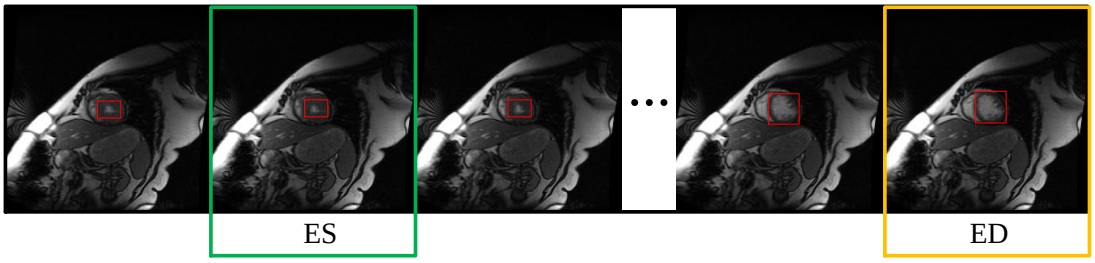

Figure 1.1: A typical cardiac MRI sequence (red, green, and yellow rectangles indicate the left ventricles, ES frame, and ED frame, respectively). Most regions do not have noticeable changes except that the left ventricle slightly changes over time.

However, structures are ubiquitous and of great importance in numerous MIA tasks. For instance, Fig. 1.1 shows a typical cardiac MRI sequence. With the contraction and

relaxation of the heart, the left ventricle (red rectangles) volume gradually diminishes and expands. The maximum and minimum left ventricular volumes correspond to the end-diastole (ED) and end-systole (ES), respectively. In a cardiac MRI sequence, most regions do not have noticeable changes except that the left ventricle slightly changes over time. The subtle differences between neighboring frames make it extremely challenging to accurately locate the ED and ES frames. Fortunately, there exist certain structures in the sequence that we can leverage to effectively guide their localization: the left ventricle volume keeps decreasing during a systole phase and increasing during a diastole phase, as is also demonstrated by Bogaert *et al.* [52].

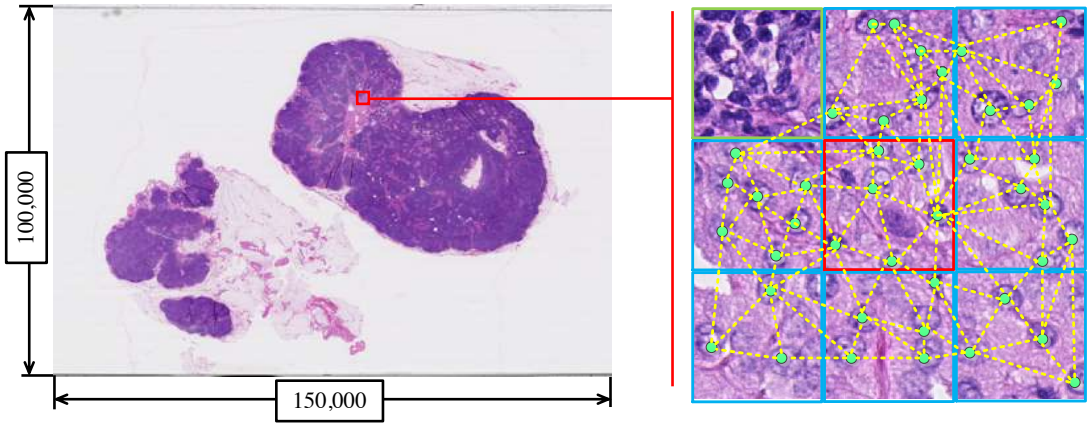

Figure 1.2: Left: a gigabyte (*e.g.*, $150,000 \times 100,000$ pixels) WSI. Right: illustration of the spatial correlations among neighboring image patches. The top left patch indicated by the green square is labeled as normal. This image region is sampled from the left WSI. The label of the center patch (red) is likely to correlate with its neighbors. The nuclei denoted by green dots also highlight similar spatial structures underlying in the center as well as its neighboring tumor patches.

Another example is cancer metastasis detection from gigabyte (*e.g.*, $150,000 \times 100,000$ pixels) whole slide images (WSIs), as is shown in the left of Fig 1.2. Existing machine learning algorithms cannot directly handle these massive images. A common solution is to divide them into small patches and tackle them individually. However, this technique doesn't consider spatial dependencies among these patches. As demonstrated in [32, 53], structural knowledge about the neighboring patches can be leveraged to boost the detection accuracy. In summary, exploiting this structural

information can be extremely important for many MIA applications.

Inspired by recent ideas for enforcing the structural information in deep MIA frameworks [54], we present a general deep structured learning framework for MIA tasks. Specifically, we observe two types of common structures in MIA tasks: the statistical inter-correlations among the input/output variables and the prior structural knowledge. In order to effectively model these structures, we introduce two key components into our structured learning framework: deep feature extractor and structural feature learner. Regarding deep feature extractor, we employ the commonly used convolutional neural networks (CNNs) such as Resnet [14] and ZFNet [15], which is shown extremely successful for automatically learning the most discriminative features from the data. This greatly facilitates subsequent analyzing steps. Additionally, the dimension of the input feature is reduced by the pooling layers to avoid the curse of dimensionality. Regarding the structural feature learner, it is capable of modeling the complex interactions in the input/output variables. Additionally, it has been shown that incorporating prior structural knowledge into medical image analytical algorithms is essential for obtaining more accurate results [55, 54] in situations like corruption and artifacts in medical images. The incorporation of prior structural knowledge into deep learning frameworks is not obvious. In this dissertation, we incorporate it into the training procedure as an additional regularisation loss term to further boost the performance.

Although the proposed method is motivated by recent work on enforcing structural coherence into deep learning frameworks, our framework is more general and less task-specific. More specifically, the proposed framework is independent of particular tasks or deep learning architectures. As a result, it is applicable to a wide range of MIA tasks and readily combined with well-established neural network architectures. Theoretically, deep neural networks are able to capture structural information without the specification of a priori. Nevertheless, this comes at the cost of a requirement of

a large-scale labeled high-quality training data, which is extremely expensive and difficult to collect in MIA. In this regard, enforcing the structural information into deep learning algorithms brings an additional benefit: alleviating the need for a significant amount of labeled data.

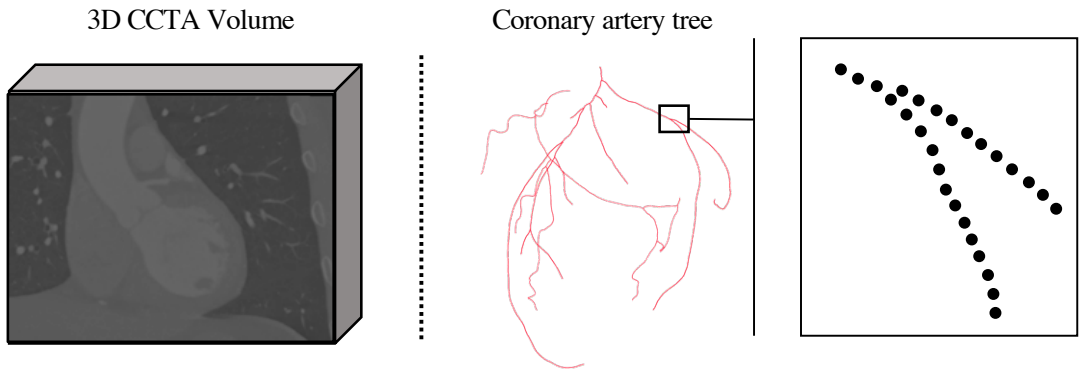

Figure 1.3: Coronary artery tree tracing in CCTA volumes. **Left:** the CCTA volume. **Right:** the traced coronary artery tree. The tracing procedure involves the anatomical structure of the coronary artery in which each node is dependent on the others.

Finally, we observe that accurately tracing object geometry/topology is crucial for a lot of applications in MIA [56], especially for the diagnosis of vascular diseases from medical images. A common practice in the analysis of these images is to build an anatomical organ structure. As an example, the coronary artery tree tracing in 3D coronary computed tomography angiography (CCTA) involves building an anatomical structure of the coronary artery in which each node is dependent on the other nodes, as is shown in Fig. 1.3. A tremendous amount of efforts [57, 58, 59] have been devoted to this line of research. In this dissertation, we provide an additional perspective on this problem. Specifically, we formulate this task as a sequential decision-making problem and approach it with deep reinforcement learning (DRL), *i.e.*, actor-critic network, a sophisticated DRL framework. Defining an effective reward function is essential for the training of the agent. In this dissertation, we show that we are able to effectively train an axon tracing agent which achieves promising results with a carefully designed reward function and a refined training procedure.

To summarize, our work for deep structured learning in MIA consists of the following contributions, which will be discussed in detail in each chapter:

1. We introduce a general structured framework for modeling the structures in MIA. In this way, the dependencies among the input/output variables are explicitly considered in hierarchical deep learning architectures. Such a deep learning architecture guides the network to learn structurally reasonable features from the training data. This framework is ignorant of specific applications and network architectures and can be used in a wide range of MIA problem settings.

2. To incorporate prior structural knowledge into deep learning frameworks, we propose to constrain the training process of the framework by introducing an additional regularisation term into the loss function. This provides us an opportunity to incorporate a prior about the structures in medical images into the training process.

3. In addition to the evaluation the proposed method on two MIA tasks, *e.g.*, cardiac MRI recognizing and cancer metastasis detection in WSIs, we consider a more challenging structured learning problem, *i.e.*, tree-structured learning. To approach this problem, we present tree-structured convolutional gated recurrent unit (GRU) to model the complex interactions between the tree nodes.

4. Finally, we observe that some MIA problems can be easily formulated as a sequential decision-making problem and introduce a sophisticated DRL framework to address this issue. Particularly, with a carefully designed reward function and a refined training procedure, we are able to effectively train an axon tracing agent to achieve promising results.

We organize the rest of the dissertation as follows. First, we give a brief introduction to the related deep learning architectures and review the related works in

chapter 2. Then, we elaborate on the formal formulation of structured learning and present a general framework to model the input/output variables' dependencies in chapter 3. We also introduce the details of incorporating prior structural knowledge into the loss function in chapter 4. As an example of special structures in the medical images, we detailedly discuss a special structure-tree structure in the coronary artery and demonstrate how to use tree-structured convolutional GRU layer to explicitly model this structure in chapter 5. Furthermore, we discuss anatomical structure reconstruction, a special structured learning problem, in MIA. We formulate this task as a sequential decision-making problem and approach this problem with DRL in chapter 6. Finally, we summarize this dissertation and provide insights for possible future directions in chapter 7.

CHAPTER 2: RELATED WORK

In this chapter, we first briefly introduce related deep learning preliminaries. Then, we review previous studies of structured learning, especially in MIA. Finally, we elaborate on recent developments on embedding structures in deep learning, followed by the discussion of differences between our methods and the proposed approach.

2.1 A Brief Introduction to Deep Learning

Work on deep learning has been done since the late seventies [60]. The use of deep learning has not until recently gathered momentum since the contribution of Krizhevsky *et al.* [16] to the ImageNet challenge [61] in 2012. We have also seen many new ideas regarding CNN architectures, such as Inception [17], ResNet [14], and U-Net [38]. In this chapter, we only introduce the basics of deep learning.

The multilayer perceptron (MLP) is a basic deep learning architecture, which consists of L layers of neurons. Every neuron in the MLP represents a non-linear transformation $g = \sigma(\boldsymbol{w}^T\boldsymbol{x} + b)$, where $\boldsymbol{x}$ and g are the input and output respectively. $\boldsymbol{w}$ and b are the learnable weights and bias respectively. Together, these neurons define a complex non-linear function, $\sigma(\boldsymbol{w}_L^T\sigma(\boldsymbol{w}_{L-1}^T...) + b_L)$. At each iteration of the training stage, the output of MLP is compared with the ground truth to compute an error per parameter, which is then used to adjust the parameter. Note that there are no preexisting assumptions about the particular task or dataset and the training is purely guided by the dataset. In the following, we elaborate on several types of neural network architectures used in this dissertation: convolutional neural network (CNN), recurrent neural network (RNN), and deep reinforcement learning (DRL).

Compared with MLP, CNN is designed to better utilize local spatial correlations

in images. As is illustrated in Fig. 2.1, three types of layers are often used in CNN: convolution, pooling, and fully-connected. CNN leverages three techniques to reduce the complexity of deep models: weights sharing, downsampling, and local receptive field. The convolution layer is designed to detect local patterns in the input feature maps. The pooling layer usually follows the convolution layer to reduce the dimension of the feature map.

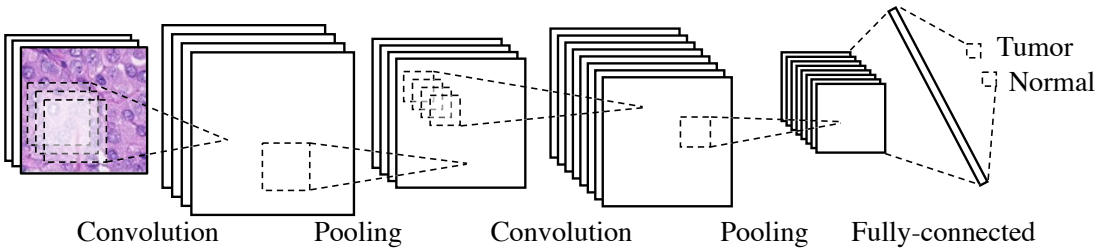

Figure 2.1: Block diagram of a typical CNN model which consists of two convolution, two pooling, and fully-connected layers. This model is used to classify the histopathological images into two categories: tumor or normal.

Recurrent Neural Network (RNN) exploits sequential patterns in sequentially formatted data. Usually, the long short-term memory (LSTM) [62] is preferred over the vanilla RNN model [63] as it significantly alleviates the notorious exploding/vanishing gradient problem. As is shown in Fig. 2.2, the LSTM contains a memory block c_t at each step t to store the history memory of the input data. It leverages three gates, *i.e.*, input gate i_t, output gate o_t, and forget gate f_t, to regulate the information flow and update the memory:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1}), \tag{2.1}$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1}), \tag{2.2}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1}), \tag{2.3}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_{xm}x_t + W_{mi}h_{t-1}), \tag{2.4}$$

$$h_t = o_t \odot \varphi(c_t), \tag{2.5}$$

where $\sigma(x) = (1 + e^{-x})^{-1}$ and $\odot$ denote sigmoid function and element-wise product respectively. W_{xi}, W_{hi}, W_{xf}, W_{hf}, W_{xo}, W_{ho}, W_{xm}, and W_{hm} are the learnable weights[1].

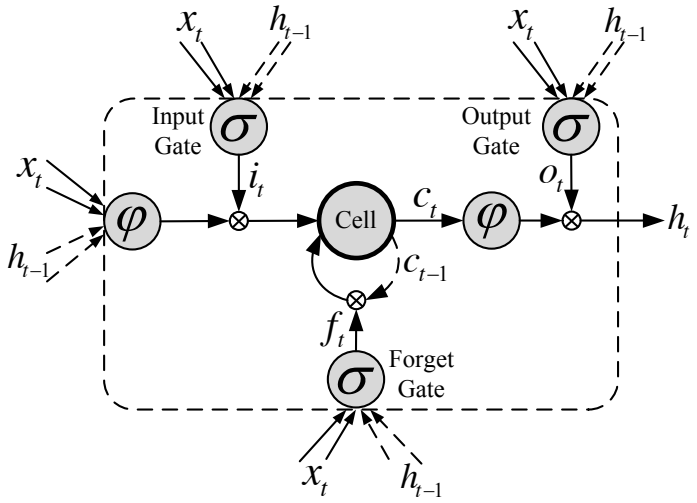

Figure 2.2: Block diagram of an LSTM unit. At each timestep t, the LSTM unit maintains a memory c_t. It takes as input the previous hidden state h_t and the current input x_t. The input, output, and forget gates are used to control the information flow inside the unit.

Reinforcement learning (RL) is a separate branch of machine learning, which aims to train an artificial agent to optimally make a sequence of decisions [64]. Formally, in the standard RL setting, an agent in RL inhabits an environment $\mathcal{E}$ and interacts with it until the terminal state is reached. More specifically, a state s_t is received by the agent from $\mathcal{E}$ at each time step t. Then, the agent samples an action $a_t \in \mathcal{A}$ based on s_t and its policy π, where $\mathcal{A}$ is the action space with all the possible actions the agent can take. Then, the environment $\mathcal{E}$ returns a reward r_t to the agent and the agent samples the next state s_{t+1} from the environment. This process is repeated until the terminal state is reached. In RL, the objective function is the total accumulated

[1]Note that all the bias terms are ignored here and all the rest RNN models in this dissertation for simplicity.

return R_t:

$$R_t = \Sigma_{k=0}^{\infty} \gamma^k r_{t+k}, \tag{2.6}$$

where $\gamma \in (0, 1]$ denotes the discount factor to trade off the influence of the immediate and later rewards. The goal of RL algorithms is to maximize R_t from each state s_t.

2.2 Traditional Approaches for Enforcing Structural Information

The aim of MIA is to extract key underlying information from medical images. Due to the existence of low contrast, extremely complex structures, noise which are commonly associated with medical images, achieving satisfactory results is challenging in clinical applications. Incorporating structural information has long been deemed as a useful approach for obtaining more plausible as well as accurate results, especially in these situations. This is because unlike the individual pixel/voxel values, structural information such as object shape and topological properties is less subjected to the influence of image corruptions, occlusions, and artifacts. Additionally, incorporating structural information in MIA could be potentially even more important compared to its adoption in natural image analysis as the anatomical structure of the objects are more distinct regarding the location and shape [54].

In standard machine learning problems, a model takes an image as input and only outputs a single output. In structured learning, the input/output contains multiple inter-dependent components. A tremendous amount of effort has been devoted to incorporating structural information into traditional MIA approaches. For instance, the structured support vector machine (structured SVM) [65] generalizes the classic SVM classifier [66] to take into account of the non-overlapping constraint of the cells. Graphical models (*e.g.*, hidden Markov models [67] and conditional random fields [68]) have also been very popular in modeling the structured output. These methods can capture the interdependencies among the input/output units by explicitly modeling

the correlations. Different types of structural information can be leveraged to improve MIA algorithms. We briefly introduce the following commonly used approaches [55]:

1. **User interaction:** Incorporating user's input into MIA pipelines is an intuitive way to better assist the computer-aided diagnosis (CAD) systems in achieving desired goals. In an interactive MIA system, the user is required to pass some input with certain important high-level prior structural knowledge. This prior high-level knowledge is considered to be unknown or extremely difficult to learn for computer algorithms. By passing the high-level prior structural knowledge into the interactive algorithm, the user does not need to worry about the low-level details of the MIA algorithms. The structural prior can take many specific forms such as object boundary specification, which requires the end-user to roughly draw a contour around an object [69, 70, 71] in a 2D image. Another convenient user input that was commonly used in algorithms such as grab-cut [72] and geodesic active contours [73] is the sub-region specification which requires a user to specify a bounding box around an object. These approaches have also been extended for 3D MIA [74, 75].

2. **Edge/boundary information:** Edge/boundary is powerful structural information of the objects in medical images. To effectively model this structural information, the drastic intensity/color change around the object edge/boundary is often leveraged. As an edge/boundary usually lies between pixels/voxels with different labels, this prior can be easily incorporated into the regularization term to decrease the penalties around the edges to allow for the discontinuities of label [76, 77]. In this way, it serves as a regulariser that confines the model to a more plausible model space. Nevertheless, these methods do not consider the direction of the intensity or color transition. To address this issue, different approaches have been proposed. For instance, Nosrati *et al.* [55] incorporate boundary polarity into the segmentation framework.

3. **Shape prior:** Shape semantically describes the objects in medical images. The shape of the object (*e.g.*, the ellipse shape of a cell) is often known as a priori. One simple approach to incorporate shape prior to MIA algorithms is to penalize any deviation of the prediction from the shape model [78]. More advanced algorithms also consider modeling the shape of non-rigid objects, *e.g.*, a heart, the shape of which gradually changes over time. For instance, the shape probability model was proposed in [79, 80] to capture the intra- and inter-subject variations.

4. **Topological Specification:** In order to obtain more accurate MIA results, the topological structure has to be preserved in many applications. There exist two common specifications of the topology: connectivity, and genus. They are often leveraged to ensure that the result is more topologically plausible. For instance, Han *et al.* [81] enforce topological preservation when conducting the level set-based segmentation. More specifically, the topology of the object contour is checked at each round of iteration to ensure that the genus keeps the same as before. In [82], Vicente *et al.* integrate the topological specification into an interactive paradigm by enforcing the connectivity.

These traditional approaches demonstrate the importance of incorporating the structural information into MIA for more plausible as well as accurate results. However, they face two major challenges. First, the structures need to be manually designed, which requires domain-specific knowledge and extensive tuning. As a result, a significant amount of time is required to tune the corresponding systems. Additionally, some underlying cues are usually difficult to be discovered, resulting in inferior results. Second, due to the high computational cost, these methods are limited to low dimensional circumstances and incapable of handling complex structured problems.

2.3 Explicit Structural Information Enforcement in Deep Learning

Deep learning, which automatically learns to capture discriminative features purely from data, has become a methodology of choice for analyzing big data since it gained popularity in 2012 [16]. Although deep neural networks (DNNs) are theoretically able to capture the structural information without the explicit specification of structural information, it comes at the cost of a requirement for a large-scale labeled high-quality training data, which is extremely expensive and difficult to collect in MIA. In this regard, enforcing structural information in deep learning algorithms brings an additional benefit: alleviating the need for a significant amount of labeled data. Early work on incorporating structural information to deep learning concentrated on enforcing sparse pixel connectivity through deep Boltzmann Machines (DBMs) [83]. Inspired by this line of work, Chen *et al.* [84] and Eslami *et al.* [85] employed DBM for the segmentation of vehicles and other objects. However, the fully connected formulation of DBM results in a complex model with a large number of parameters, easily leading to over-fitting. To address this limitation, Wu *et al.* [86] presented deep belief networks with convolutional layers to enforce the shape prior.

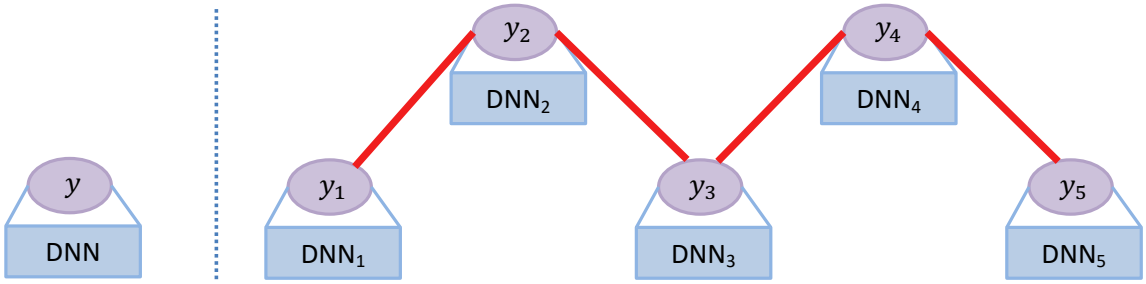

Figure 2.3: Modeling structures in the output with multiple components using deep learning. **Left:** a standard CNN. It takes an image as input and only outputs an output. **Right:** deep structured models. It takes multiple images as input and outputs multiple output variables. Note that the input/output variables are dependent on each other.

In the context deep learning and MIA, enforcing structural constraints in deep

learning has been adopted in many applications but has not been discussed in depth. Fig. 2.3 illustrates the difference between a standard DNN and deep structured models. A standard DNN takes an image as input and only outputs a single output. In deep structured models, the input/output contains multiple inter-dependent components. Existing deep structured learning for MIA can be briefly classified into two categories: modeling structural information with DNN architectures and enforcing prior structural knowledge in loss functions. Oftentimes, the input/output contains multiple inter-dependent components, as is illustrated in Fig. 2.3. Modeling the correlations in the input/output is usually achieved by explicitly designing certain network structures to consider the underlying structures in the input/output variables. For instance, in [87], an LSTM layer is used to capture the temporal cardiac dynamics in cardiac MRI sequences. There exist a considerable amount of applications with multiple interdependent input/output variables in MIA, which are briefly summarized as follows:

1. **Applications naturally requiring multiple inputs/output:** For many real-world applications, the input/output naturally contains multiple components (*e.g.*, image sequences). For example, Moeskops *et al.* [88] feed different image modalities into several CNNs. Afterward, all feature maps from these CNNs are combined to segment certain image components. Zhang *et al.* [49] generate reports for medical images, in which the outputs are sequences of words. Shin *et al.* [89] employ CNN to extract compact feature vectors from the corresponding medical images. Afterward, the feature vectors are fed into an RNN to generate sequential outputs. One classical MIA application that may need to explicitly consider structural information in the output is segmentation (*e.g.*, cells, glands, and organs), where fully convolutional neural networks (FCNN) [90] have been widely used. Every neuron in an FCNN is locally connected to the previous layer. With a series of convolutional layers, every output

unit is able to interact with other units. In this way, the correlations among the output units are considered. However, the resolution of the output degrades with the use of pooling layers. In order to address this issue, a multi-resolution approach is proposed in [91, 92]: the feature maps from different layers are concatenated together to jointly determine the final result. In this way, the low-level high-frequency information is recovered. Subsequently, Ronneberger *et al.* [38] expand on this idea one step further and introduced the so-called U-net. Stollenga *et al.* [93] alternatively choose to use the RNN to refine the results.

2. **The input/output needs to be separated:** The input/output sometimes needs to be separated into multiple parts so that different operations can be applied to each of them. One common reason is that many MIA problems involve handling images of extremely large size or higher dimensions. When dealing with these images, the sheer amount of data quickly saturate the GPU memories and the high computational requirement renders it infeasible to directly use the deep learning models on these images. A common practice is to divide the image data into multiple components. In this case, the result relies on all these input components. While features need to be extracted from all the individual components, the relationship among them should also be considered. For instance, Kong *et al.* [87] used LSTM to learn the correlations among images in a cardiac MRI sequence. Another reason is that by separating different components, individual operations can be applied to each of them. For instance, in [94], multiple ROIs of slit-lamp images are concatenated together, and the resulting fused feature map is further fed into a CNN to grade nuclear cataract. Chen *et al.* [91, 92] generate segmentation results from the gland images and then extracted gland boundaries. Afterward, the boundaries are used to refine the segmentation results.

Besides, prior structural knowledge regarding specific problems can be enforced in the loss function. For instance, Ronneberger *et al.* [38] assign the boundary pixels higher weights in the loss function to separate the overlapping cells. Kong *et al.* [87] model temporal constraints in the loss function to learn to detect ES and ED frames in cardiac MRI sequences. However, most of these discussed approaches are proposed (sometimes with brutal force) just for certain specific structured learning problems and only concentrate on solving specific problems at hand. In this dissertation, on the other hand, we investigate the general structured learning problem in MIA. We expand on the ideas in [32, 87] by presenting a general deep learning framework for structured learning in MIA.

CHAPTER 3: A GENERAL STRUCTURED LEARNING FRAMEWORK FOR MEDICAL IMAGE ANALYSIS

3.1 Motivation

MIA is the process of extracting clinically relevant information from medical images for the diagnosis, prognosis, as well as treatment planning. It involves handling images of various modalities such as positron emission tomography (PET), mammography, magnetic resonance (MR), computed tomography (CT). For the last decades, it has been demonstrated to be essential for the early diagnosis, detection, as well as treatment planning of different types of diseases [95]. When applying machine learning algorithms to MIA, the most important step is to design meaningful features for medical images. Traditionally, meaningful task-specific image features were designed by domain experts according to a specific task/dataset. As a result, it's extremely difficult to generalize to other tasks/datasets. Deep learning [96], on the other hand, integrates the feature engineering step into the whole machine learning pipeline. Specifically, instead of hand-designing image features, deep learning automatically extracts discriminative image features in a self-taught manner [97, 98]. This enables non-experts in machine learning to utilize modern deep learning tools for their specific MIA applications at hand. Recently, with the wide adoption of deep learning, many MIA tasks such as mitosis detection [5], X-ray image retrieval [6], cancer metastasis detection [7], skin cancer classification [8], fetus segmentation [9], and glaucoma assessment [10] have reached state-of-the-art performance, as is shown in Fig. 3.1.

Typically, in these applications, a standard DNN takes an image as input and only outputs a single output. However, the input/output in many MIA applications often-

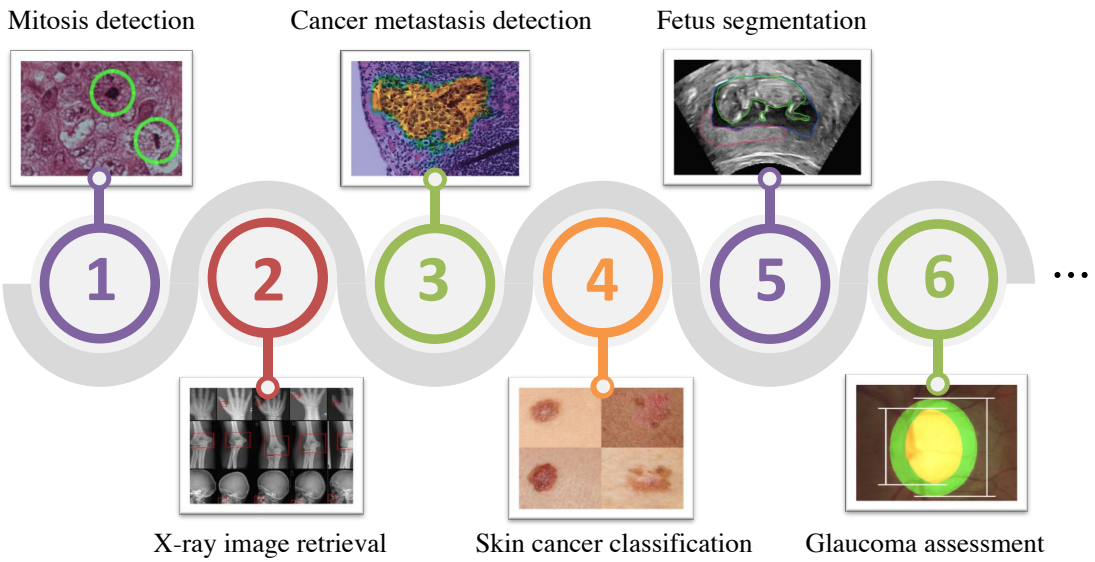

Figure 3.1: With deep learning-based methods quickly becoming a methodology of choice, MIA tasks such as mitosis detection [5], X-ray image retrieval [6], cancer metastasis detection [7], skin cancer classification [8], fetus segmentation [9], and glaucoma assessment [10] have reached the state-of-the-art performance.

times contains multiple interdependent components. A simple solution is to divide them into multiple components and analyze them individually. But the structural information, which is of great importance in these tasks, is ignored. A more advanced approach is to use simple postprocessing steps to consider simple interactions among different input/output variables. However, these techniques cannot model complex interdependencies in the input/output variables. Theoretically, deep neural networks are able to capture the structural information without the specification of a priori. Nevertheless, this comes at the cost of a requirement for a large-scale labeled high-quality training dataset, which is extremely expensive and difficult to collect in MIA. Although recent works have been proposed for deep structured learning in MIA, they are tailored for a specific application/dataset. In this dissertation, instead, we aim to propose a general and less task-specific framework for deep structured learning in MIA. The general framework is independent of particular tasks or deep learning architectures. As a result, it is applicable to a wide range of MIA tasks (*e.g.*, fetus segmentation, glaucoma assessment, and invasive cancer detection) and readily com-

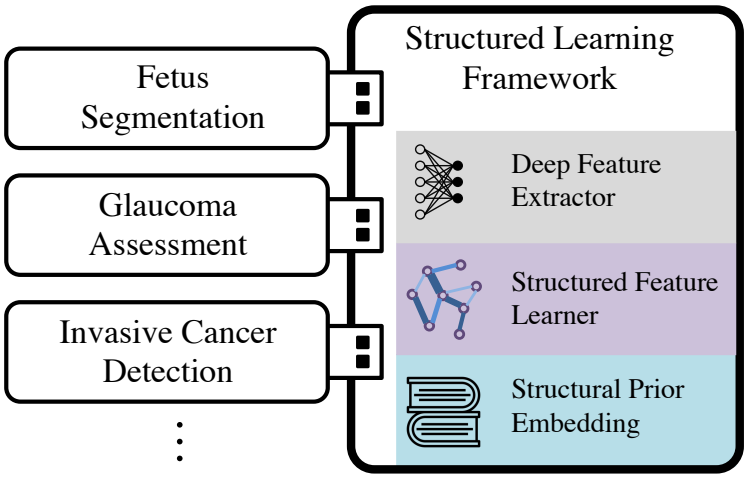

Figure 3.2: We propose a general and less task-specific framework for structured learning in MIA. The general framework is independent of particular tasks or deep learning architectures. As a result, it is applicable to a wide range of MIA tasks (*e.g.*, fetus segmentation, glaucoma assessment, and invasive cancer detection) and readily combined with the well-established neural network architectures.

bined with well-established neural network architectures, as is illustrated in Fig. 3.2.

3.2 Methodology

3.2.1 Overview

Fig. 3.3 presents an overview of the proposed deep structured learning framework. In this framework, we model the structural information in a unified neural network, which can be trained end-to-end. It has two key modules: deep feature extractor and structural feature learner. The intuition behind our framework is simple: the features in the input can be classified into two categories: features that can be extracted from individual input components and features that hold vital information about the complex interactions in the input/output variables. Thus, we leverage different neural network modules to model them respectively.

The deep feature extractor extracts discriminative features from each input component. The structural feature learner models the complex interactions and generates the final predictions. In the following, we first formally formulate the problem to be solved in section 3.2.2. Then, these two modules are expanded in section 3.2.3

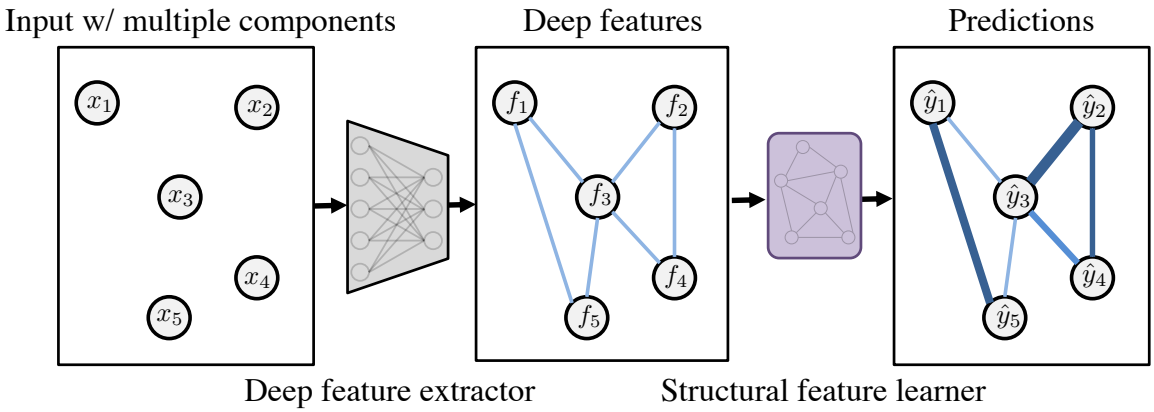

Figure 3.3: An overview of the proposed structured learning framework. In our network, we model the structural information in a unified neural network, which can be trained end-to-end. It has two key modules: deep feature extractor, structural feature learner. The deep feature extractor extracts discriminative features from the input. The structural feature learner models the complex interactions in the input/output variables and generates the final predictions.

and 3.2.4. Afterward, we present the objective function to optimize the network in section 3.2.5. Finally, two real-world applications are presented to further illustrate the proposed framework in in section 3.3.

3.2.2 Mathematical Formulation

Given the input $\mathbf{x}$ and let $\mathbf{y}$ be the output we aim to predict. The goal is to leverage the available structural information in the training data to train a strong model for the unseen testing data. In the structured learning setting, the input and output are composed of M ($M \geq 1$) and N ($N \geq 1$) inter-correlated components respectively. In other words, $\mathbf{x}$ and $\mathbf{y}$ can be decomposed into multiple components: $\mathbf{x} = (x_1, x_2 \cdots x_M)$ and $\mathbf{y} = (y_1, y_2 \cdots y_N)$. The model we are interested in learning can be denoted by a complex function η_V parametrized by V, *i.e.*, $\mathbf{y} = \eta_V(\mathbf{x})$. In this dissertation, we aim to formulate a fully differentiable deep learning framework for various structured learning tasks in MIA, so that it can be trained in an end-to-end manner.

3.2.3 Deep Feature Extractor

One key aspect of deep structured learning to extract features from each input component. Extracting discriminative features for the input images is vital as the subsequent structural feature learner is able to focus on modeling complex structural information. In the proposed deep structured learning framework, we use CNNs as the deep feature extractor to automatically learn to extract discriminative features $\mathbf{f} = (f_1, f_2 \cdots f_M)$ from the input $\mathbf{x}$, as is demonstrated in Fig. 3.3. It can be denoted by a differentiable function ϕ_U parameterized by U. We further illustrate this idea in Fig. 3.4 with a simple MIA task: classifying the pathology image patches into two categories, *i.e.*, tumor/normal. From the perspective of manifold learning, the tumor (red triangles) and normal (black circles) images are mixed together in the input manifold. After feature extraction using the proposed deep feature extractor, the tumor and normal images are separated apart.

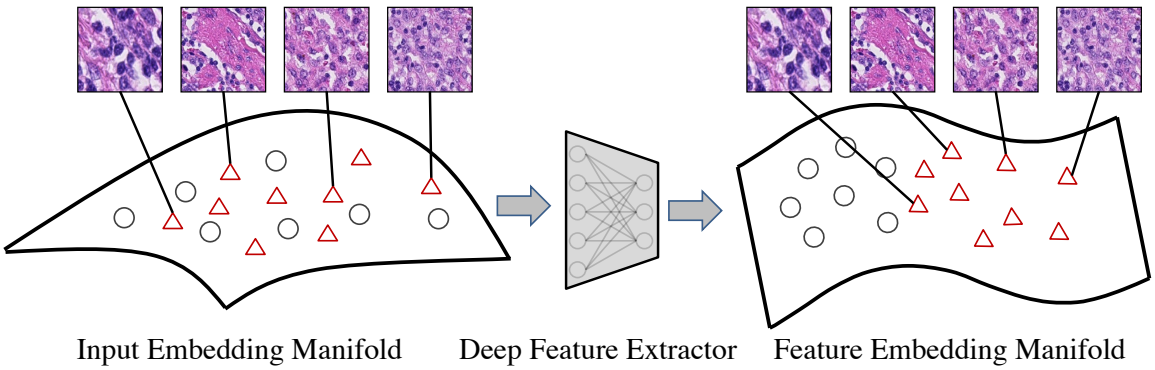

Figure 3.4: We illustrate this idea with a simple MIA task: classifying the pathology image patches into two categories, *i.e.*, tumor/normal. From the perspective of manifold learning, the tumor (red triangles) and normal (black circles) images are mixed together in the input embedding manifold. After feature extraction with the proposed deep feature extractor, the tumor and normal images are separated apart.

Apart from extracting discriminative features from the input, the dimension of the input data is significantly reduced by the pooling layers in the deep feature extractor to avoid the curse of dimensionality. Commonly, $x_1, x_2 \cdots x_M$ are of the same shape, *e.g.*, the frames in a cardiac MRI sequence. In this case, all these CNNs can share

weights to avoid overfitting. Similar architectures have also been used for deep structured learning in previous works in MIA. For example, a CNN is employed to extract discriminative features in [87] from each frame of a cardiac sequence, which are then fed into an LSTM layer to generate the final predictions. We can also interpret this from the perspective of multi-task learning [88]. Specifically, deep feature extractor is responsible for extracting different types of features from each individual input component. As a result, these auxiliary tasks introduce an inductive bias to help the model to explain multiple tasks and thus helps it to generalize better.

3.2.4 Structural Feature Learner

After extracting features from each input component, the structural feature learner considers the inter-correlations in them. From a complementary perspective, the structural feature learner complements the standard deep neural network with a module for modeling key structural information. Similar to the deep feature extractor, we also use a fully differentiable deep neural network to model the complex interactions and generate the final prediction. Specially, structural feature learner ψ_W (parameterized by W) takes as input the extracted features $\mathbf{f} = (f_1, f_2 \cdots f_M)$ and produces the final prediction $\hat{\mathbf{y}} = (\hat{y}_1, \hat{y}_2 \cdots \hat{y}_N)$. We further illustrate this idea in Fig. 3.5 with a simple task: predicting the label (tumor/normal) of the pathology image patch indicated by the red rectangle. This is a hard example. Purely judging by the features extracted by the deep feature extractor can easily lead to misclassification as its appearance/texture is very similar to the green tumor patch (red triangles in the embedding manifolds). However, considering the inter-correlation between this patch and its neighbors (black normal patches, indicated by black circles in the embedding manifolds) effectively addresses this problem. By carefully designing the structural learner, the features extracted from the hard example can be remapped to separate it from the normal image patches.

While different methods can be used to model the complex interactions, we rec-

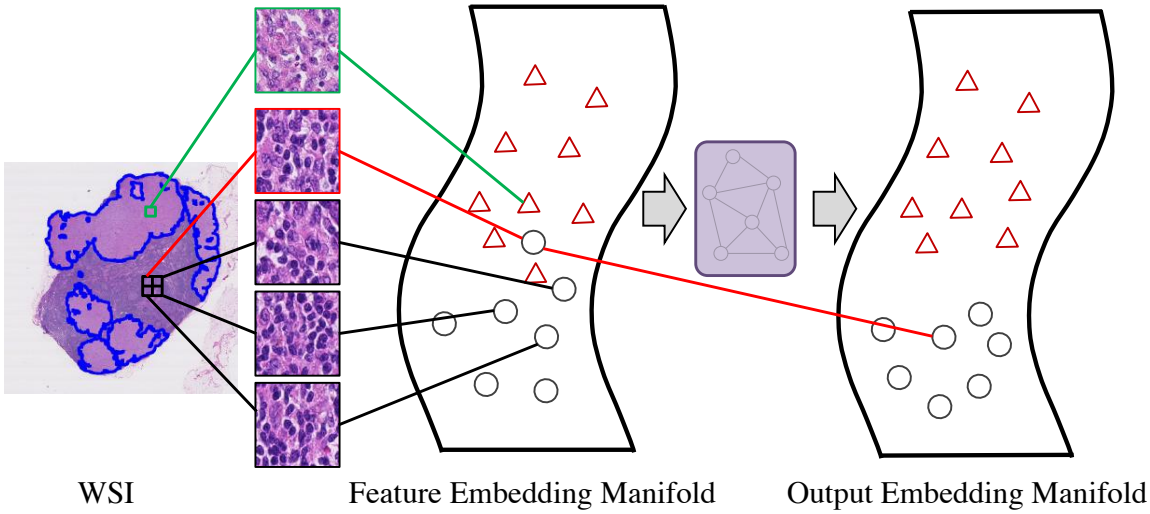

Figure 3.5: We further illustrate this idea with a simple task: predicting the label (tumor/normal) of the pathology image patch indicated by red rectangle. Purely judging by the features extracted by the deep feature extractor can easily lead to misclassification as its appearance/texture is very similar to the green tumor patch (red triangle in the feature embedding manifold). However, considering the inter-correlation between this patch and its neighbors effectively address this problem and remap the feature to separate the tumor and black normal image patches (indicated by black circles in the embedding manifolds).

ommend designing an aggregator module that can better leverage the underlying structures in the individual features. As two examples for designing effective structural feature learner, we will present two MIA applications to show how to adopt the proposed framework for regression and classification tasks in section 3.3. In these two tasks, we demonstrate how to use LSTM to model the sequential dynamics in section 3.3.1 and how to employ 2D LSTM layers to model the spatial correlations of pathology image patches in WSIs in section 3.3.2.

3.2.5 Optimization

Together, the deep feature extractor ϕ_U and structural feature learner ψ_W define a fully differentiable neural network η_V, where $\eta_V(\mathbf{x}) = (\varphi_U \circ \psi_W)(\mathbf{x})$ and $V = (U, W)$. In MIA applications, the training set $\mathbb{D} = \{(\mathbf{x}^j, \mathbf{y}^j)\}_{j=1}^J$ (here, $J = |\mathbb{D}|$ is the number of data examples in $\mathbb{D}$) is associated with a repository of medical images $\mathbf{X} = \{\mathbf{x}^j\}_{j=1}^J$, and the corresponding labels $\mathbf{Y} = \{\mathbf{y}^j\}_{j=1}^J$. After introducing

Algorithm 1 The whole training pipeline of the proposed structured learning framework η_V with parameter V.

Input: $\mathbb{D}$ = training set
Input: ϕ_U = deep feature extractor with parameter U
Input: ψ_W = structural feature learner with parameter W
Input: μ = learning rate
while not converged
  $g \leftarrow 0$
  for every training example in the sampled mini batch $(\mathbf{x}^j, \mathbf{y}^j) \in \mathbb{D}$ **do**
    extract deep features with: $\mathbf{f}^j \leftarrow \phi_U(\mathbf{x}_j)$
    model the structural information and generate final prediction: $\hat{y}^j \leftarrow \psi_W(\mathbf{f}^j)$
    accumulate gradients: $g \leftarrow g + \frac{\partial \mathcal{L}_{total}(\hat{\mathbf{y}}^j, \mathbf{y}^j)}{\partial V}$ according to equation 3.1
  end for
  $V \leftarrow V - \mu g$
end while
return parameters V

the proposed framework, the next step is to define the loss function and train our network with the training set:

$$\mathcal{L}_{total}(\eta_V(\mathbf{X}), \mathbf{Y}) = \sum_{j=0}^{J} \mathcal{L}_{task}(\eta_V(\mathbf{x}^j), \mathbf{y}^j) + \alpha \mathcal{L}_{reg}(V), \tag{3.1}$$

$$\mathcal{L}_{reg}(V) = \frac{1}{2}||V||_2^2, \tag{3.2}$$

where $\mathcal{L}_{task}(\eta_V(\mathbf{x}^j), \mathbf{y}^j)$ is the task-specific loss. For instance, cross-entropy loss can be used for classification problems. $\mathcal{L}_{reg}$ denotes the regularization term. It ensures sparse weights to avoid overfitting. The hyper-parameters α is cross-validated during the training process.

As the whole framework is fully differentiable, it can be trained end-to-end. and training the proposed deep structured learning framework is equivalent to the following optimization problem. The detailed training procedure is shown in Algorithm 1.

$$V^* = \arg\min_V \mathcal{L}_{total}(\eta_V(\mathbf{X}), \mathbf{Y}). \tag{3.3}$$

3.3 Applications of Structured Learning in MIA

In this section, we employ the proposed structured learning framework in two MIA tasks to thoroughly elucidate our approach: recognizing ED and ES frames from cardiac MRI sequences and metastasis detection in whole-slide images (WSIs).

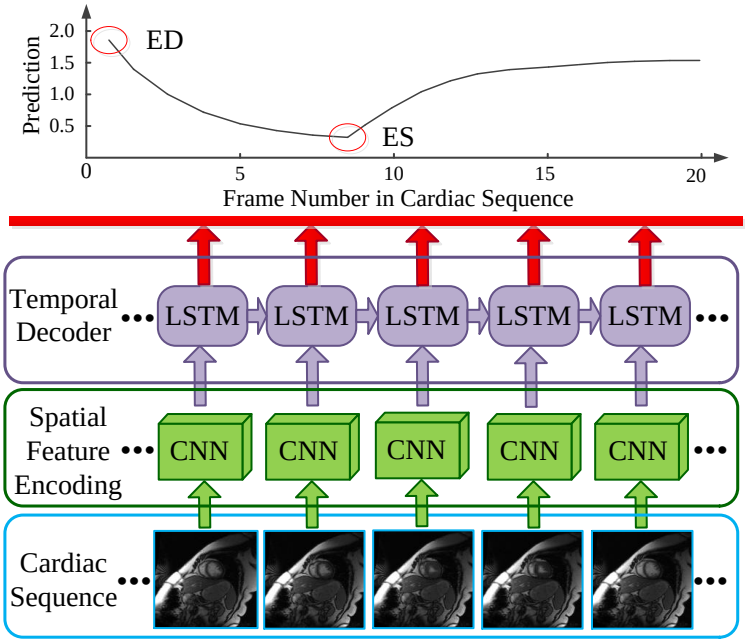

Figure 3.6: The outline of the TempReg-Net. It consists of two key components: spatial feature encoding and temporal decoder, which correspond to the deep feature extractor and structural feature learner respectively in our deep structured learning framework. The deep feature learner extracts discriminative features from the input and the structural feature learner is responsible for discovering the structures lying behind. A fully connected layer in the structural feature learner generates the final predicted values. Finally, the ES, as well as ED frames, are recognized, based on the predictions.

3.3.1 Cardiac MRI Recognizing

Recognizing ED and ES frames from cardiac MRI is the first step for many cardiac image analysis applications. Traditional methods [1, 99, 100] for this problem are based on "hand-crafted" features. Nevertheless, the results are error-prone. Applying deep learning algorithms to this task is extremely challenging, considering the almost unnoticeable changes in the cardiac image frames (see Fig. 1.1). In this chapter, based

on the deep structured learning framework, we introduce temporal regression network (TempReg-Net) to tackle this problem. Its outline is presented in Fig. 3.6. It consists of two key components: spatial feature encoding and temporal decoder, which are correspond to the deep feature extractor and structural feature learner respectively in our deep structured learning framework.

In this network, the deep feature extractor ϕ_U consists of M CNNs for extracting the spatial features from the M frames of the cardiac sequence (M is the total number of frames in the cardiac sequence), yielding the encoded features for the cardiac frames. The LSTM layer in the structural feature learner ψ_W is responsible for modeling temporal dynamics. Finally, a fully connected layer in the structural feature learner generates M predictions for the cardiac sequence (we generate a prediction for each frame, so $M = N$ in this application).

More specifically, the whole framework consists of three steps. Firstly, M CNNs are trained to extract spatial patterns from the input frames, generating M compact features vectors. Afterward, these features are fed into an LSTM model to explore the temporal correlation in the sequence. Then, the final predictions are produced by a fully-connected layer. Finally, the ED and ES frames are detected by locating the maximum and minimum predictions respectively. Note that the network is trained to regress a continuous numeric value for each frame in the cardiac sequence, where each value represents the left ventricular volume in the frame. It is noted that the weights of M CNNs are shared for all frames to avoid over-fitting.

Given the training set $\mathbb{D} = \{(\mathbf{x}^j, \mathbf{y}^j)\}_{j=1}^J$, our goal is to optimize the following loss function:

$$\mathcal{L}_{total} = \sum_{j=1}^{J} \sum_{m=1}^{M} ||y_m^j - \hat{y}_m^j||^2 + \alpha \mathcal{L}_{reg}, \tag{3.4}$$

$$\mathcal{L}_{reg} = \frac{1}{2}(||V||_2^2), \tag{3.5}$$

where $\hat{\mathbf{y}}^j = \eta_V(\mathbf{x}^j)$ is the prediction. $\mathbf{y}^j = (y_1^j, y_2^j \cdots y_M^j)$ is the synthetic ground truth, which will be discussed later. $\mathcal{L}_{reg}$ regularizes our system by controlling the complexity of TempReg-Net, *i.e.*, the sparsity of the learned weights V. α is the hyper-parameter, which is cross-validated during the training phase.

In this application, $\sum_{j=1}^{J}\sum_{m=1}^{M} ||y_m^j - \hat{y}_m^j||^2$ corresponds to the task-specific loss in equation 3.1. To model the dynamics of left ventricle volume [1], the ground truth label y_m^j is synthesized according to the following equation:

$$y_m^j = \begin{cases} \left|\frac{m-N_{es}}{N_{es}-N_{ed}}\right|^{\delta}, & \text{if} \quad N_{ed} < m \leq N_{es} \\ \left|\frac{m-N_{es}}{N_{es}-N_{ed}}\right|^{\upsilon}, & \text{otherwise} \end{cases} \tag{3.6}$$

where N_{es} and N_{ed} denote the corresponding ES and ED frame indices in the cardiac MRI sequence $\mathbf{x}^j$. The hyper-parameters δ and υ control the shape of the ventricle volume curve. They are cross-validated during the training phase.

3.3.2 Cancer Metastasis Detection in WSIs

WSIs are the golden standard in clinical diagnosis, which provide histopathological information for accurate analysis. As WSIs are massive (*e.g.*, $150,000 \times 150,000$ pixels), current practice [3] simply divides the WSIs into small patches and employs CNNs to assign a prediction value to each patch for the final diagnosis decisions. However, it does not consider the structural information in WSIs, as illustrated in Fig. 1.2. The structural information in pathology images was proven to be vital for cancer diagnosis in [101] even before the widely adoption of deep learning in MIA.

We now present our Spatio-Net for detecting cancer metastasis in WSIs. It is based on the deep structured learning framework proposed in section 3.2 to model the structural information among the image patches. The top row of Fig. 3.7 provides an overview of the cancer metastasis detection pipeline. Firstly, each WSI is divided into small image patches with fixed-size. Secondly, the Spatio-Net produces a probability

map for each WSI, indicating the malignancy probability of each image patch, through which the metastases are located.

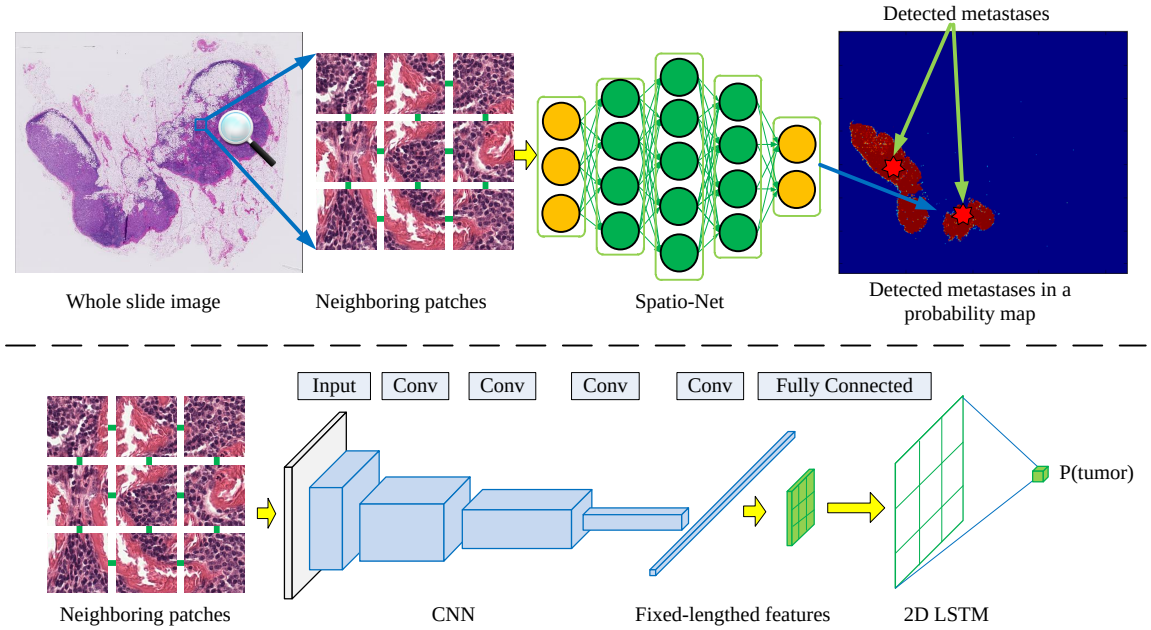

Figure 3.7: A schematic overview of the proposed Spatio-Net. For each image patch, we consider its neighboring patches. The spatio-Net generates the probability map. The metastases are located by interpreting these maps. The top and bottom row show the whole pipeline and the detailed structure of Spatio-Net respectively. The CNNs (deep feature extractor) extract features from each patch and its neighbors. 2D LSTM layers (structural feature learner) considers the inter-patch dependencies. A fully connected layer in the structural feature learner predicts a malignancy probability for each patch.

The bottom row of Fig. 3.7 shows the detailed structure of Spatio-Net. The CNN and 2D LSTM layer correspond to the deep feature extractor and structural feature learner in our deep structured learning framework respectively. The deep feature extractor ϕ_U consists of M deep residual network [102] for extracting discriminative features from each patch and its surrounding 8 patches (*i.e.*, $M = 9$). Note that these networks share the same weights. The structural feature learner ψ_W is composed of four 2D LSTM layers for aggregating the features extracted from the patches and a fully connected layer for the final prediction. Thanks to the capability of 2D LSTM layer for modeling spatial patterns, spatial information in these feature vectors can

be naturally modeled by the proposed Spatio-Net. The fully connected layer followed by a sigmoid function generates M predictions for the M patches (we generate a prediction for each frame, so $N{=}M = 9$ in this application).

Given the training set $\mathbb{D}$, we need to estimate the optimal weights for the CNN and 2D LSTM layers. We define the total loss function for the whole framework as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{cls} + \alpha\mathcal{L}_{reg}, \tag{3.7}$$

$$\mathcal{L}_{reg} = \frac{1}{2}(||V||_2^2), \tag{3.8}$$

$$\mathcal{L}_{cls} = -\sum_{x_* \in \mathbb{D}} \log(\eta_V(x_*)), \tag{3.9}$$

where $\mathcal{L}_{cls}$ is the total cross-entropy loss. $\mathcal{L}_{reg}$ regularizes our system by controlling the complexity of Spatio-Net, as discussed previously. α controls the weights of three loss terms, which is cross-validated during the training stage. V is all the learnable weights in Spatio-Net.

In this application, $\mathcal{L}_{cls}$ corresponds to the task-specific loss in equation 3.1. Following [3], two post-processing steps are employed to locate the metastases from the probability map. First, a binary mask image is generated by thresholding the probability map. Then, connected component analysis is used to label each region, with the center of each region being the predicted metastasis location and the mean probability of the region being the final score.

3.4 Summary

In this chapter, a novel deep learning framework is proposed to tackle the structured learning problems in MIA. More specifically, our framework comprises of two major components: 1) deep feature extractor, which extracts features from each input component, 2) structural feature learner, which precisely models the structural information and generates the final predictions. By employing these approaches, our

method can handle structured learning problems with a unified framework. We elucidate the proposed method on two MIA tasks: cardiac MRI recognizing and cancer metastasis detection in WSIs. As the proposed framework is independent of particular tasks or deep learning architectures it is applicable to a wide range of MIA tasks and readily combined with well-established neural network architectures.

CHAPTER 4: EMBEDDING PRIOR STRUCTURAL KNOWLEDGE IN LOSS FUNCTIONS

4.1 Motivation

In the last chapter, we demonstrated the importance of structural information in MIA and defined a general deep structured learning framework for modeling structural information in MIA systems. However, there also exist some prior structural information that can hardly be enforced in deep learning frameworks. Compared with other types of structural information, they often contain high-level understandings of human experts, which is extremely difficult for deep neural networks to capture. Therefore, solely relying on deep learning to model these high-level structured prior knowledge is not enough for producing satisfactory results. The important role of embedding structured prior in MIA systems is also highlighted in many previous research works. For instance, Ronneberger *et al.* [38] stress the boundary pixels by giving them higher weights so that touching cells can be separated. Chen *et al.* [91, 92] also demonstrate that giving higher weights to gland boundaries proves to be beneficial for separating overlapping glands. To address this issue, we complement the proposed deep structured learning framework with a novel training strategy by incorporating an additional structured loss term in the loss function. It enables the model to follow the prior structural knowledge on the solution space, thereby generating more anatomically reasonable and accurate results.

In this chapter, we demonstrate how to embed prior structural knowledge in loss functions and further illustrate our idea with more concrete examples in section 4.2. To show that incorporating this prior knowledge is indeed beneficial for the final accuracy, we conduct experiments in section 4.4 to evaluate the proposed approaches.

4.2 Methodology

4.2.1 Embedding Prior Structural Knowledge in Loss Functions

Solely relying on deep neural networks to model structural information is not enough in many MIA problems. To further enforce the high-level prior structural knowledge into the proposed deep structured learning framework, we choose to incorporate this information in the training procedure by adding an additional term in the loss function (equation 3.3). Figure 4.1 shows the training strategy of the proposed structured learning framework. Different from chapter 3, which only use the task-specific loss $\mathcal{L}_{task}$ to update the parameters in the deep structured learning framework, we also employ the prior structural loss $\mathcal{L}_{prior}$ to complement the training.

More specifically, at each iteration, predictions of the structured learning framework are compared with the ground truth labels to compute the task-specific loss $\mathcal{L}_{task}$. This loss term is used to generate the gradients $\frac{\partial \mathcal{L}_{task}}{\partial U}$ and $\frac{\partial \mathcal{L}_{task}}{\partial W}$ to update the deep feature extractor ϕ_U and the structural feature learner ψ_W respectively. This loss only consider one output component as a time and doesn't consider the dependencies in them, as illustrated in Figure 4.1.

To integrate this higher-level prior structural information, we further compute the prior structural loss $\mathcal{L}_{prior}$ and compute gradients $\frac{\partial \mathcal{L}_{prior}}{\partial U}$ as well as $\frac{\partial \mathcal{L}_{prior}}{\partial W}$ to update the parameters of ϕ_U and ψ_W respectively. In summary, the updated total loss for training the deep structured learning framework can be expressed as follows:

$$\mathcal{L}'_{total}(\eta_V(\mathbf{X}), \mathbf{Y}) = \sum_{j=0}^{J} \mathcal{L}_{task}(\eta_V(\mathbf{x}^j), \mathbf{y}^j) + \alpha \mathcal{L}_{reg} + \beta \mathcal{L}_{prior}, \tag{4.1}$$

$$\mathcal{L}_{reg} = \frac{1}{2}||V||_2^2, \tag{4.2}$$

where $\mathcal{L}_{task}$ is the task-specific loss, *e.g.*, cross-entropy loss for classification problems. $\mathcal{L}_{reg}$ denotes the regularization term. It ensures sparse weights after training to avoid

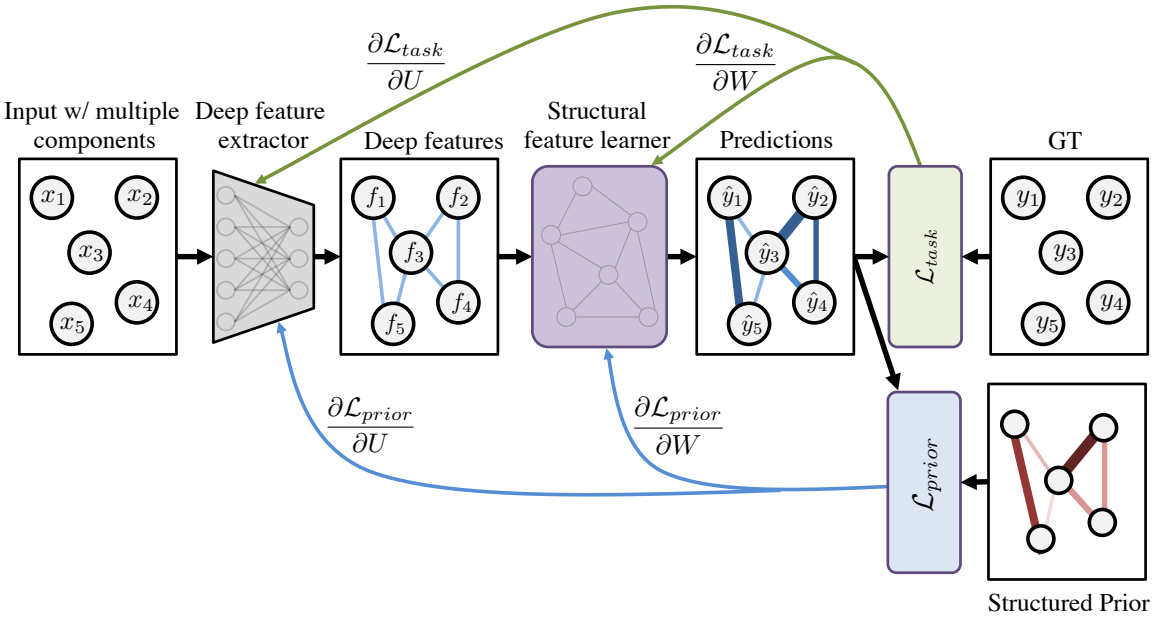

Figure 4.1: Different from chapter 3 which only use the task-specific loss $\mathcal{L}_{task}$ to update the deep structured learning framework, we also employ the prior structural loss $\mathcal{L}_{prior}$ to complement the training. More specifically, at each iteration, predictions of the structured learning framework are compared with the ground truth labels to compute the task-specific loss $\mathcal{L}_{task}$. This loss term is used to generate the gradients $\frac{\partial \mathcal{L}_{task}}{\partial U}$ and $\frac{\partial \mathcal{L}_{task}}{\partial W}$ to update the deep feature extractor ϕ_U and the structural feature learner ψ_W respectively. This loss only consider one output component as a time and doesn't consider the dependencies in them. To integrate this higher-level prior structural information, we further compute the prior structural loss $\mathcal{L}_{prior}$ and compute gradients $\frac{\partial \mathcal{L}_{prior}}{\partial U}$ as well as $\frac{\partial \mathcal{L}_{prior}}{\partial W}$ for ϕ_U and ψ_W respectively to update their parameters.

overfitting. $\mathcal{L}_{prior}$ enforces the prior structural knowledge in the loss function. The hyper-parameters, α and β , are cross-validated during the training process. As the whole framework is fully differentiable, it can be trained end-to-end with back-propagation. The whole training procedure is also illustrated in Algorithm 2.

4.2.2 Illustration with Two MIA Applications

We further illustrate our idea with two more concrete examples. As a simple example, a typical cardiac MRI sequence in illustrated in Fig. 1.1. With the contraction and relaxation of the heart, the left ventricle (red rectangles) volume gradually diminishes and expands in these cardiac image frames. The maximum and minimum

Algorithm 2 Incorporating the structured prior knowledge into the training procedure by adding an additional loss term into the loss function. The structured learning framework is denoted by η_V with parameter V.

Input: $\mathbb{D}$ = training set
Input: ϕ_U = deep feature extractor with parameter U
Input: ψ_W = structural feature learner with parameter W
Input: μ = learning rate
while not converged
  $g \leftarrow 0$
  for every training example in the sampled mini batch $(\mathbf{x}^j, \mathbf{y}^j) \in \mathbb{D}$ **do**
    extract deep features with: $\mathbf{f}^j \leftarrow \phi_U(\mathbf{x}_j)$
    model the structural information and generate final prediction: $\hat{y}^j \leftarrow \psi_W(\mathbf{f}^j)$
    accumulate gradients: $g \leftarrow g + \frac{\partial \mathcal{L}'_{total}(\hat{\mathbf{y}}^j, \mathbf{y}^j)}{\partial V}$ according to equation 4.1
  end for
  $V \leftarrow V - \mu g$
end while
return parameters V

left ventricular volumes correspond to the end-diastole (ED) and end-systole (ES), respectively. In chapter 3, we leverage the proposed deep structured learning framework to model temporal structures underlying in cardiac sequences. However, an important high-level aspect of the prior structural information regarding to the predictions is difficult to be captured solely by deep neural networks: the left ventricular volume only increase in diastole and decrease in systole phases [52]. Simply employing TempReg-Net (see Fig. 3.6) is difficult to capture this high-level knowledge. To address this issue, we complement the proposed deep structured learning framework with a generic training strategy by incorporating a new loss term in section 4.3.1. This strategy encourages the deep network to follow the prior structural knowledge on the solution space. As will be demonstrated in section 4.4.1, this significantly improves the prediction by smoothing out the results.

As another example, consider an image region in a WSI in Fig. 1.2, which includes 9 patches. In this image region, we are interested in predicting the malignancy of the center red patch. Except for the top left green patch, all the other surrounding patches

are tumorous. Without other available information, it is reasonable to infer that the center patch (red) is likely to be tumorous. This reasoning process reflects our prior knowledge about the spatial constraint or structure of image patches underlying in the WSIs. The prior knowledge about the nuclei distribution has also been leveraged for cancer diagnosis in [103].

4.2.3 Distinction with Maximum a Posteriori

This framework can be interpreted from the perspective of maximum a posteriori (MAP). Specifically, when estimating the parameters of the deep structured learning framework, we are essentially estimating the conditional probability $P(\mathbf{Y}|\mathbf{X};V)$, *i.e.*, we are interested in predicting $\mathbf{Y}$ given $\mathbf{X}$. $\mathbf{X}$ and $\mathbf{Y}$ are the input collection and target collection in training set $\mathbb{D}$ respectively. Without considering $\mathcal{L}_{reg}$, equation 4.1 is equivalent to the following objective function:

$$V_{MLE} = \arg\max_{V} P(\mathbf{Y}|\mathbf{X};V), \tag{4.3}$$

which is standard maximum likelihood estimation (MLE). Usually, equation 4.3 can be decomposed as follows, assuming that data examples in $\mathbb{D}$ are i.i.d.:

$$V_{MLE} = \arg\max_{V} \sum_{j=1}^{J} \log P(\mathbf{y}^j|\mathbf{x}^j;V). \tag{4.4}$$

In contrast, MAP works on the Bayesian posterior distribution, which can be decomposed as a prior and likelihood. Similar to equation 4.3 and 4.4, by incorporating a prior distribution, we get the following objective function:

$$V_{MAP} = \arg\max_{V} P(\mathbf{Y}|\mathbf{X};V)P(V), \tag{4.5}$$

$$= \arg\max_{V} \log P(\mathbf{Y}|\mathbf{X};V) + \log P(V), \tag{4.6}$$

$$= \arg\max_{V} \log \prod_{j=1}^{J} P(\mathbf{y}^j|\mathbf{x}^j; V) + \log P(V), \tag{4.7}$$

$$= \arg\max_{V} \sum_{j=1}^{J} \log P(\mathbf{y}^j|\mathbf{x}^j; V) + \log P(V). \tag{4.8}$$

Comparing equation 4.8 with equation 3.3 and 4.1, we can draw the conclusion that $\mathcal{L}_{task}$ and $\mathcal{L}_{prior}$ are corresponding to the first term in equation 4.8, which are guided by the labeled training data. $\mathcal{L}_{prior}$ corresponds to the second term in equation 4.8. It encompasses our prior understanding of the system. In our case, we assume that the system should not be too complex. By incorporating this term, the trained system is less likely to overfit the training dataset. Note the difference between the prior distribution in MAP and the prior structural knowledge in equation 4.1: the prior structural loss in equation 4.1 is guided by training examples.

4.3 Applications in MIA

In this section, we further consider incorporating prior structural knowledge into the training procedure of the MIA systems discussed in section 3.3.

4.3.1 Cardiac MRI Recognizing

As is mentioned in section 4.2.2, the prior knowledge about the left ventricular volume (*i.e.*, it does not decrease in a diastole stage and increase in a systole stage) can be leveraged to improve the predictions. Essentially, TempReg-Net defines a regressor η_V based on the cardiac MRI sequence $\mathbf{x}^j$ with M frames, where V is all the learnable weights in the deep feature extractor and structural feature learner. Ideally, the predicted value $\hat{\mathbf{y}}^j = \eta_V(\mathbf{x}^j)$ conform with the prior knowledge: the prediction of the m^{th} frame $\hat{y}^j_m$ $(m = 1, 2, ...M)$ should be larger than or equal to $\hat{y}^j_{m-1}$ if there are in the diastole phase (*i.e.*, $\hat{y}^j_{m-1} < \hat{y}^j_m$), and vice versa. We model this constraint in

the loss function by adding an additional term, temporal structured loss $\mathcal{L}_{temp}$:

$$\mathcal{L}_{temp} = \frac{1}{2}(\mathcal{L}_{inc} + \mathcal{L}_{dec}), \tag{4.9}$$

$$\mathcal{L}_{inc} = \frac{1}{M}\sum_{m=2}^{M} \mathbb{1}(y_m > y_{m-1}) \max(0, \hat{y}^j_{m-1} - \hat{y}^j_m), \tag{4.10}$$

$$\mathcal{L}_{dec} = \frac{1}{M}\sum_{m=2}^{N} \mathbb{1}(y_m < y_{m-1}) \max(0, \hat{y}^j_m - \hat{y}^j_{m-1}), \tag{4.11}$$

where $\mathcal{L}_{inc}$ and $\mathcal{L}_{dec}$ penalize the false predictions. More specifically, they are 0 if the predictions conform with the rules mentioned above. $\mathcal{L}_{inc}$ is positive if the predictions decrease, but the corresponding frames are in a diastole phase. $\mathcal{L}_{dec}$ is positive if the predictions increase, but the corresponding frames are in a systole phase. $\mathbb{1}(\cdot)$ denotes the indicator function.

After the definition of temporal structured loss, we further explore the loss function. Given the training set $\mathbb{D} = \{(\mathbf{x}^j, \mathbf{y}^j)\}_{j=1}^J$, our goal is to optimize the following loss function:

$$\mathcal{L}'_{total} = \sum_{j=1}^{J}\sum_{m=1}^{M} ||y^j_m - \hat{y}^j_m||^2 + \alpha\mathcal{L}_{reg} + \beta\mathcal{L}_{temp}, \tag{4.12}$$

$$\mathcal{L}_{reg} = \frac{1}{2}(||V||_2^2), \tag{4.13}$$

where $\hat{\mathbf{y}}^j = \eta_V(\mathbf{x}^j)$ is the prediction. $\mathbf{y}^j = (y^j_1, y^j_2, ..., y^j_M)$ is the synthetic ground truth. $\mathcal{L}_{reg}$ regularizes our system by controlling the complexity of TempReg-Net, *i.e.*, the sparsity of the learned weights V. α and β are the hyper-parameters of our system. They are cross-validated during the training phase.

4.3.2 Cancer Metastasis Detection in WSIs

Additionally, we further propose a new loss function $\mathcal{L}_{spatio}$ to enforce the prior structural knowledge in the training procedure of Spatio-Net. It is termed as spatially structured loss. Spatio-Net defines a classifier $\hat{\mathbf{y}}^{(i)} = (\hat{y}^{(i)}_1, \hat{y}^{(i)}_2, ..., \hat{y}^{(i)}_M) = \eta_V(\mathbf{x}^{(i)})$.

Specifically, we penalize two situations: 1) $|y_*^{(i)} - y_m^{(i)}|$ is small but $\hat{y}_*^{(i)} = \hat{y}_m^{(i)}$, which is corresponding to $\mathcal{L}_{ind}$ in the following, 2) $|y_*^{(i)} - y_m^{(i)}|$ is large but $\hat{y}_*^{(i)} \neq \hat{y}_m^{(i)}$, which is corresponding to $\mathcal{L}_{dif}$ in the following. We enforce this prior structural knowledge in the loss function:

$$\mathcal{L}_{spatio} = \frac{1}{2}(\mathcal{L}_{ind} - \mathcal{L}_{dif}), \tag{4.14}$$

$$\mathcal{L}_{ind} = \frac{1}{N}\sum_{j=1}^{J}\sum_{m\in\mathcal{N}_*}\{\mathbb{1}(y_*^{(i)} = y_m^{(i)}) \cdot |\hat{y}_*^{(i)} - \hat{y}_m^{(i)}|^2\}, \tag{4.15}$$

$$\mathcal{L}_{dif} = \frac{1}{N}\sum_{j=1}^{J}\sum_{m\in\mathcal{N}_*}\{\mathbb{1}(y_*^{(i)} \neq y_l^{(i)}) \cdot |\hat{y}_*^{(i)} - \hat{y}_m^{(i)}|^2\}, \tag{4.16}$$

where $\mathbb{1}(\cdot)$ denotes the indicator function.

Given the training set $\mathbb{D}$, we need to estimate the optimal weights for the CNN and 2D LSTM layers. It can be achieved by optimizing the following loss function:

$$\mathcal{L}'_{total} = \mathcal{L}_{cls} + \alpha\mathcal{L}_{reg} + \beta\mathcal{L}_{spatio}, \tag{4.17}$$

$$\mathcal{L}_{reg} = \frac{1}{2}(||V||_2^2), \tag{4.18}$$

$$\mathcal{L}_{cls} = -\sum_{x_*\in\mathbb{D}}\log(\eta(x_*)), \tag{4.19}$$

where $\mathcal{L}_{cls}$ is the total cross-entropy loss. $\mathcal{L}_{reg}$ regularizes our system by controlling the complexity of Spatio-Net, as discussed previously. α and β control the weights of three loss terms, and they are cross-validated during the training stage. V is all the learnable weights in Spatio-Net.

4.4 Experiments

4.4.1 Cardiac MRI Recognizing

Dataset: the cardiac MRI dataset was collected from our collaborative hospital and labeled by board-certified experts. More specifically, we gathered the cardiac sequences from four views (*i.e.*, long-axis, short-axis, four-chamber, and two-chamber)

from 420 patients, which contain around $113,000$ frames. Every cardiac MRI sequence contains 20 frames (256×256 pixels) and each patient has around 18 sequences (about 15 short-axis, a long-axis, a four-chamber, and a two-chamber view). ES and ED frames are carefully labeled by the experts in the hospital. The results are generated by performing four-fold cross-validation on this dataset.

Evaluation metrics: to quantify the accuracy of the predictions, the average frame difference (aFD) is employed, following the convention of [100, 104]. Formally, aFD is defined as:

$$aFD = \frac{1}{|\mathbb{D}|} \sum_{i=1}^{|\mathbb{D}|} |\hat{N}_i - N_i|, \tag{4.20}$$

where N_i and $\hat{N}_i$ are the ground truth frame index and predicted value of i^{th} cardiac MRI sequence in the testing dataset. $|\mathbb{D}|$ denotes the total number of evaluated cardiac MRI sequences.

Implementation details: TempReg-Net uses the Zeiler-Fergus (ZF) model [15] as the deep feature extractor, as it makes an excellent trade-off between the performance and the computational cost. Each gray-scale frame is squashed to the range of $[0, 255]$ and replicated two times, resulting in a three-channel image. In order to avoid overfitting, we fine-tune the pre-trained LRCN network [105] (originally trained on ImageNet [61]) on our dataset. The learning rate of the last fully-connected layer (the layer that follows the LSTM model) is set to be 10 times larger than the learning rates of the rest layers. We initialize all the parameters in the LSTM in the range of $[-0.01, 0.01]$. The hyper-parameters are cross-validated during the training stage. We randomly crop the resized cardiac frames in order to artificially augment our datasets.

Quantitative Results: We first compare the proposed framework with the state-of-the-art (Reg-based: CNN+Reg): a similar regression-based method. This method

differs from our method only in two aspects: 1) it does not use an LSTM layer (structural feature learner) to model the inter-frame dependencies, 2) the prior knowledge regarding the predicted values is not enforced in the loss function. The results are shown in Table 4.1. According to Table 4.1, our TempReg-Net can achieve competitive results (*i.e.*, 0.47 for identifying ED and 0.52 for locating ES, respectively) even without enforcing the prior knowledge in the loss (*i.e.*, temporal structured loss $\mathcal{L}_{temp}$). After adding $\mathcal{L}_{temp}$, the performance improves significantly, *i.e.*, 0.44 for ES and 0.38 for ED, increasing the accuracy by around 15%. The result demonstrates the effectiveness of the proposed structured learning framework. In terms of the computational cost, only 1.4 seconds is required for TempReg-Net to process a cardiac MRI sequence. Due to its efficiency, TempReg-Net can potentially be integrated with cardiac analysis platforms.

Table 4.1: Quantitative comparisons of the proposed TempReg-Net with the state-of-the-art (Reg-based: CNN+Reg), segmentation based methods (level set [1] and graph cut [2]), and TempReg-Net without temporal structured loss $\mathcal{L}_{temp}$.

Methods		Seg-based: Level Set [1]	Seg-based: Graph Cut [2]	Reg-based: CNN+Reg	TempReg-Net (w/o $\mathcal{L}_{temp}$)	TempReg-Net
aFD	ED	1.54	2.27	1.30	0.47	**0.38**
	ES	1.24	1.65	1.97	0.52	**0.44**
STD	ED	1.93	2.89	1.77	0.49	**0.39**
	ES	1.64	1.96	2.42	0.53	**0.46**
Time (s)		2.9	3.5	1.5	**1.4**	**1.4**

Additionally, we compare the proposed framework with other related methods. For example in [106], the left ventricle is first segmented from each frame. Then, the ES and ED frames are identified by comparing the areas of these regions. A similar method is developed in this dissertation to segment the left ventricle, employing variations of graph cut [2], a method proposed recently that can accurately achieve the task of myocardium segmentation with level set [1]. This type of method is very intuitive, therefore being widely used. Nevertheless, there exist several limitations in these approaches; *e.g.*, high computational cost, significant segmentation errors,

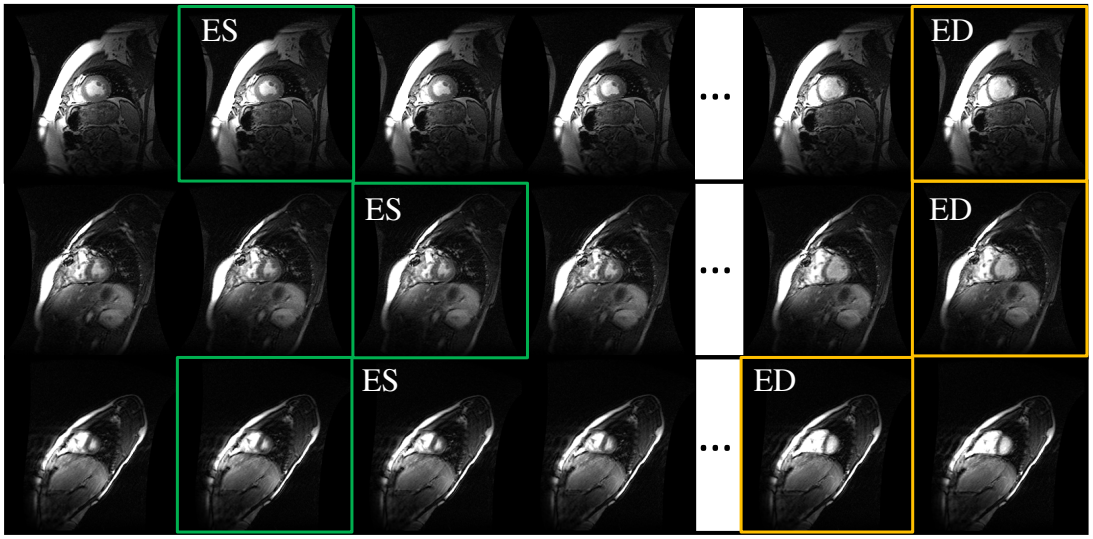

Figure 4.2: Three predicted results by the proposed TempReg-Net. The ground truth annotations are illustrated in the top left corner of the corresponding frames. Green and yellow frames are the predicted ES and ED frames, respectively.

and requirements of human interactions. In the experiments, 3.5 and 2.9 seconds are required to segment the left ventricle from the frames in a cardiac MRI sequence employing graph cut and level set, respectively. In other words, this system is significantly slower than the proposed method, assuming the fact that the time of human interactions for initializing the segmentation (*e.g.*, the background and/or foreground has to be manually defined in the graph cut based approach) is not counted.

Regarding the accuracy, the aFD is 1.24 and 1.54 for ES and ED respectively, when level set is employed. When graph cut is employed, the aFD is 1.65 and 2.27 for ES and ED respectively. In comparison, the proposed method achieves much better performance. The reason resides in the segmentation procedure: these segmentation algorithms are not able to generate perfect segmentation results, while even a small segmentation error substantially reduce the final result due to the subtle differences among adjacent frames.

Evaluation of the Temporal Structured Constraint: In order to obtain a deeper understanding of the influence the structured constraint has on the results, we randomly selected a cardiac MRI sequence from the testing data and compare the

results of the proposed methods with the state-of-the-art (Reg-based: CNN+Reg), which does not consider the temporal structured constraint (TSC). In this sequence, the ED and ES frames are the 8th and 1st frame, respectively. Fig. 4.3 shows the visual comparison results. The result of TempReg-Net is consistent with our prior knowledge. On the other hand, the predictions of the method without TSC fluctuate at several places (2^{nd}, 5^{th}, 9^{th}, and 12^{th} frames). Because of these fluctuations, the predicted ED and ES will differ from the ground truth by one and two frames, respectively. The visual comparisons provide additional evidence for the superiority of our method.

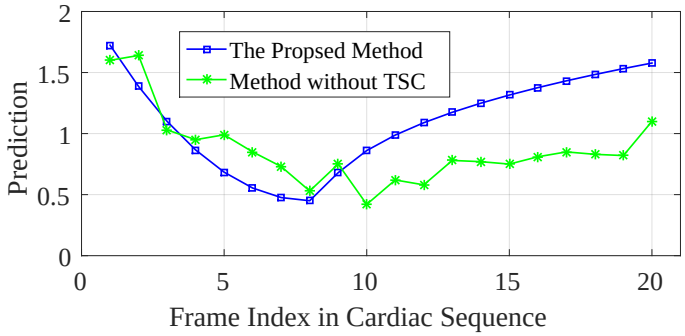

Figure 4.3: Comparison of the predicted values of a cardiac MRI sequence generated by the state-of-the-art method without temporal structured constraint (TSC) and the proposed structured learning framework.

4.4.2 Cancer Metastasis Detection in WSIs

Dataset & Evaluation metrics: in the CAMELYON16[1] dataset, the training set includes 110 tumor WSIs and 160 normal WSIs. All these images are carefully annotated by pathologists. The testing set is composed of 130 WSIs. To maximally leverage image information, all experiments are carried out on the 40× magnification. The detection performance is evaluated by average FROC (Ave. FROC) [107]. A higher average FROC value suggests better detection performance.

Implementation details: To effectively extract image features, residual neural network [102] (ResNet101) was used for feature extraction. Four 2D LSTM layers

[1]https://camelyon16.grand-challenge.org/

are employed to model the dependencies in the neighboring image patches. For fair comparison, we follow Wang *et al.* [3] to sample the training image patches and train the neural networks. In the testing stage, the stride of sliding window is set as 64.

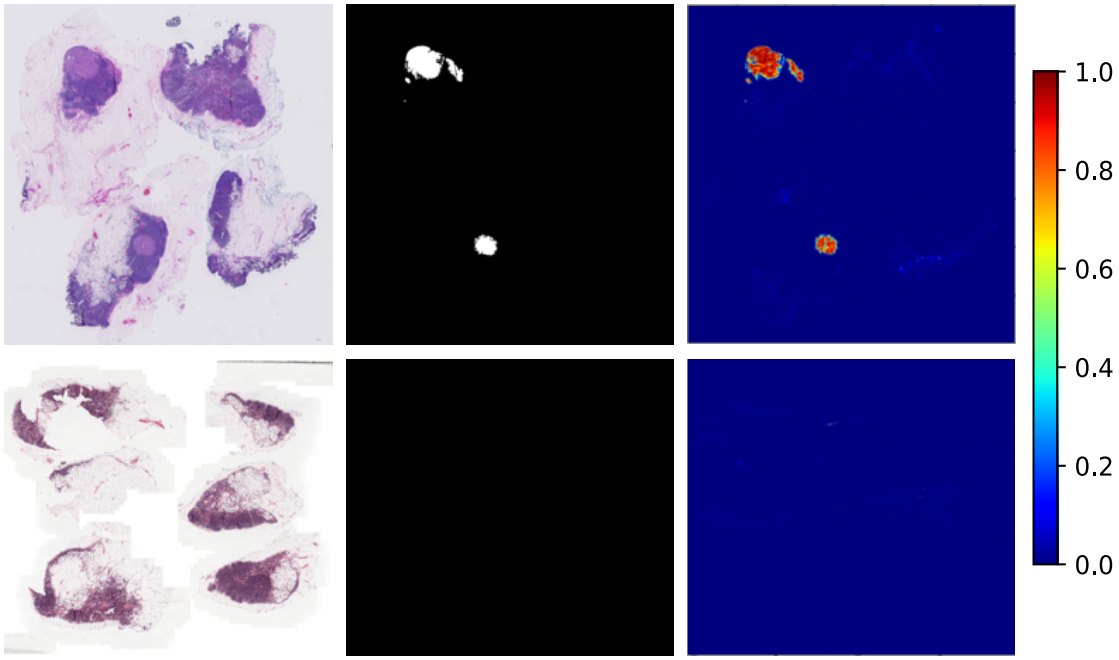

Figure 4.4: Predicted probability maps of WSIs with cancer metastases (top row) and without cancer metastasis (bottom row). The first, second, and third columns show the original WSIs, ground truth annotations, and the probability maps generated by Spatio-Net.

Quantitative Results: We carried out multiple experiments to evaluate Spatio-Net. Firstly, we compare Spatio-Net with the baseline [3], the state-of-the-art architecture. The difference between the baseline and our framework is two-fold: 1) no extra 2D-LSTM layers is used to model the inter-patch dependencies, and 2) spatially structured loss $\mathcal{L}_{spatio}$ is not considered in the loss function. Smoothing is often used as a simple approach to consider the correlation of neighboring predictions. Thus, we further compare our method with baseline + smoothing for postprocessing. Fig. 4.4 shows some generated probability maps.

The final results of the above methods are summarized in Fig. 4.5 and Table 4.2. According to Table 4.2, our approach, Spatio-Net significantly outperforms [3] (more

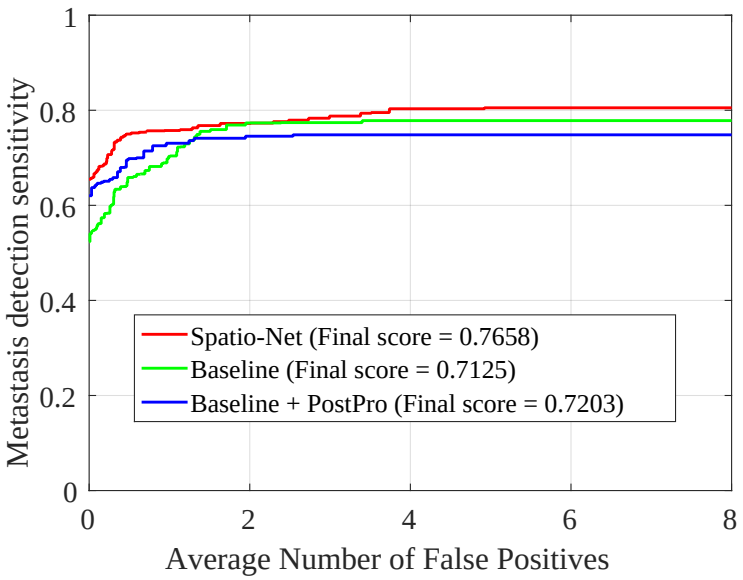

Figure 4.5: FROC curves of different methods on the testing set.

than 5%), demonstrating the effectiveness of the proposed structured learning framework. Furthermore, Spatio-Net also outperforms Baseline+post-processing by more than 4%. This is because although post-processing bring certain benefits, it is not fully integrated into our structured learning framework. In contrast, our approach models the structural information in a systematic way.

Table 4.2: Quantitative comparisons of the proposed methods and the baselines: Wang *et al.* [3] and Wang *et al.* [3]+Postprocessing.

Methods	Wang *et al.* [3]	Wang *et al.* [3]+PostPro	Spatio-Net
Ave. FROC	0.7125	0.7203	**0.7658**

Finally, we conduct an additional experiment to test if the proposed structured learning framework is able to benefit different types of CNN architectures. Specifically, three types of CNN models are tested: ResNet101 [102], GoogleNet [17], and ZFNet [15]. For all these models, we evaluate their performance with or without enforcing structured learning. Their performances are summarized in Table 4.3. Two key conclusions can be drawn. First, stronger CNN architecture has higher detection performance. More specifically, the detection accuracy improves from ZFNet to

GoogleNet and ResNet101. We guess this is because we have enough training data to train a high capacity CNN network. In this case, higher capacity network means better generalization. Second, structured learning consistently helps these CNN models to boost the detection accuracy. Note that the Ave. FROC increased by 4.9% in average when structured learning is enforced in these CNN models.

Table 4.3: Detection results of different methods: CNN models only or CNN models with structured learning (SL).

Methods	ZFNet	ZFNet (with SL)	GoogleNet	GoogleNet (with SL)	ResNet101	ResNet-101 (with SL)
Ave. FROC	0.6938	0.7354	0.7026	0.7543	0.7125	0.7658

4.5 Summary

For many MIA applications, prior structural knowledge about a specific task is also extremely important for obtaining more accurate results. Apart from proposing a general structured learning framework for MIA, we enforce the prior structural knowledge in loss functions to regularize the training of the deep structured learning framework in this chapter. We extensively evaluate the proposed structured learning approach with two MIA tasks: cardiac MRI recognizing and cancer metastasis detection in WSIs. The superior performance demonstrates the effectiveness of the proposed method.

CHAPTER 5: MODELING TREE STRUCTURES WITH TREE-STRUCTURED CONVOLUTIONAL GRU

In chapter 3, we demonstrated two use cases of the proposed deep structured learning framework: cardiac MRI recognizing and cancer metastasis detection in WSIs. In this chapter, we consider a more challenging structured learning problem: tree-structured learning and its application in coronary artery segmentation.

5.1 Motivation

Over the past two decades, coronary artery segmentation has drawn greater and greater attention because it not only greatly facilitates the reviewing process but also provides quantitative function analysis [108]. Unfortunately, the segmentation procedure still heavily relies on semi-automatic approaches, which are still time-consuming and error-prone. This is because fully-automatic approaches cannot produce sufficiently accurate results, as the coronary arteries exhibit extremely complex structures. Therefore, it is essential to accurately as well as efficiently segment coronary arteries.

Currently, it is a standard procedure to evaluate coronary artery diseases with computed tomography angiography (CTA) as it provides high-resolution 3D imaging with non-invasiveness. The focus of this chapter is accurate segmentation of the coronary artery in 3D coronary computed tomography angiography (CCTA) volumes, as illustrated in Fig. 5.1. Multiple reasons account for the difficulty of coronary artery segmentation. First, the boundaries between the artery and background are often highly fuzzy, as is shown in Fig. 5.1 (a). Second, the tubular structure of the coronary artery is extremely complex: the cross-section area changes gradually along

the artery and there exist a large number of bifurcations (see Fig. 5.1 (b)). Third, the appearance and geometry of the coronary artery may vary considerably from one patient to another. Plus, the buildup of the plaque or calcification (extremely high-intensity regions in Fig. 5.1 (c)) inside the coronary artery wall may further cause the variability from one patient to another [108]. Finally, the image acquisition process may further introduce inherent image noise and artifacts [109], making the segmentation even more challenging.

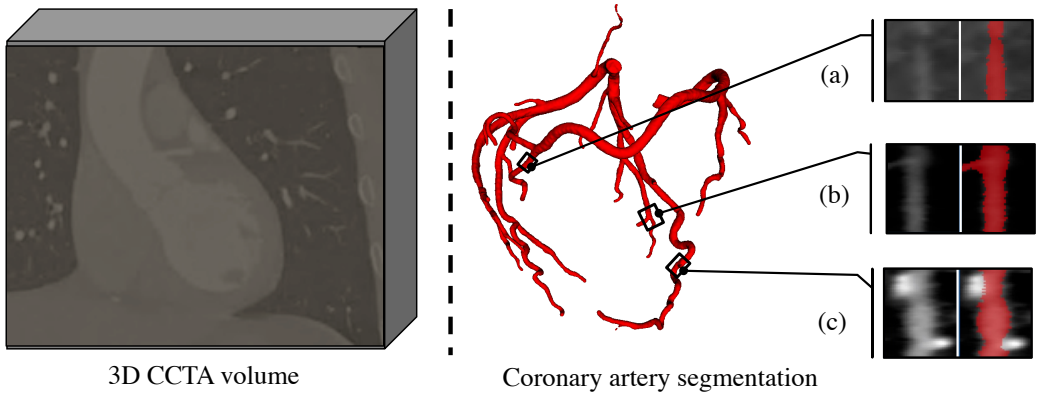

Figure 5.1: From left to right: a 3D CCTA volume, the corresponding coronary artery segmentation, and three longitudinal views of the coronary artery. The coronary artery segmentation is denoted in red.

A substantial body of research has been devoted to the segmentation of the coronary arteries. Most of them [110, 111] are only based on domain knowledge about the voxel intensity distributions, which suffer from multiple issues, *e.g.*, holes and noisy contours. Additionally, they often fail to build a global tree structure as they only rely on local intensity information. To address this issue, geometry and topology prior have been employed to generate more anatomically reasonable segmentation result [112]. Nevertheless, introducing these priors requires domain-specific expertise. Recently, deep learning has been introduced to the segmentation of tree-like objects [113, 114, 115]. Compared with traditional methods for MIA [116, 117, 118, 119], deep learning-based approaches achieve better performance and at the same time obviate hand-crafting features, as the hierarchical neural networks automatically learn

the most discriminative features for the coronary artery purely from the training data. However, these methods either ignore the underlying anatomical structure in the coronary artery [113] or simply use traditional methods to post-process the segmentation results [114], which requires domain-specific knowledge and extensive tuning.

Inspired by the proposed deep structured learning framework in chapter 3, we propose to explicitly model the anatomical structure of the coronary artery with a unified network. It consists of a fully convolutional network (FCN) model to extract discriminative features from CCTA dataset and a tree-structured ConvGRU layer to model the anatomical structure of the coronary arteries. We summarize the essential contributions as follows:

- A novel convolutional recurrent neural network (ConvRNN) layer, tree-structured convolutional gated recurrent unit (ConvGRU), is proposed to explicitly model the topological structure of the coronary artery.
- Accordingly, an end-to-end deep learning-based framework, consisting of a tree-structured ConvGRU layer and an FCN, is presented to accurately segment coronary arteries from 3D CCTA data.
- Four large-scale CCTA datasets are employed to extensively evaluate the performance of the proposed framework. The results demonstrate that the proposed framework outperforms other baseline methods.

5.2 Methodology

5.2.1 Convolutional RNN Models

Vessels with tubular structures and bifurcations gradually change geometry and elongation from proximal to the distal end. In this chapter, we strive to use deep learning to model this special anatomical structure. Recurrent neural networks (RNNs) are great candidates for modeling long-term dependence [36, 32]. Until now, most of

the past studies have used long short-term memory (LSTM) to deal with the notorious issue of vanishing or exploding gradients [120], which is a significant problem when training the vanilla RNN models. By incorporating several sophisticated gating functions, LSTM alleviates this issue. Nevertheless, the input-to-state and state-to-state changes are based on fully-connected layers in LSTM, which neglects local spatial correlations in input data. It is therefore not appropriate for the analysis of image sequences. The recently proposed convolutional LSTM (ConvLSTM) replaces the vector multiplication in LSTM with convolutional operations by preserving the spatial topology of the input while introducing sparsity and locality to the LSTM to reduce over-parameterization and overfitting. Unfortunately, vessels with highly branching and tubular structures are extremely complex, and ConvLSTM, which is originally designed for image sequence analysis, cannot deal with such complicated tree structures. While the tree-structured LSTM [121] is proposed for the analysis of tree-structured data (specifically, natural language processing), the vector multiplication used in the tree-structured LSTM unit is not appropriate for image analysis. In contrast, our tree-structured ConvGRU design addresses both issues, *i.e.*, a lack of consideration of complex tree structures and the local spatial correlation in the input data.

The input-to-state as well as the state-to-state transitions are conducted by vector multiplications in the standard LSTM. It ignores the local spatial correlations in the input by vectorizing the input feature map. Therefore, it is not suitable for image sequence analysis. To address this issue, the vector multiplications are replaced by convolutions in ConvLSTM [11], to maintain the local correlations in the image sequence data. It defines a new mechanism to update the input-to-state as well as

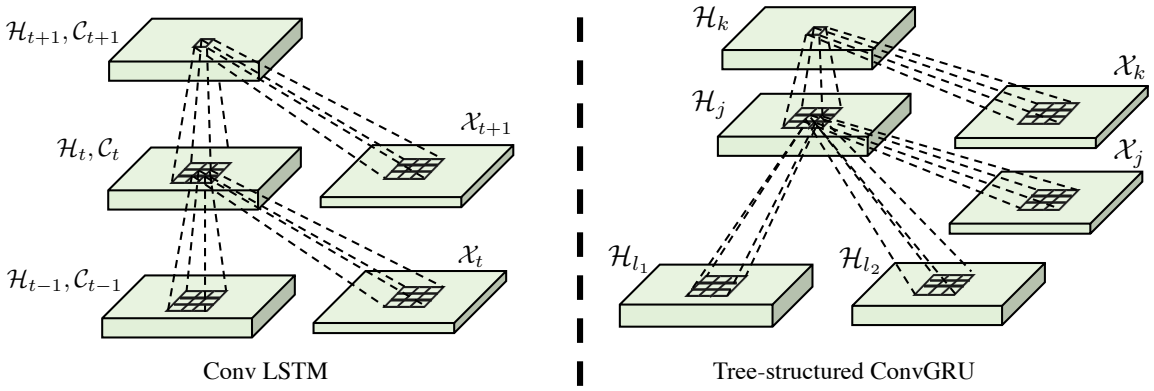

Figure 5.2: From left to right: sequential ConvLSTM [11] and the proposed tree-structured ConvGRU. In ConvLSTM, the information, including the input $\mathcal{X}_t$, previous hidden state $\mathcal{H}_{t-1}$, and previous memory $\mathcal{C}_{t-1}$, is passed sequentially (from $t-1$ to t and then to $t+1$). As with tree-structured ConvGRU, there is no memory cell. The information is passed from all the children nodes to the parent node. For instance, node j in this figure incorporates the information (hidden state $\mathcal{H}_{l_1}$ and $\mathcal{H}_{l_2}$ from both its children l_1 and l_2 and the current input $\mathcal{X}_j$) to produce the current hidden state $\mathcal{H}_j$. Node k incorporates the information (hidden state $\mathcal{H}_j$ from its child j and its input $\mathcal{X}_k$) to produce the current hidden state $\mathcal{H}_k$. Note that although we only show one or two child nodes for the tree-structured ConvGRU model, it is capable of handling more than two child nodes.

state-to-state transition:

$$i_t = \sigma\left(W_i * \mathcal{X}_t + U_i * \mathcal{H}_{t-1}\right), \tag{5.1}$$

$$f_t = \sigma\left(W_f * \mathcal{X}_t + U_f * \mathcal{H}_{t-1}\right), \tag{5.2}$$

$$o_t = \sigma\left(W_o * \mathcal{X}_t + U_o * \mathcal{H}_{t-1}\right), \tag{5.3}$$

$$\mathcal{M}_t = \tanh\left(W_m * \mathcal{X}_t + U_m * \mathcal{H}_{t-1}\right), \tag{5.4}$$

$$\mathcal{C}_t = f_t \odot \mathcal{C}_{t-1} + i_t \odot \mathcal{M}_t, \tag{5.5}$$

$$\mathcal{H}_t = o_t \odot \tanh(\mathcal{C}_t), \tag{5.6}$$

where $*$ indicates convolution, $\mathcal{X}_t$ is the current input image at time step t. The memory cell and hidden state are denoted by $\mathcal{C}_t$ and $\mathcal{H}_t$, respectively.

5.2.2 Tree-structured ConvGRU

Sequential ConvRNNs [122] can not handle tree-structured data. For this reason, we propose a novel tree-structured ConvRNN network for extracting tree-structured anatomical information, in which the parent node selectively aggregates features from all its child nodes. Desirably, this tree-structured ConvRNN model is capable of automatically learning to emphasize important information in the data. For instance, it is desirable to emphasize the geometry and direction of the main artery when there exists a much thinner artery merging with the main branch artery. In this chapter, we mainly focus on the extension of GRU, considering its lower computational requirement [123] than LSTM. Also, the experimental results demonstrate its superior performance than the LSTM extension on our datasets. Unlike LSTM, there is no memory cell or forget gate in GRU. Rather, for each node j in the tree, the memory cell is integrated into the hidden state $\mathcal{H}_j$ and the reset gate r_j controls the updating of the previous memory. As one unit may have multiple child nodes, we use a distinct rest gate r_{jk} for each child node to remove unimportant past information from each individual child node's memory. The whole procedure is detailed as follows:

$$\mathcal{H}'_j = \sum_{k \in \mathcal{N}_j} \mathcal{H}_k, \tag{5.7}$$

$$u_j = \sigma(W_z * \mathcal{X}_j + U_z * \mathcal{H}'_j), \tag{5.8}$$

$$r_{jk} = \sigma(W_r * \mathcal{X}_j + U_r * \mathcal{H}_k), \tag{5.9}$$

$$\tilde{\mathcal{H}}_j = \tanh(\sum_{k \in \mathcal{N}_j} r_{jk} \odot U * \mathcal{H}_k + W * \mathcal{X}_j), \tag{5.10}$$

$$\mathcal{H}_j = (1 - u_j) \odot \tilde{\mathcal{H}}_j + u_j \odot \mathcal{H}'_j, \tag{5.11}$$

where W_z, U_z, W_r, U_r, W, and U are the learnable parameters.

5.2.3 Artery Centerline Extraction

First, we extract the coronary artery centerline from the CCTA data, which captures the anatomical structure of the coronary artery. We use our earlier published approach [124] for centerline extraction. It is a deep learning-based method, which is able to produce accurate (the error is within a single voxel) centerlines. The brief pipeline is summarized here. We refer the readers to [124] for more details.

- We pre-segmented coronary arteries with 3D U-Net [39]. The anatomical structure is captured by pre-segmentation. Nevertheless, there exists a lot of erroneous predictions (see Fig. 5.5 for more details). As the proposed tree-structured segmentation framework is comparatively resistant to imperfect segmentation, precise pre-segmentation is not needed.

- The endpoints and distance map of the centerline are simultaneously predicted by a trained multi-task FCN network.

- The ultimate artery centerline is generated by minimal path algorithm. The generated centerline can be defined by a tree structure $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the nodes (representing the centerline points) and adjacency matrix (representing connections among the centerline points) are denoted by $\mathcal{V}$ and $\mathcal{E}$, respectively.

5.2.4 Tree-structured Segmentation Network Architecture

In this chapter, the coronary artery segmentation is formulated as a tree-structured segmentation problem, in which the training set is a collection of coronary artery trees and the predictions are also organized as a tree structure. The input tree is produced as follows. For each node j in the artery tree $\mathcal{G}$, a cross-sectional view is cropped from the CCTA volume in the centerline's perpendicular direction. We further normalize this small patch with the aorta intensity and calcification threshold respectively to highlight both of these important regions. Finally, the normalized

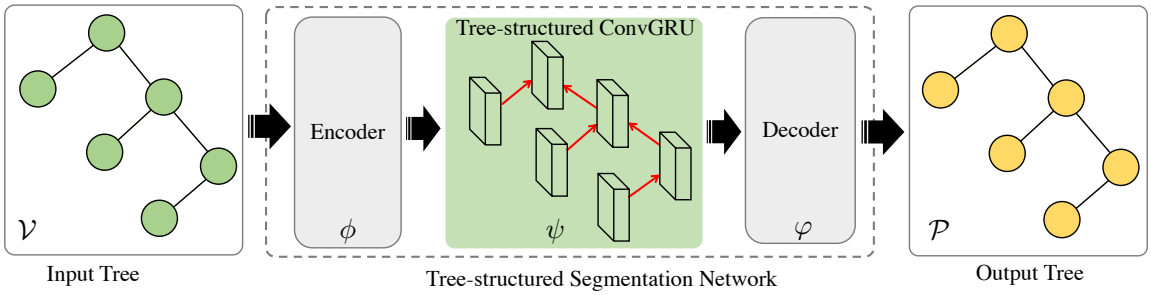

Figure 5.3: An overview of the proposed tree-structured segmentation network. The input of the system is a input tree $\mathcal{V}$, *i.e.*, images organized as a tree structure. The output $\mathcal{P}$ is also organized as a tree structure. The tree-structured segmentation network consists of two components: an FCN backbone with an encoder ϕ for discriminative feature learning and a decoder φ for prediction, and a tree-structured ConvGRU layer ψ for anatomical structure modeling. The FCN backbone and tree-structured ConvGRU layer are shared by all tree nodes. The detailed information is illustrated in Fig. 5.4.

patches are concatenated with the original patch. The result is a three-channel image $\mathbf{x}_j$ associated with node j. Formally, the goal is to learn a non-linear function, $(\mathcal{H}_1, ..., \mathcal{H}_J) = \sigma_W(\mathbf{x}_1, ..., \mathbf{x}_J)$, to map the tree-structured input to the tree-structured output, where J and W represent the number of nodes in the tree and the parameters to be learned.

Fig. 5.3 presents an overview of the proposed tree-structured segmentation framework. In our network, we model the structured information in a unified neural network, which can be trained end-to-end. It has three modules: an encoder, a tree-structured ConvGRU, and a decoder. This architecture is motivated by the proposed deep structured learning framework in chapter 3. The encoder corresponds the deep feature extractor. The tree-structured ConvGRU and decoder correspond to the structural feature learner. Specifically, the encoder ϕ extracts discriminative features from the input data, yielding a multi-scale representation $\mathcal{X}_j$ for each node j. The tree-structured ConvGRU module ψ models the anatomical structure of the coronary artery, generating a feature map $\mathcal{H}_j$, encoding the newly-extracted anatomically related features. Based on the feature map generated by the encoder and tree-structured

ConvGRU, the decoder φ generates the final prediction $\mathcal{P}_j$.

5.2.5 Discriminative Feature Learning & Tree-structured Output Generation

Fig. 5.4 illustrates the backbone network for feature extraction and final prediction. It's based on the U-Net [38] architecture. The encoder ϕ and decoder φ divide the whole segmentation procedure into three separate stages: discriminative feature learning, anatomical structure modeling, and tree-structured output generation. During the discriminative feature learning stage, the image $\mathbf{x}_j$ associated with each node j is fed into the encoder, which includes several 3×3 convolutional layers (each is followed by a ReLU layer). Two 2×2 layers are also used to downsample the feature map. The encoder is able to extract discriminative features from the input $\mathcal{X}_j = \phi(\mathbf{x}_j)$. After the anatomical structure modeling stage, a hidden state $\mathcal{H}_j$ is generated by the tree-structured ConvGRU layer (will be detailed in Sec. 5.2.6), the decoder progressively rescale the feature maps to the original dimension using deconvolution and at the same time incorporate the information passed from the encoder, yielding the final prediction $\mathcal{P}_j = \varphi(\mathcal{X}_j, \mathcal{H}_j)$ (see Eq. (17) to (21) for more details). The details of the encoder and decoder are shown in Fig. 5.4.

5.2.6 Anatomical Structure Modeling

The introduction of the tree-structured ConvGRU ψ is motivated by the fact that there exist inherent anatomical structures in the coronary artery tree. For instance, tubular artery gradually changes from the proximal to the distal end, with the elongation and radius changes smoothly from node to node. Using tree-structured ConvGRU in our system brings two benefits. First, by feeding the features extracted by the encoder to the tree-structured ConvGRU, the context information is propagated among the tree nodes. As a result, the final encoder makes prediction not solely based by the features of one node but considering the topological changes along the coronary artery tree. Second, as mentioned in section 5.2.2, there may exist multiple branches

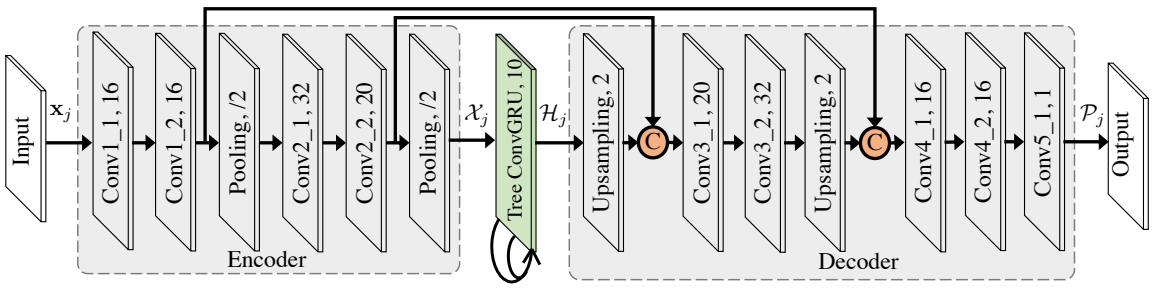

Figure 5.4: Details of the proposed tree-structured segmentation network. Both the encoder and decoder consist of multiple convolutional layers (each is followed by a ReLU layer, which is ignored for simplicity). For the input image $\mathbf{x}_j$ associated with node j, it is passed into several convolutional layers and progressively downsampled by the pooling layers in the encoder, generating the feature map $\mathcal{X}_j$. The tree-structured ConvGRU layer takes input $\mathcal{X}_j$ and produces the hidden state $\mathcal{H}_j$. In the decoder, $\mathcal{H}_j$ from the tree-structured ConvGRU layer is progressively upsampled to the original dimension and at the same time incorporates the information passed from the encoder, yielding the final prediction $\mathcal{P}_j$.

at each tree node. In these special locations, our system is capable of modeling these transitions. The tree-structured ConvGRU layer takes input $\mathcal{X}_j$ and produces the hidden state $\mathcal{H}_j = \psi(\mathcal{X}_j)$.

5.2.7 Loss Function

The forward pass of the proposed tree-structured segmentation network for one input tree is illustrated in Algorithm 3. The proposed tree-structured segmentation forms a differentiable system, which can be trained end-to-end. Dice loss is applied node-wise and the final loss is the average dice loss, as defined as follows:

$$\mathscr{L}(\mathcal{P}, \mathcal{G}) = \frac{1}{J} \sum_{j=1}^{J} \frac{2|\mathcal{P}_j \cap \mathcal{G}_j|}{|\mathcal{P}_j| + |\mathcal{G}_j|}, \tag{5.12}$$

where the output tree and all the ground truth segmentation are represented by $\mathcal{P} = (\mathcal{P}_1, ..., \mathcal{P}_J)$ and $\mathcal{G} = (\mathcal{G}_1, ..., \mathcal{G}_J)$, respectively. $\mathcal{P}_j$ and $\mathcal{G}_j$ are the prediction and ground truth for node j, respectively.

Algorithm 3 The forward pass of the proposed tree-structured segmentation network for one input tree.

Input: $\mathcal{G}$ = input tree $(\mathcal{V}, \mathcal{E})$
Input: ϕ = encoder
Input: ψ = tree-structured ConvGRU layer
Input: φ = decoder
$\mathcal{P} \leftarrow \varnothing$
for j in $[1, num\_nodes]$ **do**
  sample the input image $\mathbf{x}_j$ associated with node $j \in \mathcal{V}$
  extract features from $\mathbf{x}_j$ with $\mathcal{X}_j \leftarrow \phi(\mathbf{x}_j)$
  generate the hidden state using $\mathcal{H}_j \leftarrow \psi(\mathcal{X}_j)$
  produce the final prediction using $\mathcal{P}_j \leftarrow \varphi(\mathcal{X}_j, \mathcal{H}_j)$
  $\mathcal{P}[j] \leftarrow \mathcal{P}_j$
end for
return $\mathcal{P}$

5.3 Experiments

5.3.1 Dataset, Evaluation Metrics, and Implementation Details

We collected four large datasets (916 CT scans in total) from four hospitals. These collaborating hospitals are selected from different areas to represent the diversity of healthcare settings. 80%, 5%, and 15% scans were used for training, validation, and testing, respectively. The data splitting was carried out on the patient level. The ground truth was obtained by a semi-automatic approach. First, a vesselness based approach combined with the dynamic programming algorithm was used to obtain an initial entire coronary artery. Then, the generated masks were refined by two medical image analysts, respectively. Finally, the better one was chosen as the ground truth by a more experienced expert. To the best of our knowledge, this dataset is largest available for evaluating coronary artery segmentation algorithms. These datasets are dubbed CTA1, CTA2, CTA3, and CTA4 in this chapter and they include 516, 546, 446, 324 scans, respectively. The details of these datasets are shown in Table 5.1. To measure the performance of the segmentation methods, we use the

average dice score of all the tree nodes. All the methods were trained and evaluated on a workstation equipped with a Tesla P40 GPU. To train the neural networks, the Adam optimizer [125] was used. The initial learning rate, weight decay, and momentum are 0.001, 0.0005, and 0.9, respectively. Additionally, early-stopping was used to combat over-fitting.

Table 5.1: Detailed information of our datasets (CTA1, CTA2, CTA3, and CTA4). Apart from providing the number of training scans in each dataset, the average number of tree nodes and branches are also given.

Dataset	Number of of scans	Number of Nodes	Number of Branches
CTA1	258	727	12.6
CTA2	273	806	11.1
CTA3	223	802	13.2
CTA4	162	694	12.9
Total	916	774	12.4

5.3.2 Main Results

First, the proposed approach is compared with a recently-introduced 3D object segmentation framework, 3D volumetric convnet (DenseVox) [4]. For DenseVox, a $41 \times 41 \times 41$ subvolume around each tree node is fed into the a DenseVox network. Unlike our approach, DenseVox doesn't consider long-range inter-node dependencies in the artery tree or the tree structure underlying in the artery tree. According to Table 5.2, the proposed tree-structured ConvGRU based segmentation framework (TreeConvGRU) consistently surpasses DenseVox (1.24%, 0.98%, 1.12% and 1.01%, and 1.01% on CTA1, CTA2, CTA3, and CTA4 respectively), indicating the essential role of modeling the long-range inter-node dependency and tree-structure in coronary artery segmentation.

Next, TreeConvGRU is compared with its sequential version, sequential ConvGRU

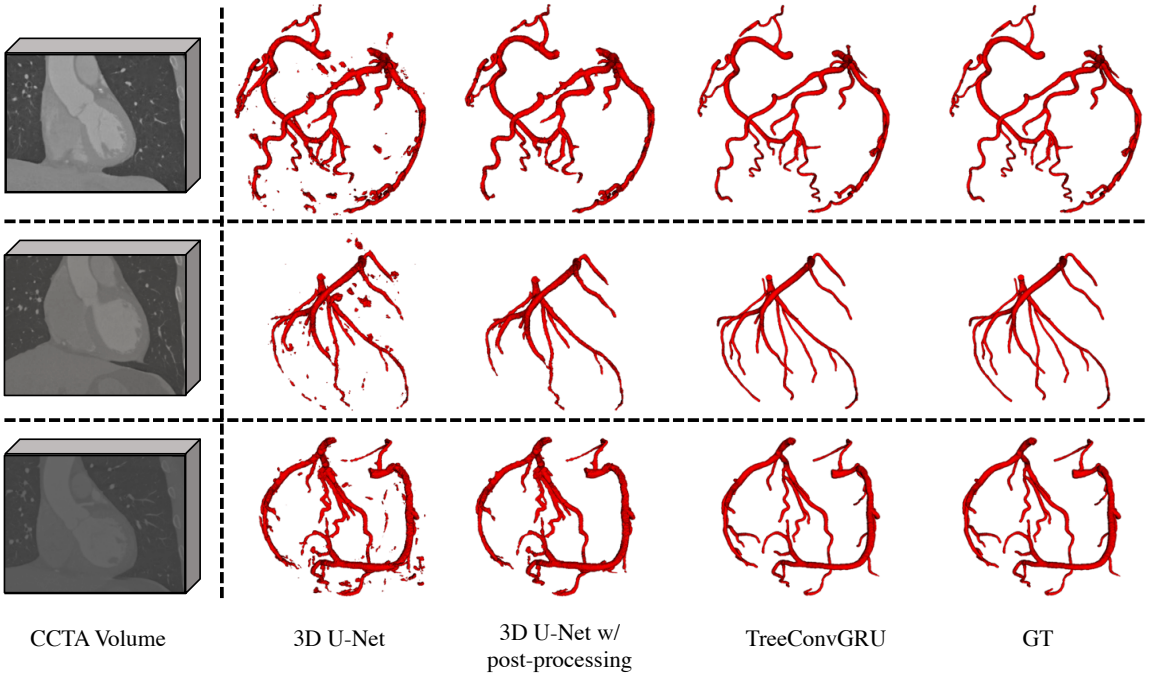

Figure 5.5: Qualitative coronary artery segmentation result of 3D U-Net, 3D U-Net with post-processing, and the proposed method. From left right shows: the input 3D CCTA volumes, segmentation results of 3D U-Net based method [12], segmentation results of 3D U-Net with post-processing, segmentation results of the proposed tree-structured segmentation network, and the ground truth.

(ConvGRU). Compared with TreeConvGRU, the tree structures are ignored by ConvGRU and the segmentation results are generated independently for each path in the tree. The results once again suggest the superiority of the proposed method over sequential models in modeling the inter-node dependency in tree structures: the average dice score of TreeConvGRU is better than ConvGRU by 0.95%, 0.76%, 1.09%, and 0.46% on CTA1, CTA2, CTA3, and CTA4, respectively. Additionally, to test the scalability of the proposed method, we evaluate the performance of the above methods on the aggregated dataset of CTA1, CTA2, CTA3, and CTA4, which are named Total. As is shown in Table 5.2, TreeConvGRU still consistently overperform ConvGRU and DenseVox (0.69% and 1.65%, respectively). We also provide some qualitative coronary artery segmentation results of our approach in Fig. 5.5.We compare the qualitative results of our network with a 3D U-Net based network [12], *i.e.*,

Table 5.2: Main comparison results. The proposed tree-structured segmentation network (TreeConvGRU) is compared with the recently proposed 3D densely-connected volumetric convnets (DenseVox) [4], sequential version of our tree-structured segmentation network (ConvGRU). All these methods are evaluated by the average dice loss.

Methods	DenseVox [4]	ConvGRU	TreeConvGRU
CTA1	0.8370	0.8399	0.8494
CTA2	0.8405	0.8427	0.8503
CTA3	0.8433	0.8436	0.8545
CTA4	0.8182	0.8237	0.8283
Total	0.8518	0.8614	0.8683

the pre-segmentation of the coronary artery. As the segmentation is applied on every single voxel of the CCTA volume, the network is extremely sensitive to local perturbations. Therefore, the results suffer from a significant amount of false positives and false negatives. Even after post-processing (erosion, dilation, and connected component analysis), the false predictions on the coronary artery cannot be corrected. On the contrary, our network efficiently leverage the anatomical structure of coronary artery to guide its segmentation, generating a much more accurate segmentation result.

Lastly, we also compare our method with another tree-structured extension of ConvRNN model, tree-structured ConvLSTM (TreeConvLSTM). This approach is different from our approach by substituting the GRU operations with LSTM. This change slightly decreases (0.14% in average on all datasets) the performance of our framework. This result matches the findings in [123] regarding the comparison of non-convolution versions of LSTM and GRU. In this chapter, the information propagation is conducted from the root to the leaf nodes. It's possible to extend the proposed method to conduct the propagation in both directions with the technique in [126], *i.e.*, from tree leaves to the root as well as from the root to leaves. However,

the overall performance degraded by 0.04%. Here is one possible explanation for this performance degradation: the anatomical structure can be sufficiently modeled by the one-directional tree-structured ConvGRU, without needing to resort to more complex RNN models. In contrast, a more complex system may render the learning process even harder.

5.3.3 Comparisons on Bifurcation Nodes

Intuitively, it's much more challenging for the segmentation framework to generate good prediction at bifurcation nodes, compared with non-bifurcation nodes. This is because the dynamics around these nodes are much more complex. We conducted an extra experiment on Total to verify this hypothesis. In this experiment, we only evaluate the performance of the segmentation approaches on the nodes within 4 nodes' distance from bifurcation nodes. According to Table 5.3, our method consistently exceeds DenseVox and ConvGRU (7.31% and 3.14%, respectively). The results demonstrate the importance of introducing the tree structure. DenseVox ignores the inter-node dependencies in the artery tree while ConvGRU only considers the dependencies along each vessel path. The proposed TreeConvGRU fully utilizes the tree structures, thus yielding the best performance at the bifurcation location.

Table 5.3: Comparison of the segmentation accuracy around the bifurcation nodes (within 4 nodes' distance) on the testing set of the aggregated dataset (Total). The compared methods are: DenseVox [4], ConvGRU, and TreeConvGRU.

Methods	DenseVox	ConvGRU	TreeConvGRU
Average Dice	0.7806	0.8223	0.8537

5.4 Summary

Extensive studies of coronary artery segmentation have been spurred by the arising concerns regarding cardiovascular diseases. However, owing to the complex nature

of its anatomical structure, local image perturbations, and appearance or geometry variability, it is still challenging to apply fully automatic algorithms in clinical practices. Inspired by the proposed deep structured learning framework in chapter 3, we use the anatomical information of the coronary artery tree to guide its segmentation. In this way, our network only needs to focus on the local artery segmentation. The reconstructed tree is a collection of nodes, with each of them highly dependent on others. Therefore, we propose tree-structured ConvGRU models to model the inter-node dependency. Accordingly, a tree-structured segmentation network is presented. Augmented with the tree-structured formulation to explicitly model the tree structure, our framework is able to achieve the state-of-the-art performance on four CCTA datasets, demonstrating the effectiveness of the proposed method in the segmentation of complex tree-structured objects.

CHAPTER 6: ANATOMICAL STRUCTURE TRACING WITH DEEP REINFORCEMENT LEARNING

6.1 Motivation

From chapter 3 to 5, we discussed the proposed deep structured learning framework in detail and extensively validated it on several MIA applications. In this chapter, we discuss a special structured learning problem in MIA, anatomical structure tracing. Accurate tracing anatomical structures is crucial for a lot of applications in MIA [56]. Although other preprocessing steps such as segmentation and detection are also commonly used in MIA, anatomical structure tracing provides other key structural information, which is vital for a lot of MIA applications. Anatomical structure tracing is especially useful when medical images lack structural visibility due to occlusion and lack of contrast. Take the diagnosis of coronary artery diseases as an example. A common first step is to build an anatomical structure for the coronary artery, as is illustrated in Fig. 1.3. This greatly facilitates the subsequent analyzing steps such as plaque identification and stenosis detection [57].

A tremendous amount of efforts [57, 58, 59] have been devoted to this line of research. We briefly divide them into three categories. The first category [127, 128] involves computing a minimal cost path between the starting and ending points. This approach results in a high overlap between the prediction and the ground truth structures, but at the cost of potentially suffering from shortcuts [57]. Methods in the second category use an object segmentation [129] or localization [130] to guide the tracing procedure. However, a thorough analysis of the medical images is required, which is extremely time-consuming. The third category consists of approaches that iteratively delineate the anatomical structure [131]. These methods usually have lower

computational overhead. However, they typically suffer from gaps and discontinuities in the results.

In this chapter, we provide an additional perspective on this problem. Specifically, we formulate this task as a sequential decision-making problem and approach it with deep reinforcement learning (DRL), *i.e.*, actor-critic network, a sophisticated DRL framework. Integrating deep learning into reinforcement learning (RL) has been extremely popular since the successful adoption of reinforcement learning (RL) in the past few years [132], including biomedical imaging [133, 134, 135, 136]. It combines the power of feature learning in deep learning and a more sophisticated objective function in sequential decision making. For instance, DRL has been used to detect landmarks in [133, 134, 137, 138], which requires finding a specific structure in an image. It has also been used for image registration [135, 136] and view planning [139], which requires the agent to align two images and locate optimal 2D views in 3D images respectively. Defining an effective reward function is essential for the training of the agent. We show that we are able to effectively train an axon tracing agent which achieves promising results with a carefully designed reward function and a refined training procedure. Instead of using the average integral intensity [140], we show that axon tracing accuracy can be significantly boosted by carefully designing the reward function.

6.2 Methodology

6.2.1 Overview

We now take axon tracing as an example to illustrate how to use DRL to trace anatomical structures in medical images. In this section, we first introduce the mathematical formulation of the axon tracing problem in section 6.2.2. Then, we discuss the essential elements of our DRL system. More specifically, we first discuss the environment and actor in section 6.2.3. Finally, we introduce the reward function used in the axon tracing example and demonstrate how to train the DRL system in

section 6.2.4.

6.2.2 Mathematical Formulation

Given a 2D image of axon image $\mathbf{I}$ and the ground truth axon centerline points $\mathcal{G} = [g_0, g_1, ..., g_n]$, the aim is to train a agent that traces the axon in the image. This is a sequential decision-making problem. In this chapter, we use actor-critic algorithm to solve this problem. Specifically, the agent interacts with the environment over a period of time. At each timestep t, the agent receives a state s_t from the environment and selects an action a_t based on a policy π. Accordingly, a scalar reward r_t is issued by the environment to measure the accuracy of the action a_t. The goal is to minimize the total accumulated return $R_t = \Sigma_{k=0}^{\infty} \gamma^k r_{t+k}$.

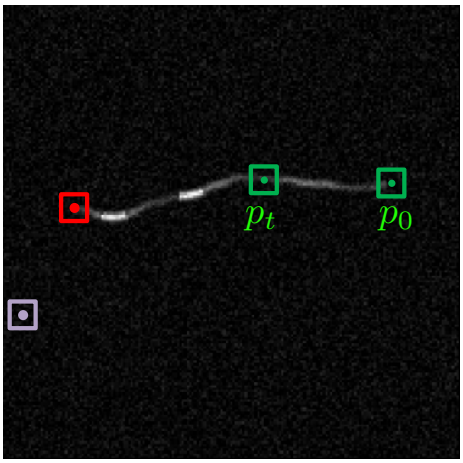

Figure 6.1: The environment of the axon tracing problem. The squares denote the positions of the actor at different timesteps. The actor begins at the start position p_0. p_t denotes the position of the actor's state at timestep t. The red and purple squares denote two possible terminal states. The red one means that the axon is successfully traced and the purple denotes that the actor fails to trace the full axon.

6.2.3 Environment, State Space, and Actor

Environment: As is illustrated in Fig. 6.1, the environment in the axon tracing problem is a 2D greyscale image, as is discussed detailedly in section 6.3.1. Following Dai *et al.* [140], we choose the episode length as 200. It means that the episode

is terminated after at most 200 timesteps. The squares denote the positions of the actor at different timesteps. The actor begins at the start position p_0. p_t denotes the position of the actor's state at timestep t. The red and purple squares denote two possible terminal states. The red means that the axon is successfully traced and the purple one denotes that the actor fails to trace the full axon. During the training stage, after each terminal state, the environment is reset and a new axon image is generated to train the DRL network.

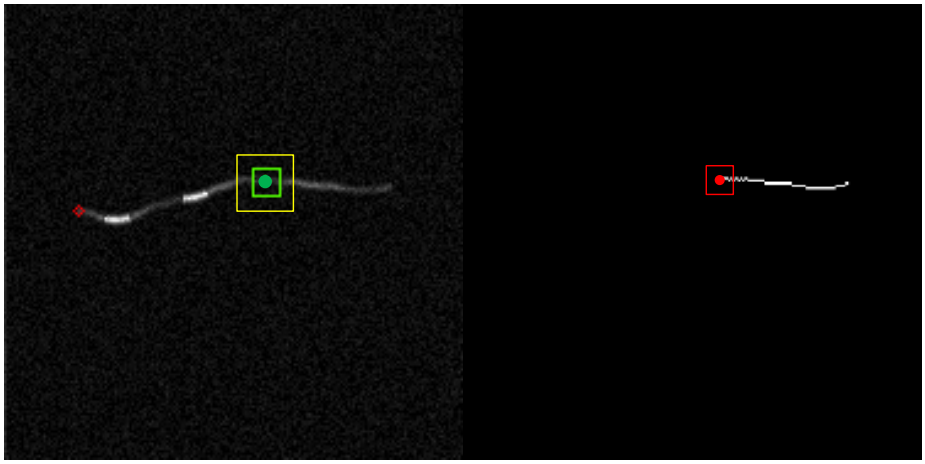

Figure 6.2: The state space in the axon tracing problem. At each timestep t, a three-channel image is generated from the image for both the actor and critic networks. Specifically, a actor-centric view of size 11×11 pixels (green square in the left) is extracted from the original image. Afterward, a larger view of size 21×21 pixels (yellow square in the left) is extracted and downsampled to 11×11 pixels. This technique is used to aid the actor to consider the scale variance. At the same time, the historical path containing all the previous positions of the actor is recorded in a separate image (right). From this image, a 11×11 pixels (red square) is extracted from this image. These three images are concatenated together to form a three-channel state s_t.

State Space: The state space in the axon tracing problem is illustrated in Fig. 6.2. At each timestep t, a three-channel image is generated from the image for both the actor and critic networks. Specifically, an actor-centric view of size 11×11 pixels (green square in the left) is extracted from the original image. Afterward, a larger view of size 21×21 pixels (yellow square in the left) is extracted and downsampled to

11×11 pixels. This technique is used to aid the actor to consider the scale variance. At the same time, the historical path containing all the previous positions of the actor is recorded in a separate image (right). From this image, a 11×11 pixels (red square) is extracted from this image. These three images are concatenated together to form a three-channel state s_t.

Policy and Critic Networks: In the actor-critic algorithms, the policy and value function are modeled by two separate neural networks. In the axon tracing task, the policy network estimates the actor's movement at each timestep t. In our case, the actor can move to one of its neighbors in the 8-neighbors setting. Thus, our policy network outputs a distribution over the 8 possible positions with the last fully-connected layer. Additionally, the policy network consists of two convolutional layers with kernel size 5×5 before the fully-connected layer for feature extraction. The critic network is used as a strong signal to guide the training of the policy network. The critic network is also composed of two convolutional layers with kernel size 5×5. Its final fully-connected layer generates a scalar value for each state s_t.

6.2.4 Reward Function & Training

Reward function: As is illustrated in Fig. 6.3, we consider two scenarios when calculating the reward: the actor is close or too far away from the axon. If the actor is too far away from the axon (left of Fig. 6.3), we want to pull the actor back to the ground truth centerline $\mathcal{G}$. In this scenario, if the axon further moves away from the axon, it receives a negative reward. Otherwise, it receives a positive reward. If the actor is close to the axon (right of Fig. 6.3), the reward is simple: it's the distance the actor moves along the axon. In summary, the reward function is as follows:

$$r_t = \begin{cases} d(\mathcal{G}, p_t) - d(\mathcal{G}, p_{t+1}), & \text{if} \quad d(\mathcal{G}, p_t) > T \\ d(\mathcal{G}, p_t) - d(\mathcal{G}, p_{t+1}) + \vec{e}_t \cdot (p_{t+1} - p_t), & \text{otherwise} \end{cases} \tag{6.1}$$

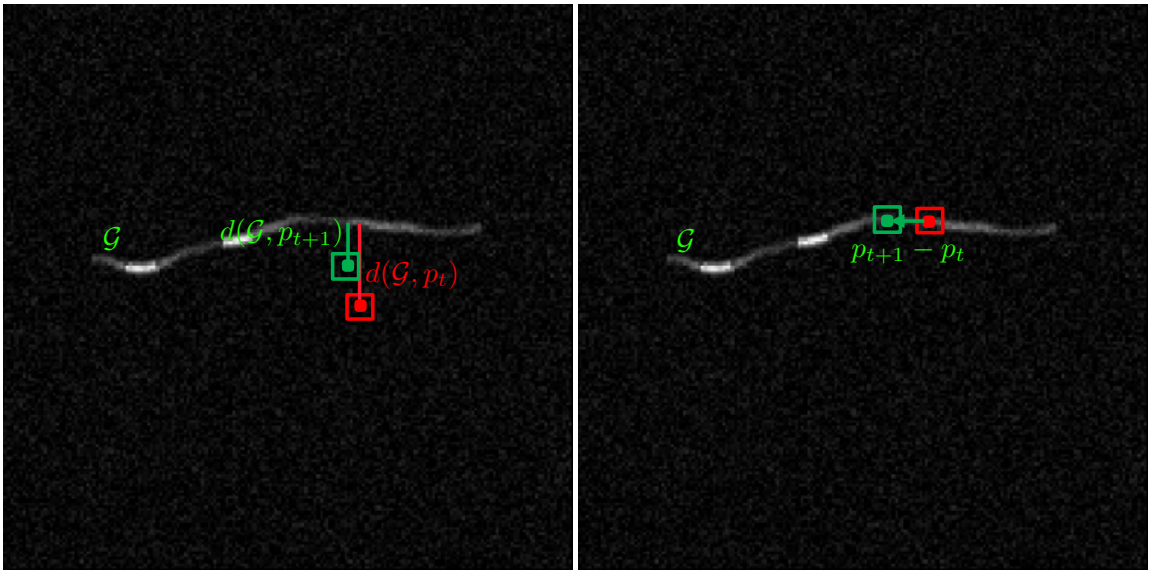

Figure 6.3: Illustration of two scenarios of reward function calculation. **Left:** when the actor is too far away from the axon, the goal is to pull the actor back to the axon. **Right:** when the actor is close the axon, the reward is simple.

where $d(\mathcal{G}, p_t)$ represents the distance between p_t and its closest point in $\mathcal{G}$. T is a predefined threshold, which is empirically chosen as 2.

The actor-critic learning algorithm combines both the power of policy gradient and value function [64]. More specifically, the policy network learns to take the optimal action for each state. While solely using policy gradient optimization can result in high variance, the learned value function is leveraged to reduce the variance [141]. In summary, the actor-critic training procedure is illustrated in Fig. 6.4. At each timestep t, a state s_t is sampled from the environment, which is fed into both the policy and value function networks. The policy network estimates the probability of each action based on state s_t. The value function, on the other hand, estimates the value function of the state s_t regarding each action a_t. After selecting action a_t, the next state s_{t+1} is sampled. The above procedure is repeated until the end of each episode. The detailed training procedure is also shown in Algorithm 4. The actor and value function networks are both modeled by CNNs. This allows them to directly extract features and make inference from images.

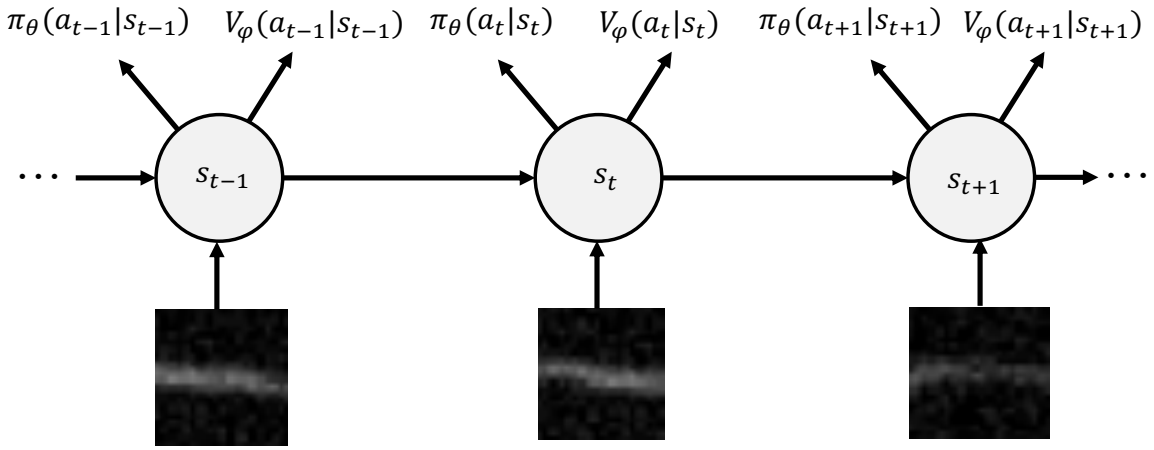

Figure 6.4: Illustration of the procedure of actor-critic learning algorithm. At each timestep t, a state s_t is sampled from the environment, which is fed into both the policy and value function networks. The policy network estimates the probability of each action based on state s_t. The value function, on the other hand, estimates the value function of the state s_t regarding each action a_t. After selecting action a_t, the next state s_{t+1} is sampled. The above procedure is repeated until the end of each episode.

Algorithm 4 The actor-critic training algorithm for axon tracing.

Input: π_θ = policy network with parameter θ
Input: V_φ = value function network with parameter φ
initialize both π_θ and V_φ
while not converged
  for i in $[1, batch\_size]$ **do**
    reset the environment
    for t in $[1, episode\_length]$ **do**
      sample action $a_t = \pi_\theta(s_t)$
      implement a_t and sample the next state s_{t+1} and the reward r_t
      store the transition (s_t, s_{t+1}, r_t)
    end for
  end for
  update π_θ according to equation 6.1
  update V_φ
end while
return π_θ

6.3 Experiments

6.3.1 Datasets & Evaluation Metrics

Our algorithm is evaluated on a synthetic axon dataset, which is similar to [140]. More specifically, all the evaluated networks are trained on a training set with 32,000

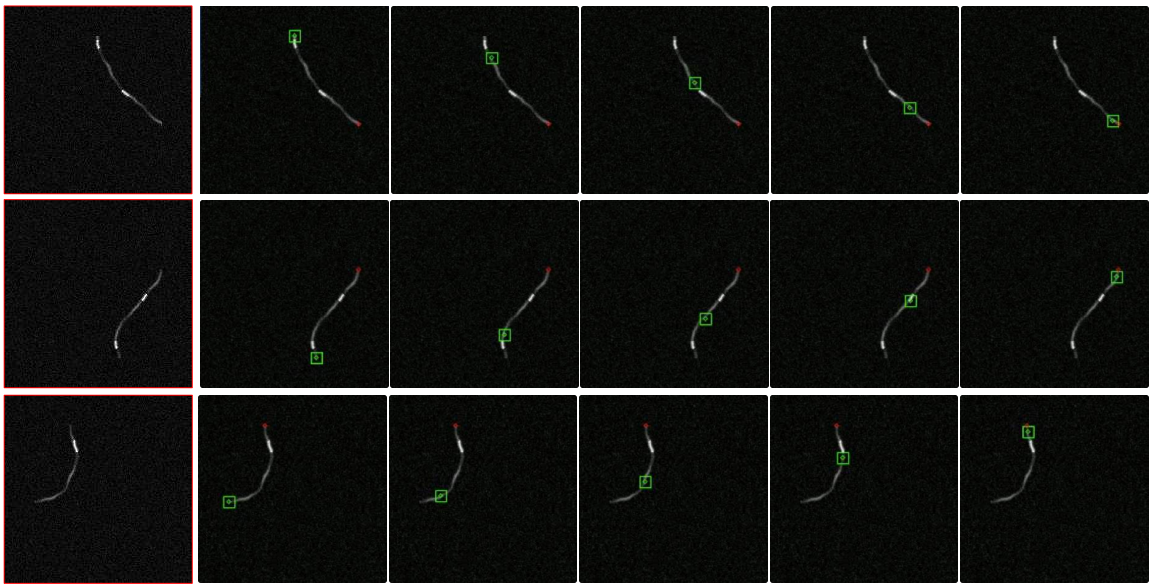

Figure 6.5: Three axon tracing results. The leftmost shows the axon images. Column 2 to column 6 show the agent's positions (denoted by green squares) during the tracing procedure. Red circles indicate the ending points in the axon images.

axon images and tested on a testing set with 1,000 axon images. Each of these images are generated as follows. First, a starting point is randomly selected from the border of the axon image. Afterward, a series of points are selected following the starting point until touching the image border. In order to make the generated axons more realistic, we fit a polynomial spline for these points and randomly added some Gaussian noises to the image. The first column of Fig. 6.5 shows some examples of the generated axon images. The tracing results are evaluated according to the distance between the ground truth axon centerlines and the predictions, which is defined as follows:

$$D(\mathcal{G}, \mathcal{P}) = \frac{1}{n} \sum_{i=1}^{n} \min_{p \in \mathcal{P}} ||g_i - p|| \tag{6.2}$$

Intuitively, the above equation defines the mean distance one has to travel from each point of the ground truth centerline $\mathcal{G}$ to its closest point in the prediction $\mathcal{P}$.

6.3.2 Results

To train the policy network, we randomly select an axon image from the training set and start tracing from the starting point (see section 6.3.1) and train the policy and value function networks according to Algorithm 4. During the testing stage, we also start from the selected starting points and set the initial state of the policy network. We terminate the tracing procedure if the agent reaches the boundary of the axon image or the agent stops moving.

The quantitative results are provided in Table 6.1. Our results significantly outperform the baselin [140], demonstrating the effectiveness of the proposed reward function. We also provide some qualitative tracing results in Fig. 6.5. The leftmost shows the axon images. Column 2 to column 6 show the agent's positions (denoted by green squares) during the tracing procedure. Red circles indicate the ending points in the axon images. Obviously, our method significantly outperforms the baseline approach [121]. This is because our method offers more stability with the new reward function: the agent can be pulled back to the centerline if if deviates from it.

Table 6.1: Quantitative tracing result on the axon images. The results are evaluated in terms of equation 6.2.

Methods	Dai *et al.* [140]	Ours
$D(\mathcal{G}, \mathcal{P})$	1.87	0.93

6.4 Summary

In this chapter, we elaborated on a special structured learning problem, *i.e.*, anatomical structure tracing. Instead of relying on previous approaches like minimal cost path presegmentation, we propose an alternative approach using DRL. We show that we are able to effectively train an axon tracing agent which achieves promising results with a carefully designed reward function and a refined training procedure. Instead

of using the average integral intensity [140], we show that axon tracing accuracy can be significantly boosted by carefully designing the reward function.

CHAPTER 7: CONCLUSIONS AND FUTURE DIRECTION

In this dissertation, a novel framework is proposed to tackle the structured learning problems in MIA. More specifically, our framework comprises of two major components: **(1)** deep feature extractor, which automatically extracts discriminative features from the input image, **(2)** structural feature learner, which models the complex interactions in the input/output variables. Additionally, the prior structural knowledge can be enforced in loss functions to regularize the training to further improve the performance. By employing these approaches, our method can handle structured learning problems in a unified framework. We validated the proposed method on two benchmarks. The superior results demonstrate its effectiveness. As our method is general and ignorant of specific applications/datasets, we expect that it can benefit many other structured MIA tasks. Finally, we observe that some MIA problems can be easily formulated as a sequential decision-making problem and introduce a sophisticated DRL framework to address this issue. Particularly, with a carefully designed reward function and a refined training procedure, we are able to effectively train an axon tracing agent to achieve promising results. There also exist several limitations in this dissertation.

First, in the current setting, we implicitly assume that the training and testing examples are drawn from the same distribution. However, in clinical practices, this may not be the case. For instance, the retinal fundus images in the top and bottom of Fig. 7.1 are acquired by different fundas cameras. As a result, the appearance of these two datasets may vary significantly and the models trained on the first dataset may not generalize to the second one, resulting in their poor performance. The distribution shift between the training and testing data is formally referred to as the

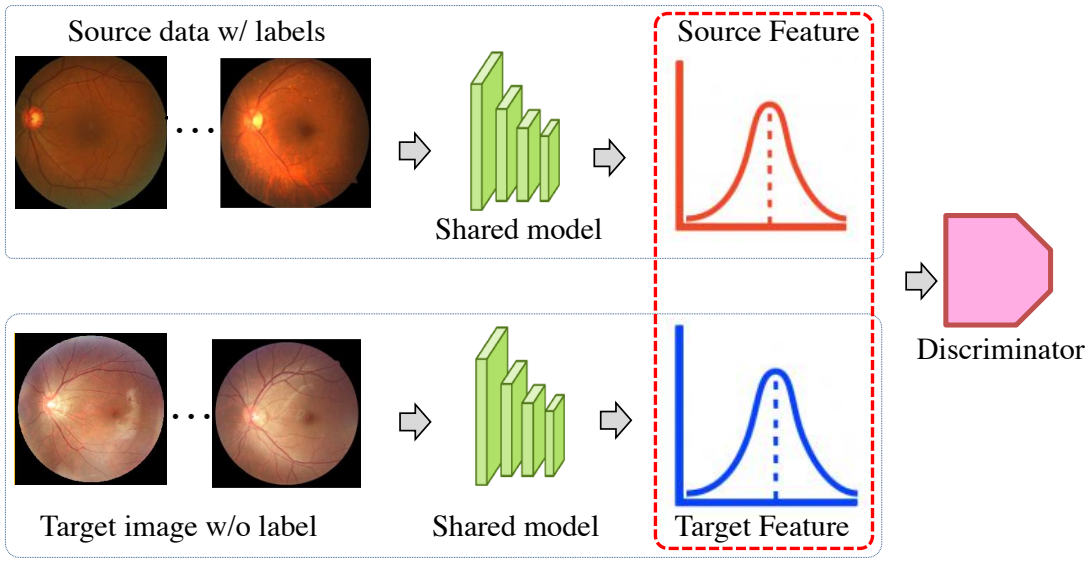

Figure 7.1: The segmentation network can be trained to consider the domain shift problem by forcing the source and target features to lie in the same distribution with adversarial training. Note that labels are not required for the target images.

domain shift. It would be greatly desirable for one model trained on one dataset from one hospital to be generalizable to the datasets from other hospitals. In [142], we have demonstrated that the segmentation network can be trained to consider the domain shift problem by forcing the source and target features to lie in the same distribution with adversarial training. In the future, we plan to explore using more domain adaptation techniques [143, 144, 142] to address this problem.

Second, as hierarchically structured DNNs are extremely complex, they are usually regarded as "black boxes" [145]. If provided with a sufficient amount of training data, they can be trained to produce accurate enough predictions. However, in the field of MIA, the reasoning process is also vitally important. This has led to a surge of research in the direction of interpretability of DNNs [146, 147], including "Explainable AI" launched by DARPA. There is also a second line of research that tries to make DNNs more trustworthy by producing predictions as well as uncertainty estimates. For instance, Bayesian deep learning [148, 149] integrates Bayesian inference with deep learning to estimate uncertainty. Interestingly, in addition to generating un-

certainty estimates, these approaches can also be used to defend against adversarial attacks [149].

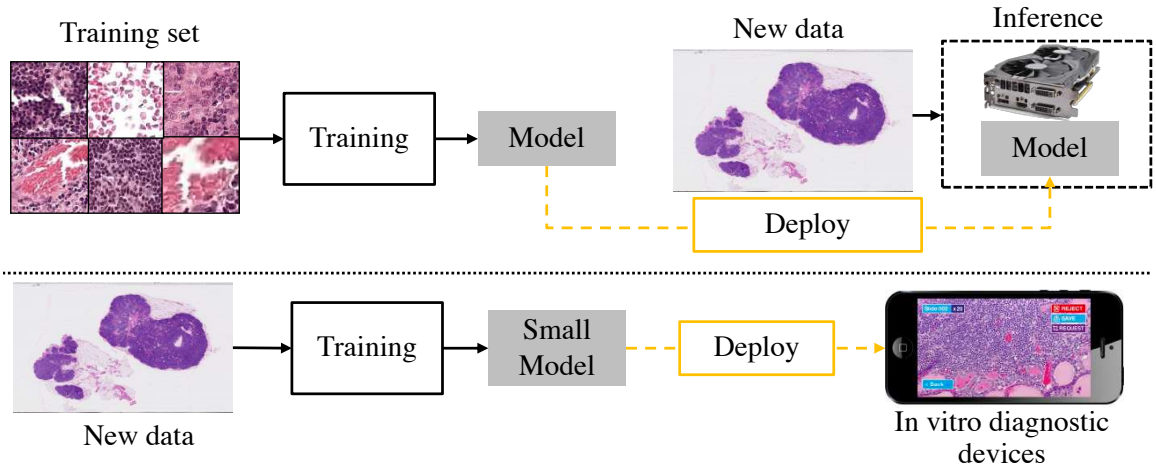

Figure 7.2: **Top:** In the standard machine learning pipeline, a large deep model is trained and then deployed into a server with high-performance computing resources. **Bottom:** In the MIA setting, the reasonable processing time is required to apply CAD algorithms in the clinical setting. It is more desirable that the trained model can be deployed into a small device without high-performance computing resources, *e.g.*, in vitro diagnostic devices.

Third, one challenge often encountered in MIA is dealing with very large images, such as megabyte WSI [150] and 3D CT images. As is illustrated in Fig. 7.2, in the standard machine learning pipeline (top), a large deep model is trained and then deployed into a server with high-performance computing resources. However, in the MIA setting, the reasonable processing time is required to apply CAD algorithms in the clinical setting [151]. It is more desirable that the trained model can be deployed into a small device without high-performance computing resources (bottom), *e.g.*, in vitro diagnostic devices. Therefore, reducing the processing time without losing accuracy is pivotal. How to maintain the prediction accuracy while at the same time make the MIA systems significantly more efficient? We presented a preliminary framework with two key techniques to answer this question in [33]. First, a significant amount of speedup can be achieved by designing a compact network. Second, in order to

maintain accuracy, a large-capacity network trained on the training set is employed to supervise the training of the compact network. Together, these two approaches ensure the efficiency of our network without too much loss of the performance. Recently, neural architecture search [152] has become increasingly popular. It has been widely used to search for small yet effective deep learning architectures. We expect it to be even more popular in the upcoming years.

REFERENCES

[1] S. Darvishi, H. Behnam, M. Pouladian, and N. Samiei, "Measuring left ventricular volumes in two-dimensional echocardiography image sequence using level-set method for automatic detection of end-diastole and end-systole frames," *Research in Cardiovascular Medicine*, vol. 2, no. 1, p. 39, 2013.

[2] M. Uzunbaş, S. Zhang, K. Pohl, D. Metaxas, and L. Axel, "Segmentation of myocardium using deformable regions and graph cuts," in *Proceedings of the IEEE International Symposium on Biomedical Imaging*, vol. 2012, pp. 254–258, 2012.

[3] D. Wang, A. Khosla, R. Gargeya, H. Irshad, and A. H. Beck, "Deep learning for identifying metastatic breast cancer," *arXiv preprint arXiv:1606.05718*, 2016.

[4] L. Yu, J.-Z. Cheng, Q. Dou, X. Yang, H. Chen, J. Qin, and P.-A. Heng, "Automatic 3d cardiovascular mr segmentation with densely-connected volumetric convnets," in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 287–295, Springer, 2017.

[5] D. C. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Mitosis detection in breast cancer histology images with deep neural networks," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 411–418, Springer, 2013.

[6] X. Liu, H. R. Tizhoosh, and J. Kofman, "Generating binary tags for fast medical image retrieval based on convolutional nets and radon transform," in *International Joint Conference on Neural Networks*, pp. 2872–2878, IEEE, 2016.

[7] B. E. Bejnordi, M. Veta, P. J. Van Diest, B. Van Ginneken, N. Karssemeijer, G. Litjens, J. A. Van Der Laak, M. Hermsen, Q. F. Manson, M. Balkenhol, *et al.*, "Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer," *Jama*, vol. 318, no. 22, pp. 2199–2210, 2017.

[8] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, p. 115, 2017.

[9] X. Yang, L. Yu, S. Li, H. Wen, D. Luo, C. Bian, J. Qin, D. Ni, and P.-A. Heng, "Towards automated semantic segmentation in prenatal volumetric ultrasound," *IEEE Transactions on Medical Imaging*, vol. 38, no. 1, pp. 180–193, 2018.

[10] J. I. Orlando, H. Fu, J. B. Breda, K. van Keer, D. R. Bathula, A. Diaz-Pinto, R. Fang, P.-A. Heng, J. Kim, J. Lee, *et al.*, "Refuge challenge: A unified framework for evaluating automated methods for glaucoma assessment from fundus photographs," *Medical Image Analysis*, vol. 59, p. 101570, 2020.

[11] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in Neural Information Processing Systems*, pp. 802–810, 2015.

[12] Y. Shen, Z. Fang, Y. Gao, N. Xiong, C. Zhong, and X. Tang, "Coronary arteries segmentation based on 3d fcn with attention gate and level set function," *IEEE Access*, vol. 7, pp. 42826–42835, 2019.

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

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[15] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision*, pp. 818–833, Springer, 2014.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

[17] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2015.

[18] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[19] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Advances in Neural Information Processing Systems*, pp. 153–160, 2007.

[20] Y. Li, M.-Y. Liu, X. Li, M.-H. Yang, and J. Kautz, "A closed-form solution to photorealistic image stylization," in *Proceedings of the European Conference on Computer Vision*, pp. 453–468, 2018.

[21] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, G. Liu, A. Tao, J. Kautz, and B. Catanzaro, "Video-to-video synthesis," *arXiv preprint arXiv:1808.06601*, 2018.

[22] A. Pumarola, A. Agudo, A. M. Martinez, A. Sanfeliu, and F. Moreno-Noguer, "Ganimation: Anatomically-aware facial animation from a single image," in *Proceedings of the European Conference on Computer Vision*, pp. 818–833, 2018.

[23] X. Zhang, F. Zhou, Y. Lin, and S. Zhang, "Embedding label structures for fine-grained feature representation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1114–1123, 2016.

[24] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proceedings of the International Conference on Machine Learning*, pp. 609–616, ACM, 2009.

[25] R. Socher, B. Huval, B. Bath, C. D. Manning, and A. Y. Ng, "Convolutional-recursive deep learning for 3d object classification," in *Advances in Neural Information Processing Systems*, pp. 656–664, 2012.

[26] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the International Conference on Machine learning*, pp. 160–167, 2008.

[27] J. Hirschberg and C. D. Manning, "Advances in natural language processing," *Science*, vol. 349, no. 6245, pp. 261–266, 2015.

[28] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *Proceedings of the International Conference on Machine Learning*, pp. 173–182, 2016.

[29] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury, *et al.*, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, vol. 29, 2012.

[30] C. Xia, X. Li, X. Wang, B. Kong, Y. Chen, Y. Yin, K. Cao, Q. Song, S. Lyu, and X. Wu, "A multi-modality network for cardiomyopathy death risk prediction with cmr images and clinical information," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 577–585, Springer, 2019.

[31] E. Hosseini-Asl, R. Keynton, and A. El-Baz, "Alzheimer's disease diagnostics by adaptation of 3d convolutional network," in *IEEE International Conference on Image Processing*, pp. 126–130, IEEE, 2016.

[32] B. Kong, X. Wang, Z. Li, Q. Song, and S. Zhang, "Cancer metastasis detection via spatially structured deep network," in *International Conference on Information Processing in Medical Imaging*, pp. 236–248, Springer, 2017.

[33] B. Kong, S. Sun, X. Wang, Q. Song, and S. Zhang, "Invasive cancer detection utilizing compressed convolutional neural network and transfer learning," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 156–164, 2018.

[34] S. Hwang and H.-E. Kim, "Self-transfer learning for weakly supervised lesion localization," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 239–246, Springer, 2016.

[35] C. F. Baumgartner, K. Kamnitsas, J. Matthew, S. Smith, B. Kainz, and D. Rueckert, "Real-time standard scan plane detection and localisation in fetal ultrasound using fully convolutional neural networks," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 203–211, Springer, 2016.

[36] B. Kong, Y. Zhan, M. Shin, T. Denny, and S. Zhang, "Recognizing end-diastole and end-systole frames via deep temporal regression network," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 264–272, 2016.

[37] Q. Song, B. Kong, and S. Sun, "Systems and methods for detecting cancer metastasis using a neural network," Apr. 18 2019. US Patent App. 16/049,809.

[38] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, Springer, 2015.

[39] Ö. Çiçek and et al., "3d u-net: learning dense volumetric segmentation from sparse annotation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 424–432, Springer, 2016.

[40] M. Ghafoorian, N. Karssemeijer, T. Heskes, I. W. van Uden, C. I. Sanchez, G. Litjens, F.-E. de Leeuw, B. van Ginneken, E. Marchiori, and B. Platel, "Location sensitive deep convolutional neural networks for segmentation of white matter hyperintensities," *Scientific Reports*, vol. 7, no. 1, p. 5110, 2017.

[41] T. Brosch, L. Y. Tang, Y. Yoo, D. K. Li, A. Traboulsee, and R. Tam, "Deep 3d convolutional encoder networks with shortcuts for multiscale feature integration applied to multiple sclerosis lesion segmentation," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1229–1239, 2016.

[42] B. Kong, X. Wang, J. Bai, Y. Lu, F. Gao, K. Cao, J. Xia, Q. Song, and Y. Yin, "Learning tree-structured representation for 3d coronary artery segmentation," *Computerized Medical Imaging and Graphics*, p. 101688, 2019.

[43] M. Simonovsky, B. Gutiérrez-Becker, D. Mateus, N. Navab, and N. Komodakis, "A deep metric for multimodal registration," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 10–18, Springer, 2016.

[44] S. Miao, Z. J. Wang, and R. Liao, "A cnn regression approach for real-time 2d/3d registration," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1352–1363, 2016.

[45] X. Yang, R. Kwitt, M. Styner, and M. Niethammer, "Quicksilver: Fast predictive image registration–a deep learning approach," *NeuroImage*, vol. 158, pp. 378–396, 2017.

[46] A. Janowczyk, A. Basavanhally, and A. Madabhushi, "Stain normalization using sparse autoencoders (stanosa): Application to digital pathology," *Computerized Medical Imaging and Graphics*, vol. 57, pp. 50–61, 2017.

[47] A. Benou, R. Veksler, A. Friedman, and T. R. Raviv, "De-noising of contrast-enhanced mri sequences by an ensemble of expert deep neural networks," in *Deep Learning and Data Labeling for Medical Applications*, pp. 95–110, Springer, 2016.

[48] P. Kisilev, E. Sason, E. Barkan, and S. Hashoul, "Medical image description using multi-task-loss cnn," in *Deep Learning and Data Labeling for Medical Applications*, pp. 121–129, Springer, 2016.

[49] Z. Zhang, Y. Xie, F. Xing, M. McGough, and L. Yang, "Mdnet: A semantically and visually interpretable medical image diagnosis network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6428–6436, 2017.

[50] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, pp. 91–99, 2015.

[51] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[52] J. Bogaert, S. Dymarkowski, and A. M. Taylor, *Clinical cardiac MRI*. Taylor & Francis US, 2005.

[53] Y. Li and W. Ping, "Cancer metastasis detection with neural conditional random field," in *Medical Imaging with Deep Learning*, 2018.

[54] O. Oktay, E. Ferrante, K. Kamnitsas, M. Heinrich, W. Bai, J. Caballero, S. A. Cook, A. De Marvao, T. Dawes, D. P. O'Regan, *et al.*, "Anatomically constrained neural networks (acnns): application to cardiac image enhancement and segmentation," *IEEE Transactions on Medical Imaging*, vol. 37, no. 2, pp. 384–395, 2017.

[55] M. S. Nosrati and G. Hamarneh, "Incorporating prior knowledge in medical image segmentation: a survey," *arXiv preprint arXiv:1607.01092*, 2016.

[56] B. Kong, X. Wang, J. Bai, Y. Lu, F. Gao, K. Cao, Q. Song, S. Zhang, S. Lyu, and Y. Yin, "Attention-driven tree-structured convolutional lstm for high dimensional data understanding," *arXiv preprint arXiv:1902.10053*, 2019.

[57] J. M. Wolterink, R. W. van Hamersvelt, M. A. Viergever, T. Leiner, and I. Išgum, "Coronary artery centerline extraction in cardiac ct angiography using a cnn-based orientation classifier," *Medical Image Analysis*, vol. 51, pp. 46–60, 2019.

[58] G. Tetteh, V. Efremov, N. D. Forkert, M. Schneider, J. Kirschke, B. Weber, C. Zimmer, M. Piraud, and B. H. Menze, "Deepvesselnet: Vessel segmentation, centerline prediction, and bifurcation detection in 3-d angiographic volumes," *arXiv preprint arXiv:1803.09340*, 2018.

[59] D. Lesage, E. D. Angelini, I. Bloch, and G. Funka-Lea, "A review of 3d vessel lumen segmentation techniques: Models, features and extraction schemes," *Medical Image Analysis*, vol. 13, no. 6, pp. 819–845, 2009.

[60] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in *Competition and Cooperation in Neural Nets*, pp. 267–285, Springer, 1982.

[61] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[62] W. Zaremba and I. Sutskever, "Learning to execute," *arXiv preprint arXiv:1410.4615*, 2014.

[63] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the International Conference on Machine Learning*, pp. 369–376, ACM, 2006.

[64] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT Press, 2018.

[65] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun, "Support vector machine learning for interdependent and structured output spaces," in *Proceedings of the Twenty-first International Conference on Machine Learning*, pp. 104–112, ACM, 2004.

[66] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[67] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[68] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of*

the International Conference on Machine Learning, pp. 282–289, Morgan Kaufmann Publishers Inc, 2001.

[69] J. Wang, M. Agrawala, and M. F. Cohen, "Soft scissors: an interactive tool for realtime high quality matting," *ACM Transactions on Graphics*, vol. 26, no. 3, p. 9, 2007.

[70] C. Xu and J. L. Prince, "Gradient vector flow: A new external force for snakes," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 66–71, IEEE, 1997.

[71] W. A. Barrett and E. N. Mortensen, "Interactive live-wire boundary extraction," *Medical Image Analysis*, vol. 1, no. 4, pp. 331–341, 1997.

[72] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 309–314, 2004.

[73] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *International Journal of Computer Vision*, vol. 22, no. 1, pp. 61–79, 1997.

[74] G. Hamarneh, J. Yang, C. McIntosh, and M. Langille, "3d live-wire-based semi-automatic segmentation of medical images," in *Medical Imaging 2005: Image Processing*, vol. 5747, pp. 1597–1603, International Society for Optics and Photonics, 2005.

[75] A. Top, G. Hamarneh, and R. Abugharbieh, "Active learning for interactive 3d image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 603–610, Springer, 2011.

[76] L. Grady, "Targeted image segmentation using graph methods," *Image Processing and Analysis with Graphs*, pp. 111–135, 2012.

[77] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, 2000.

[78] M. Rousson and N. Paragios, "Shape priors for level set representations," in *European Conference on Computer Vision*, pp. 78–92, Springer, 2002.

[79] S. M. Pizer, P. T. Fletcher, S. Joshi, A. Thall, J. Z. Chen, Y. Fridman, D. S. Fritsch, A. G. Gash, J. M. Glotzer, M. R. Jiroutek, *et al.*, "Deformable m-reps for 3d medical image segmentation," *International Journal of Computer Vision*, vol. 55, no. 2-3, pp. 85–106, 2003.

[80] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active shape models-their training and application," *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995.

[81] X. Han, C. Xu, and J. L. Prince, "A topology preserving level set method for geometric deformable models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 6, pp. 755–768, 2003.

[82] S. Vicente, V. Kolmogorov, and C. Rother, "Graph cut based image segmentation with connectivity priors," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, 2008.

[83] S. M. A. Eslami, N. Heess, C. K. I. Williams, and J. Winn, "The shape boltzmann machine: A strong model of object shape," *International Journal of Computer Vision*, vol. 107, pp. 155–176, Apr 2014.

[84] F. Chen, H. Yu, R. Hu, and X. Zeng, "Deep learning shape priors for object segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition*, June 2013.

[85] S. Eslami and C. Williams, "A generative model for parts-based object segmentation," in *Advances in Neural Information Processing Systems*, pp. 100–107, Curran Associates, Inc., 2012.

[86] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *The IEEE Conference on Computer Vision and Pattern Recognition*, June 2015.

[87] B. Kong, Y. Zhan, M. Shin, T. Denny, and S. Zhang, "Recognizing end-diastole and end-systole frames via deep temporal regression network," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 264–272, 2016.

[88] P. Moeskops, J. M. Wolterink, B. H. van der Velden, K. G. Gilhuijs, T. Leiner, M. A. Viergever, and I. Išgum, "Deep learning for multi-task medical image segmentation in multiple modalities," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 478–486, Springer, 2016.

[89] H.-C. Shin, K. Roberts, L. Lu, D. Demner-Fushman, J. Yao, and R. M. Summers, "Learning to read chest x-rays: recurrent neural cascade model for automated image annotation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2497–2506, 2016.

[90] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, 2015.

[91] H. Chen, X. Qi, L. Yu, and P.-A. Heng, "Dcan: Deep contour-aware networks for accurate gland segmentation," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2487–2496, 2016.

[92] Y. Xu, Y. Li, M. Liu, Y. Wang, M. Lai, I. Eric, and C. Chang, "Gland instance segmentation by deep multichannel side supervision," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 496–504, Springer, 2016.

[93] M. F. Stollenga, W. Byeon, M. Liwicki, and J. Schmidhuber, "Parallel multi-dimensional lstm, with application to fast biomedical volumetric image segmentation," in *Advances in Neural Information Processing Systems*, pp. 2998–3006, 2015.

[94] X. Gao, S. Lin, and T. Y. Wong, "Automatic feature learning to grade nuclear cataracts based on deep learning," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 11, pp. 2693–2701, 2015.

[95] H. Brody, "Medical imaging," *Nature*, vol. 502, no. 7473, pp. S81–S81, 2013.

[96] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[97] Y. Bengio *et al.*, "Learning deep architectures for ai," *Foundations and Trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[98] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[99] C. R. Dominguez, N. Kachenoura, S. Mulé, A. Tenenhaus, A. Delouche, O. Nardi, O. Gérard, B. Diebold, A. Herment, and F. Frouin, "Classification of segmental wall motion in echocardiography using quantified parametric images," in *International Workshop on Functional Imaging and Modeling of the Heart*, pp. 477–486, Springer, 2005.

[100] P. Gifani, H. Behnam, A. Shalbaf, and Z. A. Sani, "Automatic detection of end-diastole and end-systole from echocardiography images using manifold learning," *Physiological Measurement*, vol. 31, no. 9, p. 1091, 2010.

[101] X. Zhang, H. Dou, T. Ju, and S. Zhang, "Fusing heterogeneous features for the image-guided diagnosis of intraductal breast lesions," in *IEEE International Symposium on Biomedical Imaging*, pp. 1288–1291, IEEE, 2015.

[102] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[103] S. Doyle, S. Agner, A. Madabhushi, M. Feldman, and J. Tomaszewski, "Automated grading of breast cancer histopathology using spectral clustering with textural and architectural image features," in *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 496–499, IEEE, 2008.

[104] A. Shalbaf, H. Behnam, P. Gifani, and Z. Alizadeh-Sani, "Automatic detection of end systole and end diastole within a sequence of 2-d echocardiographic images using modified isomap algorithm," in *Middle East Conference on Biomedical Engineering*, pp. 217–220, IEEE, 2011.

[105] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2625–2634, 2015.

[106] A. A. Abboud, R. W. Rahmat, S. B. Kadiman, M. Z. B. Dimon, L. Nurliyana, M. I. Saripan, and H. H. Khaleel, "Automatic detection of the end-diastolic and end-systolic from 4d echocardiographic images," *Journal of Computer Science*, vol. 11, no. 1, pp. 230–240, 2015.

[107] J. Shiraishi, Q. Li, K. Suzuki, R. Engelmann, and K. Doi, "Computer-aided diagnostic scheme for the detection of lung nodules on chest radiographs: localized search method based on anatomical classification," *Medical Physics*, vol. 33, no. 7, pp. 2642–2653, 2006.

[108] D. P. Zhang, *Coronary artery segmentation and motion modelling.* PhD thesis, Imperial College London, 2010.

[109] F. E. Boas and D. Fleischmann, "Ct artifacts: causes and reduction techniques," *Imaging in Medicine*, vol. 4, no. 2, pp. 229–240, 2012.

[110] D. Lesage, E. D. Angelini, G. Funka-Lea, and I. Bloch, "Adaptive particle filtering for coronary artery segmentation from 3d ct angiograms," *Computer Vision and Image Understanding*, vol. 151, pp. 29–46, 2016.

[111] Z. Gao, X. Liu, S. Qi, W. Wu, W. K. Hau, and H. Zhang, "Automatic segmentation of coronary tree in ct angiography images," *International Journal of Adaptive Control and Signal Processing*, 2017.

[112] P. Strandmark, J. Ulén, F. Kahl, and L. Grady, "Shortest paths with curvature and torsion," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 2024–2031, 2013.

[113] F. Chen, Y. Li, T. Tian, F. Cao, and J. Liang, "Automatic coronary artery lumen segmentation in computed tomography angiography using paired multiscale 3d cnn," in *Medical Imaging 2018: Biomedical Applications in Molecular, Structural, and Functional Imaging*, vol. 10578, p. 105782R, International Society for Optics and Photonics, 2018.

[114] D. Jin, Z. Xu, A. P. Harrison, K. George, and D. J. Mollura, "3d convolutional neural networks with graph refinement for airway segmentation using incomplete data labels," in *International Workshop on Machine Learning in Medical Imaging*, pp. 141–149, Springer, 2017.

[115] Z. Jiang, H. Zhang, Y. Wang, and S.-B. Ko, "Retinal blood vessel segmentation using fully convolutional network with transfer learning," *Computerized Medical Imaging and Graphics*, vol. 68, pp. 1–15, 2018.

[116] Z. Yan, S. Zhang, C. Tan, H. Qin, B. Belaroussi, H. J. Yu, C. Miller, and D. N. Metaxas, "Atlas-based liver segmentation and hepatic fat-fraction assessment for clinical trials," *Computerized Medical Imaging and Graphics*, vol. 41, pp. 80–92, 2015.

[117] S. Zhao, Z. Gao, H. Zhang, Y. Xie, J. Luo, D. Ghista, Z. Wei, X. Bi, H. Xiong, C. Xu, *et al.*, "Robust segmentation of intima–media borders with different morphologies and dynamics during the cardiac cycle," *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 5, pp. 1571–1582, 2017.

[118] Z. Gao, H. Xiong, X. Liu, H. Zhang, D. Ghista, W. Wu, and S. Li, "Robust estimation of carotid artery wall motion using the elasticity-based state-space approach," *Medical Image Analysis*, vol. 37, pp. 1–21, 2017.

[119] L. Xu, X. Huang, J. Ma, J. Huang, Y. Fan, H. Li, J. Qiu, H. Zhang, and W. Huang, "Value of three-dimensional strain parameters for predicting left ventricular remodeling after st-elevation myocardial infarction," *The International Journal of Cardiovascular Imaging*, vol. 33, no. 5, pp. 663–673, 2017.

[120] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proceedings of the International Conference on Machine Learning*, pp. 1310–1318, 2013.

[121] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," *arXiv preprint arXiv:1503.00075*, 2015.

[122] W. Shi, F. Jiang, S. Zhang, and D. Zhao, "Deep networks for compressed image sensing," in *IEEE International Conference on Multimedia and Expo*, pp. 877–882, 2017.

[123] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *NIPS 2014 Workshop on Deep Learning*, 2014.

[124] Z. Guo, J. Bai, Y. Lu, X. Wang, K. Cao, Q. Song, M. Sonka, and Y. Yin, "Deepcenterline: A multi-task fully convolutional network for centerline extraction," in *International Conference on Information Processing in Medical Imaging*, pp. 441–453, Springer, 2019.

[125] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2015.

[126] S. Zhang, D. Zheng, X. Hu, and M. Yang, "Bidirectional long short-term memory networks for relation classification," in *Proceedings of the Pacific Asia Conference on Language, Information and Computation*, pp. 73–78, 2015.

[127] K. Krissian, H. Bogunovic, J. Pozo, M. Villa-Uriol, and A. Frangi, "Minimally interactive knowledge-based coronary tracking in cta using a minimal cost path," *The Insight Journal*, 2008.

[128] O. Wink, A. F. Frangi, B. Verdonck, M. A. Viergever, and W. J. Niessen, "3d mra coronary axis determination using a minimum cost path approach," *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, vol. 47, no. 6, pp. 1169–1175, 2002.

[129] G. Yang, P. Kitslaar, M. Frenay, A. Broersen, M. J. Boogers, J. J. Bax, J. H. Reiber, and J. Dijkstra, "Automatic centerline extraction of coronary arteries in coronary computed tomographic angiography," *The International Journal of Cardiovascular Imaging*, vol. 28, no. 4, pp. 921–933, 2012.

[130] Y. Zheng, H. Tek, and G. Funka-Lea, "Robust and accurate coronary artery centerline extraction in cta by combining model-driven and data-driven approaches," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 74–81, Springer, 2013.

[131] S. Cetin and G. Unal, "A higher-order tensor vessel tractography for segmentation of vascular structures," *IEEE Transactions on Medical Imaging*, vol. 34, no. 10, pp. 2172–2185, 2015.

[132] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[133] F. C. Ghesu, B. Georgescu, T. Mansi, D. Neumann, J. Hornegger, and D. Comaniciu, "An artificial agent for anatomical landmark detection in medical images," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 229–237, Springer, 2016.

[134] G. Maicas, G. Carneiro, A. P. Bradley, J. C. Nascimento, and I. Reid, "Deep reinforcement learning for active breast lesion detection from dce-mri," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 665–673, Springer, 2017.

[135] R. Liao, S. Miao, P. de Tournemire, S. Grbic, A. Kamen, T. Mansi, and D. Comaniciu, "An artificial agent for robust image registration," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

[136] J. Krebs, T. Mansi, H. Delingette, L. Zhang, F. C. Ghesu, S. Miao, A. K. Maier, N. Ayache, R. Liao, and A. Kamen, "Robust non-rigid registration through agent-based action learning," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 344–352, Springer, 2017.

[137] W. A. Al and I. D. Yun, "Partial policy-based reinforcement learning for anatomical landmark localization in 3d medical images," *arXiv preprint arXiv:1807.02908*, 2018.

[138] F.-C. Ghesu, B. Georgescu, Y. Zheng, S. Grbic, A. Maier, J. Hornegger, and D. Comaniciu, "Multi-scale deep reinforcement learning for real-time 3d-landmark detection in ct scans," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 1, pp. 176–189, 2017.

[139] A. Alansary, L. Le Folgoc, G. Vaillant, O. Oktay, Y. Li, W. Bai, J. Passerat-Palmbach, R. Guerrero, K. Kamnitsas, B. Hou, *et al.*, "Automatic view planning with multi-scale deep reinforcement learning agents," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 277–285, Springer, 2018.

[140] T. Dai, M. Dubois, K. Arulkumaran, J. Campbell, C. Bass, B. Billot, F. Uslu, V. de Paola, C. Clopath, and A. A. Bharath, "Deep reinforcement learning for subpixel neural tracking," in *International Conference on Medical Imaging with Deep Learning*, pp. 130–150, 2019.

[141] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.

[142] P. Liu, B. Kong, Z. Li, S. Zhang, and R. Fang, "CFEA: Collaborative feature ensembling adaptation for domain adaptation in unsupervised optic disc and cup segmentation," *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2019.

[143] Q. Dou, C. Ouyang, C. Chen, H. Chen, and P.-A. Heng, "Unsupervised cross-modality domain adaptation of convnets for biomedical image segmentations with adversarial loss," in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 691–697, AAAI Press, 2018.

[144] C. S. Perone, P. Ballester, R. C. Barros, and J. Cohen-Adad, "Unsupervised domain adaptation for medical imaging segmentation with self-ensembling," *NeuroImage*, 2019.

[145] D. Castelvecchi, "Can we open the black box of ai?," *Nature News*, vol. 538, no. 7623, p. 20, 2016.

[146] C. Olah, A. Satyanarayan, I. Johnson, S. Carter, L. Schubert, K. Ye, and A. Mordvintsev, "The building blocks of interpretability," *Distill*, vol. 3, no. 3, p. e10, 2018.

[147] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," *Digital Signal Processing*, vol. 73, pp. 1–15, 2018.

[148] S. M. Murray, "An exploratory analysis of multi-class uncertainty approximation in bayesian convolutional neural networks," Master's thesis, The University of Bergen, 2018.

[149] Y. Li and Y. Gal, "Dropout inference in bayesian neural networks with alpha-divergences," in *Proceedings of the International Conference on Machine Learning*, pp. 2052–2061, 2017.

[150] B. Kong, Z. Li, and S. Zhang, "Toward large-scale histopathological image analysis via deep learning," in *Biomedical Information Technology*, pp. 397–414, Elsevier, 2020.

[151] S. Zhang and D. Metaxas, "Large-scale medical image analytics: Recent methodologies, applications and future directions," *Medical Image Analysis*, vol. 33, pp. 98–101, 2016.

[152] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," in *International Conference on Learning Representations*, 2017.