DATA-EFFICIENT AND ROBUST DEEP LEARNING BASED APPROACHES FOR BIOMEDICAL IMAGE SEGMENTATION AND RELATED PROBLEMS

A Dissertation

Submitted to the Graduate School
of the University of Notre Dame
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy

by

Hao Zheng

Danny Z. Chen, Co-Director

Chaoli Wang, Co-Director

Graduate Program in Computer Science and Engineering
Notre Dame, Indiana

January 2022

DATA-EFFICIENT AND ROBUST DEEP LEARNING BASED APPROACHES FOR BIOMEDICAL IMAGE SEGMENTATION AND RELATED PROBLEMS

Abstract

by

Hao Zheng

Image segmentation is a fundamental problem in computer vision and has been studied for decades. It is also an essential preliminary step for quantitative biomedical image analysis and computer-aided diagnoses and studies. Recently, deep learning (DL) based methods have witnessed huge success on various image analysis tasks in terms of accuracy and generality. However, it is not straightforward to apply known semantic segmentation algorithms directly to biomedical images due to different imaging techniques and special application scenarios (e.g., volumetric images, multi-modal data, small amounts of annotated data, domain knowledge from experts). In this dissertation, I develop new deep learning methods to save annotation efforts, improve model efficacy, and generalize well to different biomedical image segmentation tasks.

First, I introduce advanced model architectures and training algorithms to make use of abundant 3D information from volumetric images (e.g., MR and CT images) for delineating detailed structures. The heterogeneous feature aggregation network utilizes anisotropic 3D convolutional kernels to explicitly extract and fuse contextual information from orthogonal geometric views. I also devise a new ensemble learning framework to unify the merits of both 2D and 3D DL models and boost segmentation performance significantly. Second, I introduce the representative annotation method to only select the most effective areas/samples for annotation, thus saving manual ef-

forts. Our method decouples the selection process from the segmentation process and makes a one-shot suggestion. It can achieve comparable performance to full annotation and active learning based methods. Third, noticing that using sparse annotation leads to huge performance degradation, I introduce two semi-supervised methods to leverage unlabeled images and utilize automatically generated labels (i.e., pseudo labels) in model training. Specifically, I propose combining representative annotation and ensemble learning to bridge the performance gap compared with full annotation methods. I also propose a method to estimate the uncertainty of pseudo labels and use them to guide iterative self-training. Fourth, I present a new self-supervised learning framework to extract generic knowledge directly from unlabeled data and demonstrate its high robustness and efficiency on diverse downstream segmentation tasks.

DEDICATION

To my parents Jinzhu Zheng and Qiufang Liu.

CONTENTS

FIGURES

TABLES

ACKNOWLEDGMENTS

It is no small thing to complete a dissertation in the midst of a global pandemic, and such an achievement would not have been possible for me without the support of many people. First of all, I would like to express my sincere gratitude to my advisor, Dr. Danny Z. Chen, whose support and patience I have benefited from for several years and whose erudition and kindness will be models for years to come. He shows by example how a good researcher thinks critically, faces challenges and overcomes difficulties, and stays curious, persistent, and enthusiastic about research all the time. His insightful guidance, constant encouragement, and the freedom he gave me at all stages helped me to become one. I also extend profuse thanks to my co-advisor, Dr. Chaoli Wang. He has significantly encouraged me to broaden my research and participate in diverse projects, provided unwavering support and perceptive feedback for my research projects, and made tremendous efforts to revise my papers. I benefit immensely from his attitude and passion for research. His unflagging trust and confidence in me encouraged me to improve constantly.

I would like to thank Dr. Joan T. Richtsmeier for her support and help in our collaborative project on chondrocranium development. Appreciation is due to Dr. Susan M. Motch Perrine, Dr. M. Kathleen Pitirri, and Dr. Kazuhiko Kawasaki from the Richtsmeier lab. Their expertise has broadened my horizons and helped my research career greatly. Their invaluable comments and feedback inspired me to seek new research directions and strive for a breakthrough. And my visits to Penn State were pleasant and memorable.

I also want to give thanks to my committee members, Dr. Yiyu Shi, Dr. Walter

Scheirer, Dr. Meng Jiang, Dr. Joan T. Richtsmeier, and my advisors, who took the time to assess my work and provide insightful feedback. I am indebted to the staff in the Computer Science and Engineering Department, especially our administrative assistant, Mrs. Joyce Yeats, as well as the people at the University of Notre Dame, who helped me survive the toughest times throughout this journey.

I am immensely grateful to Lin Yang, Yizhe Zhang, and Jianxu Chen. It has been a privilege to benefit from the generous professional and personal support and advice of these gifted researchers. And it is my honor to work with my colleagues and friends, Shenglong Zhu, Jun Han, Hongxiao Wang, Zhuo Zhao, Peixian Liang, Pengfei Gu, Yejia Zhang, Yaopeng Peng, and Chengtao Peng. I learned a lot from them, and the time we spent together at Notre Dame was beautiful and invaluable. Our friendship is a lifelong treasure.

The research in this dissertation was supported in part by the National Science Foundation (NSF) through grants CCF-1617735, CNS-1629914, IIS-1455886, DUE-1833129, and IIS-1955395, and the National Institute of Health (NIH) through grant R01 DE027677.

Finally, and most importantly, I would like to thank my family and friends for their endless solicitude and support, which helped me through the most difficult times. And I dedicate this dissertation to my beloved parents.

CHAPTER 1

INTRODUCTION

1.1 Background

Biomedical image analysis performs computing on images to obtain qualitative and quantitative information for biological discoveries [48, 166, 182] and medical research [160, 185]. Applications include image registration, tissue or organ detection and segmentation, morphological and pathological analysis, disease diagnosis, surgical planning, and so on [31, 49, 65, 66, 78, 160]. Image segmentation is a fundamental problem in computer vision that partitions a digital image into multiple segments in which pixels with the same label share certain characteristics. Automated segmentation methods have significantly advanced the biomedical image analysis.

Deep learning frees people from designing hand-crafted features by data-driven representation learning (Fig. 1.1 (a)) and has achieved tremendous success in many domains, including natural language processing (e.g., machine translation and language modeling) and computer vision (e.g., classification, detection, and segmentation) [81]. These advances also accelerate the development of biomedical image analysis [185]. There are two types of challenges in applying deep learning to biomedical image segmentation tasks. (1) Special imaging properties. Medical images have multiple modalities, such as computed tomography (CT), magnetic resonance (MR), ultrasound (US), and histopathology images, so image contrast and intensity distribution are very diverse. Volumetric images (e.g., 3D(+T) MR and CT) are routinely invented and some images are of large size. Medical images are heterogeneous and

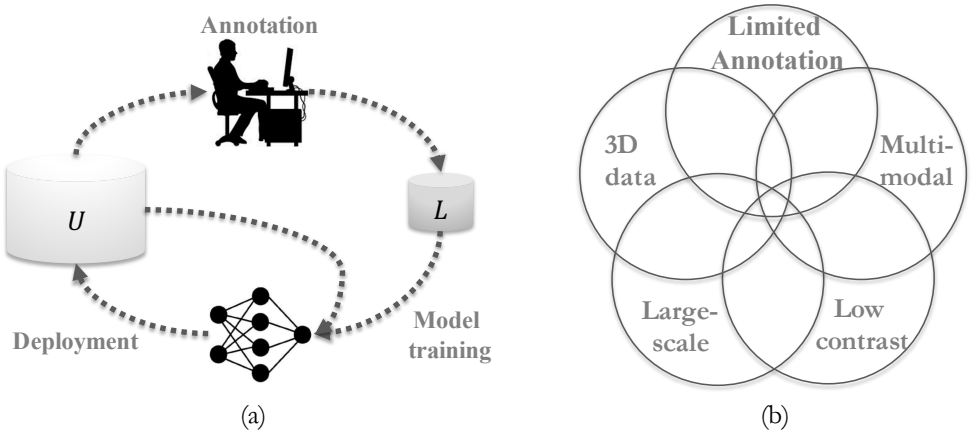

Figure 1.1. (a) The deep learning pipeline. After dataset U (unlabeled) is collected, experts annotate a certain number of them (i.e., labeled set L). The neural network is trained with L (and some unlabeled data). A good model should be able to achieve good performance on unseen data. (b) Typical biomedical image analysis challenges, which are entangled in real-world applications.

imbalanced because they are collected at different facilities, by different protocols, at different times, and the incidence of disease typically has a long-tail distribution. (2) Limited amount of training data. Because manual annotation requires high expertise and is labor-intensive and time-consuming, it is usually difficult to collect a sufficiently large annotated dataset. Moreover, annotation could be inconsistent and noisy due to various experiences and ambiguities in images. Combining these two factors together, we usually encounter several difficulties simultaneously when we deploy deep learning algorithms in real-world applications (e.g., five common obstacles we encounter are shown in Fig. 1.1 (b)) and experts always have higher requirements for us, such as model robustness and interpretability due to medical safety and ethics concerns.

These challenges hinder the deployment of deep learning models in various healthcare applications and scientific research. There is a great demand for *data-efficient and robust deep learning techniques that can make the most of both labeled and un-*

labeled data and generalize to diverse biomedical tasks without requiring intensive manual annotation effort.

In this dissertation, I address these challenges from the following three specific aspects of the deep learning pipeline (Fig. 1.1 (a)): (1) Model training. Given training data (labeled or unlabeled), we are supposed to utilize them, advanced model architectures, and training algorithms to achieve as good performance as possible. I propose a heterogeneous feature aggregation network to explicitly extract and fuse contextual information from orthogonal geometric views of volumetric data (Chapter 2). And I devise a new ensemble learning framework to unify the merits of both 2D and 3D deep learning models and boost segmentation performance significantly (Chapter 4). (2) Necessary annotation and efficient training. I introduce the representative annotation method to only select the most effective areas/samples for annotation, thus saving manual effort while maintaining high performance (Chapter 3). Moreover, I leverage unlabeled images and introduce two semi-supervised methods to (a) bridge the performance gap with respect to full annotation methods by ensemble learning (Chapter 5), and (b) estimate the uncertainty of pseudo labels and use them to guide iterative self-training (Chapter 6). (3) Deployment. The model should generalize well on diverse new data, so I present a new self-supervised learning framework to extract generic knowledge directly that is beneficial for diverse downstream segmentation tasks (Chapter 7).

1.2 Overview of Main Results

In this section, I will briefly discuss the motivation, ideas, and main results of the 3D network architecture design (Section 1.2.1), training method (Section 1.2.3), annotation selection and efficient training work (Section 1.2.2 and Section 1.2.4), uncertainty-guided cartilage segmentation (Section 1.2.5), and self-supervised pre-training framework (Section 1.2.6).

1.2.1 Heterogeneous Feature Aggregation Network for 3D Cardiovascular Image Segmentation

Automatic and accurate cardiovascular image segmentation is important in clinical applications. However, due to large variations in different subjects, ambiguous borders, subtle structures (e.g., thin myocardium), and inhomogeneous appearance and image quality, parsing fine-grained structures in 3D cardiovascular images is very challenging.

In Chapter 2, we propose a novel deep *heterogeneous feature aggregation network* (HFA-Net) to fully exploit complementary information from multiple views of 3D cardiac data [180]. First, we utilize asymmetrical 3D kernels and pooling to obtain heterogeneous features in parallel encoding paths. Thus, from a specific view, distinguishable features are extracted and indispensable contextual information is kept (rather than quickly diminished after symmetrical convolution and pooling operations). Then, we employ a content-aware multi-planar fusion module to aggregate meaningful features to boost segmentation performance. Further, to reduce the model size, we devise a new DenseVoxNet model by sparsifying residual connections, which can be trained in an end-to-end manner. We demonstrate the effectiveness and robustness of our new HFA-Net on several MRI/CT datasets, achieving state-of-the-art performance.

1.2.2 Biomedical Image Segmentation via Representative Annotation

Due to the diversity and complexity of biomedical image data, manual annotation for training common deep learning models is very time-consuming and labor-intensive, especially because normally only biomedical experts can annotate image data well. Human experts are often involved in a long and iterative process of annotation, as in active learning type annotation schemes.

In Chapter 3, we propose *representative annotation* (RA), a new deep learning

framework for reducing annotation effort in biomedical image segmentation [179]. RA uses unsupervised networks for feature extraction and selects representative image patches for annotation in the latent space of learned feature descriptors, which implicitly characterizes the underlying data while minimizing redundancy. A fully convolutional network (FCN) is then trained using the annotated selected image patches for image segmentation. Our RA scheme offers three compelling advantages: (1) It leverages the ability of deep neural networks to learn better representations of image data; (2) it performs one-shot selection for manual annotation and frees annotators from the iterative process of common active learning based annotation schemes; (3) it can be deployed to 3D images with simple extensions. We evaluate our RA approach using three datasets (two 2D and one 3D) and show our framework yields competitive segmentation results comparing with state-of-the-art methods.

1.2.3 A New Ensemble Learning Framework for 3D Biomedical Image Segmentation

Many 2D and 3D deep learning models have achieved state-of-the-art segmentation performance on 3D biomedical image datasets. Yet, 2D and 3D models have their own strengths and weaknesses, and by unifying them together, one may be able to achieve more accurate results.

In Chapeter 4, we propose a new *ensemble learning* framework for 3D biomedical image segmentation that combines the merits of 2D and 3D models [181]. First, we develop a fully convolutional network based meta-learner to learn how to improve the results from 2D and 3D models (base-learners). Then, to minimize over-fitting for our sophisticated meta-learner, we devise a new training method that uses the results of the base-learners as multiple versions of "ground truths". Furthermore, since our new meta-learner training scheme does not depend on manual annotation, it can utilize abundant unlabeled 3D image data to further improve the model.

Extensive experiments on two public datasets show that our approach is effective under fully-supervised, semi-supervised, and transductive settings, and attains superior performance over state-of-the-art image segmentation methods.

1.2.4 An Annotation Sparsification Strategy for 3D Medical Image Segmentation via Representative Selection and Self-Training

Data annotation is a big bottleneck to Deep learning (DL) based 3D segmentation because (1) DL models tend to need a large amount of labeled data to train, and (2) it is highly time-consuming and label-intensive to voxel-wise label 3D medical images. Significantly reducing annotation effort while attaining good performance of DL segmentation models remains a major challenge. We find that, using partially labeled datasets, there is indeed a large performance gap with respect to using fully annotated training datasets.

In Chapter 5, we propose a new DL framework for reducing annotation effort and bridging the gap between full annotation and sparse annotation in 3D medical image segmentation [183]. We achieve this by (i) selecting representative slices in 3D images that minimize data redundancy and save annotation effort, and (ii) self-training with pseudo-labels automatically generated from the base-models trained using the selected annotated slices. Extensive experiments using two public datasets show that our framework yields competitive segmentation results comparing with state-of-the-art DL methods using less than $\sim 20\%$ of annotated data.

1.2.5 Embryonic Cartilage Segmentation in High-Resolution 3D Micro-CT Images with Very Sparse Annotation

Craniofacial syndromes often involve skeletal defects of the head. Studying the development of the *chondrocranium* (the part of the endoskeleton that protests the brain and other sense organs) is crucial to understanding genotype-phenotype relationships

and early detection of skeletal malformation as the chondrocranium forms prior to mineralization of cranial bones of the skull. Our goal is to segment craniofacial cartilages in 3D micro-CT images of embryonic mice stained with phosphotungstic acid. However, due to high image resolution, complex object structures, and low contrast, delineating fine-grained structures in these images is very challenging, even manually. Specifically, only experts can differentiate cartilages, and it is unrealistic to manually label whole volumes for training deep learning models.

In Chapter 6, we propose a new framework to progressively segment cartilages in high-resolution 3D micro-CT images using extremely sparse annotation (e.g., annotating only a few selected slices in a volume) [182]. Specifically, to deal with such high-dimensional data, our method consists of a lightweight fully convolutional network (FCN) to accelerate the training and generate pseudo labels (PLs) for unlabeled slices. Meanwhile, we take into account the reliability of PLs by devising a bootstrap ensemble based uncertainty quantification method. Next, our framework gradually learns from the PLs with the guidance of the uncertainty estimation via self-training.

1.2.6 Hierarchical Self-Supervised Learning for Medical Image Segmentation Based on Multi-Domain Data Aggregation

A large labeled dataset is a key to the success of supervised deep learning, but for medical image segmentation, it is highly challenging to obtain sufficient annotated images for model training. In many scenarios, unannotated images are abundant and easy to acquire. Self-supervised learning (SSL) has shown great potentials in exploiting raw data information and representation learning.

In Chapter 7, we propose Hierarchical Self-Supervised Learning (HSSL), a new self-supervised framework that boosts medical image segmentation by making good use of unannotated data [184]. Unlike the current literature on task-specific self-supervised pretraining followed by supervised fine-tuning, we utilize SSL to learn

task-agnostic knowledge from heterogeneous data for various medical image segmentation tasks. Specifically, we first aggregate a dataset from several medical challenges, then pre-train the network in a self-supervised manner, and finally fine-tune on labeled data. We develop a new loss function by combining contrastive loss and classification loss, and pre-train an encoder-decoder architecture for segmentation tasks. Our extensive experiments show that multi-domain joint pre-training benefits downstream segmentation tasks and outperforms single-domain pre-training significantly. Compared to learning from scratch, our method yields better performance on various tasks (e.g., +0.69% to +18.60% in Dice with 5% of annotated data). With limited amounts of training data, our method can substantially bridge the performance gap with respect to denser annotations (e.g., 10% vs. 100% annotations).

1.3 Author Contribution

To demonstrate the effectiveness of our methods, we conducted extensive experiments on various public challenge datasets and a real application provided by our collaborators. The micro-CT mouse images and manual annotations are provided by Dr. Joan T. Richtsmeier, Dr. Susan M. Motch Perrine, Dr. M. Kathleen Pitirri, and Dr. Kazuhiko Kawasaki (Department of Anthropology at Pennsylvania State University). All approaches and implementations presented in this dissertation are my original work, except for the following collaboration. In the ensemble learning framework (Chapter 4), the random-fit module and the experiment on the mouse piriform cortex dataset should be partially credited to Lin Yang (Department of Computer Science and Engineering at the University of Notre Dame), and the implementation and design of the NN-fit module should be credited to Yizhe Zhang (Department of Computer Science and Engineering at the University of Notre Dame). In the HSSL framework (Chapter 7), the implementation and experiments of baselines (rotation and in-painting; MoCo) should be partially credited to Jun Han (Department of

Computer Science and Engineering at the University of Notre Dame) and Hongxiao Wang (Department of Computer Science and Engineering at the University of Notre Dame).

1.4 Organization of the Dissertation

The rest of this dissertation is organized as follows: Chapters 2 presents a new 3D neural network to explicitly extract and fuse contextual information from orthogonal geometric views to delineate detailed structures. Chapter 3 presents a new one-shot representative selection framework to reduce annotation efforts by directly identifying the most diverse and informative samples. Chapter 4 presents a new ensemble learning framework to unify the merits of 2D and 3D deep learning models for 3D biomedical image segmentation. Chapter 5 presents a new framework that combines representative selection and ensemble learning to bridge the performance gap of 3D biomedical image segmentation between using sparse annotation and full annotation. Chapter 6 presents a new semi-supervised method that estimates the uncertainty of generated pseudo labels efficiently and utilizes them for refining segmentation via iterative self-training. Chapter 7 presents a new self-supervised learning framework that can extract task-agnostic knowledge from heterogeneous unlabeled data. Chapter 8 summarizes the main results and discusses some future directions to follow based on the work in this dissertation.

CHAPTER 2

HETEROGENEOUS FEATURE AGGREGATION NETWORK FOR 3D CARDIOVASCULAR IMAGE SEGMENTATION

A paper published in *2019 22nd International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)* [180]

2.1 Backgrounds

Cardiovascular disease is a leading cause of death globally. Segmenting the whole heart in cardiovascular images is a prerequisite for morphological and pathological analysis, disease diagnosis, and surgical planning [116]. However, automatic and accurate cardiovascular image segmentation remains very challenging due to large variations in different subjects, missing/ambiguous borders, and inhomogeneous appearance and image quality (e.g., see Fig. 2.1(a-b)).

Recent studies showed that deep learning based methods [45, 90, 93, 168, 181] can learn robust contextual and semantic features and achieve state-of-the-art segmentation performance. 3D fully convolutional networks (FCNs) are a mainstream approach for cardiac segmentation due to their ability to integrate both inter- and intra-slice information in 3D images. However, two key factors have not been well explored: (1) the imaging qualities in different anatomical planes are not the same, and thus the degrees of segmentation difficulty from different views are unequal; (2) subtle structures (e.g., myocardium, pulmonary artery) have different orientations in different anatomical planes. Symmetrical convolutional and pooling operations may cause quick diminishment of subtle structures or boundaries, incurring segmentation

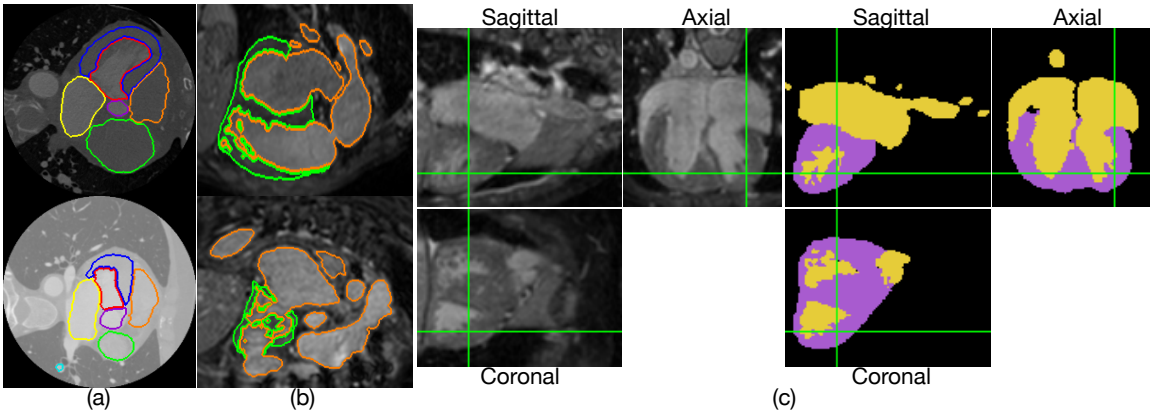

Figure 2.1. Examples of cardiovascular images from (a) the MM-WHS CT dataset [193] in the axial plane and (b) the HVSMR dataset [116] in the sagittal plane. (c) Myocardium boundaries in the axial plane are easier to recognize.

errors. As shown in Fig. 2.1(c), myocardium boundaries in the axial plane are easier to recognize; with asymmetrical pooling along the longitudinal axis, more complementary inter-slice information can be kept which in return benefits segmentation in the axial plane.

Many recent studies tried to tackle the anisotropic issue of 3D biomedical images. But still, they could not segment myocardium or pulmonary artery well. Known methods that explored anisotropic 3D kernels in FCNs can be categorized into two types. (1) The methods in [45, 124] focused on designing repeatable cell structures and replaced all 3D convolutions systematically, called *short-range asymmetrical cell*. However, symmetrical pooling was used and deep features were fused periodically (with distinctive features vanishing quickly). (2) The methods in [90, 96] dealt with the anisotropic problem in 3D images using 2D FCNs to extract intra-slice features and 3D FCNs to aggregate inter-slice features. But, they did not exploit the fact that complementary information in the other planes (xz- and yz-planes) can also benefit the xy-plane, especially in less anisotropic 3D data (e.g., when the spacing resolution

in the z-axis is only $3 \sim 5\times$ larger than that of the x- and y-axes).

To address the above two key factors, we propose a new *heterogeneous feature aggregation network* (HFA-Net), which is able to fully exploit complementary information in multiple views of 3D cardiac images and aggregate heterogeneous features to boost segmentation performance. To handle the issue in [45, 124], we utilize long-range asymmetrical branches to maintain distinguishable features associated with a specific view. Besides asymmetrical convolutional operations, we also apply asymmetrical pooling operations to maintain spatial resolution in the other planes. To address the issue in [90, 96], we utilize parallel encoding paths to extract heterogeneous features from multiple geometric views of the 3D data (i.e., the axial, coronal, and sagittal planes). There is a good chance that an object can be distinguished from at least one of the geometric views. Thus, we encourage richer contextual and semantic features to be extracted. Further, to improve the parameter-performance efficiency and reduce GPU memory usage, we devise a sparsified densely-connected convolutional block for our model, and our HFA-Net thus designed can be trained end-to-end.

Experiments on three public challenge datasets [116, 161, 193] show that our new method achieves competitive segmentation results over state-of-the-art methods.

2.2 Method

Our HFA-Net has three main components (see Fig. 2.2): (1) Long-range asymmetrical branches (LRABs) that preserve subtle structures via asymmetrical convolutions and poolings; (2) a content-aware fusion module (CAFM) that combines multiple asymmetrical branches together, utilizing both raw images and feature maps from LRABs; (3) a new 3D sparse aggregation block (SAB) to reduce GPU memory usage and enable end-to-end training of the entire network.

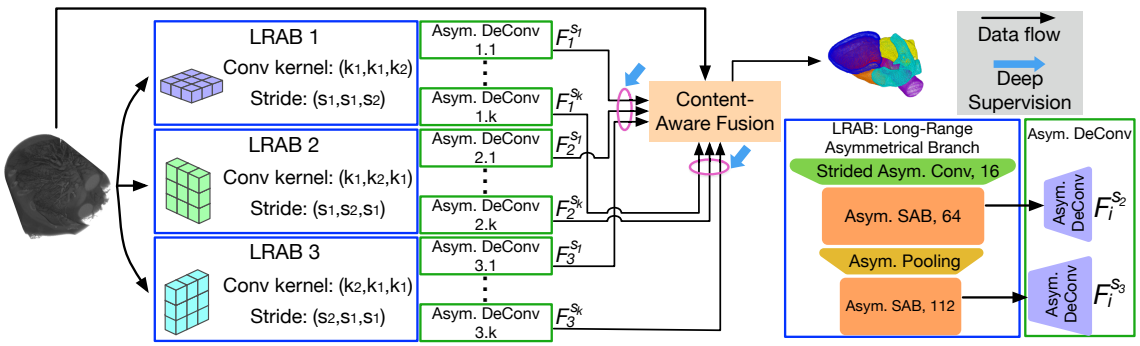

Figure 2.2. An overview of our new HFA-Net framework.

2.2.1 Long-Range Asymmetric Branch (LRAB)

A straightforward way to exploit multiple geometric views of 3D images is to replace conventional 3D convolutional (Conv) layers by *short-range asymmetrical cell* (SRAC) [45, 124]. As shown in Fig. 2.3(a), a 3D Conv kernel is decomposed into m parallel streams, each having n pseudo 2D kernels and a corresponding orthogonal pseudo 1D kernel. But, the typical decompositions they exploited are $\{m = 1, 2; n = 1, 2\}$, which may not make the best out of all geometric properties of 3D data. Further, such SRAC only governs the specific layer-wise computation but neglects the outer branch/network level which controls spatial resolution changes. Most importantly, feature maps are added together periodically after each SRAC, which causes homogeneous feature maps in deeper layers and that parallel streams do not benefit richer feature extraction anymore. To address these issues, our method aims to fully exploit all the three orthogonal views and encourage extracting heterogeneous features from different scales. For this goal, we need to carefully design both the layer-level and branch-level operations.

Notation. We denote a 3D Conv layer as $\text{Conv}(\mathcal{K}_{k_1,k_2,k_3}/\mathcal{S}_{s_1,s_2,s_3})$, where k_i and s_i are the kernel size and stride step size in each direction. Conventionally, $k_1 = k_2 = k_3$ and $s_1 = s_2 = s_3$. A 3D kernel $\mathcal{K}_{3,3,3}$ can be decomposed into an

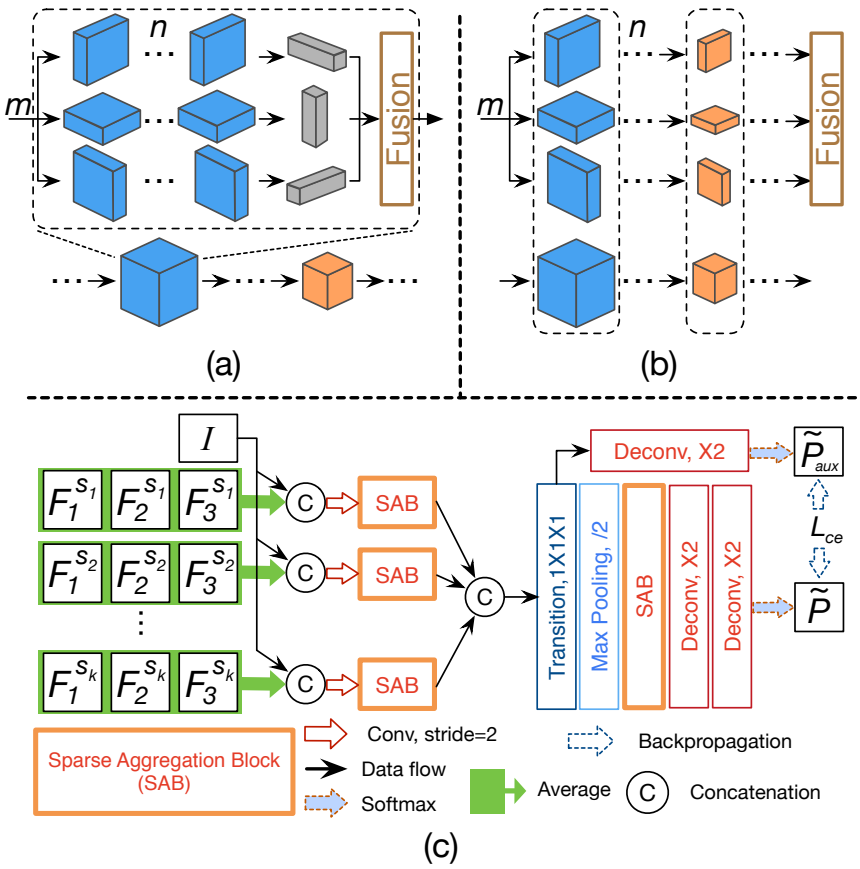

Figure 2.3. (a) Short-Range Asymmetric Cell; (b) Long-Range Asymmetric Branch; (c) Content-Aware Fusion Module. I: raw image; $F_i^{s_j}$: feature maps (see Sect. 2.2.2).

SRAC (with $m = 1$ and $n = 1$) by $\mathcal{K}_{3,3,1} \otimes \mathcal{K}_{1,1,3}$, $\mathcal{K}_{3,1,3} \otimes \mathcal{K}_{1,3,1}$, or $\mathcal{K}_{1,3,3} \otimes \mathcal{K}_{3,1,1}$, where $\otimes$ is convolution. Similarly, we denote a 3D deconvolutional (DeConv) layer as DeConv($\mathcal{K}_{k_1,k_2,k_3} \times \mathcal{S}_{s_1,s_2,s_3}$). A pooling layer is denoted as $\mathcal{P}_{s_1,s_2,s_3}$.

Fig. 2.3(b) shows the concept of our *long-range asymmetrical branch* (LRAB). We utilize three LRABs ($m = 3$) to operate on three orthogonal geometric views separately, thus increasing the independency among m parallel encoding paths. The original symmetrical Conv($\mathcal{K}_{k_a,k_a,k_a}/\mathcal{S}_{s_a,s_a,s_a}$) is replaced by an asymmetrical counterpart in each branch (i.e., ($\mathcal{K}_{k_a,k_a,1}/\mathcal{S}_{s_a,s_a,1}$), ($\mathcal{K}_{k_a,1,k_a}/\mathcal{S}_{s_a,1,s_a}$), or ($\mathcal{K}_{1,k_a,k_a}/\mathcal{S}_{1,s_a,s_a}$)). Also, the consecutive 3D Conv kernel ($\mathcal{K}_{k_b,k_b,k_b}/\mathcal{S}_{s_b,s_b,s_b}$) is decomposed in the same

orientation in each branch. Besides, since in each LRAB, Conv kernels are along the same orientation, conventional symmetrical pooling is no longer suitable (otherwise, inter-slice features may vanish quickly before being extracted). In our problem, cardiovascular segmentation is highly challenging especially due to the missing/ambiguous boundaries between the regions of interest and background or among various sub-structures. Thus, asymmetrical pooling (i.e., $\mathcal{P}_{s,s,1}$, $\mathcal{P}_{s,1,s}$, or $\mathcal{P}_{1,s,s}$) is utilized to maintain spatial resolution in the orthogonal direction so that there is a bigger chance that subtle distinguishable features can be kept in at least one of the geometric views.

For example, a $T \times T \times T$ tensor after three $\mathcal{P}_{2,2,2}$ becomes a $\frac{T}{8} \times \frac{T}{8} \times \frac{T}{8}$ tensor but becomes $\frac{T}{8} \times \frac{T}{8} \times T$ after three $\mathcal{P}_{2,2,1}$. Hence, additional information of subtle structures along the z-axis is kept and will be utilized by subsequent processing. Observe that the designs in [90, 96] can be viewed as special cases of our LRAB since these methods only used (pre-trained) 2D FCN to extract deep feature maps from 3D data slice by slice independently with $m = 1$. Thus, our method is more cautious in heterogeneous feature aggregation. Specifically, as shown in Fig. 2.2, our first LRAB is composed of stacking layers of Conv($\mathcal{K}_{3,3,1}/\mathcal{S}_{2,2,1}$), SAB($\mathcal{K}_{3,3,1}/\mathcal{S}_{1,1,1}$), $\mathcal{P}_{2,2,1}$, and SAB($\mathcal{K}_{3,3,1}/\mathcal{S}_{1,1,1}$), where SAB($\mathcal{K}_{3,3,1}/\mathcal{S}_{1,1,1}$) refers to sparse aggregation block (SAB) composed of stacked Conv($\mathcal{K}_{3,3,1}/\mathcal{S}_{1,1,1}$). We will present SAB in Sect. 2.2.3. In the i^{th} LRAB, feature maps from different scales ($s_j, j = 1, 2, \ldots, k$) are recovered by asymmetrical DeConv layers accordingly, denoted by $F_i^{s_j}$. We will discuss how to aggregate useful information from these heterogeneous feature maps in Sect. 2.2.2.

2.2.2 Content-Aware Fusion Module (CAFM)

To maximally exploit the extracted heterogeneous features maps $F_i^{s_j}$ from parallel LRABs, we need to selectively leverage the correct information and suppress the incorrect one. It is quite possible that each voxel is correctly classified in at least one

geometric view; thus, a key challenge is how to deal with agreement and disagreement in different views. For this, we present a content-aware fusion module (CAFM, see Fig. 2.3(c)) to generate aggregated deep features.

The input of CAFM includes two parts: a raw image I and heterogeneous feature maps $F_i^{S_j}$ of the same shape, where i is for the i^{th} LRAB and S_j is for the selected scales in LRABs. HFA-Net has $m = 3$ LRABs; thus $i \in \{1, 2, 3\}$. There are three scales in each LRAB and we choose the last two scales; thus $j \in \{2, 3\}$. To recover the asymmetrical feature maps to the original resolution of the input image I, we use asymmetrical DeConv accordingly (e.g., we use stacked $\{\text{DeConv}(\mathcal{K}_{4,4,1} \times \mathcal{S}_{2,2,1}), \text{DeConv}(\mathcal{K}_{4,4,1} \times \mathcal{S}_{2,2,1})\}$ to obtain $F_1^{S_3}$ for the 1^{st} LRAB). Then we average the feature maps from the same scale but different branches together to obtain hierarchical features $F^{S_j} = \frac{1}{m}\sum_{i=1}^{m} F_i^{S_j}$. This averaging provides a compact representation of all $F_i^{S_j}$'s while still showing the image areas where the heterogeneous features have agreement or disagreement. Next, each F^{S_j} is concatenated with the raw image I and fed to an encoder SAB, and all the intermediate feature maps are integrated in the middle of CAFM for extracting better representations. The raw image I provides a reference for helping further find detailed features and guide the feature aggregation process.

The loss function is computed as:

$$\ell(X, Y; \theta) = \ell_{mse}(\widetilde{P}, Y) + \lambda_1 \ell_{mse}(\widetilde{P}_{aux}, Y) + \sum_i \sum_j \lambda_{ij} \ell_{mse}(S(F_i^{S_j}), Y), \qquad (2.1)$$

where Y is the corresponding ground truth of each training sample X, ℓ_{mse} is the multi-class cross-entropy loss and $S(\cdot)$ is the softmax function. Detailed structure of HFA-Net is shown in Fig. 2.4.

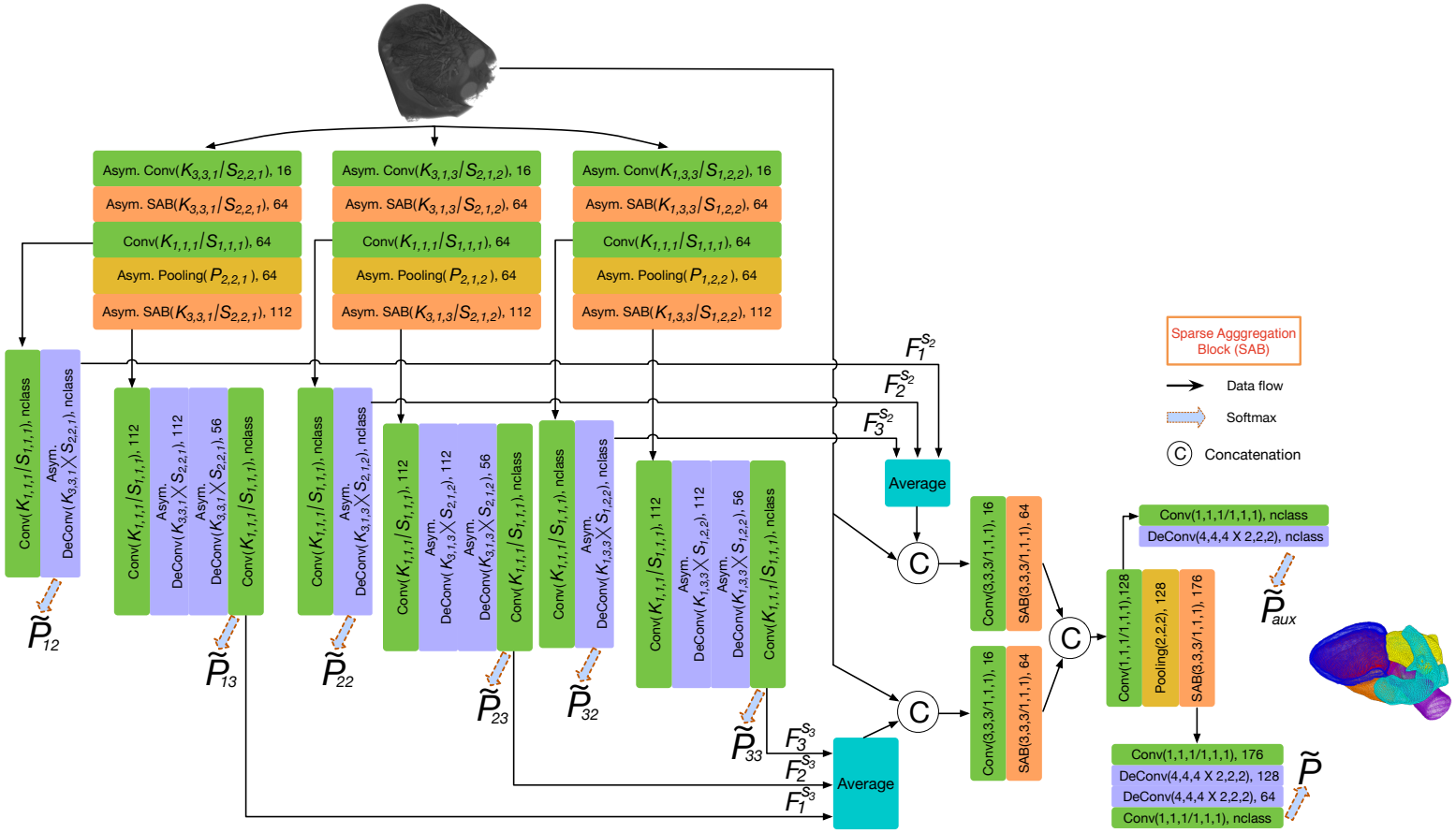

Figure 2.4. Detailed structure of HFA-Net. $F_i^{s_j}$: feature maps from the j^{th} scale in the i^{th} long-range asymmetrical branch (LRAB) (see Sect. 2.2.1). $\widetilde{P}_{ij}$: prediction from the j^{th} scale in the i^{th} LRAB; $\widetilde{P}_{aux}$ is auxiliary prediction in the content-aware module (CAFM) (see Sect. 2.2.2); $\widetilde{P}$ is our main prediction in CAFM. Finally, Eq. 2.1 is computed to train the model.

2.2.3 Sparse Aggregation Block (SAB)

DenseVoxNet [168] is a state-of-the-art model for cardiovascular image segmentation, built on DenseBlock with dense residual connections. It aggregates all the previously computed features to each subsequent layer, computed as:

$$x_\ell = H_\ell([x_0, x_1, \ldots, x_{\ell-1}]), \tag{2.2}$$

where x_0 is the input, x_ℓ is the output of layer ℓ, $[\cdot]$ is the concatenation operation, and $H_\ell(\cdot)$ is a composite of operations such as Conv, Pooling, BN, and ReLU. The dense connections help transfer useful features from shallower to deeper layers, and in turn, allow each shallow layer to receive direct supervision signal, thus alleviating the gradient vanishment issue in training deep ConvNets and achieving better parameter-performance efficiency.

However, for a DenseBlock of depth N, the number of skip connections and parameters grows quadratically asymptotically (i.e., $O(N^2)$). This means that each layer generates only a few new outputs to an ever-widening concatenation of previously seen feature representations. Thus, it is hard for the model to make full use of all the parameters and dense skip connections [192].

To further ease the training of our HFA-Net, we devise a new sparsified densely-connected convolutional block, called sparse aggregation block (SAB), to improve parameter-performance efficiency. The output x_ℓ of layer ℓ is computed as

$$x_\ell = H_\ell([x_{\ell-c^0}, x_{\ell-c^1}, x_{\ell-c^2}, x_{\ell-c^3}, \ldots, x_{\ell-c^k}]), \tag{2.3}$$

where $c > 1$ is an integer and $k \geq 0$ is the largest integer such that $c^k \leq \ell$. For an SAB of total depth N, this sparse aggregation introduces no more than $\log_c(N)$ incoming links per layer, for a total of $O(N \log(N))$ connections and parameters. We

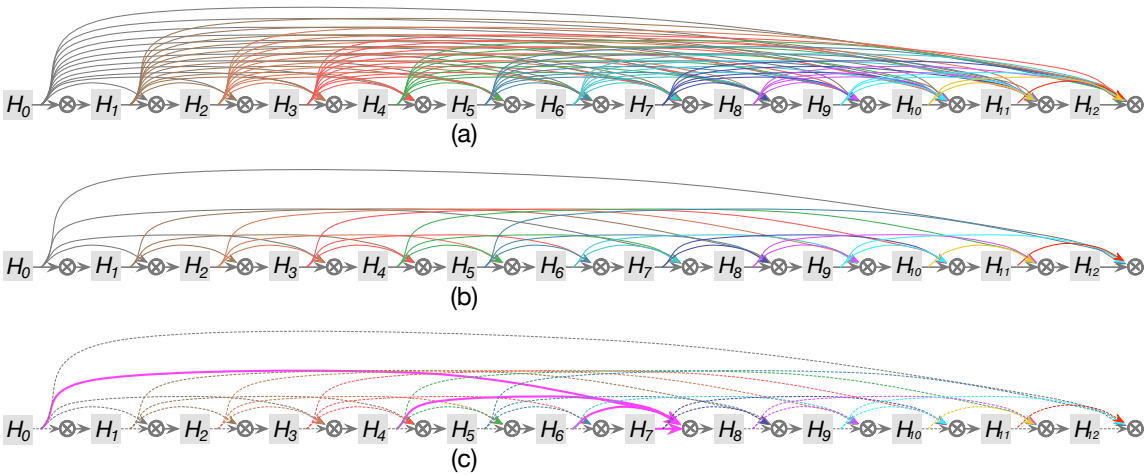

Figure 2.5. Comparison between Dense Block and our Sparse Aggregation Block (SAB), where $\otimes$ is the concatenation operation. (a) Dense Block: $x_\ell = H_\ell([x_0, x_1, \ldots, x_{\ell-1}])$, which takes all previous layers into consideration. (b) SAB: $x_\ell = H_\ell([x_{\ell-c^0}, x_{\ell-c^1}, x_{\ell-c^2}, x_{\ell-c^3}, \ldots, x_{\ell-c^k}])$, which only takes selected previous layers into consideration. (c) For example, the input of H_8 is concatenation of feature maps x_7, x_6, x_4, and x_0.

TABLE 2.1

DATASETS AND TRAINING DETAILS[1]

Dataset	Train		Test		# Class	Optimizer	# Iter.	Learning rate policy
	# stack	GT	# stack	GT				
2016 HVSMR [116]	10	✓	10	✗	2	Adam: $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon =$ 1e-10	45,000	$L_r \times \left(1 - \frac{iter}{\#iter}\right)^{0.9}$
2017 MM-WHS CT [193]	16	✓	4	✓	7		60,000	
2017 AAPM CT [161]	36	✓	12	✗	5		60,000	

use $c = 2$ and $N = 12$ in all experiments. Fig. 2.5 shows more details of SAB.

[1] "GT = ✗": the ground truth of the data is kept by the organizers for fair comparison. The initial learning rate $L_r = 5 \times 10^{-4}$.

2.3 Experiments and Results

Three 3D Datasets. (1) The *2016 HVSMR dataset* [116] aims to segment myocardium and great vessels (blood pool) in cardiovascular MRIs. The results are evaluated using three criteria: Dice coefficient, average surface distance (ADB), and symmetric Hausdorff distance. A score $S = \sum_{class}(\frac{1}{2}Dice - \frac{1}{4}ADB - \frac{1}{30}Hausdorff)$ is used to measure the overall accuracy of the results and for ranking. (2) The *2017 MM-WHS CT dataset* [193] aims to segment seven cardiac structures (the left/right ventricle blood cavity (LV/RV), left/right atrium blood cavity (LA/RA), myocardium of the left ventricle (LV-myo), ascending aorta (AO), and pulmonary artery (PA)). Following the setting in [38], we randomly split the dataset into the training (16 subjects) and testing (4 subjects) sets, which are fixed throughout all experiments. (3) The *2017 AAPM CT dataset* [161] aims to segment five thoracic structures (esophagus, spinal cord, left/right lung, and heart); esophagus and spinal cord are highly difficult cases.

Implementation Details. Our proposed method is implemented with Python using the TensorFlow framework and trained on an NVIDIA Tesla V100 graphics card with 32GB GPU memory. All the models are initialized using a Gaussian distribution and trained with the "poly" learning rate policy. We perform data augmentation to reduce overfitting. More details can be found in Table 2.1.

Quantitative Results. Table 2.2 shows quantitative comparison of HFA-Net against other methods from the 2016 HVSMR Challenge Leaderboard, including a conventional atlas-based method [131] and 3D FCN based methods [93, 168]. First, our re-implementation of DVN achieves the state-of-the-art performance and our S-DVN with SAB achieves competitive results while reducing the number of parameters by $\sim$60% (4.3M *vs.* 1.6M). Second, recall the two types of the known anisotropic 3D methods (see Sect. 2.1). We choose at least one typical method from each type for comparison. The method [45] is based on the short-range asymmetrical cell design,

which utilizes 3D kernel decomposition on the orthogonal planes to predict a class label for each voxel. The method [90] extracts features from the xy-plane by a 2D FCN and applies a 3D FCN to fuse inter-slice information. Our HFA-Net outperforms these methods across nearly all the metrics with a very high overall score of 0.239.

The results for the 2017 MM-WHS CT dataset are given in Table 2.3. First, our baselines (DVN and S-DVN) already achieve better results than the known state-of-the-art methods [38, 121]. Second, our HFA-Net further improves the accuracy on most the categories across nearly all the metrics, especially for subtle structures such as LV-myo and AO.

To further show that our method is robust and effective in delineating subtle structures, we experiment with HFA-Net on the 2017 AAPM CT dataset. Quantitative results in Table 2.4 show promising performance gain, especially for esophagus and spinal cord (2% gain in Dice coefficient).

TABLE 2.2

SEGMENTATION RESULTS ON THE 2016 HVSMR DATASET

Method	Myocardium			Blood Pool			Overall Score
	Dice	ADB [mm]	Hausdorff [mm]	Dice	ADB [mm]	Hausdorff [mm]	
Shahzad et al. [131]	0.747	1.099	5.091	0.885	1.553	9.408	-0.330
3D Unet [93]	0.762	0.943	5.618	0.932	0.826	7.015	-0.016
DVN [168]	0.821	0.964	7.294	0.931	0.938	9.533	-0.161
DVN (ours)	0.829	0.701	3.431	0.933	0.921	8.489	0.078
S-DVN	0.822	0.689	3.729	0.936	0.900	8.770	0.065
Gonda et al. [45]	0.793	0.783	4.002	0.934	0.853	7.043	0.087
Li et al. [90]	0.802	0.876	4.243	0.930	0.978	7.481	0.012
HFA-Net	**0.837**	**0.627**	**3.301**	**0.942**	**0.751**	**5.875**	**0.239**

TABLE 2.3

SEGMENTATION RESULTS ON THE 2017 MM-WHS CT DATASET

Model	Metrics	Structures							Mean
		LV	RV	LA	RA	LV-myo	AO	PA	
Payer et al. [121]	Dice	0.918	0.909	0.929	0.888	0.881	0.933	**0.840**	0.900
Dou et al. [38]	Dice	0.888	-	0.891	-	0.733	0.813	-	-
DVN	Dice	0.942	0.891	**0.933**	0.879	0.908	0.959	0.824	0.905
	Jacard	0.891	0.806	**0.874**	0.786	0.832	0.922	0.713	0.832
	ADB[voxel]	0.084	**0.448**	**0.199**	0.459	**0.180**	0.132	1.710	0.459
	Hausdorff[voxel]	**6.752**	39.156	71.189	101.570	**35.422**	27.810	59.982	48.840
S-DVN	Dice	0.929	0.890	0.914	**0.899**	0.895	0.956	0.828	0.902
	Jaccard	0.870	0.805	0.843	**0.817**	0.811	0.916	0.718	0.826
	ADB[voxel]	0.610	0.666	1.384	**0.307**	0.362	0.210	**1.594**	0.733
	Hausdorff[voxel]	21.214	55.473	85.726	73.757	62.053	80.511	77.181	65.131
HFA-Net	Dice	**0.946**	**0.893**	0.925	0.897	**0.910**	**0.964**	0.830	**0.909**
	Jaccard	**0.898**	**0.810**	0.861	0.816	**0.836**	**0.930**	**0.722**	**0.839**
	ADB[voxel]	**0.076**	0.562	0.210	0.334	0.225	**0.103**	1.685	**0.456**
	Hausdorff[voxel]	7.148	**33.128**	**42.173**	**22.903**	36.954	**12.075**	**37.845**	**27.461**

TABLE 2.4

SEGMENTATION RESULTS ON THE 2017 CT AAPM DATASET

Model	Metrics	Structures					Mean
		Esophagus	Spinal Cord	Lung_R	Lung_L	Heart	
DVN [93]	Dice	0.676	0.851	0.960	0.960	0.917	0.873
	ADB[mm]	2.227	0.867	**1.212**	1.295	2.418	1.604
	Hausdorff[mm]	7.748	2.298	**3.938**	4.100	6.781	4.973
HFA-Net	Dice	**0.697**	**0.874**	**0.962**	**0.964**	**0.920**	**0.883**
	ADB[mm]	**1.974**	**0.766**	1.266	**0.967**	**2.336**	**1.462**
	Hausdorff[mm]	**5.883**	**2.190**	4.149	**3.370**	**6.557**	**4.430**

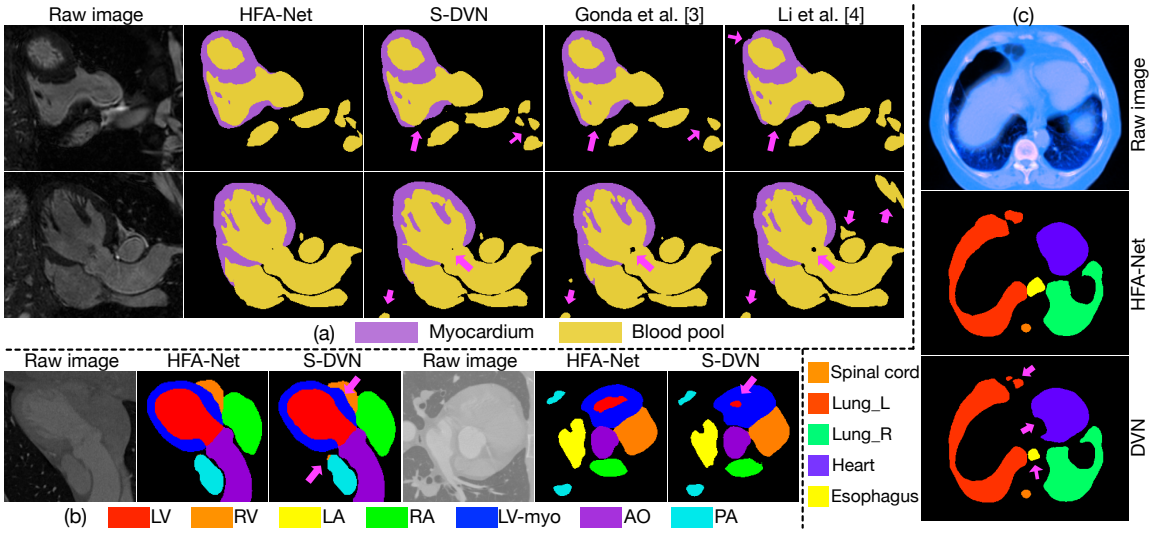

Figure 2.6. Visual qualitative results: the 2016 HVSMR dataset (a), 2017 MM-WHS CT dataset (b), and 2017 CT AAPM dataset (c) (some errors marked by magenta arrows).

Qualitative Results. As shown in Fig. 2.6, our HFA-Net attains better results and shows a strong capability of delineating missing/ambiguous boundaries.

2.4 Conclusions

In this chapter, we presented a new deep *heterogeneous feature aggregation network* (HFA-Net) for cardiovascular segmentation in 3D CT/MR images. Our proposed HFA-Net extracts rich heterogeneous features using long-range asymmetrical branches and aggregates diverse contextual and semantic deep features using a content-aware fusion module. Sparse aggregation block is utilized to give HFA-Net a better parameter-performance efficiency. Comprehensive experiments on three open challenge datasets demonstrated the efficacy of our new method.

CHAPTER 3

BIOMEDICAL IMAGE SEGMENTATION VIA REPRESENTATIVE ANNOTATION

A paper published in *2019 33rd AAAI Conference on Artificial Intelligence (AAAI)* [179]

3.1 Backgrounds

Image segmentation is a central task in diverse biomedical imaging applications. Recently, deep learning (DL) has been successfully applied to many image segmentation tasks and achieved state-of-the-art or even human-level performance [15, 18, 128, 159, 174]. It is well known that the amount and variety of data that DL networks use for model training drastically affect their performance. However, it is often quite difficult to acquire sufficient training data for DL based biomedical image segmentation tasks, because biomedical image annotation highly depends on expert experience and variations in biomedical data (e.g., different modalities and object types) can be large. With limited resources (e.g., money, time, and available experts), reducing annotation efforts while maintaining the best possible performance of DL models becomes a critical problem.

Currently, there are two main categories of methods for alleviating the burden of annotation. The methods in the first category aim to utilize unannotated data by leveraging weakly/semi-supervised learning methods [27, 94, 163]. Though promising, the performance of such methods is still far from that of supervised learning methods.

Accuracy in biomedical analysis is of high importance and thus performance is a big concern.

The methods in the second category aim to identify and annotate only the most valuable image areas that contribute to the final segmentation accuracy. To achieve this goal, such methods usually explore the following two properties of biomedical images. (1) Biomedical images for a certain type of applications are usually *similar* to one another (e.g., gland segmentation, heart segmentation). Thus, a great deal of redundancy may exist in biomedical image datasets. Fig. 3.1(a) and (c) show some frequent patterns in glands and heart MR images, respectively. (2) Although regions of interest (ROIs) in biomedical images may have *different* appearances, we notice that they can be roughly divided into a certain number of groups (e.g., see Fig. 3.1(b)). Hence, it is helpful to select representative samples to cover the diverse cases in order to achieve good segmentation performance.

Up to date, the most popular approaches [67, 162] designed to leverage these two properties are all based on active learning (AL). In general, AL based approaches iteratively conduct two steps: *selecting informative samples from unlabeled sets* and *querying labels from human experts*. The ability of AL on reducing annotation cost while maintaining good learning performance hinges on the fact that it can iteratively add the most diverse and influential samples from unlabeled sets for learning a better model and simultaneously update its selection strategy to help human experts reduce labeling redundant samples. However, this iterative process is usually quite time-consuming and not practical in real-world applications for several reasons. (1) It is implied that human experts should be *constantly* and readily available for labeling whenever new unlabeled samples are queried. (2) The AL process needs to be *suspended* until newly queried samples are annotated. (3) In *each* round of the AL process, the model needs to be applied to *all* unannotated images, which can take a large amount of time, especially for 3D biomedical images.

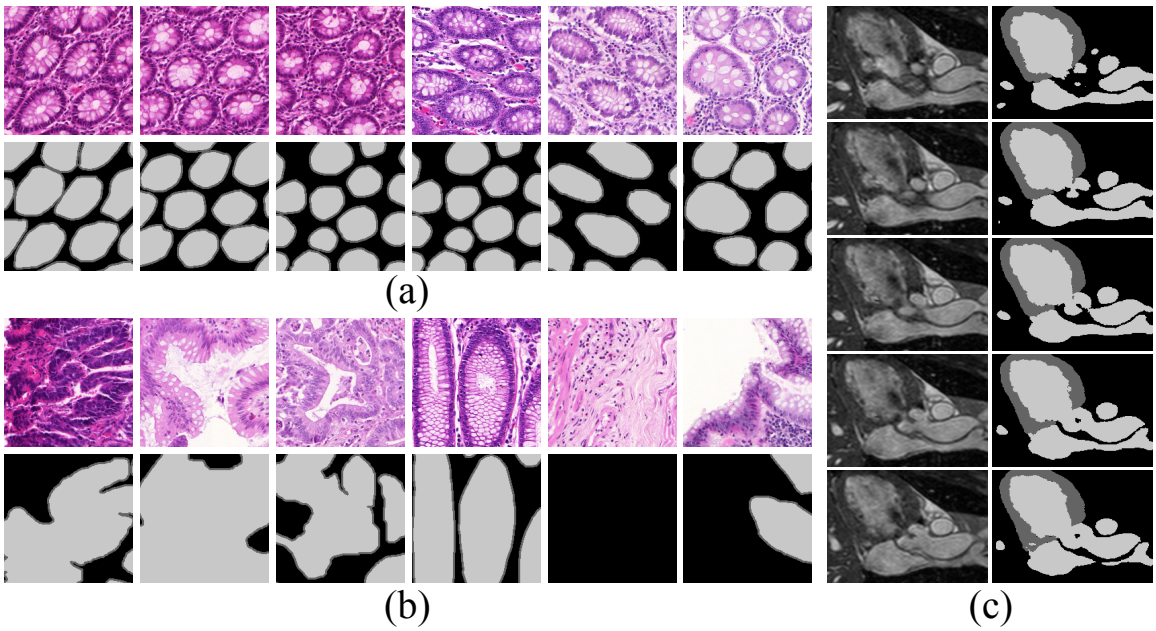

Figure 3.1. (a)-(b) Example patches showing similarity and diversity in the gland dataset. The samples in (b) are queried by the active learning (AL) based method [162]. (c) Similarity in consecutive slices of the 3D heart dataset of HVSMR 2016 (slices #80, #82, . . . , #88 in the xz plane).

To address these issues, in this chapter, we propose a new DL framework, representative annotation (RA), to directly select effective instances with *high influence and diversity* for biomedical image segmentation in **one-shot** (i.e., no iterative process and only training a DL model once). To achieve one-shot selection, we need to address two main challenges. (1) Comparing to AL, in which the model has access to manual annotation and can be trained in a supervised manner to extract informative features, the image feature extraction component in our framework has only raw image data and can only be trained in an unsupervised manner. (2) AL methods mainly rely on uncertainty estimation of unannotated images which is not used in our framework. Instead, we need to develop a new criterion for valuable ROIs.

For the first challenge, we investigate and tune various predominant unsupervised models that can be applied to extract image features: autoencoder (AE) [130], gen-

erative adversarial networks (GANs) [46], and variational autoencoder (VAE) [76]. For the second challenge, we develop an effective geometry based data selection approach that combines a clustering based method and a max-cover based method. The clustering based method divides the whole dataset into K clusters and selects the most representative samples from each cluster. To a large extent, it reduces intra-cluster redundancy, but the number of clusters, K, is usually not given. The max-cover based method forms a candidate set containing selected samples such that the coverage score for the whole dataset is maximized, which implies that both influential samples from large clusters and diverse samples from different clusters have a chance to be selected. But, the max-cover problem is NP-hard and the performance of approximation algorithms may degrade a lot when the size of the whole dataset increases. To combine the advantages of both these methods, we leverage the clustering based method to reduce intra-cluster redundancy and utilize the max-cover approach to reduce inter-cluster redundancy without sacrificing inter-cluster diversity. In this way, representative (i.e., high influential and diverse) image samples are selected. Fig. 3.2 outlines our main idea and steps. Further, our one-shot framework enables efficient annotation selection for 3D images.

We conduct extensive experiments, and the results show that our framework outperforms state-of-the-art methods.

Our new RA framework reduces annotation efforts for biomedical image segmentation while maintaining good performance. Our main contributions are as follows.

- We decouple representative selection from segmentation, and achieve "one-shot" selection, alleviating the key issue of keeping human experts standby in AL schemes.
- We introduce a clustering-based representative selection method to select representatives for human annotation.
- Our experiments demonstrate that our approach yields higher efficiency and considerably improves the results of state-of-the-art methods on two 2D datasets. Further, we show that our RA framework is effective for a 3D dataset.

3.2 Related Work

Semantic Segmentation and Network Structures. Since FCNs [99], an array of DL networks has been proposed and significantly improved performance by adapting state-of-the-art deep convolutional neural network (CNN) based image classifiers to semantic segmentation. ResNet-based approaches [52] achieve higher accuracy with substantially deeper structures [15, 128]. To further increase information flow, DenseNets [61] replace identity mapping in the residual block by concatenation operation, so that new feature learning can be reinforced while keeping old feature re-usage. The idea of dense connections has been extended to semantic segmentation [68, 90, 168]. In line with this view, CliqueNets [164] incorporate recurrent connections and attention mechanism into CNNs by allowing information flow between any pair of layers inside each block (of the same scale). In this study, we make use of most of these advanced techniques to design our 2D/3D FCNs for segmentation.

Active Learning (AL). Active learning was not incorporated with DL for image classification and segmentation to reduce annotation efforts until recently. Among various variants, different active selection schemes were proposed to iteratively query annotators to label the most informative examples from unlabeled data and re-train the model. Besides the aforementioned inherent drawbacks of AL-based methods, recent advanced approaches also had their own constraints. Jain *et al.* [67] needed a series of preprocessing to generate region proposals and descriptors which are not always easy to obtain due to large variations in biomedical images. Yang *et al.* [162] utilized the last convolutional layer of FCNs to generate image descriptors, and multiple FCNs were trained to estimate the uncertainty of segmentation results, which used considerable computational resources. Besides, using *random sampling* to initialize their data selection also makes the initialization unstable, which may considerably influence the final performance. Zhou *et al.* [188] proposed to find worthy candidates via a combination criterion of the entropy and diversity of patches based

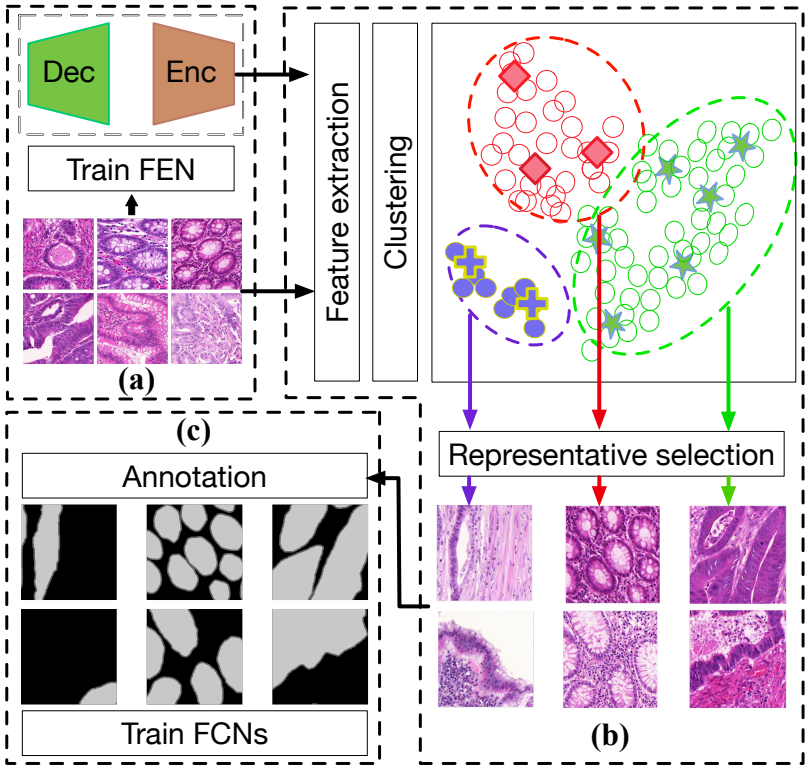

Figure 3.2. An overview of our representative annotation (RA) framework: (a) Feature extraction network (FEN) training (Enc: encoder, Dec: decoder); (b) feature extraction and clustering-based representative selection (RS); (c) annotation and fully convolutional network (FCN) training.

on the prediction of CNNs. But, it is not clear how to extend their method from image classification to segmentation. To overcome these drawbacks, we develop a new "one-shot" RA framework that consists of an unsupervised feature extraction network (FEN) and a representative selection (RS) scheme.

3.3 Representative Annotation

Our RA framework (see Fig. 3.2) has three key components: (1) an unsupervised feature extraction network (FEN) that maps each image patch to a low-dimensional feature descriptor; (2) a clustering-based algorithm for selecting representatives from

training data; (3) an FCN for segmentation.

3.3.1 Feature Extraction Networks (FENs)

Clustering methods group similar data into a cluster and can be used to reduce intra-cluster redundancy [3]. In our problem, to map input data to a clustering-friendly feature space, data representation learning is vital. Many unsupervised methods have been proposed for representation learning. We explore the predominant models (i.e., AE, GAN, and VAE) to design our FEN so that it has good ability for generalization and is fast and stable to train.

Autoencoder (AE). AE can be used to learn efficient data encoding in an unsupervised manner [130]. It consists of two networks that *encode* an input sample $\boldsymbol{x}$ to a latent representation $\boldsymbol{z}$ and *decode* the latent representation back to reconstruct the sample in the original space, as follows:

$$\boldsymbol{z} \sim Enc(\boldsymbol{x}) = q_\phi(\boldsymbol{z}|\boldsymbol{x}),\ \tilde{\boldsymbol{x}} \sim Dec(\boldsymbol{z}) = p_\theta(\boldsymbol{x}|\boldsymbol{z}). \tag{3.1}$$

Training an AE involves finding parameters $\{\theta, \phi\}$ that minimize the reconstruction loss, $\mathcal{L}_{AE}$, on the given dataset X; the objective is given as:

$$\theta^*, \phi^* = \arg\min_{\theta,\phi} \mathcal{L}_{AE}(X,\ (\phi \circ \theta)X). \tag{3.2}$$

Generative Adversarial Networks (GANs). GANs [46] are explicitly set up to optimize for generative tasks. A GAN consists of a generator G and a discriminator D (similar structures as a decoder and an encoder of AE, respectively). In training, the generator $G = G(\boldsymbol{z}) \sim p_g$ takes a random noise $\boldsymbol{z} \sim p_{\boldsymbol{z}}$ as input and generates an image. The discriminator D takes an image as input and outputs the probability that the image comes from real data rather than from G. Ideally, at the end of training, p_g can be shown to match p_{data} (i.e., G converges to a good estimator of p_{data}). The

objective function of the min-max game between G and D can be formulated as:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]. \tag{3.3}$$

Variational Autoencoder (VAE). Although VAE consists of an *encoder* and a *decoder* network, it is quite different from other types of AE models. It makes a strong assumption concerning the distribution of latent neurons and tries to minimize the difference between a posterior distribution and the distribution of latent neurons with the difference measured by the Kullback-Leibler divergence [76]. Typically, the latent distribution $p(\boldsymbol{z})$ is a predefined Gaussian distribution, such as $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. The VAE loss is minus the sum of the expected log likelihood (the reconstruction error) and a prior regularization term:

$$\mathcal{L}_{VAE} = -\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\Big[\log \frac{p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})}{q(\boldsymbol{z}|\boldsymbol{x})}\Big] = \mathcal{L}_{llike}^{pixel} + \mathcal{L}_{prior} \tag{3.4}$$

with

$$\mathcal{L}_{llike}^{pixel} = -\mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}[\log p(\boldsymbol{x}|\boldsymbol{z})] \tag{3.5}$$

and

$$\mathcal{L}_{prior} = D_{KL}(q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})), \tag{3.6}$$

where D_{KL} is the Kullback-Leibler divergence.

All these three models are predominant unsupervised representation learning methods and have been utilized in many applications. One common technique for evaluating the quality of these methods is to use the feature descriptors extracted by them on supervised datasets and evaluate the performance on top of these features. In our scenario, the extracted features reflect how well we capture the characteristics of image data and directly decide how representative our selected images are

with respect to the whole dataset, thus affecting the final segmentation performance. Hence, we evaluate these methods by the segmentation performance. To our best knowledge, we are the first to explore in this direction. We use all these methods as backbone for feature extractors and conduct extensive experiments to compare their potentials (denoted by AE-/GAN-/VAE-FEN below). Our VAE-FEN largely follows the structures in deep convolutional GAN (DCGAN) [125]. We re-use the *encoder* and *decoder* in the AE-/GAN-based FENs for fair comparison. Experimental results are shown in Table 3.1.

3.3.2 Representative Selection for 2D Images

Our goal is to select a representative set, S_r, from the whole input unannotated image set, S_u, as suggested samples for human annotation. We call this selection process **representative selection (RS)**. Below we will first analyze two intuitive methods, *clustering based RS* (denoted by **Cls-RS**) and *max-cover based RS* (denoted by **MC-RS**), and then explain why we propose our geometry based selection approach (denoted by **ClsMC-RS**) that combines the benefits of Cls-RS and MC-RS and addresses their drawbacks.

Cls-RS is a straightforward strategy that utilizes clustering to reduce intra-cluster redundancy. It first conducts clustering of the input images and then selects one representative image from each cluster to form S_r. A main drawback of this method is that we may need to know the number of clusters, K, beforehand, which is usually unavailable. K directly decides how many images to annotate; thus we should not choose K arbitrarily. As a result, we may run the risk of over-clustering or under-clustering, and need to deal with unbalanced data. For example, in the gland dataset, normal glands are the majority, and are mainly of a roughly round shape and similar to one another; but, abnormal glands are quite different. Even if we use a large number of clusters, normal glands are still in one cluster while different abnormal

glands are distinctly separated. Consequently, in the final candidate set, normal glands become a minority.

MC-RS is another intuitive strategy, inspired by suggestive annotation (SA) [162]. Each image in S_u has a representativeness score, and SA aims to find a subset $S_r \subseteq S_u$ such that, for a given budget $|S_r| \leqslant B$, the total coverage score $|F(S_r, S_u)|$ is maximized. The active learning based SA [162] uses uncertainty estimation to select a subset $S_a \subset S_u$ as an intermediate step. In our scenario, since we decouple the feature extraction process from the supervised FCN model, no such uncertainty estimation could be used. Thus, SA degenerates to MC-RS: Each time, among all the unannotated images of S_u, we select the most representative *one* to add to S_r such that the coverage score is maximized over the whole set S_u. One advantage of this one-by-one selection is that it inherently gives an order list of all unannotated images in which better representative images have higher priorities for manual annotation. But, MC-RS has two obvious disadvantages. First, the maximum set cover problem is NP-hard and cannot be approximated within $1 - \frac{1}{e} \approx 0.632$ under standard assumptions [57]. Our experiments show that, without using uncertainty measures, the performance of the greedy max-cover algorithm is largely jeopardized. Second, MC-RS is applied to the whole dataset at once; so it still runs the risk of selecting redundant images from certain groups of large sizes due to unbalanced image patterns.

Hence, based on the above observations and analysis, we propose our two-stage **ClsMC-RS** that combines clustering based and max-cover based methods. In the first stage, we first conduct agglomerative clustering and use the resulted dendrogram to determine a proper number of clusters, K. Second, we apply the greedy max-cover strategy to select a certain number of images from each cluster to form a temporal candidate set, S_c. In this way, (1) we need not know K beforehand (K directly decides the final S_r), (2) the whole dataset is divided into multiple clusters of smaller sizes, and max-cover selection works better on smaller sets so that it reduces intra-cluster

Algorithm 1: The Representative Selection Algorithm

Input: $C = \{C_i | i = 1, \ldots, M\}, C_i = \{I_{ij} | j = 1, \ldots, N_i\}, \delta, r, S_c = \emptyset, S_r = \emptyset$;

for C_i *in* C **do**
  $S_{i1} = \emptyset$, $S_{i2} = C_i$;
  while $|F(S_{i1}, C_i)| < \delta \cdot |C_i|$ **do**
    $s^* = \arg\max_{s \in S_{i2}} (F(S_{i1} \cup \{s\}, C_i) - F(S_{i1}, C_i))$;
    $S_{i1} = S_{i1} \cup \{s^*\}$, $S_{i2} = S_{i2} \setminus \{s^*\}$;
  $S_c = S_c \cup S_{i1}$;
$S_a = \emptyset, S'_c = S_c, Num_c = |S_c|$;
for $i = 1, \ldots, Num_c$ **do**
  $s^* = \arg\max_{s \in S'_c} (F(S_a \cup \{s\}, S_c) - F(S_a, S_c))$;
  $S_a = S_a \cup \{s^*\}$, $S'_c = S'_c \setminus \{s^*\}$;
  $L[i][1] = s^*$;
  $L[i][2] = PixelRatio(S_a)$;
for $i = 1, \ldots, Num_c$ **do**
  if $L[i][2] < r \leq L[i+1][2]$ **then**
    $S_r = S_r \cup L[i][1]$;
return S_r

redundancy while maintaining inter-cluster diversity, and (3) we maintain a balance among different clusters, so that scarce samples from small-size clusters would not be neglected in the greedy selection. In the second stage, we apply max-cover selection on S_c. We select a most representative image from S_c one by one to form the final S_r (S_r essentially forms an order list). Consequently, (a) since $|S_c| < |S_u|$, the max-cover algorithm works on a smaller set; (b) many images share similar patterns (e.g., nearly round shape glands are common) but could still be divided into several clusters, and this stage helps further reduce inter-cluster redundancy; (c) since considerable intra-cluster redundancy is reduced in the first stage, the data unbalanced issue is alleviated for the second stage.

Our ClsMC-RS: Clustering + Max-cover. After training FEN, we can make use of it by feeding an image patch I to the *encoder* model; the output feature vector, I^f, of the last fully-connected layer (fc) can be viewed as a high-level representation of I. In Algorithm 1, we can measure the similarity between two images I_i and I_j as:

$$\text{sim}(I_i, I_j) = Cosine\_similarity(I_i^f, I_j^f) \tag{3.7}$$

To measure the representativeness of a set S_x of image patches for a patch I of another set S_y, we define:

$$f(S_x, I) = \max_{I_i \in S_x} \text{sim}(I_i, I) \tag{3.8}$$

It means I is represented by its most similar patch I_i in S_x.

After patch clustering, each cluster C_i ($i = 1, \ldots, M$) contains some number of image patches, $C_i = \{I_{ij} \mid j = 1, \ldots, N_i\}$. First, we choose a subset, $S_{i1} \subset C_i$, which is the most representative for C_i. To measure how representative S_{i1} is for C_i, we define the coverage score of S_{i1} for C_i as:

$$F(S_{i1}, C_i) = \sum_{I_j \in C_i} f(S_{i1}, I_j) \tag{3.9}$$

When forming a candidate set S_c, it is desired that its overall coverage score approximates a fraction δ of each cluster, i.e., $S_{i1} \subset C_i$, $S_{i1} \subset S_c$, and $|F(S_{i1}, C_i)| \approx \delta \cdot |C_i|$, where δ controls the size of S_c and the reduced redundancy in the clusters. Empirically, δ is above the "elbow" point in the coverage score curve (i.e., the coverage score increases fast at the beginning and is much flatter at the end).

Having obtained the candidate set S_c, we find a subset $S_r \subseteq S_c = S_c'$ that has the highest coverage score. Iteratively, we choose one image patch from S_c' and put it in S_r:

$$I^* = \arg\max_{I \in S_c'} (F(S_r \cup \{I\}, S_c) - F(S_r, S_c)) \tag{3.10}$$

The selection of the patches I^* essentially sorts the patches in S_c based on their representativeness. With more patches selected, the pixel ratio for annotation increases monotonically. We use an array L to record the order of the selected patches for annotation and the corresponding pixel ratio.

Finally, experts can label image patches according to the order of L, until a certain pixel ratio r is reached. In our comparative experiments of RA, $r = 30\%$ or 50%.

3.3.3 Representative Selection for 3D Images

Compared to 2D image annotation, annotating 3D images is more challenging, partially due to a exponential increase in data volume. Yet, neighboring 2D slices in 3D biomedical image stacks are often quite similar (e.g., see Fig. 3.1(c)); thus one can potentially exploit this to reduce annotation efforts. Intuitively, there are two kinds of selection methods for 3D images: *sub-volume based* selection and *slice based* selection. The former method directly extends our 2D patch-based selection method to 3D datasets. However, this is impractical due to two issues: (1) 3D FEN is very costly, thus making the size of sub-volumes selected quite small [154]; (2) human can only label 2D images well. Even if a sub-volume is selected, experts would have to choose a certain plane (e.g., xy, xz, or yz plane) and label a set of consecutive 2D slices (possibly similar to their neighbors). The latter method, proposed in [28], trains a *sparse 3D FCN model* with some annotated 2D slices. But, a key issue to this method is *where* to annotate. Besides the redundancy among consecutive slices, we also observe that some neighboring slices can vary a lot. Our RA can address these issues. Hence, we propose to directly extend our RA framework to 3D datasets and select some 2D slices from each orthogonal plane for manual annotation.

Specifically, a 3D image can be analyzed from three orthogonal directions. By splitting each volume along the xy, xz, and yz directions, we obtain three sets of 2D slices. We train three FENs simultaneously on these three sets of 2D slices. For example, given an annotation ratio, r_a, our budget of annotating slices in the z-axis is $k = \lfloor D/r_a \rfloor$, where D is the number of voxels along the z-axis. We can use our 2D RA approach to select the top k representative slices along the z-axis. After obtaining annotation from human experts, we then train a sparse 3D FCN for segmentation.

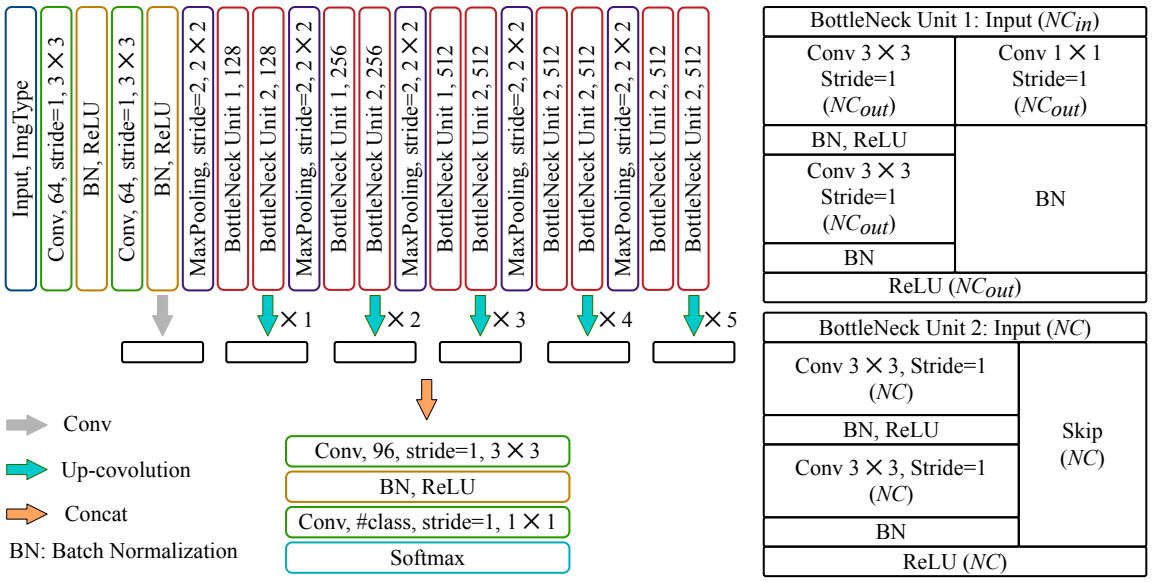

Figure 3.3. The 2D FCN architecture. "ImageType = 1" if the input image is a gray-scale image; "ImageType = 3" if the input image is an RGB image. In the bottleneck structure, if the number of channels of the input (NC_{in}) is equal to the number of channels of the output (NC_{out}), then *BottleNeck Unit 2* is used; otherwise, *BottleNeck Unit 1* is applied.

3.3.4 FCN Models for Supervised Segmentation

2D FCN Model. Since 2D FCNs for biomedical image segmentation are well studied, we focus on developing our RA framework for annotation in this chapter. To validate the effectiveness of our framework, we adopt the FCN network architecture as in SA [162] for fair comparison (detailed network architecture is shown in Fig. 3.3). Our baseline performance using full annotation matches the corresponding performance given in SA (see Table 3.1).

3D FCN Model. 3D FCN structure design is more challenging, due to the limits of computing resources that are still not well addressed. Inspired by recent advances on network architectures, clique block was proposed in CliqueNet [164]. We propose a new 3D FCN model, **CliqueVoxNet**, for segmentation. First, it uses the standard encoding-decoding FCN diagram to fully incorporate 3D image cues and geometric

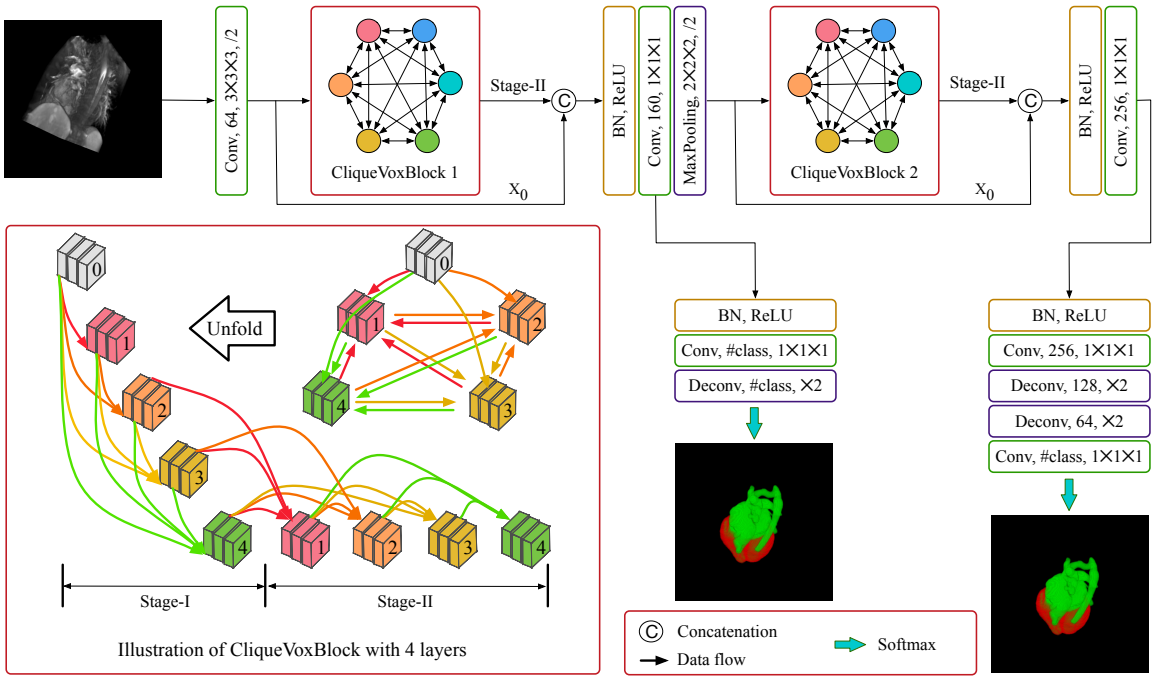

Figure 3.4. The architecture of our CliqueVoxNet. It consists of two CliqueVoxBlocks. The input layer and the Stage-II feature are concatenated to form the final block feature. The block feature passes through transition layers (including a convolution and an max-pooling) and becomes the input of the next block. Left-bottom: An illustration of a CliqueVoxBlock with 4 layers. Any layer is both the input and output of another one. Node 0 is the input layer of this block.

cues for effective volume-to-volume prediction. Second, it utilizes the state-of-the-art clique block to improve information flow and parameter efficiency, and maintain abundant (both low- and high-level) features for segmenting complicated biomedical structures. Third, it takes advantage of auxiliary side paths for deep supervision [37] to improve the gradient flow within the network and stabilize the learning process. Detailed network architecture is shown in Fig. 3.4

Given sparsely labeled 3D volume, the parameters W of the 3D FCN model are

optimized by minimizing the following total loss function:

$$\mathcal{L}(\mathcal{X}; W) = \sum_{x \in \mathcal{X}} (\psi(x, l(x)) \cdot \Delta(x)), \tag{3.11}$$

where $\psi(x, l(x))$ denotes the cross entropy loss regarding the true label $l(x)$ for pixel x in the image space $\mathcal{X}$, and $\Delta(x) = 1$ if and only if x is annotated (otherwise, $\Delta(x) = 0$).

3.4 Experiments

To show the effectiveness and efficiency of our RA framework, we evaluate RA on two 2D datasets and one 3D dataset: the MICCAI 2015 Gland Segmentation Challenge (GlaS) dataset [136], a fungus dataset [174], and the HVSMR 2016 Challenge dataset [116]. For our representative selection (RS), we only need a training set to train our feature extraction network (FEN). Then we train our FCN with annotated images and evaluate its segmentation on a test set.

2D GlaS Dataset. The GlaS dataset contains 85 training images (37 benign (BN), 48 malignant (MT)) and 80 test images (33 BN and 27 MT in Part A, 4 BN and 16 MT in Part B). Each image is of size 775×522 with pixel-wise annotation. To train our FEN, we randomly crop patches of size 384×384 from the given training set and downsample into 64×64 patches, as training data for FEN. Having trained FEN, we crop patches from each training image with a 75% ratio of overlapping with neighboring patches, and form a set of 1,530 patches for representative selection. The results are evaluated with three criteria, F1 score, object Dice index, and Hausdorff distance [136].

2D Fungus Dataset. The fungus dataset has 84 fully annotated images of size 1658×1658. As in [174], we use 4 images as the training set and 80 images as the test set. We randomly crop patches of size 450×450 from the training set and

downsample into 64×64 patches to train FEN. We crop patches from each training image with a step size of 100 pixels and form a set of 784 patches for representative selection. Results are evaluated using F1 score.

3D HVSMR Dataset. The HVSMR 2016 dataset aims to segment myocardium and great vessel (blood pool) in cardiovascular MR images. 10 3D MR images and their ground truth annotation are provided as training data. The test data, containing another 10 3D MR images, are publicly available; yet their ground truth is kept secret for fair comparison. The results are evaluated using three criteria: Dice coefficient, average surface distance (ADB), and symmetric Hausdorff distance. Finally, a score S, computed as $S = \sum_{class}(\frac{1}{2}Dice - \frac{1}{4}ADB - \frac{1}{30}Hausdorff)$, is used to reflect the overall accuracy of the results and for ranking.

Implementation Details. Our FENs and 2D FCN are implemented with PyTorch [119] and Torch7 [30], respectively. An NVIDIA Tesla P100 GPU with 16GB GPU memory is used for both training and testing. The training of FENs and FCN uses similar setups as in [125] and [162], respectively. Our 3D CliqueVoxNet is implemented with TensorFlow [2]. All the models are initialized using a Gaussian distribution ($\mu = 0$, $\sigma = 0.01$) and trained with the Adam optimization [75] ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon =$ 1e-10). We also adopt the "poly" learning rate policy with the power variable equal to 0.9 and the max iteration number equal to 50k. To leverage the limited training data, we perform data augmentation (i.e., random rotation with 90, 180, and 270 degrees, as well as image flipping along the axial planes) to reduce overfitting.

3.4.1 Main Experimental Results

We first show the state-of-the-art segmentation performance on all the three datasets with full annotation, and then show the effectiveness of our representative annotation (**RA**) on two aspects: the saved human annotation and the corresponding

TABLE 3.1

SEGMENTATION RESULTS ON THE GLAS DATASET[1]

Anno.	Method	F1 Score		Object Dice		Object Hausdorff	
		Part A	Part B	Part A	Part B	Part A	Part B
Full	CUMedVision [15]	0.912	0.716	0.897	0.781	45.418	160.347
	Multichannel [159]	0.893	0.843	**0.908**	0.833	**44.129**	116.821
	SA[162]	**0.921**	**0.855**	0.904	**0.858**	44.736	**96.976**
30%	SA[162]	0.901	0.827	**0.894**	0.835	–	–
	AE-RA	0.903	0.810	0.892	0.823	48.7781	111.5563
	DCGAN-RA	0.900	0.828	0.883	0.837	56.833	117.088
	VAE-RA	**0.909**	**0.843**	0.890	**0.855**	**48.611**	**91.486**
50%	SA[162]	**0.917**	0.828	**0.906**	0.837	–	–
	AE-RA	0.911	0.831	0.899	0.826	48.170	120.234
	DCGAN-RA	0.914	0.848	0.903	0.852	**44.912**	99.093
	VAE-RA	0.916	**0.862**	0.897	**0.856**	45.859	**91.922**

segmentation performance compared with the state-of-the-art active learning based method, suggestive annotation (**SA**) [162]. Specifically, we measure annotation effort using the number of pixels selected as representatives by our representative selection (RS) method.

Table 3.1 gives the segmentation results on the GlaS dataset. First, for fairness of comparison, we use the same FCN model as that in SA and achieve comparable performance as SA with *full annotation*. One can see that it attains state-of-the-art

[1] X-RA stands for using X-based FEN and RS in our RA framework.

TABLE 3.2

SEGMENTATION RESULTS ON THE FUNGUS DATA[2]

Anno.	Method	Recall	Precision	F1 Score
Full	DAN [174]	0.9020	0.9287	0.9152
	Ours (baseline)	0.9118	0.9379	**0.9247**
30%	VAE*	0.9254	0.9211	0.9232
	VAE-RA	0.9285	0.9219	**0.9252**
50%	VAE*	0.9268	0.9220	0.9244
	VAE-RA	0.9288	0.9226	**0.9257**

performance. Second, using the same FCN structure, we train FCNs with partial annotation with different pixel ratios (30% and 50%). Table 3.1 shows that our approach (VAE-RA) achieves competitive or much better results comparing to SA. It is worth noting that, compared to SA with 50% of annotated data, our segmentation results are better than SA ($\sim$ 2.5%) on Part B (which contains more malignant samples) while retaining nearly the same performance on Part A. More importantly, our 50% VAE-RA closely approaches the performance of full SA on all the three metrics (while there are still some gaps between 50% SA and full SA).

Table 3.2 gives the segmentation results on the fungus dataset. First, our FCN can achieve slightly better performance than the state-of-the-art methods using full annotation. Second, our framework (VAE-RA) can achieve state-of-the-art performance using only 30% of the training data, which implies that the fungus dataset is probably less challenging than the gland dataset. Indeed, the fungus dataset con-

[2]VAE* = VAE-FEN + Cls-RS; VAE-RA = VAE-FEN + ClsMC-RS.

tains fewer variations, and its F1 scores on average are higher than those of the GlaS dataset.

Table 3.3 gives the segmentation results on the 3D heart dataset. First, compared to the state-of-the-art DenseVoxNet, our CliqueVoxNet achieves considerable improvement on all the metrics. Then, we implement sparse 3D FCN models based on CliqueVoxNet. We use *uniform annotation* (UA) as baseline. Let s_k denote the setting of labeling one slice out of every k slices (i.e., the annotation ratio is $\sim 1/k$). In this dataset, a heart almost occupies the entire stack (see Fig. 3.1(c)); thus UA is a fairly strong baseline. From Table 3.3, one can see: (1) With a lower annotation ratio, the overall segmentation performance decreases accordingly (the lower, the faster); (2) the results are not very stable. For example, s_{10} of UA is slightly better than s_2. The reason is that UA cannot ensure that all the slices selected in the setting s_{10} also belong to s_2 (due to the $\lfloor \cdot \rfloor$ operation for computing slice indices). On the contrary, our RA does not suffer this issue, because inherently it gives an order of slices for annotation and the slices annotated in s_j always belong to s_i ($i < j$). As shown in Table 3.3, overall, our RA achieves much better performance than UA on the same sampling ratios.

In summary, the segmentation results on all the three datasets demonstrate the effectiveness of our representative annotation framework (X-FEN + ClsMC-RS), which achieves state-of-the-art segmentation performance and saves annotation efforts considerably.

TABLE 3.3

SEGMENTATION RESULTS ON THE HVSMR 2016 DATASET USING UNIFORM ANNOTATION AND REPRESENTATIVE ANNOTATION

Model	Sample Rate	Myocardium			Blood Pool			Overall Score
		Dice	ADB[mm]	Hausdorff[mm]	Dice	ADB[mm]	Hausdorff[mm]	
DenseVoxNet	Full	0.821	0.964	7.294	0.931	0.938	9.533	-0.161
CliqueVoxNet		0.827	0.924	6.679	0.935	0.797	5.032	0.06
Sparse-CliqueVoxNet + Uniform Annotation (UA)	s_2	0.792	0.877	5.050	0.926	0.946	7.601	-0.019
	s_{10}	0.814	0.826	4.608	0.931	0.961	7.997	0.005
	s_{20}	0.791	0.988	6.470	0.934	0.900	6.437	-0.04
	s_{40}	0.780	1.334	11.365	0.930	0.942	8.435	-0.374
	s_{80}	0.739	1.472	10.227	0.917	1.082	8.932	-0.449
Sparse-CliqueVoxNet + Representative Annotation (RA)	s_2	0.806	0.928	5.710	0.930	0.871	6.276	0.019
	s_{10}	0.812	0.895	5.820	0.928	0.896	6.360	0.016
	s_{20}	0.809	0.984	6.874	0.924	0.933	6.470	-0.057
	s_{40}	0.786	0.908	4.711	0.916	1.057	8.365	-0.076
	s_{80}	0.733	1.250	7.447	0.923	1.010	8.715	-0.276

3.4.2 Discussions

On FEN Structures. As shown in Table 3.1, using features extracted by VAE-based FEN is more beneficial for the subsequent representation selection, leading to better segmentation results. We think the reasons are: (1) Compared with AE, VAE is a generative model that was originally designed to learn the underlying data distribution and generate new data, while AE learns how to compress data into a condensed vector with only reconstruction loss; (2) compared with GAN, the output of the *encoder* in VAE is used to generate a new vector for the *decoder* to generate a new image, while the output of the *discriminator* in GAN is fed to a classifier to differentiate real and fake data. Thus more information could be kept in VAE-extracted features.

On RS Strategies. As shown in Table 3.4, our ClsMC-RS is better than the other two baselines. First, clustering of image patches reduces intra-cluster redundancy. Inside each cluster, we select abundant representatives and the number of patches is controlled by the coverage score (i.e., $\delta \cdot |C_i|$) rather than the size of the cluster. Thus, much redundancy is eliminated. Second, the "max-cover selection" incrementally chooses the most representative patches, one by one, which further reduces inter-cluster redundancy without sacrificing inter-cluster diversity. Hence, the final representative set for annotation is both influential and diverse. Besides, our ClsMC-RS has two more benefits. (1) Inherently, in the second step, our ClsMC-RS outputs an ordered list, thus enabling experts to label "better" samples incrementally. (2) After the first step, the size of the candidate set S_c is largely reduced compared to the whole input set S_u (i.e., $|S_c| < |S_u|$), which could help save more time in the second step.

On Time Efficiency. Compared with the state-of-the-art suggestive annotation (SA) [162], our RA has better time efficiency. Suppose we need to make annotation suggestion for 50% of data. The iterative SA training takes 16 rounds, but our

TABLE 3.4

SEGMENTATION RESULTS ON THE GLAS DATASET USING DIFFERENT SELECTION SCHEMES.

Anno.	Method	F1 Score		Object Dice		Object Hausdorff	
		Part A	Part B	Part A	Part B	Part A	Part B
30%	SA	0.901	0.827	**0.894**	0.835	–	–
	Cls-RS	0.908	0.838	**0.894**	0.846	50.207	101.547
	MC-RS	0.906	0.833	0.891	0.834	49.773	106.990
	ClsMC-RS	**0.909**	**0.843**	0.890	**0.855**	**48.611**	**91.486**
50%	SA	**0.917**	0.828	**0.906**	0.837	–	–
	Cls-RS	0.912	0.855	0.893	0.852	47.565	96.644
	MC-RS	0.912	0.850	0.900	0.848	**45.628**	100.706
	ClsMC-RS	0.916	**0.862**	0.897	**0.856**	45.859	**91.922**

training finishes in one-shot. Each SA round takes $\sim$ 10 minutes to train FCNs; between every two rounds, experts annotate more data based on SA suggestion. More importantly, if we directly apply SA to 3D datasets, the waiting time between two consecutive rounds would increase dramatically. With our method, experts do not start annotation until FEN and RS complete, and need not wait for FCN training round after round as in SA. Thus, our training scheme is much more expert-friendly.

Cluster Visualization. Fig. 3.5(a) shows the distribution of 2D points for the image patches of the GlaS dataset produced by t-SNE in the latent space. Using the dendrogram of agglomerative clustering, we cluster the set of FEN-extracted feature descriptors for the GlaS dataset into 10 clusters (see Fig. 3.5(a)). Fig. 3.5(b)

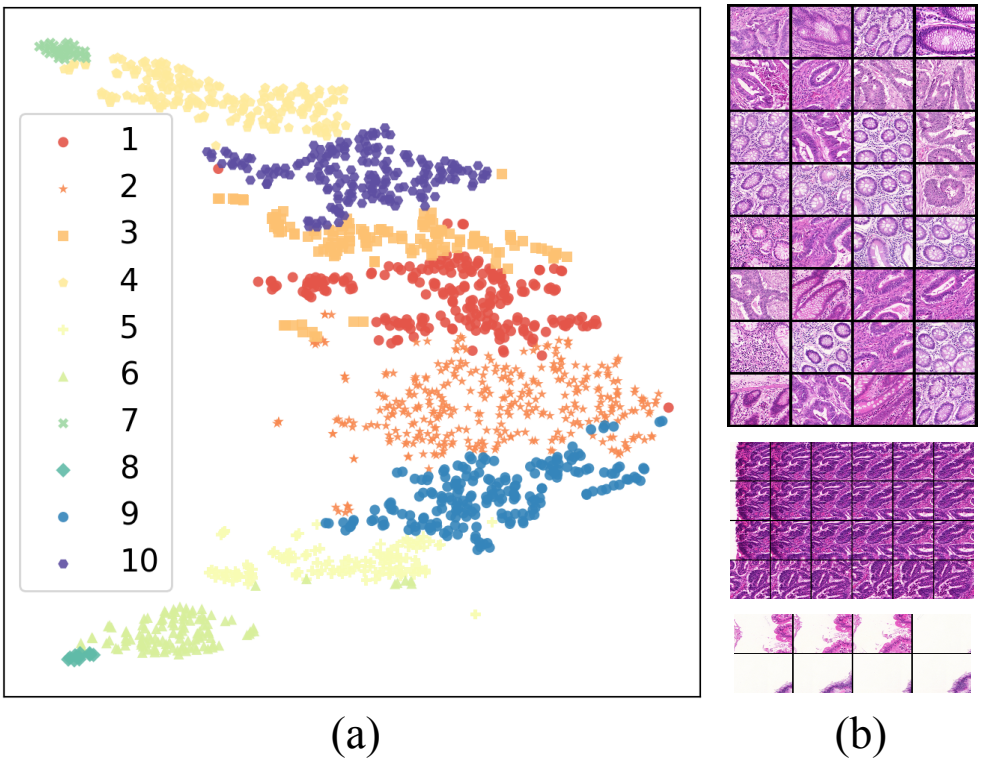

Figure 3.5. 2D t-SNE visualization of feature descriptors of the GlaS dataset: (a) FEN-generated feature descriptors; (b) the corresponding image patches in cluster-2 (top), cluster-7 (middle), and cluster-8 (bottom) for (a).

shows the corresponding patches in three typical clusters. One can see that the consistency within each cluster and the dissimilarity among different clusters are both high, indicating a desired representative capability of FEN-extracted feature descriptors.

3.5 Conclusions

In this chapter, we presented a new deep learning framework, representative annotation (RA), for reducing annotation effort in biomedical image segmentation. RA combines unsupervised feature extraction for representative selection and supervised FCNs for image segmentation. Extensive experimental results on three datasets (two

2D and one 3D) show that RA achieves competitive performance as the state-of-the-art suggestive annotation (SA) method [162] while using one-shot selection of representatives for annotation. Further, RA can be easily extended to 3D datasets and experimental results show great potentials of our method.

CHAPTER 4

A NEW ENSEMBLE LEARNING FRAMEWORK FOR 3D BIOMEDICAL IMAGE SEGMENTATION

A paper published in *2019 33rd AAAI Conference on Artificial Intelligence (AAAI)* [181]

4.1 Backgrounds

3D image segmentation plays an important role in biomedical image analysis (e.g., segmenting the whole heart to diagnose cardiac diseases [116, 168] and segmenting neuronal structures to identify cortical connectivity [84, 132]). With recent rapid advances in deep learning, many 2D [132, 152] and 3D [17, 28, 168] convolutional neural networks (CNNs) have been developed to attain state-of-the-art segmentation results on various 3D biomedical image datasets [116, 132]. However, due to the limitations of both GPU memory and computing power, when designing 2D/3D CNNs for 3D biomedical image segmentation, the trade-off between the field of view and utilization of inter-slice information in 3D images remains a major concern. For example, 3D CNNs attempt to fully utilize 3D image information but only have a limited field of view (e.g., $64 \times 64 \times 64$ [168]), while 2D CNNs can have a much larger field of view (e.g., 572×572 [128]) but are not able to fully explore inter-slice information.

Many methods have been proposed to circumvent this trade-off by carefully designing the structures of 2D and 3D CNNs. Their main ideas can be broadly classified into two categories. The models in the first category selectively choose the input data. For example, the tri-planar schemes [123, 152] use only three orthogonal planes (i.e.,

the xy, yz, and xz planes) instead of the whole 3D image, aiming to utilize inter-slice information without sacrificing the field of view. The models in the second category first summarize intra-slice information using 2D CNNs and then use the distilled information as an (extra) input to their 3D network component. For example, in [84], intra-slice information is first extracted using a 2D CNN (VD2D) and then passed to the 3D component (VD2D3D) via recursive training. In [18], its recurrent neural network (RNN) component directly uses the results of 2D CNNs as input to compute 3D segmentation.

However, these methods still have considerable drawbacks. Tri-planar schemes [123, 152] use only a small fraction of 3D image information and the computation cost is not reduced but shifted to the inference stage (a tri-planar scheme can only predict one voxel at a time, which is very slow when predicting new 3D images). For models that first summarize intra-slice information, the asymmetry nature of the network design (first 2D, then 3D) may hinder a full utilization of 3D image information (2D results may dominate since they are much easier to be interpreted than raw images in the 3D stage).

In this chapter, we explored a different perspective. Instead of designing new network structures to circumvent the trade-off between field of view and inter-slice information, we address this difficulty by developing a new *ensemble learning* framework for 3D biomedical image segmentation which aims to retain and combine the merits of 2D/3D models. Fig. 4.1 gives an overview of our framework.

Due to the heterogeneous nature of our 2D and 3D models (base-learners), we use the idea of stacking [151] (i.e., training a meta-learner to combine the results of multiple base-learners). Given a set of image samples, $X = \{x_1, x_2, \ldots, x_n\}$, their corresponding ground truth, $Y = \{y_1, y_2, \ldots, y_n\}$, and a set of base-learners, $F = \{f_1, f_2, \ldots, f_m\}$, a common design of a meta-learner is to learn the prediction of $(x_i, \hat{y}_i) = f_{meta}(f_1(x_i), f_2(x_i), \ldots, f_m(x_i))$, by fitting y_i. Since the results from the

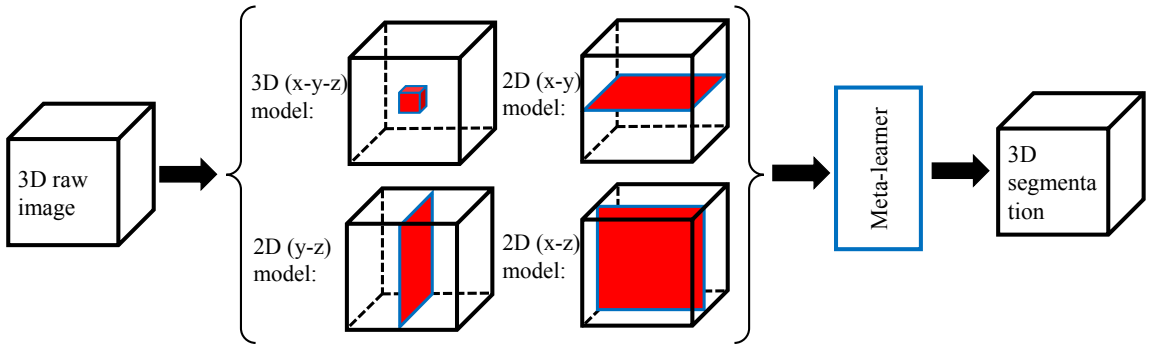

Figure 4.1. An overview of our proposed framework. Red box/planes show the effective fields of view of the corresponding 3D/2D base-learners. Our meta-learner works on top of all the base-learners.

base-learners can be very close to the ground truth, training the meta-learner by directly fitting y_i is susceptible to over-fitting. Many measures have been proposed to address this issue: (1) simple meta-learner structure designs (e.g., in [69], the meta-learner was implemented as a single 1×1 convolution layer, and in [122], the meta-learner was implemented using the XGBoost classifier [22]); (2) excluding raw image information from the input [191]; (3) splitting the training data into multiple folds and training the meta-learner by minimizing the cross-validated risk [142].

However, in our 3D biomedical image segmentation scenario, these meta-learner designs may not work well due to the following reasons. First, each of our individual base-learners (2D and 3D models) has its distinct merit; in many difficult image areas, it is quite likely that only one of the base-learners could produce the correct results. Thus, our meta-learner should be sophisticated enough in order to capture the merits of all the base-learners. Second, since extensive annotation efforts are often needed to produce full 3D annotation, not many 3D training images are available (e.g., 3 in [84] and 10 in [116]) in common 3D biomedical image datasets. Splitting the already scarce training data can largely lower the accuracy of the base-learners and meta-learner.

Hence, we propose a new stacking method that includes (1) a deep learning based meta-learner to combine and improve the results of the base-learners, and (2) a new meta-learner training method that can train a sophisticated meta-learner with less risk of over-fitting.

A Deep Learning Based Meta-Learner. Comparing with image classification, the output domain of image segmentation is much more structural. However, recent studies have not leveraged this property to design a better meta-learner for image segmentation. For example, in [69, 107], only linear combination of base-learners was explored. We develop a new fully convolutional network (FCN) based meta-learner to capture the merits of our base-learners and produce spatially consistent results.

Minimizing the Risk of Over-Fitting. A key idea of our meta-learner training method is to use the results of the base-learners as *pseudo-labels* [83] and compute ensemble by finding a balance among these pseudo-labels through the training process (in contrast to directly fitting y_i). More specifically, for each input sample x_i, there are multiple versions of pseudo-labels (from the individual base-learners). During the iterative meta-learner training process, in each iteration, we randomly choose one pseudo-label from all the versions and use it as "ground truth" to compute the loss and update meta-learner parameters. Iteration by iteration, the pseudo-labels with small disagreement would provide a more consistent supervision and the pseudo-labels with large disagreement would request the meta-learner to find a balanced solution by minimizing the overall loss.

Our method can minimize the risk of over-fitting in two aspects. (1) Intuitively, over-fitting occurs when a model over-explains the ground truth. Since our method uses multiple versions of "ground truths" (pseudo-labels), the meta-learner is unlikely to over-fit any one of them. (2) Since our meta-learner training uses only model-generated results, unlabeled data can be easily incorporated into the training process; this will allow us to further reduce over-fitting.

Compared with previous methods that combine 2D and 3D models, our main contributions are: (a) a new ensemble learning framework for tackling 3D biomedical image segmentation from a different perspective, and (b) an effective meta-learner training method for ensemble learning that minimizes the risk of over-fitting and makes use of unlabeled data. Extensive experiments on two public datasets (the HVSMR 2016 Challenge dataset [116] and the mouse piriform cortex dataset [84]) show that our framework is effective under fully-supervised, semi-supervised, and transductive settings, and attains superior performance over the state-of-the-art methods [17, 132, 168].

4.2 Method

Our proposed approach has two main components: (1) a group of 2D and 3D base-learners that are trained to explore the training data from different geometric perspectives; (2) an ensemble learning framework that uses a deep learning based meta-learner to combine the results from the base-learners. A schematic overview of our proposed framework is shown in Fig. 4.1.

In Section 4.2.1, we illustrate how to design our 2D and 3D base-learners to achieve a set of accurate and diverse results. In Section 4.2.2, we discuss our new deep learning based meta-learner that can considerably improve the results from the base-learners. In Section 4.2.3, we present our new method for training a more powerful meta-learner while preventing over-fitting.

4.2.1 2D and 3D Base-Learners

To achieve the best possible ensemble results, it is commonly desired that individual base-learners be as *accurate* and as *diverse* as possible [191]. In this section, we show how to design our 2D and 3D base-learners to satisfy these two criteria.

For Accurate Results. Our 2D model basically follows the structure of that

in [162]. We choose this structure because it is based on a well-known FCN [15] which has attained lots of successes in biomedical image segmentation and has been integrated in recent advances of deep learning network design structures, such as batch normalization [64], residual networks and bottleneck design [52]. It generalizes well and is fast to train. As for the 3D model, we use DenseVoxNet [168], for three reasons. First, it adopts a 3D FCN architecture, and thus can fully incorporate 3D image cues and geometric cues for effective volume-to-volume prediction. Second, it utilizes the state-of-the-art dense connectivity [61] to accelerate the training process, improve parameters and computational efficiency, and maintain abundant (both low- and high-complexity) features for segmenting complicated biomedical structures. Third, it takes advantage of auxiliary side paths for deep supervision [37] to improve the gradient flow within the network and stabilize the learning process. For further details of these 2D and 3D models, the readers are referred to [162] and [168].

For Diverse Results. Our key idea to achieve diverse results is to let each of our base-learners have a unique geometric view of the data. As discussed in Section 4.1, 2D models can have large fields of view in 2D slices while 3D models can better utilize 3D image information in a smaller field of view. Our mix of 2D and 3D base-learners creates the first level of diversity. To further boost diversity, within the group of 2D base-learners, we leverage the 3D image data to create multiple 2D views (representations) of the 3D images (e.g., xy, xz, and yz views). The different 2D representations of the 3D images create 2D models with diverse strengths (e.g., large fields of view for different planes) and thus generate diverse 2D results. Note that the 2D representations are not limited to being orthogonal to each of the major axes. But, based on our trial studies, we found that the results from the xy, xz, and yz views are the most accurate (probably because no interpolation is needed when extracting these 2D slices from 3D images) and already create good diversity.

Thus, in our framework, we use the following four base-learners: a 3D Den-

seVoxNet [168] for utilizing full 3D information; three 2D FCNs [162] for large fields of view in the xy, xz, and yz planes.

4.2.2 Deep Meta-Learner Structure Design

Since our base-learners have distinct model architectures and work on different geometric perspectives of 3D images to produce diverse predictions, for difficult image areas, there is a better chance that one of the base-learners would give correct predictions (see Fig. 4.3). In order to attain a meta-learner to pick up the correct predictions, we need a model that is, architecture and complexity wise, capable of learning robust visual features for jointly utilizing the diverse prediction results (from the base-learners) as well as the raw image information. It is known that simple models (e.g., linear models, shallow neural networks) are not powerful enough to learn/extract robust and comprehensive features for difficult vision tasks [191]. Furthermore, our learning task is a segmentation problem that requires spatially consistent output. Thus, we employ a state-of-the-art 3D FCN (DenseVoxNet [168]) for building our meta-learner. The input of the network is the base-learners' results and the raw image, and the output of the network is the computed ensemble. Below we describe how to construct the input of our deep meta-learner and the details of the deep meta-learner's model architecture.

Given a set of image samples, $X = \{x_1, x_2, \ldots, x_n\}$, and a set of base-learners, $F = \{f_1,\ f_2,\ \ldots, f_m\}$, a *pseudo-label set* for each x_i can be obtained as $PL_i = \{f_1(x_i), f_2(x_i), \ldots, f_m(x_i)\}$. The input of our meta-learner $\mathcal{H}$ includes two parts: x_i and $S(PL_i)$, where S is a function of PL_i that forms a representation of PL_i. There are multiple design choices for constructing S. For example, (1) concatenating all the elements of PL_i, or (2) averaging all the elements of PL_i. Concatenation allows the meta-learner to gain full information from the base-learners (no information is added or lost). Averaging provides a more compact representation of all pseudo-

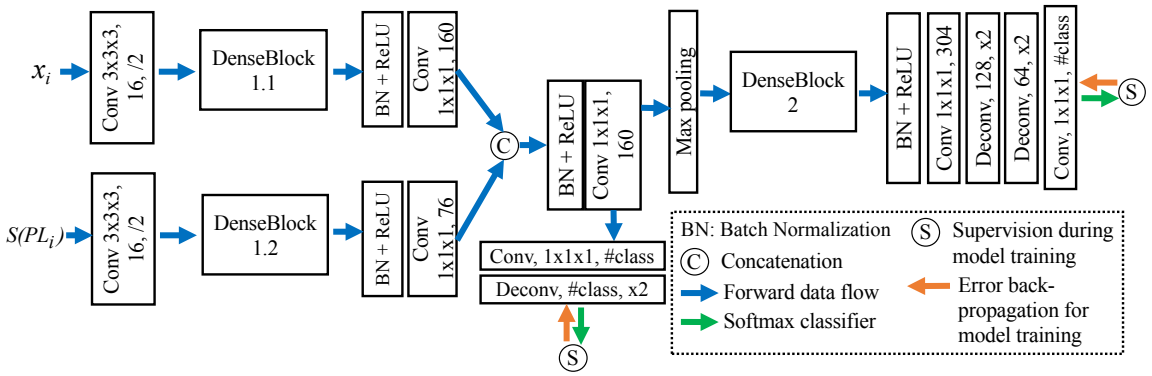

Figure 4.2. Our deep meta-learner (a variant of 3D DenseVoxNet [168]). Since $S(PL_i)$ and x_i are of different nature, we use separate encoding blocks (i.e., DenseBlock 1.1 and DenseBlock 1.2) for extracting information from $S(PL_i)$ and x_i, respectively, before the information fusion. The auxiliary loss in the side path can improve the gradient flow within the network.

labels, while still showing the image areas where the pseudo-labels hold agreement or disagreement. Furthermore, using the average of all the pseudo-labels of x_i to form part of the meta-learner's input can be viewed as a preliminary ensemble of the base-learners. We have experimented with both these design choices and found that making S an averaging function of the elements of PL_i gives slightly better results. The overall model specification of our proposed deep meta-learner is shown in Fig. 4.2.

4.2.3 Meta-Learner Training Using Pseudo-Labels

A major goal of our training procedure is to train a powerful meta-learner, while minimizing the risk of over-fitting. To achieve this goal, instead of using the ground truth to supervise the meta-learner training, we use the pseudo-labels produced by our base-learners (as discussed in Section 4.2.1) to form the supervision signals. Because there are multiple possible targets (pseudo-labels) for the meta-learner to fit, the meta-learner is unlikely to over-fit any fixed target. The base-learners can also

be applied to generate pseudo-labels for unlabeled data. Thus, our method is also capable of using unlabeled data for deep meta-learner training (which can further reduce over-fitting).

Suppose X, PL_i, and $S(PL_i)$ are given, for $i = 1, 2, \ldots, n$. Ideally, the learning objective of the meta-learner would be: (1) finding the "best" pseudo-labels in PL_i, and (2) training the meta-learner $\mathcal{H}$ to fit the pseudo-labels found in (1). However, the best pseudo-labels are not clearly defined and can be difficult to find. Based on different evaluation criteria, the "best" choices can be different. Even when a criterion is given, using the most accurate pseudo-labels can likely lead to a higher chance of suffering over-fitting. Furthermore, when training using unlabeled data, it is in general quite difficult to determine which pseudo-label gives more accurate predictions than the others. One could set up a hand-crafted algorithm based on a predefined criterion to select the "best" pseudo-labels for training. The meta-learner, however, could very likely over-fit the algorithm's choices and hence likely not be able to generalize well to future unseen image data.

Rather than explicitly defining the full learning objective for meta-learner training, we initially train the meta-learner in order to set up a near-optimal (or sub-optimal) configuration: The meta-learner is aware of all the available pseudo-labels, and its position in the hypothesis space is influenced by the raw image and the pseudo-label data distribution. Next, the meta-learner itself chooses the nearest pseudo-labels to fit (based on its current model parameters) and updates its model parameters based on its current choices. This nearest-neighbor-fit process iterates until the meta-learner fits the nearest neighbors well enough. Thus, our meta-learner training consists of two phases: (1) random-fit, and (2) nearest-neighbor-fit. We describe these two training phases below.

Random-Fit. In the first training phase (which aims to train the meta-learner $\mathcal{H}$ to reach a near-optimal solution), we seek to minimize the overall cross-entropy

Algorithm 2: Random-Fit

Input: $(x_i, PL_i = \{f_1(x_i), f_2(x_i), \ldots, \ f_m(x_i)\}, S(PL_i)), i = 1, 2, \ldots, n$;
Output: A trained meta-learner $\mathcal{H}$;
initialize a meta-learner $\mathcal{H}$ with random weights;
mini-batch = $\emptyset$;
while *stopping condition not met* **do**
  for k = *1 to batch-size* **do**
    $p = rand\text{-}int(1, n)$;
    $q = rand\text{-}int(1, m)$;
    add training sample $\{(x_p, S(PL_p)), f_q(x_p)\}$ to the mini-batch;
  update $\mathcal{H}$ using training samples in the mini-batch with forward and backward propagation;
  mini-batch = $\emptyset$;

loss for all the image samples with respect to all the pseudo-labels:

$$\ell(\theta_{\mathcal{H}}) = \sum_{i=1}^{n} \sum_{j=1}^{m} \ell_{mce}(\theta_{\mathcal{H}}(x_i, S(PL_i)), f_j(x_i)), \tag{4.1}$$

where $\theta_{\mathcal{H}}$ is the meta-learner's model parameters and ℓ_{mce} is a multi-class cross-entropy criterion. The above loss ensures that the meta-learner training process in this phase works on all the available pseudo-labels. Since the loss function itself does not impose any favor towards any particular pseudo-labels produced by the base-learners, our meta-learner is unlikely to over-fit any pseudo-labels. Exploring the overall raw image and the pseudo-label data distribution, the meta-learner obtained by minimizing the above loss may have different tendencies towards different pseudo-labels.

To effectively optimize the loss function in Eq. (4.1), we develop a random-fit algorithm. In the SGD-based optimization, for one image sample x_i, our algorithm randomly chooses a pseudo-label from PL_i and sets it as the current "ground truth" for x_i (see Algorithm 2). This ensures the supervision signals not to impose any bias towards any base-learner, and allows image samples with diverse pseudo-labels to have a better chance to be influenced by other image samples. Our experiments

Algorithm 3: Nearest-Neighbor-Fit (NN-Fit)

Input: $(x_i, PL_i = \{f_1(x_i), f_2(x_i), \ldots, \ f_m(x_i)\}, S(PL_i))\,, i = 1, 2, \ldots, n$, meta-learner $\mathcal{H}$ (obtained from random-fit);

Output: A refined meta-learner $\mathcal{H}$;

```
mini-batch = ∅;
while stopping condition not met do
    for k = 1 to batch-size do
        p = rand-int(1, n);
        ŷ = H(x_p, S(PL_p));
        q̂ = arg min_{q=1,2,...,m} L_mce(ŷ, f_q(x_p));
        add training sample {(x_p, S(PL_p)), f_q̂(x_p)} to the mini-batch;
    update H using training samples in the mini-batch with forward and
     backward propagation;
    mini-batch = ∅;
```

show that our random-fit algorithm is effective for learning with diverse pseudo-labels.

Nearest-Neighbor-Fit (NN-Fit). Unlike image classification problems, the label space of segmentation problems is with high spatial dimensions and not all solutions in the label space are meaningful. For example, a union or intersection of two prediction maps (pseudo-labels) may incur a risk of yielding strange shapes or structures that are quite likely incorrect. Even when all pseudo-labels of a particular image sample are close to the true solution, the trained meta-learner, if not fitting any of the pseudo-labels appropriately, can still have a risk of producing new types of errors.

Thus, to help the model training process converge, in the second training phase, we aim to train the meta-learner to fit the nearest pseudo-label. Since the overall training loss is based on cross-entropy, to make NN-fit have direct effects on the convergence of the model training, we use cross-entropy to measure difference between a meta-learner's output and a pseudo-label. The details of our NN-fit algorithm are presented in Algorithm 3. Our experiments show that NN-fit can effectively improve the performance of the deep meta-learner (see Fig. 4.3, and Tables 4.1 and 4.2).

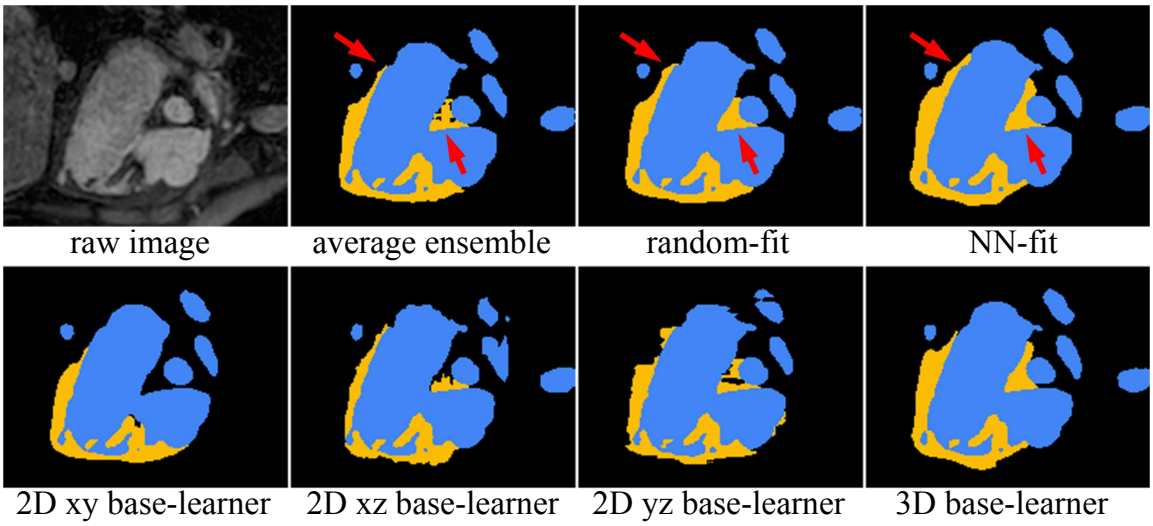

Figure 4.3. Visual comparison of segmentation results (yellow: myocardium; blue: blood pool). With NN-fit, our meta-learner can achieve more accurate segmentation of myocardium (red arrows).

4.3 Evaluation Datasets and Implementation Details

We evaluate our approach using two public datasets: (1) the HVSMR 2016 Challenge dataset [116] and (2) the mouse piriform cortex dataset [84].

HVSMR 2016. The objective of the HVSMR 2016 Challenge [116] is to segment the myocardium and great vessel (blood pool) in cardiovascular magnetic resonance (MR) images. 10 3D MR images and their corresponding ground truth annotation are provided by the challenge organizers as training data. The test data, consisting of another 10 3D MR images, are publicly available, yet their ground truths are kept secret for fair comparison. The results are evaluated using three criteria: (1) Dice coefficient, (2) average surface distance (ADB), and (3) symmetric Hausdorff distance. Finally, a score S, computed as $S = \sum_{class}(\frac{1}{2}Dice - \frac{1}{4}ADB - \frac{1}{30}Hausdorff)$, is used to reflect the overall accuracy of the results and for ranking.

Mouse Piriform Cortex. Our approach is also evaluated on the mouse piriform cortex dataset [84] for neuron boundary segmentation in serial section EM images.

This dataset contains 4 stacks of 3D EM images. Following the previous practice [84, 132], the 2nd, 3rd, and 4th stacks are used for model training, and the 1st stack is used for testing. Also, as in [84, 132], the results are evaluated using the Rand F-score (the harmonic mean of the Rand merge score and the Rand split score).

Implementation Details. All our networks are implemented using TensorFlow [2]. The weights of our 2D base-learners are initialized using the strategy in [51]. The weights of our 3D base-learner and meta-learner are initialized with a Gaussian distribution ($\mu = 0$, $\sigma = 0.01$). All our networks are trained using Adam [75] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon =$ 1e-10. The initial learning rates are all set as 5e-4. Our 2D base-learners reduce the learning rates to 5e-5 after 10k iterations; our 3D base-learner and meta learner adopt the "poly" learning rate policy [168] with the power variable equal to 0.9 and the max iteration number equal to 40k. To leverage the limited training data, standard data augmentation techniques (i.e., random rotation with 90, 180, and 270 degrees, as well as image flipping along the axial planes) are employed to augment the training data.

For the HVSMR 2016 Challenge dataset, due to large intensity variance among different images, all the cardiac images are normalized to have zero mean and unit variance. We also employ spatial resampling to $1mm$ isotropically. For the mouse piriform cortex data, since the 3D EM images are highly anisotropic ($7 \times 7 \times 40nm$), the 2D base-learners in the xz and yz views did not converge well. Thus, we only use the 3D base-learner and the 2D base-learner in the xy view for this dataset.

4.4 Experiments

Because our meta-learner training does not require any manual-labeled data, our method can be easily adapted to the semi-supervised and transductive settings. Thus, we experiment with the following three main settings to demonstrate the effectiveness of our method.

1. To achieve fair comparison with the known state-of-the-art methods that cannot leverage unlabeled data, under the first setting, we train our meta-learner using only training data (the "only training data" entries in Tables 4.1 and 4.2).

2. We show that our model can be improved under the semi-supervised setting in which we use additional unlabeled images to train our meta-learner (Table 4.3).

3. We show that improved results can be obtained under the tranductive setting in which we allow our meta-learner to utilize test data (the "transductive" entries in Tables 4.1 and 4.2). We emphasize that, although it might be less common to use test data for training in natural scene image segmentation, the transductive setting plays an important role in many biomedical image segmentation tasks (e.g., for making biomedical discoveries). For example, after biological experiments are finished, one may have all the raw images available and the sole remaining goal is to train a model to attain the best possible segmentation results for all the data to achieve accurate quantitative analysis.

4.4.1 Comparison with State-of-the-Art Methods When Only Using Training Data

HVSMR 2016. Table 4.1 shows a quantitative comparison with other methods in the leader board of the HVSMR 2016 Challenge [116]. Recall the two categories of the known deep learning based 3D segmentation methods (discussed in Section 4.1). We choose at least one typical method from each category for comparison. (1) [152] is based on the tri-planar scheme [123], which utilizes *three* 2D ConvNets on the orthogonal planes to predict a class label for each voxel. (2) VFN [155] first trains *three* 2D models with slices that are split from three orthogonal planes, respectively, and then applies a 3D ConvNet to fuse 2D results together. Besides, we compare our approach with state-of-the-art models (including 3D U-Net [28], VoxResNet [17], and DenseVoxNet [168]). Without using unlabeled data, our meta-learner outperforms these methods on nearly all the metrics and has a very high overall score, 0.215 (ours) *vs.* -0.161 (DenseVoxNet), -0.036 (tri-planar), and 0.108 (VFN).

TABLE 4.1

QUANTITATIVE ANALYSIS ON THE HVSMR 2016 DATASET[1]

Method	Myocardium			Blood Pool			Overall Score
	Dice	ADB [mm]	Hausdorff [mm]	Dice	ADB [mm]	Hausdorff [mm]	
3D U-Net [28]	0.694 ± 0.076	1.461 ± 0.397	10.221 ± 4.339	0.926 ± 0.016	0.940 ± 0.192	8.628 ± 3.390	-0.419
VoxResNet [17]	0.774 ± 0.067	1.026 ± 0.400	6.572 ± 0.013	0.929 ± 0.013	0.981 ± 0.186	9.966 ± 3.021	-0.202
DenseVoxNet [168]	0.821 ± 0.041	0.964 ± 0.292	7.294 ± 3.340	0.931 ± 0.011	0.938 ± 0.224	9.533 ± 4.194	-0.161
Wolterink *et al.*[152]	0.802 ± 0.060	0.957 ± 0.302	6.126 ± 3.565	0.926 ± 0.018	0.885 ± 0.223	7.069 ± 2.857	-0.036
VFN* [155]	0.773 ± 0.098	0.877 ± 0.318	4.626 ± 2.319	0.935 ± 0.009	0.770 ± 0.098	5.420 ± 2.152	0.108
Base-learner 2D (xy)	0.789 ± 0.076	0.852 ± 0.265	4.231 ± 1.908	0.930 ± 0.016	0.794 ± 0.153	5.295 ± 1.671	0.13
Base-learner 2D (xz)	0.736 ± 0.093	1.000 ± 0.260	5.417 ± 1.604	0.924 ± 0.015	0.932 ± 0.113	7.951 ± 2.820	-0.098
Base-learner 2D (yz)	0.756 ± 0.082	0.870 ± 0.181	4.169 ± 0.632	0.928 ± 0.012	0.812 ± 0.111	**5.229 ± 1.721**	0.108
Base-learner 3D	0.809 ± 0.069	0.785 ± 0.235	4.121 ± 1.870	0.937 ± 0.008	0.799 ± 0.145	6.285 ± 3.108	0.13
Average ensemble	0.805 ± 0.073	0.708 ± 0.184	**3.211 ± 0.923**	0.936 ± 0.011	0.752 ± 0.119	5.960 ± 2.526	0.2
Our meta-learner (Only training data)	0.823 ± 0.060	0.685 ± 0.164	3.224 ± 1.096	0.935 ± 0.010	0.763 ± 0.120	5.804 ± 2.670	0.21
Our meta-learner (Transductive)	**0.833 ± 0.054**	**0.681 ± 0.178**	3.285 ± 1.370	**0.939 ± 0.008**	**0.733 ± 0.143**	5.670 ± 2.808	**0.234**

[1]VFN*: For fair comparison, we use DenseVoxNet [168] as backbone, which is the same as our 3D base-learner.

TABLE 4.2

QUANTITATIVE RESULTS ON THE MOUSE PIRIFORM CORTEX DATASET

Method	V_{Fscore}^{Rand}
N4 [29]	0.9304
VD2D [84]	0.9463
VD2D3D [84]	0.9720
M^2FCN [132]	0.9866
Our 2D base-learner	0.9948
Our 3D base-learner	0.9956
Average ensemble of 2D and 3D	0.9959
Random-fit (only training data)	0.9963
NN-fit (only training data)	0.9967
Random-fit (transductive)	0.9967
NN-fit (transductive)	**0.9970**

Mouse Piriform Cortex. Owning to the advanced components used in our base-learners (e.g., ResNet components [52] and DenseNet components [61]), our 2D and 3D base-learners already achieve better results than the known state-of-the-art methods (Table 4.2). Nevertheless, from Table 4.2, one can see that our meta-learner is able to (1) further improve the accuracy of the base-learners, and (2) achieve a result that is considerably better than the known state-of-the-art methods (0.9967 *vs.* 0.9866).

TABLE 4.3

SEMI-SUPERVISED SETTING ON HVSMR 2016 DATASET

Method	Method	Myocardium			Blood Pool			Overall Score
		Dice	ADB [mm]	Hausdorff [mm]	Dice	ADB [mm]	Hausdorff [mm]	
A	Base-learner 3D	0.772	0.923	5.559	0.932	0.862	7.546	-0.036
	Meta-learner	0.785	0.874	5.095	0.935	0.805	6.223	0.063
B	Base-learner 3D	0.777	0.886	5.008	0.926	0.942	8.175	-0.045
	Meta-learner	0.800	0.825	4.077	0.929	0.882	7.928	0.038

4.4.2 Utilizing Unlabeled Data

Semi-Supervised Setting. We conduct semi-supervised learning experiments on the HVSMR 2016 dataset. The training set of HVSMR 2016 is randomly divided into two groups evenly, S_a and S_b.

We conduct two sets of experiments. Under the setting of "Group A", we first use S_a to train base-learners using the original manual annotation; we then use $S_a \cup S_b$ to train our meta-learner with pseudo labels generated by the trained base-learners. For the overall training procedure, S_a is labeled data and S_b is unlabeled data. Testing phase utilizes the original test images in HVSMR 2016 dataset. The training & testing procedures for "Group B" follows the same protocol except that base-learners are trained with S_b. As shown in Table 4.3, by leveraging unlabeled images, our approach can improve the model accuracy and generalize well to unseen test data.

Transductive Setting. In this setting, we use the full training data to train our base learners, and use the training and testing data to train our meta-learner. As discussed at the beginning of this section, the transductive setting is important for biomedical image segmentation applications and research. The ability to refine the model after seeing the raw test data (no annotation for test data) is another advantage of our framework. From Table 4.1 and Table 4.2, one can see that our meta-learner can achieve further improvements than using only the training data (0.234 *vs.* 0.215 on the HVSMR dataset, and 0.9970 *vs.* 0.9967 on the piriform dataset).

4.4.3 Ablation Study

Average Ensemble *vs.* Naïve Meta-Learner *vs.* Our Best. The results of the average ensemble of all the base-learners (the 2D and 3D models) are shown in Tables 4.1 and 4.2. One can see that the average ensemble is consistently worse than our meta-learner ensemble. We also compare our meta-learner with the naïve meta-learner implementation (in which the outputs of the base-learners are used as

input and the ground truths of the training set are used to train the meta-learner). Table 4.4 shows the results (the S1 row). One can see that the naïve meta-learner implementation is even worse than the average ensemble (probably due to over-fitting). This demonstrates the effectiveness of our meta-learner structure design and training strategy.

Random-Fit + NN-Fit *vs.* Random-Fit Alone. Random-fit + NN-fit performs significantly better than Random-fit alone (Table 4.4: S7>S6, S5>S4, S9>S8; Table 4.2), which demonstrates that NN-fit can help the training procedure converge and thus improve the segmentation quality.

Model Training Using Pseudo-Labels *vs.* Ground Truth. One may concern that our meta-learner training method totally discards manual-labeled ground truth even when it is available. This ablation study shows that our method can perform better without using any manual ground truth. We explore the following ways of utilizing ground truth. When using only the training data, we compare the difference between only ground truth (S2) and only pseudo-labels (S7). Table 4.4 shows that our training method can achieve better results ($0.215 > 0.192$) when not using ground truth. When utilizing the test data (the transductive setting), we compare the difference between (1) only ground truth (S3), (2) mix of ground truth and pseudo-labels, i.e., using ground truth as the 5th version (S4 & S5), and (3) only pseudo-labels (S8 & S9). In Table 4.4, one can see that (a) using pure ground truth or pure pseudo-labels achieves better results than mixing them together (probably due to the different nature of ground truth and pseudo-labels), and (b) using only pseudo-labels is still better than using ground truth (S8 > S3). We think the reason that our method can work well with only pseudo-labels is because the pseudo-labels have already effectively distilled the knowledge from ground truth [55].

4.5 Conclusions

In this chapter, we present a new *ensemble learning* framework for 3D biomedical image segmentation that can retain and combine the merits of 2D and 3D models. Our approach consists of (1) diverse and accurate base-learners by leveraging diverse geometric and model-architecture perspectives of multiple 2D and 3D models, (2) a fully convolutional network (FCN) based meta-learner that is capable of learning robust visual features/representations to improve the base-learners' results, and (3) a new meta-learner training method that can minimize the risk of over-fitting and utilize unlabeled data to improve performance. Extensive experiments on two public datasets show that our approach can achieve superior performance over the state-of-the-art methods.

TABLE 4.4

ABLATION EXPERIMENTS ON THE HVSMR 2016 DATASET[2]

Setting	Inputs		Supervision of Training Set	Transductive Learning	Supervision of Testing Set	Training		Overall Score
	Raw Image (x_i)	$S(PL_i)$				Random-Fit	NN-Fit	
S1		✓	GT					0.075
S2	✓	✓	GT					0.192
S3	✓	✓	GT	✓	PL	✓		0.217
S4	✓	✓	GT + PL	✓	PL	✓		0.205
S5	✓	✓	GT + PL	✓	PL	✓	✓	0.224
S6	✓	✓	PL			✓		0.199
S7	✓	✓	PL			✓	✓	0.215
S8	✓	✓	PL	✓	PL	✓		0.218
S9	✓	✓	PL	✓	PL	✓	✓	**0.234**

[2] "GT" represents ground truth and "PL" represents pseudo labels. Transductive learning setting: Test image data are involved as unlabeled data in model training.

TABLE 4.5

DETAILED RESULTS OF THE "ABLATION STUDY" IN THE TABLE 4.4

Method	Myocardium			Blood Pool			Overall Score
	Dice	ADB $[mm]$	Hausdorff $[mm]$	Dice	ADB $[mm]$	Hausdorff $[mm]$	
S1	0.742 ± 0.103	0.905 ± 0.271	4.665 ± 1.764	0.930 ± 0.014	0.811 ± 0.118	5.285 ± 1.738	0.075
S2	0.800 ± 0.074	0.737 ± 0.189	3.476 ± 1.100	0.936 ± 0.010	0.751 ± 0.107	5.652 ± 2.385	0.192
S3	0.831 ± 0.054	0.659 ± 0.149	3.098 ± 0.917	0.936 ± 0.010	0.767 ± 0.120	6.208 ± 2.592	0.217
S4	0.820 ± 0.062	0.686 ± 0.165	3.200 ± 0.946	0.935 ± 0.010	0.766 ± 0.110	6.100 ± 2.451	0.205
S5	0.829 ± 0.058	0.713 ± 0.183	3.282 ± 1.129	0.937 ± 0.009	0.743 ± 0.145	5.566 ± 2.829	0.224
S6	0.819 ± 0.063	0.682 ± 0.167	3.246 ± 0.974	0.935 ± 0.010	0.763 ± 0.116	6.254 ± 2.554	0.199
S7	0.823 ± 0.060	0.685 ± 0.164	3.224 ± 1.096	0.935 ± 0.010	0.763 ± 0.120	5.804 ± 2.670	0.215
S8	0.832 ± 0.053	0.662 ± 0.136	3.037 ± 0.870	0.935 ± 0.010	0.764 ± 0.118	6.248 ± 2.515	0.218
S9	0.833 ± 0.054	0.681 ± 0.178	3.285 ± 1.370	0.939 ± 0.008	0.733 ± 0.143	5.670 ± 2.808	**0.234**

CHAPTER 5

AN ANNOTATION SPARSIFICATION STRATEGY FOR 3D MEDICAL IMAGE SEGMENTATION VIA REPRESENTATIVE SELECTION AND SELF-TRAINING

A paper published in *2020 34th AAAI Conference on Artificial Intelligence (AAAI)* [183]

5.1 Backgrounds

3D image segmentation is one of the most important tasks in medical image applications, such as morphological and pathological analysis [58, 84], disease diagnosis [116], and surgical planning [78]. Recently, 3D deep learning (DL) models have been widely used in medical image segmentation and achieved state-of-the-art performance [93, 128, 168], most of which were trained with fully annotated 3D image stacks. The performance of DL models (when applied to testing images) is highly dependant on the amount and variety of labeled data used in model training. However, obtaining medical image annotation data is highly difficult and expensive, and full annotation of 3D medical images is a monotonous, labor-intensive, and time-consuming job. For example, a typical 3D abdominal CT scan is of size $300 \times 512 \times 512$, and would take hours of a medical expert to label certain objects of interest in it. How to reduce annotation effort (e.g., cost, time, and available experts) while attaining the best possible performance of DL models remains a challenging problem for 3D medical image segmentation.

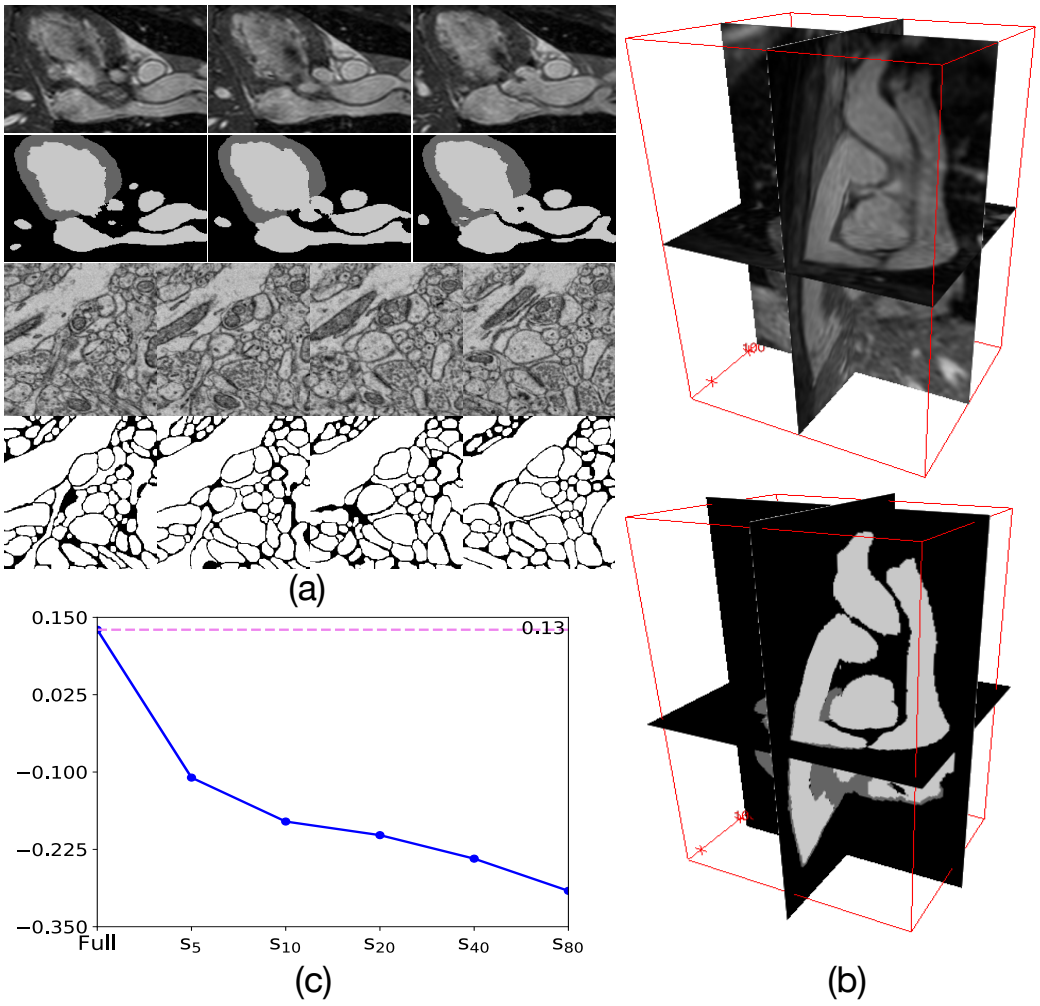

Figure 5.1. (a) Examples showing similarity in consecutive slices of the HVSMR 2016 heart dataset and of the neuron dataset of mouse piriform cortex. (b) Sparse annotation in a 3D image (top: image, bottom: annotation); only selected slices are manually annotated to train deep learning models. (c) Performance on the HVSMR 2016 dataset using different amounts of annotated training data. Let s_k denote the setting of selecting slices at an equal distance (i.e., label one out of every k slices). The segmentation performance drops drastically as the annotation ratio s_k decreases.

A common method to alleviate annotation burden is *sparse 3D fully convolutional networks (FCNs)* [28]. As shown in Fig. 5.1(a), there can be a great deal of redundancy in consecutive 2D slices along an axis of a 3D image, and it is unnecessary to annotate each and every one of them. [28] showed that a small number of annotated 2D slices could be used as supervision (see Fig. 5.1(b)) to train a 3D FCN, and satisfactory segmentation performance was obtained. Compared with conventional 3D FCN models, when calculating the loss, sparse 3D FCN models take only annotated voxels into consideration and perform back-propagation to optimize the networks. However, there are two major issues. (1) The more sparsely one annotates the data, the worse the performance becomes. In our preliminary experiments, we use *equal-interval annotation* (EIA) as a baseline. Although unseen testing stacks can be segmented during inference, the performance decreases drastically if fewer slices are annotated compared with FCNs trained with full annotation (see Fig. 5.1(c)). (2) Which slices are most valuable for annotation? This is not well addressed. A subset of selected slices should be both *informative* and *diverse* so that the subset would cover typical patterns/topology of 3D objects and reduce redundancy. Although a series of sample selection based methods [162, 179, 187] were proposed to deal with 2D image segmentation, for 3D images, this is not well studied.

Another line of related approaches is based on semi-supervised learning (SSL) [173, 186], where abundant and easily-obtainable unannotated data are utilized for training to boost performance. However, the focus of conventional SSL-based methods is somewhat different from our goal to reduce annotation effort: SSL has an underlying assumption that annotated data should be representative enough to cover the true data distribution, but which data samples should be selected for annotation is neglected in previous work. Besides, selected 3D stacks still need dense voxel-wise annotation. Our aim is complementary to SSL-based approaches; we can further reduce annotation effort, and SSL could in turn improve performance by adding more

unannotated data in a later stage.

In this chapter, we proposed a new framework to adapt an annotation sparsification strategy into semi-supervised segmentation. For an unannotated 3D image, we select effective slices with *high influence and diversity* using a representative selection algorithm, which allows a considerable relief of manual annotation. Then we train light-weight networks using sparsely annotated data to perform segmentation on the remaining, unannotated slices and obtain pseudo-labels, which fills the annotation gap in the 3D image. Finally, we use these pseudo-labels as *dense* supervision to conduct self-training with the original training data. To achieve this goal, we need to address three vital challenges: (1) How to provide useful clues about the most influential and diverse slices for manual annotation? (2) How to make the most out of the sparse annotation and generate high quality pseudo-labels? (3) How to conduct self-training using dense pseudo-labels?

For the first challenge, we leverage a pre-trained network to extract image features, and devise a max-cover based method to select the most representative slices. For the second challenge, we observe that the generated pseudo-labels (PLs) by an FCN with sparse annotation contain noise, and different types of FCNs possess different characteristics. For example, inferred PLs from 2D FCNs along the three axes may be inconsistent with one another, but 2D FCNs have a quite large field of view thus large structures could be recognized. In contrast, inferred PLs from 3D FCNs are much smoother since 3D image information could be utilized, but some regions-of-interest may be missing due to their limited field of view. Hence, we adopt the predictions of both 2D and 3D FCNs as supervision for better knowledge distillation. Such heterogeneous predictions are likely to get closer to the correct labels of unannotated slices, and thus the performance gap can be reduced accordingly. For the third challenge, we utilize a self-training based network to combine the merits of multiple sets of PLs, which offers the benefits of weakening noisy labels and reducing over-

fitting.

In summary, our contribution is three-fold. (a) We propose a new training strategy based on representative slice selection and self-training for 3D medical image segmentation. (b) The most representative slices are selected for manual annotation, thus saving annotation effort. (c) Self-training using heterogeneous pseudo-labels bridges the performance gap with respect to full annotation. Extensive experiments show that using only less than 20% annotated slices, our model achieves comparative results as fully-supervised methods.

5.2 A Brief Review of Related DL Techniques

3D Medical Image Segmentation. An array of 2D [128, 132, 152] and 3D [28, 93, 168, 180] FCNs has been developed that significantly improved segmentation performance on various 3D medical image datasets [116, 132]. Scale-level [128] and block-level [52, 61] skip-connections allow substantially deeper architecture design and ease the training by alleviating the vanishing gradient problem. Other advances such as batch normalization [64] and deep supervision [82] also help network training and optimization. In this study, we utilize these advanced techniques in our 2D and 3D FCNs for segmentation.

Sparse Medical Image Annotation. Sparse annotation was not well addressed in medical image segmentation until recently. Where to annotate and how to utilize sparse annotation for training are two basic issues. Active learning (AL) based frameworks [162, 187] reduced annotation effort by incrementally selecting the most informative samples from unlabeled sets and querying human experts for annotation iteratively. Recently, [179] decoupled these two iterative steps in AL frameworks by applying unsupervised networks to encode input samples and extract latent vectors, and ordering the samples based on their representativeness in one-shot, achieving competitive performance. These approaches succeeded in dealing with 2D images

because repeated patterns appear over and over again (e.g., cells, glands, etc), but are not potent enough for a large portion of 3D image datasets which have more complex object topology and fewer samples (see Fig. 5.1(a)). A pioneer work [28] shed some light on sparse 3D FCN training using 2D annotated slices and yielded good performance. Our framework combines these previous methods to address the two basic issues for sparse annotation to obtain good segmentation performance.

Weakly-/Semi-Supervised Learning. Weakly-supervised learning (WSL) based methods explore various weak annotation forms (e.g., points [5], scribbles [94], and bounding boxes [73, 163, 177]). But, none of them is suitable for a large portion of 3D medical images. For example, not all cardiovascular substructures are convex and an object could be wrapped by another (e.g., myocardium and blood pool in Fig. 5.1(a)), or objects are closely packed and are in arbitrary orientation (e.g., neuron cells in Fig. 5.1(a)). Semi-supervised learning (SSL) based methods exploit additional unannotated images to improve segmentation performance. The self-training approach is the earliest SSL one and recently became popular in DL schemes [126, 173]. It uses the predictions of a model on unlabeled data to re-train the model itself iteratively. Another array of work is based on multi-view learning [9] which splits a dataset based on different attributes and utilizes the agreement among different learners. [186] incorporated multi-view learning using multi-view properties of 3D medical data to achieve better performance. However, a major limitation of WSL/SSL based approaches is that they still require annotation of a certain amount of *full* 3D stacks.

We embed a new annotation sparsification strategy into the self-training scheme to address the problem. It further makes use of the underlying assumptions of self-training: the independent and identical distribution of labeled and unlabeled data, and the smoothness of manifold in high-dimensions [108]. Consequently, sparse annotation in each 3D stack would produce accurate pseudo-labels.

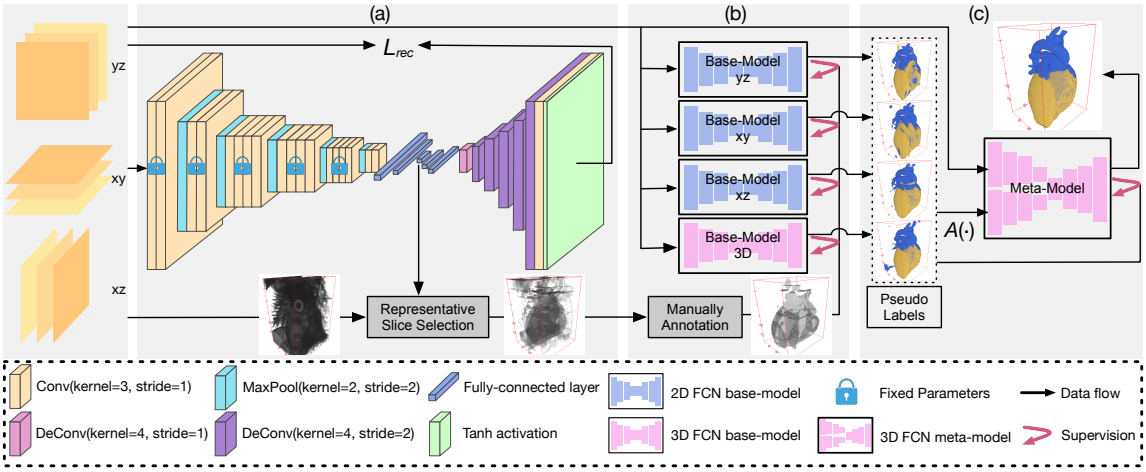

Figure 5.2. An overview of our proposed framework. (a) Representative slice selection. (b) Manual annotation and Pseudo-label (PL) generation from the base-models using sparse annotation. (c) Meta-model training using PLs.

5.3 Methodology

We propose a new annotation sparsification approach which saves considerable annotation effort via representative slice selection from each 3D stack and improves segmentation performance via self-training using pseudo-labels (PLs).

Problem Formulation: Under the fully-supervised setting, given a set of 3D images, $X = \{\mathcal{X}_i\}_{i=1}^{m}$, and their corresponding ground-truth $Y = \{\mathcal{Y}_i\}_{i=1}^{m}$, consider a 3D image $\mathcal{X}_i \in \mathbb{R}^{W \times H \times D}$ with its associated ground-truth C-class segmentation masks, $\mathcal{Y}_i \in \{1, 2, \ldots, C\}^{W \times H \times D}$, where W, H, and D are the numbers of voxels along the x-, y-, and z-axis of $\mathcal{X}_i$ respectively and $\mathcal{Y}_i^{(w,h,d)} = [\mathcal{Y}_i^{(w,h,d,c)}]_c$ provides the label of voxel (w, h, d) as a one-hot vector.

Conventionally, when training a 2D FCN, we can split a 3D volume $\mathcal{X}_i$ along an orthogonal direction. For example, $\{\mathcal{X}_i^V = \{\mathbf{I}_{i,n}^V\}_{n=1}^{N_V}\}_{V \in \{xy,xz,yz\}}$, where N_V is the number of 2D slices obtained from plane V and $\mathbf{I}_{i,n}^V$ is a 2D slice from plane V (e.g., $\mathbf{I}_{i,n}^{xy} \subset \mathbf{R}^{W \times H}$ and $N_V = D$ if $V = xy$). Similarly, $\{\mathcal{Y}_i^V = \{\mathbf{Y}_{i,n}^V\}_{n=1}^{N_V}\}_{V \in \{xy,xz,yz\}}$. If the

3D data are approximate-isotropic, we can split each volume in the xy, xz, and yz planes respectively, and get three sets of 2D slices. Each set $S = \{(\mathbf{I}_\ell, \mathbf{Y}_\ell)\}_{\ell=1}^{L}$, where L is the total number of slices. The goal of segmentation is to design a function $\mathcal{H}$ so that $\hat{\mathbf{Y}}_\ell = \mathcal{H}(\mathbf{I}_\ell)$ is close to $\mathbf{Y}_\ell$. The parameters $\theta_\mathcal{H}$ of $\mathcal{H}$ are learned to minimize the segmentation loss $\mathcal{L}_{seg}(\mathbf{I}_\ell, \mathbf{Y}_\ell) = -\sum \mathbf{Y}_\ell \log \hat{\mathbf{Y}}_\ell$ on the whole set S. Under the sparse annotation setting, only a subset $S' \subseteq S$ is annotated, and the objective is:

$$\min_{\theta_\mathcal{H}} \frac{1}{|S'|} \sum_{\mathbf{I}_\ell \in S'} \mathcal{L}_{seg}(\mathbf{I}_\ell, \mathbf{Y}_\ell) \tag{5.1}$$

When training a 3D FCN, the parameters $\theta_\mathcal{H}$ are optimized by minimizing the loss $\mathcal{L}_{seg}(\mathcal{X}_i, \mathcal{Y}_i) = -\sum \mathcal{Y}_i \log \hat{\mathcal{Y}}_i$ over the whole set $\{(\mathcal{X}_i, \mathcal{Y}_i)\}_{i=1}^{m}$. Under the sparse annotation setting, only a part of all the voxels is annotated. Following [28], the objective function is:

$$\min_{\theta_\mathcal{H}} \frac{1}{|\mathcal{M}(X)|} \sum_{\mathcal{X}_i \in X} \mathcal{L}_{seg}(\mathcal{X}_i, \mathcal{Y}_i) \cdot \mathcal{M}(\mathcal{X}_i) \tag{5.2}$$

where $\mathcal{M}(\mathcal{X}_i) = \mathbb{1}_{\Delta(v)}$ and $\Delta(v) = 1$ if and only if a voxel v in $\mathcal{X}_i$ is annotated (otherwise, $\Delta(v) = 0$). Similarly, it is for $\mathcal{M}(X)$ in the dataset. As shown in Fig. 5.2, our proposed approach consists of three steps:

- Step I: Representative Slice Selection. Pre-train an auto-encoder (AE) using $\{\mathcal{X}_i^V\}_{i=1}^{m}$, and extract the compressed vector from AE as the feature vector of each input 2D slice $\mathbf{I}_{i,n}^V$. Select image slices according to their representativeness captured by the feature vectors.
- Step II: Pseudo-Label (PL) Generation. Train 2D and 3D base-models by Eq. (5.1) and Eq. (5.2) using sparsely annotated 2D slices. The trained base-models are applied to $\{\mathcal{X}_i\}_{i=1}^{m}$ to get corresponding PLs $\{\hat{\mathcal{Y}}_i^V\}_{V \in \{xy, xz, yz, 3D\}}$.
- Step III: FCN self-training. A 3D FCN is trained with noisy PLs to learn from multiple-views of the 3D medical images.

5.3.1 Representative Selection

Intuitively, one could annotate 3D images by a *sub-volume based* method or a *slice based* method. The former method could be impractical in real-world applications for several reasons: (1) human can only annotate 2D slices well; (2) even if a sub-volume is selected, experts have to choose a certain plane (e.g., the xy, xz, or yz plane) and annotate consecutive 2D slices one by one, where a lot of redundancy may exist (e.g., see Fig. 5.1(a)). The latter method, proposed in [28], trains a *sparse 3D FCN model* with some annotated 2D slices, which is more practical and expert-friendly. Considering that regions-of-interest have various topology shapes and feature patterns in different views of 3D data, we hence propose to select some 2D slices from each orthogonal plane for manual annotation.

Feature Extractor with a Pre-trained VGG-19. Auto-encoder (AE) can be used to learn efficient data encoding in an unsupervised manner [130]. It consists of two sub-networks: an *encoder* that takes an input sample $\boldsymbol{x}$ and compresses it into a latent representation $\boldsymbol{z}$, and a *decoder* that reconstructs the sample from the latent representation back to the original space.

$$\boldsymbol{z} \sim Enc(\boldsymbol{x}) = q_\phi(\boldsymbol{z}|\boldsymbol{x}),\ \tilde{\boldsymbol{x}} \sim Dec(\boldsymbol{z}) = p_\psi(\boldsymbol{x}|\boldsymbol{z}) \tag{5.3}$$

where $\{\phi, \psi\}$ are network parameters and the optimization objective is to minimize the reconstruction loss, $\mathcal{L}_{rec}$, on the given dataset X:

$$\psi^*, \phi^* = \arg\min_{\psi,\phi} \mathcal{L}_{rec}(\boldsymbol{x},\ (\phi \circ \psi)\boldsymbol{x}). \tag{5.4}$$

To accelerate the training process and extract rich features, in our implementation, we use the VGG-19 [133] model pre-trained on ImageNet [32] as the backbone network. To further facilitate the customized dataset, we fine-tune the model with

our medical images. More specifically, we tile a few fully-connected (FC) layers to the last convolution layer of the VGG-19 network, and add a light-weight decoder to form an AE. The parameters of the convolution layers of the VGG-19 are fixed, and the remaining network is fine-tuned with the combination of images from the three orthogonal planes.

Representative Slice Selection. Having trained the feature extractor, we feed an image I to the *encoder* model, and the output feature vector, I^f, of the last FC layer can be viewed as a high-level representation of the image I. We can measure the similarity between two images I_i and I_j as:

$$\text{sim}(I_i, I_j) = Cosine\_similarity(I_i^f, I_j^f) \tag{5.5}$$

To measure the representativeness of a set S_x of images for a single image I in another set S_y, we define:

$$f(S_x, I) = \max_{I_i \in S_x} \text{sim}(I_i, I) \tag{5.6}$$

It means I is represented by its most similar image I_i in S_x.

In our scenario, we need to find a subset S_i^V of slices from every 3D stack along each plane (i.e., $S_i^V \subset \mathcal{X}_i^V = \{\mathbf{I}_{i,n}^V\}_{n=1}^{N_V}$, where $V \in \{xy, xz, yz\}$) such that S_i^V is the most representative for the corresponding $\mathcal{X}_i^V$. To measure how representative S_i^V is for $\mathcal{X}_i^V$, we define the coverage score of S_i^V for $\mathcal{X}_i^V$ as:

$$F(S_i^V, \mathcal{X}_i^V) = \sum_{I_j \in \mathcal{X}_i^V} f(S_i^V, I_j) \tag{5.7}$$

This forms a maximum set cover problem which is known to be NP-hard. Its best possible polynomial time approximation solution is based on a greedy method with an approximation ratio $1 - \frac{1}{e}$ [57]. Therefore, we iteratively choose one image slice

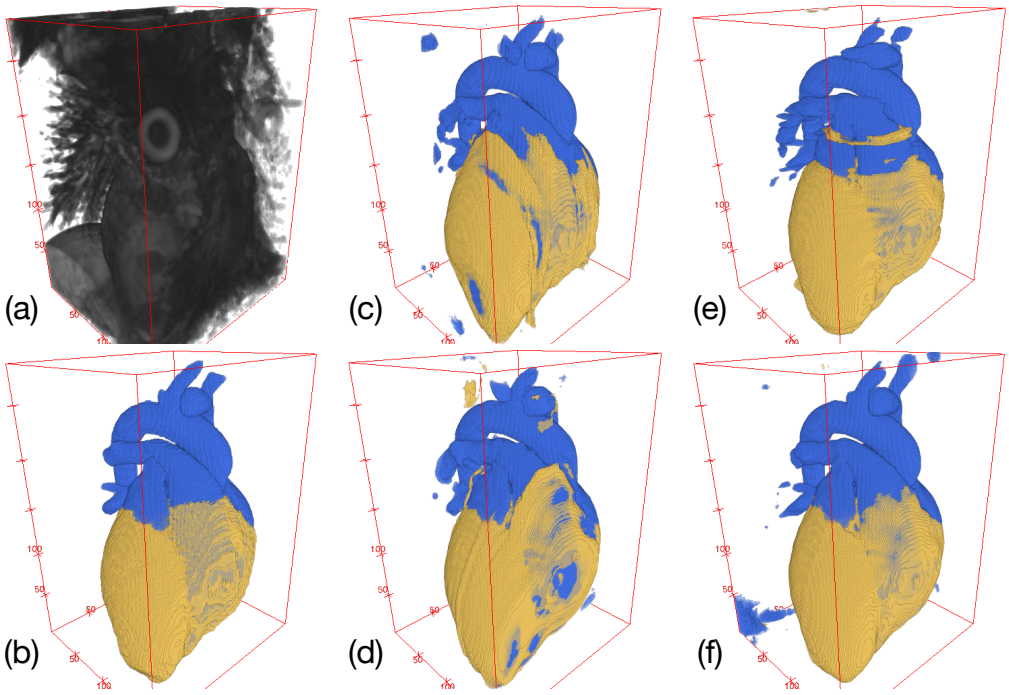

Figure 5.3. Pseudo-labels generated with an annotation budget s_{20}. (a) A raw image $\mathcal{X}_1$; (b) manual annotation $\mathcal{Y}_1$; (c)-(f) $\{\hat{\mathcal{Y}}_1^V\}_{V\in\{xy,xz,yz,3D\}}$, respectively.

from $\mathcal{X}_i^V$ and put it into S_i^V:

$$I^* = \underset{I\in\mathcal{X}_i^V\setminus S_i^V}{\arg\max}(F(S_i^V\cup\{I\},\mathcal{X}_i^V) - F(S_i^V,\mathcal{X}_i^V)) \tag{5.8}$$

This selection process essentially sorts the image slices in $\mathcal{X}_i^V$ based on their representativeness decreasingly. We record the order of the selected slices. The better representative slices have higher priorities for manual annotation.

Under the *equal-interval annotation* (EIA) setting, we select slices at an equal distance, i.e., labeling one out of every k slices, denoted by s_k. The number of EIA-selected slices along the z-axis is $K = \lfloor D/s_k \rfloor$, where D is the number of voxels along the z-axis. Given the same annotation budget, s_k, in our *representative annotation* (RA) setting, we select the K most representative slices along the z-axis.

5.3.2 Pseudo-Label Generation

After obtaining sparse annotation from human experts, following [28], we can train a sparse 3D FCN by Eq. (5.2). Although 3D FCNs can better utilize 3D image information, they adopt a sliding-window strategy to avoid the out of memory problem, thus having a relatively small field of view. Compared with 3D FCNs, 2D FCNs take 2D images as input and can be much deeper and have a larger field of view using the same amount of computational resources. Hence, we propose to utilize 2D FCNs as well (by Eq. (5.1)), which make the most out of multiple sets of 2D slices to capture heterogeneous features from different views of 3D data. Naturally, we can train three 2D FCNs on three sets of 2D slices separately. The drawbacks are: (1) multiple versions of 2D models are trained, and (2) each 2D model only observes the 3D volume from a specific view and does not explore full geometric distribution of the 3D data. Thus, we treat the three 2D slice sets $\{\{\mathcal{X}_i^V\}_{V\in\{xy,xz,yz\}}\}_{i=1}^m$ equally. In each forward pass of a 2D FCN model, it randomly chooses a stack $\mathcal{X}_i$ and a plane V, and crops a patch from a slice as input. This resembles data augmentation that forces the 2D model to learn more from the 3D data. During inference, we apply the trained 2D FCNs to all the sets of 2D slices respectively, and obtain three sets of predictions in the three orthogonal directions respectively, i.e., $\{\{\hat{\mathcal{Y}}_i^V\}_{V\in\{xy,xz,yz\}}\}_{i=1}^m$. Besides, the trained sparse 3D FCN can produce the fourth set of predictions, $\{\hat{\mathcal{Y}}_i^{3D}\}_{i=1}^m$. We use all these as pseudo-labels (PLs) for the next step. As shown in Fig. 5.3, PLs generated with sparse annotation contain noise, and different types of FCNs possess different characteristics: PLs from the 2D FCNs are inconsistent in the third orthogonal direction, but more structures could be recognized; PLs from the 3D FCN are much smoother, but some regions-of-interest may be missing.

5.3.3 Self-Training with Pseudo-Labels

In the previous steps, we obtain four sets of PLs, $\hat{Y} = \{\{\hat{\mathcal{Y}}_i^V\}_{V\in\{xy,xz,yz,3D\}}\}_{i=1}^m$ for the training set $X = \{\mathcal{X}_i\}_{i=1}^m$. Here we aim to train a meta-model that summarizes the noisy PLs and attains better prediction accuracy.

Following the practice in [181], our meta-model is designed as a Y-shape DensVoxNet [168] (see Fig. 5.4), which takes two pieces of input, $\mathcal{X}_i$ and $A(\hat{\mathcal{Y}}_i)$. $A(\cdot)$ is the averaging function that forms a compact representation of $\hat{\mathcal{Y}}_i$ of the PLs. This representation shows the image areas where the PLs hold agreement or disagreement (i.e., average prediction values close to 1 or 0). In addition, using the average of all the PLs of $\mathcal{X}_i$ to form part of the meta-model's input can be viewed as a preliminary ensemble of the base-models and ease the training of the meta-model.

Rather than defining a fixed learning objective for the meta-model training, we train the meta-model in two main stages: (1) Initially, we train the meta-model in order to set up a near-optimal (or sub-optimal) configuration: The meta-model is aware of all the available PLs, and its position in the hypothesis space is influenced by the raw image and the PL data distribution; (2) In the second training stage, we train the meta-model to fit the nearest PLs to help the training process converge. More technical details are given below.

In the first training stage, we seek to minimize the overall cross-entropy loss for all the image samples with respect to all the PLs:

$$\min_{\theta_{\mathcal{H}}} \sum_{i=1}^{m} \sum_{V} \ell_{mce}(\theta_{\mathcal{H}}(\mathcal{X}_i, A(\hat{\mathcal{Y}}_i)), \hat{\mathcal{Y}}_i^V), \tag{5.9}$$

where $\theta_{\mathcal{H}}$ is the meta-model's parameters and ℓ_{mce} is a multi-class cross-entropy loss. In every training iteration, for one image sample $\mathcal{X}_i$, we randomly choose a set of PLs from $\hat{\mathcal{Y}}_i^V$ ($V \in \{xy, xz, yz, 3D\}$) and set it as the "ground truth" for $\mathcal{X}_i$ in the current training iteration. Randomly choosing PLs for the model to fit ensures the

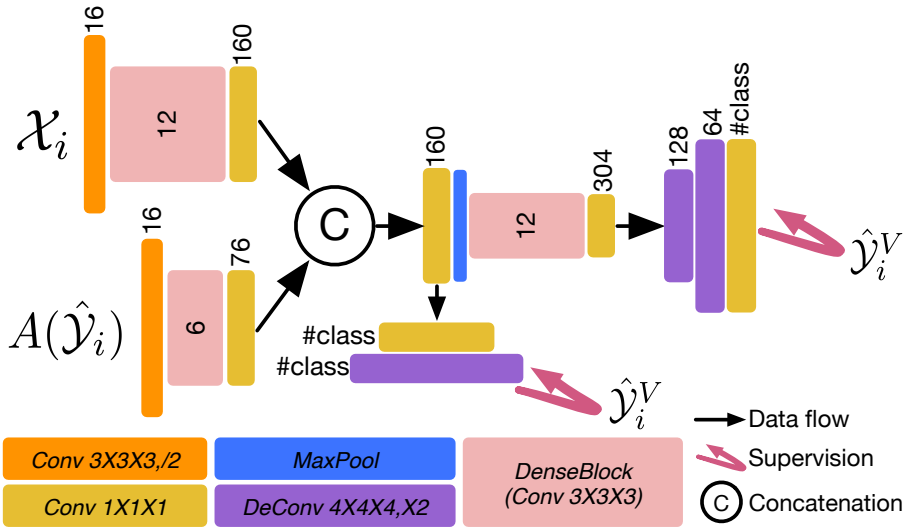

Figure 5.4. The meta-model structure. For readability, BN and ReLU are omitted, the number of channels is given above each unit, and the number of Conv units in each DenseBlock is shown in the block.

supervision signals not to impose any bias towards any base-model, and allows image samples with diverse PLs to have a better chance to be influenced by other image samples.

In the second training stage, the meta-model itself chooses the nearest PLs to fit (based on its current model parameters), and updates its model parameters based on its current choices. This nearest-neighbor-fit (NN-fit) process iterates until the meta-model fits the nearest neighbors well enough. Since the overall training loss is based on cross-entropy, to make the NN-fit have direct effects on the convergence of the model training, we use cross-entropy to measure the "distance" between a meta-model's output and a PL.

5.4 Experiments

To show the effectiveness and efficiency of our new framework, we evaluate it on two public datasets: the HVSMR 2016 Challenge dataset [116] and the mouse

piriform cortex dataset [84].

3D HVSMR Dataset. The HVSMR 2016 dataset consists of 10 3D MR images (MRIs) for training and another 10 MRIs for testing. The goal is to segment myocardium and great vessel (blood pool) in cardiovascular MRIs. The ground truth of the testing data is kept secret by the organizers for fair comparison. The results are evaluated using three criteria: Dice coefficient, average distance of boundaries (ADB), and symmetric Hausdorff distance. Finally, an overall score is computed as $\sum_{class}(\frac{1}{2}Dice - \frac{1}{4}ADB - \frac{1}{30}Hausdorff)$ for ranking, which reflects the overall accuracy of the results.

Mouse Piriform Cortex Dataset. The mouse piriform cortex dataset aims to segment neuron boundaries in serial section EM images. This dataset contains 4 stacks of 3D EM images. Following the setting in [84, 132], we split the dataset into the training set (the 2nd, 3rd, and 4th stacks) and testing set (the 1st stack), which are fixed throughout all experiments. Also, as in [84, 132], the results are evaluated using the Rand F-score (the harmonic mean of the Rand merge score and the Rand split score).

Implementation Details. Our feature extractor network is implemented with PyTorch. The decoder is initialized with a Gaussian distribution ($\mu = 0$, $\sigma = 0.01$) and trained with 2k epochs (with batch size 128; input sizes 128^2 and 256^2 for the HVSMR and mouse piriform cortex datasets, respectively). All our FCNs are implemented using TensorFlow. The weights of our 2D base-models are initialized using the strategy in [51]. The weights of our 3D base-model and meta-model are initialized with a Gaussian distribution ($\mu = 0$, $\sigma = 0.01$). All our networks are trained using Adam [75] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon =$ 1e-10 on an NVIDIA Tesla V100 graphics card with 32GB GPU memory. The initial learning rates are all set as 5e-4. Our 2D base-models decrease the learning rates to 5e-5 after 10k iterations; our 3D base-model and meta-model adopt the "poly" learning rate policy with the power

variable equal to 0.9 [168]. To leverage the limited training data, standard data augmentation techniques (i.e., image flipping along the axial planes and random rotation with 90, 180, and 270 degrees) are employed to augment the training data. Due to large intensity variance among different images, all the images are normalized to have zero mean and unit variance before feeding to the networks.

5.4.1 Main Experimental Results

Our approach consists of two major components: representative annotation (RA) and self-training (ST). To evaluate the effectiveness of our proposed strategy, we first compare our approach using sparse annotation (denoted by **RA+ST**) with the state-of-the-art methods using full annotation on the two datasets. Then, we demonstrate the robustness of our method under different annotation budgets (e.g., $\mathrm{s}_k, k = 5, 10, 20, 40, 80$ for the HVSMR dataset) comparing to the state-of-the-art DenseVoxNet (DVN) [168].

Table 5.1 gives the segmentation results on the HVSMR 2016 dataset. Note that among the state-of-the-art methods on the leaderboard, DVN achieves the highest Dice score and outdoes others on the overall score. Our re-implementation DVN* of DVN is an enhanced version and outperforms other methods by a large margin. We use DVN* as the baseline for all our experiments, for fair comparison. First, compared with the fully supervised DVN*, we obtain a significant improvement on nearly all the metrics, which demonstrates that our method is more effective. More importantly, if we measure annotation effort using the number of voxels selected as representatives by our method, s_5 is equivalent to $\sim 60\%$ of all voxels, which shows the efficiency of our method. Compared with sparse 3D DVN*, our method bridges the performance gap between sparse and full annotations. Second, our approach can further save more annotation effort. We conduct experiments with different annotation ratios; the results are shown in Fig. 5.5. One can note that the performance gap between the

sparse- and fully-annotated 3D DVN* is reduced by our approach with even sparser annotation. Our RA+ST-s_{40} and RA+ST-s_{20} closely approach or outperform the fully supervised DVN*, i.e., our method is able to save up to $\sim 85\%$ of voxel-wise annotation. Some qualitative results are shown in Fig. 5.6. One can see that our method (RA+ST) achieves superior performance than the 2D and 3D base-models, and approaches that of the fully supervised FCN (using more annotation).

TABLE 5.1

QUANTITATIVE RESULTS ON THE HVSMR 2016 DATASET[1]

Model	Annotation Budget	Myocardium			Blood Pool			Overall Score (↑)
		Dice (↑)	ADB[mm] (↓)	Hausdorff[mm] (↓)	Dice (↑)	ADB[mm] (↓)	Hausdorff[mm] (↓)	
3D U-Net [28]	Full	0.694	1.461	10.221	0.926	0.940	8.628	-0.419
VoxResNet [17]		0.774	1.026	6.572	0.929	0.981	9.966	-0.202
Wolterink *et al.* [152]		0.802	0.957	6.126	0.926	0.885	7.069	-0.036
DVN [168]		0.821	0.964	7.294	0.931	0.938	9.533	-0.161
DVN*		0.809	0.785	4.121	**0.937**	0.799	**6.285**	0.13
Sparse DVN* w/ RA	s_5	0.792	1.024	6.906	0.932	0.898	7.396	-0.095
Sparse DVN* w/ RA+ST (Ours)		**0.830**	**0.678**	**3.614**	**0.937**	**0.770**	7.034	**0.166**

[1]DVN*: For fair comparison, we re-implement it and achieve better performance than what was reported in the original paper, and we use it as the backbone in all our experiments. The up arrows (↑) indicate that higher values are better for the corresponding metrics, and vice versa.

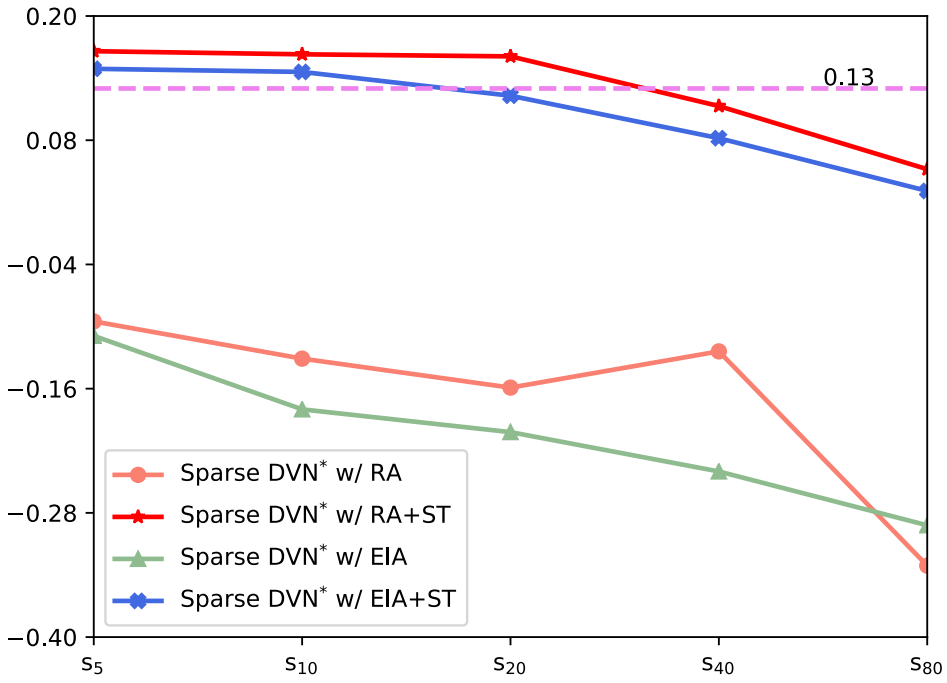

Figure 5.5. Evaluation of several methods on the HVSMR 2016 dataset with different annotation budgets s_k. Given an s_k, RA and EIA select different sets of slices for annotation and FCN training. "Sparse DVN* w/ RA" and "Sparse DVN* w/ EIA" are baselines. The dashed line is the performance using the fully supervised DVN*.

We further evaluate our method on the mouse piriform cortex dataset, using similar experimental settings as those for the HVSMR 2016 dataset. Table 5.2 shows such results. First, we compare our method with an array of 3D FCN-based models, which are all trained with full annotation. Table 5.2 demonstrates that our method with sparse annotation surpasses each such single 3D FCN with full annotation. Second, one can see that with different annotation ratios, the performance gap is reduced consistently. In particular, our RA+ST-s_{64} < DVN*-Full < RA+ST-s_{16}, that is, our method can save up to $\sim 80\%$ of voxel-wise annotation.

TABLE 5.2

QUANTITATIVE RESULTS ON THE MOUSE PIRIFORM CORTEX DATASET[2]

Method	Annotation Budget	V_{Fscore}^{Rand} (↑)
N4 [29]	Full	0.9304
VD2D [84]		0.9463
VD2D3D [84]		0.9720
M^{2}FCN [132]		0.9866
DVN*		0.9959
DVN*	s_4	0.9970
DVN* w/ RA+ST (Ours)		**0.9971**
DVN*	s_{16}	0.9940
DVN* w/ RA+ST (Ours)		**0.9961**
DVN*	s_{64}	0.9951
DVN* w/ RA+ST (Ours)		**0.9957**

5.4.2 Analysis and Discussions

On Representative Annotation (RA). As shown in Fig. 5.5, we compare our strategy with a different annotation strategy: equal-interval annotation (EIA). One can see that "RA+ST" is better than "EIA+ST", which demonstrates that our representative slice selection algorithm helps select more informative and diverse samples to represent the data (see Fig. 5.6(c)). Given the same annotation budget, these RA-selected slices are more valuable for expert annotation.

[2]The up arrow (↑) indicates that higher values are better for the V_{Fscore}^{Rand} metric.

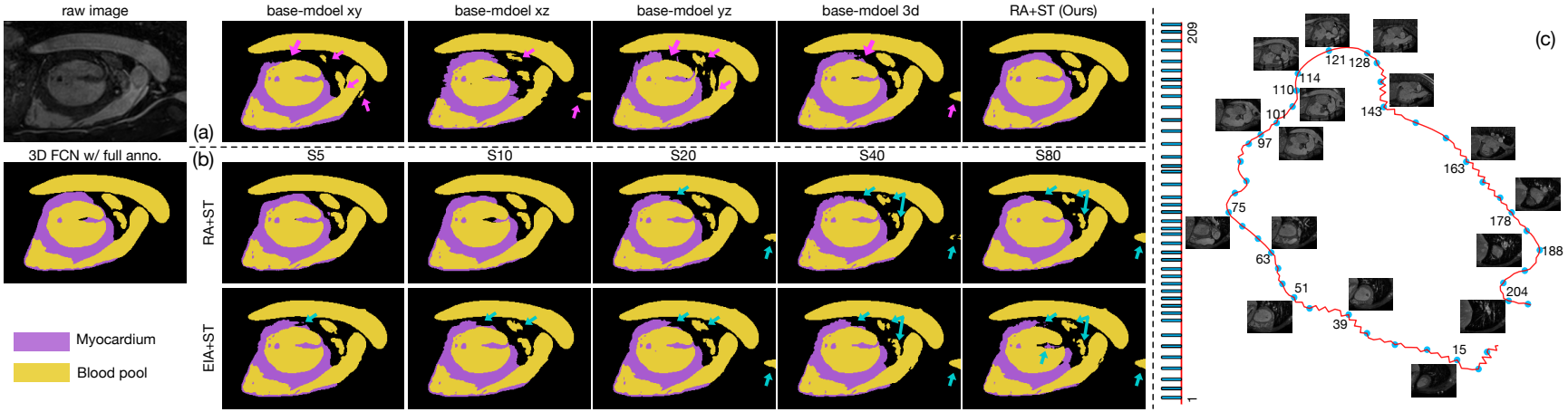

Figure 5.6. Some visual qualitative results on the HVSMR 2016 dataset (some errors are marked by arrows). (a) Results of the 2D and 3D base-models using annotated slices selected by RA. After self-training using pseudo-labels, our approach produces more accurate results which are comparative to that generated by 3D FCN with full annotation. (b) By comparing our strategy RA+ST (the top row of (b)) with EIA+ST (the bottom row of (b)), using slices selected by RA yields superior performance. (c) We show some slices selected by RA (for an s_5 budget) from a 3D stack with the xy-plane. After being projected to 2D space by t-SNE, each point represents one selected slice and the consecutive points form a curve. Selected slices are marked with blue dots and those shown along with thumbnails are labeled with their slice IDs. We also indicate the index positions of the slices selected by RA along the z-axis, as shown by the vertical line on the left of (c) that represents the z-axis of the stack.

On Self-Training. As shown in Fig. 5.5, by comparing "Sparse DVN* w/ RA+ST" with "Sparse DVN* w/ RA", and "Sparse DVN* w/ EIA+ST" with "Sparse DVN* w/ EIA", one can see that utilizing pseudo-labels (PLs) for self-training, the performance is significantly improved. It demonstrate that though PLs generated from sparse annotation may be noisy, they fill the spatial gaps of voxel-wise supervision in the 3D stack. Thus our self-training utilizes the PLs and bridges the final performance gap with respect to full annotation.

5.5 Conclusions

In this chapter, we proposed a new annotation sparsification strategy for 3D medical image segmentation based on representative annotation and self-training. The most valuable slices are selected for manual annotation, thus saving annotation effort. Heterogeneous 2D and 3D FCNs are trained using sparse annotation, which generate diverse pseudo-labels (PLs) for unannotated voxels in 3D data. Self-training utilizing PLs further improves the segmentation performance and bridges the performance gap with respect to full annotation. Our extensive experiments on two public datasets show that using less than 20% annotated data, our new strategy obtains comparative results with fully supervised training.

CHAPTER 6

EMBRYONIC CARTILAGE SEGMENTATION IN HIGH-RESOLUTION 3D MICRO-CT IMAGES WITH VERY SPARSE ANNOTATION

A paper published in *2020 23rd International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)* [182]

6.1 Backgrounds

Approximately 1% of babies born with congenital anomalies have syndromes including skull abnormalities [104]. Anomalies of the skull invariably require treatments and care, imposing high financial and emotional burdens on patients and their families. Although prenatal development data are not available for study in humans, the deep conservation of mammalian developmental systems in evolution means that laboratory mice give access to embryonic tissues that can reveal critical molecular and structural components of early skull development [10, 127]. The precise delineation of 3D chondrocranial anatomy is fundamental to understanding dermatocranium development, provides important information to the pathophysiology of numerous craniofacial anomalies, and reveals potential avenues for developing novel therapeutics. An embryonic mouse is tiny ($\sim 2cm^3$), and thus we dissect and reconstruct the *chondrocranium* from 3D micro-computed tomography (micro-CT) images of specially stained mice. However, delineating fine-grained cartilaginous structures in these images is very challenging, even manually (e.g., see Fig. 6.1).

Although deep learning has achieved great success in biomedical image segmentation [93, 99, 128, 150, 180], there are three main challenges when applying existing

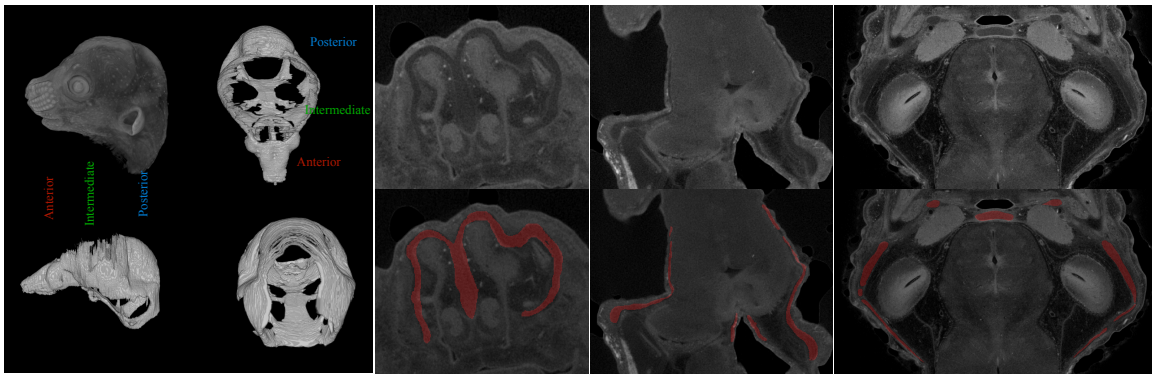

Figure 6.1. Examples of micro-CT images of stained mice. (a) A raw 3D image and its manual annotation. The shape variations are large: the front nasal cartilage is relatively small (i.e., 300^2); the cranial vault is very big (i.e., 900×500) but extremely thin like a half-ellipsoid surface. (b) A 2D slice from the nasal cartilage (top) and its associated label (bottom); the image contrast is low and there are many hard mimics in surrounding areas. (c) Two 2D slices from the cranial vault (top) and their associated labels (bottom); the cartilage is very thin. Best viewed in color.

methods to cartilage segmentation in our high-resolution micro-CT images. (1) The topology variations of craniofacial cartilages are very large in the anterior, intermediate, and posterior of the skull (as shown in Fig. 6.1(a)). Known methods for segmenting articular cartilages in knees [4, 123] only deal with relatively homogeneous structures. (2) Such methods deal with images of much lower resolutions (e.g., 200×512^2), and simple scaling-up would precipitate huge computation requirements. Micro-CT scanners work at the level of one micron (i.e., $1\mu m$, our image pixels range from 6 to 10 microns), and a typical scan of ours is of size 1500×2000^2. In Fig. 6.1(c), the cropped sub-region is of size 400^2, and the region-of-interest (ROI) is only 5 pixels thick. (3) More importantly, only experts can differentiate cartilages, and it is unrealistic to manually label whole volumes for training fully convolution networks (FCNs) [99]. While some semi-supervised methods [169, 183] were studied very recently, how to acquire and make the most out of very sparse annotation is seldom

explored, especially for real-world complex cartilage segmentation tasks.

To address these challenges, we propose a new framework that utilizes FCNs and uncertainty-guided self-training to gradually boost the segmentation accuracy. We start with extremely sparsely annotated 2D slices and train an FCN to predict pseudo labels (PLs) for unseen slices in the training volumes and the associated uncertainty map, which quantifies pixelwise prediction confidence. Guided by the uncertainty, we iteratively train the FCN with PLs and improve the generalization ability of FCN in unseen volumes. Although the above process seems straightforward, we must overcome three difficulties. (1) The FCN should have a sufficiently large receptive field to accommodate such high-resolution images yet needs to be lightweight for efficient training and inference due to the large volumes. (2) Bayesian-based uncertainty quantification requires a linear increase of either space or time during inference. We integrate FCNs into a bootstrap ensemble based uncertainty quantification scheme and devise a K-head FCN to balance efficiency and efficacy. (3) The generated PLs contain noises. We consider the quality of PLs and propose an uncertainty-guided self-training scheme to further refine segmentation results.

Experiments show that our proposed framework achieves an average Dice of 78.98% in segmentation compared to prior arts and obtains performance gains by iterative self-training (from 78.98% to 83.16%).

6.2 Method

As shown in Fig. 6.2, our proposed framework contains a new FCN, which can generate PLs and uncertainty estimation at the same time, and an iterative uncertainty-guided self-training strategy to boost the segmentation results.

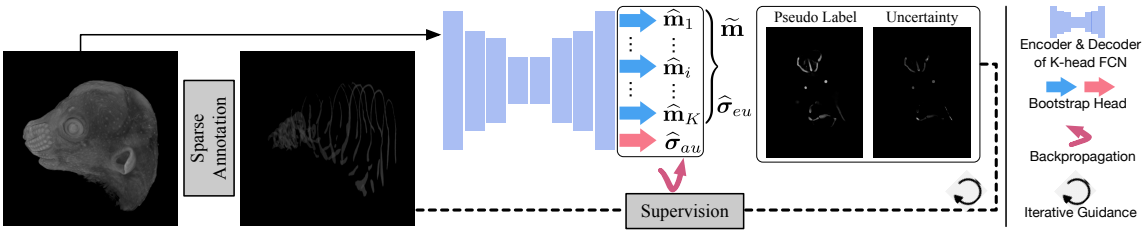

Figure 6.2. An overview of our proposed framework.

6.2.1 K-Head FCN

Initial Labeling and PL Generation. We consider two sets of 3D data, $\mathcal{A} = \{\mathcal{A}_i\}_{i=1}^{L}$ and $\mathcal{B} = \{\mathcal{B}_i\}_{i=1}^{U}$, for training and testing respectively, where each $\mathcal{A}_i$ (or $\mathcal{B}_i$) is a 3D volume and L (or U) is the number of volumes in $\mathcal{A}$ (or $\mathcal{B}$). Each 3D volume can be viewed as a series of 2D slices, i.e., $\mathcal{A}_i = \{\mathbf{A}_i^j\}_{j=1}^{i_Q}$, where i_Q is the number of slices in $\mathcal{A}_i$. To begin with, experts chose representative slices in each $\mathcal{A}_i$ from the anterior, intermediate, and posterior of the skull and annotated them at the pixel level. Due to the high resolution of our micro-CT images, the annotation ratio is rather sparse (e.g., 25 out of 1600 slices). Thus, each $\mathcal{A}_i$ can be divided into two subsets $\mathcal{A}l_i = \{\mathbf{l}_i^j\}_{j=1}^{i_P}$ and $\mathcal{A}u_i = \{\mathbf{u}_i^j\}_{j=1}^{i_R}$, where each slice $\mathbf{l}_i^j$ has its associate label $\mathbf{m}_i^j$, and $i_Q > i_R \gg i_P$. Conventionally, using such sparse annotation, a trained FCN lacks generalization ability to the unseen volumes $\mathcal{B}$. Hence, a key challenge is how to make the most out of the labeled slices. We will show that an FCN can delineate ROIs in unseen slices of the training volumes (i.e., $\mathcal{A}u_i$) with very sparsely labeled slices. For this, we propose to utilize these true labels (TLs) and generate PLs to expand the training data.

Uncertainty Quantification. Since FCN here is not trained by standard protocol, its predictions may be unreliable and noisy. Thus, we need to consider the reliability of the PLs (which may otherwise lead to meaningless guidance). Bayesian methods [71] provided a straightforward way to measure uncertainty quantitatively

by utilizing Monte Carlo sampling in forward propagation to generate multiple predictions. Prohibitively, the computational cost grows linearly (either time or space). Since our data are large volumes, such cost is unbearable. To avoid this issue, we need to design a method that is both time- and space-efficient. Below we illustrate how to design a new FCN for this purpose.

There are two main types of uncertainty in Bayesian modelling [72, 110]: *epistemic uncertainty* captures uncertainty in the model (i.e., the model parameters are poorly determined due to the lack of data/knowledge); *aleatoric uncertainty* captures genuine stochasticity in the data (e.g., inherent noises). Without loss of generality, let $f_\theta(x)$ be the output of a neural network, where θ is the parameters and x is the input. For segmentation tasks, following the practice in [72], we define pixelwise likelihood by squashing the model output through a softmax function $\mathcal{S}$: $p(y|f_\theta(x), \sigma^2) = \mathcal{S}(\frac{1}{\sigma^2} f_\theta(x))$. The magnitude of σ determines how 'uniform' (flat) the discrete distribution is. The log likelihood for the output is:

$$\begin{aligned}
\log p(y = c|f_\theta(x), \sigma^2) &= \frac{1}{\sigma^2} f_\theta^c(x) - \log \sum_{c'} \exp(\frac{1}{\sigma^2} f_\theta^{c'}(x)) \\
&= \left(\frac{1}{\sigma^2} f_\theta^c(x) - \frac{1}{\sigma^2} \log \sum_{c'} \exp(f_\theta^{c'}(x)) \right) \\
&- \left(\log \sum_{c'} \exp(\frac{1}{\sigma^2} f_\theta^{c'}(x)) - \frac{1}{\sigma^2} \log \sum_{c'} \exp(f_\theta^{c'}(x)) \right) \\
&= \frac{1}{\sigma^2} \log \frac{\exp(f_\theta^c(x))}{\sum_{c'} \exp(f_\theta^{c'}(x))} - \log \frac{\sum_{c'} \exp(\frac{1}{\sigma^2} f_\theta^{c'}(x))}{\left(\sum_{c'} \exp(f_\theta^{c'}(x))\right)^{\frac{1}{\sigma^2}}} \\
&\approx \frac{1}{\sigma^2} \log S(f_\theta(x))^c - \frac{1}{2} \log \sigma^2,
\end{aligned} \tag{6.1}$$

where $f_\theta^c(x)$ is the c-th class of output $f_\theta(x)$, and we use the explicit simplifying assumption $\left(\sum_{c'} \exp(f_\theta^{c'}(x))\right)^{\frac{1}{\sigma^2}} \approx \frac{1}{\sigma} \sum_{c'} \exp(\frac{1}{\sigma^2} f_\theta^{c'}(x))$. The objective is to minimize

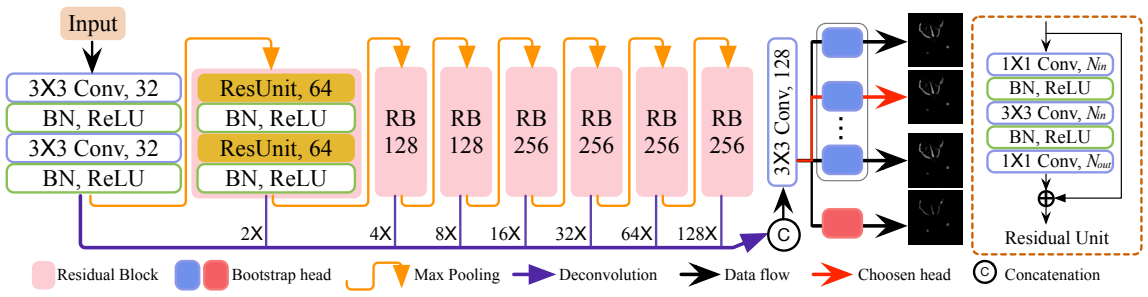

Figure 6.3. The network architecture of our proposed method, K-head FCN. The output layer branches out to K bootstrap heads and an extra log-variance output.

the loss given by the negative log likelihood:

$$\mathcal{L}_{UC}(\theta, \sigma^2) = -\frac{1}{N}\sum_{i}^{N}\sum_{m}^{M} \mathbb{1}_{m=c}\log(p(y_i = c|f_\theta(x_i), \sigma^2)), \tag{6.2}$$

where N is the number of training samples and $\mathbb{1}_{m=c}$ is the one-hot vector of class c. In practice, we make the network predict the log variance $s := \log\sigma^2$ for numerical stability. Now, the aleatoric uncertainty is estimated by e^{-s}, and we can quantify the epistemic uncertainty by the predictive variance:

$$\frac{1}{K}\sum_{k}^{K}\hat{y}_k^2 - \left(\frac{1}{K}\sum_{k}^{K}\hat{y}_k\right)^2, \tag{6.3}$$

where $\hat{y}_k = f_\theta(x)$ is the k-th sample from the output distribution.

K-head FCN. To sample K samples from the output distribution, we adopt the bootstrap method into the FCN design. A naïve way would be to maintain a set of K networks $\{f_{\theta_k}\}_{k=1}^{K}$ independently on K different bootstrapped subsets (i.e., $\{D_k\}_{k=1}^{K}$) of the whole dataset D and treat each network f_{θ_k} as independent samples from the weight distribution. However, it is computationally expensive, especially when each neural net is large and deep. Hence, we propose a single network that consists

of a shared backbone architecture with K lightweight bootstrapped heads branching on/off independently. The shared network learns a joint feature representation across all the data, while each head is trained only on its bootstrapped sub-sample of the data. The training and inference of this type of bootstrap can be conducted in a single forward/backward pass, thus saving both time and space. Besides, in contrast to previous methods where σ^2 is assumed to be constant for all inputs, we estimate it directly as an output of the network [71, 110]. Thus, our proposed network consists of a total of $K+1$ branches — K heads corresponding to the segmentation prediction map and an extra head corresponding to σ^2. In all the experiments, K is set as 5, and the input image size is 512×512.

Fig. 6.3 shows the detailed structure of our new K-head FCN. There are 7 residual blocks (RBs) and max-pooling operations in the encoding-path to deliver larger reception fields, each RB containing 2 cascaded residual units as in ResNet [52]. To save parameters, we maintain the number of channels in each residual unit and a similar number of feature channels at the last 4 scales. Rich contextual and semantic information is extracted in shallower and deeper scales in the encoding-path and is up-sampled to maintain the same size for the input and output and then concatenated to generate the final prediction. The output layer splits near the end of the model for two reasons: (1) ease the training difficulty and improve the convergence speed; (2) incur minimal computation resource increases (both time and space) in training and inference. To train the network, we randomly choose one head in each iteration and compute the cross-entropy loss $\mathcal{L}_{CE}$. It is combined with the uncertainty loss $\mathcal{L}_{UC}$ to update the parameters in the chosen head branch and the shared backbone only (i.e., freezing the other $K-1$ head branches). Specifically, $\mathcal{L} = \mathcal{L}_{CE} + 0.04\mathcal{L}_{UC}$.

6.2.2 Iterative Uncertainty-Guided Self-Training

Since both $\mathcal{A}l_i$ and $\mathcal{A}u_i$ come from the same volume $\mathcal{A}_i$ and are based on the assumption that the manifolds of the seen/unseen slices (of $\mathcal{A}_i$) are smooth in high dimensions [108], our generated PLs bridge the annotation gap. However, the K predictions, $\{\widehat{\mathbf{m}}_i^{j,k}\}_{k=1}^{K}$, obtained from the output distribution for each $\mathbf{u}_i^j \in \mathcal{A}u_i$ could be unreliable and noisy. Thus, we propose an uncertainty-guided scheme to reweight PLs and rule out unreliable (highly uncertain) pixels in subsequent training. Specifically, we calculate the voxel-level cross-entropy loss weighted by the epistemic uncertainty $\boldsymbol{\sigma}_i^j$ for $\mathbf{u}_i^j$:

$$\mathcal{L}_{CE}(\overline{\mathbf{m}}_i^j, \widetilde{\mathbf{m}}_i^j) = \frac{\sum_v e^{-\sigma_v} \mathcal{L}_{ce}(\overline{m}_v, \widetilde{m}_v)}{\sum_v e^{-\sigma_v}}, \tag{6.4}$$

where $\overline{\mathbf{m}}_i^j$ is the prediction at the current iteration and $\widetilde{\mathbf{m}}_i^j = \sum_{k=1}^{K} \widehat{\mathbf{m}}_i^{j,k}$; $\overline{m}_v$ and $\widetilde{m}_v$ are the values of the v-th pixel (for simplicity, we omit i and j); σ_v is the sum of normalized epistemic and aleatoric uncertainties at the v-th pixel; $\mathcal{L}_{ce}$ is the cross-entropy error at each pixel. Note that we do not choose a hard threshold to convert the average probability map $\widetilde{\mathbf{m}}_i^j$ to a binary mask, as inspired by the "label smoothing" technique [106] which may help prevent the network from becoming over-confident and improve generalization ability.

With the expansion of the training set (TLs $\cup$ PLs), our FCN can distill more knowledge about the data (e.g., topological structure, intensity variances), thus becoming more robust and generalizing better to unseen data $\mathcal{B}$. However, due to the extreme sparsity of annotation at the very beginning, not all the generated PLs are evenly used (i.e., highly uncertain and assigned with low weights). Hence, we propose to conduct this process iteratively.

Overall, with our iterative uncertainty-guided self-training scheme, we can further refine the PLs and FCN at the same time. In practice, it needs 2 or 3 rounds, but

we do not have to train from scratch, incurring not too much cost.

6.3 Experiments

Data Acquisition. Mice were produced, sacrificed, and processed in compliance with animal welfare guidelines approved by the Pennsylvania State University (PSU). Embryos were stained with phosphotungstic acid (PTA), as described in [85]. Data were acquired by the PSU Center for Quantitative Imaging using the General Electric v|tom|x L300 nano/micro-CT system with a 180-kV nanofocus tube and were then reconstructed into micro-CT volumes with a resulting average voxel size of $5\mu m$ and volume size of 1500×2000^2. Seven volumes are divided into the training set $\mathcal{A} = \{\mathcal{A}_i\}_{i=1}^4$ and test set $\mathcal{B} = \{\mathcal{B}_i\}_{i=1}^3$. Only a very small subset of slices in each $\mathcal{A}_i$ is labeled for training (denoted as $\mathcal{A}l_i$) and the rest unseen slices $\mathcal{A}u_i$ and $\mathcal{B}$ are used for the test. Four scientists with extensive experience in the study of embryonic bones/cartilages were involved in image annotations. They first annotated slices in the 2D plane and then refined the whole annotation by considering 3D information of the neighboring slices.

Evaluation. In the 3D image regions not considered by the experts, we select 11 3D subregions (7 from $\mathcal{B}$ and 4 from $\mathcal{A}u_i$), each of an average size 30×300^2 and containing at least one piece of cartilages. These subregions are chosen for their representativeness, i.e., they cover all the typical types of cartilages (e.g., nasal capsule, Meckel's cartilage, lateral wall, braincase floor, etc). Each subregion is manually labeled by experts as ground truth. The segmentation accuracy is measured by Dice-Sørensen Coefficient (DSC).

Implementation Details. All our networks are implemented with TensorFlow [2], initialized by the strategy in [51], and trained with the Adam optimizer [75] (with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon =$ 1e-10). We adopt the "poly" learning rate policy, $L_r \times \left(1 - \frac{iter}{\#iter}\right)^{0.9}$, where the initial rate $L_r =$ 5e-4 and the max iteration number is

TABLE 6.1

CARTILAGE SEGMENTATION RESULTS[1]

Method	Anterior	Intermediate	Posterior	Overall
U-Net* [128] (TL)	80.03	81.19	64.39	76.06
DCN* [16] (TL)	80.87	81.68	64.07	76.42
K-head FCN (TL)	82.23	84.46	67.52	78.98
1-head FCN-R3 (TL∪PL)	85.15	87.53	69.46	81.69
K-head FCN-R3 (TL∪PL)	85.77	88.34	70.30	82.45
K-head FCN-R3-U (TL∪PL)	86.31	89.17	70.98	83.16

set as 60k. To leverage the limited training data and reduce over-fitting, we augment the training data with standard operations (e.g., random crop, flip, rotation in 90°, 180°, and 270°). Due to large intensity variance among different images, all images are normalized to have zero mean and unit variance.

6.3.1 Main Experimental Results

The results are summarized in Table 6.1. To our best knowledge, there is no directly related work on cartilage segmentation from embryonic tissues. We compare our new framework with the following methods. (1) A previous work which utilizes U-Net [128] to automatically segment knee cartilages [4]. We also try another robust FCN model DCN [16]. For a fair comparison, we scale up U-Net [128] and DCN [16] to accommodate images of size 512^2 as input and match with the number of parameters of our K-head FCN (denoted as U-Net* and DCN*). (2) A semi-supervised method

[1]DSC (%) comparison of cartilages in the anterior, intermediate, and posterior skull, w/ annotation ratio of 3.0%. TL: true labels; PL: pseudo labels.

that generates PLs and conducts self-training (i.e., 1-head FCN-R3).

First, compared with known FCN-based methods, our K-head FCN yields better performance for cartilages in different positions. We attribute this to its deeper structures and multi-scale extracted feature fusion design, which leads to larger receptive fields and richer spatial and semantic features. Hence, our backbone model can capture significant topology variances in skull cartilages (e.g., relatively small but thick nasal parts, and large but thin shell-like cranial base and vault).

Second, to show that our K-head FCN is comparable with Monte Carlo sampling based Bayesian methods, we implement 1-head FCN and conduct sampling K times to obtain PLs. Repeating the training process 3 times (denoted as '-R3'), we observe that using PLs, K-head FCN-R3 achieves similar performance as 1-head FCN-R3. However, in each forward pass, we obtain K predictions at once, thus saving $\sim K\times$ the time/space costs. Qualitative results are shown in Fig. 6.4.

Third, we further show that under the guidance of uncertainty, our new method (K-head FCN-R3-U) attains performance gain (from 82.45% to 83.16%). We attribute this to that unreliable PLs are ruled out, and the model optimizes under cleaner supervisions. Qualitative results are shown in Fig. 6.5.

6.3.2 Analysis and Discussions

Iteration Numbers. We measure DSC scores on both unseen slices in the training volumes ($\{\mathcal{A}u_i\}_{i=1}^{L}$) and unseen slices in the test volumes ($\{\mathcal{B}_i\}_{i=1}^{U}$) during the training of "K-head FCN-R3-U" (see Table 6.2 left). We notice significant performance gain after expanding the training set (i.e., TLs $\rightarrow$ TLs $\cup$ PLs, as Iter-1 $\rightarrow$ Iter-2). Meanwhile, because the uncertainty of only a small amount of pixels changes during the whole process, the performance gain is not substantial from Iter-2 to Iter-3.

Annotation Ratios. As shown in Table 6.2 right, the final segmentation results can be improved using more annotation, but the improvement rate decreases when

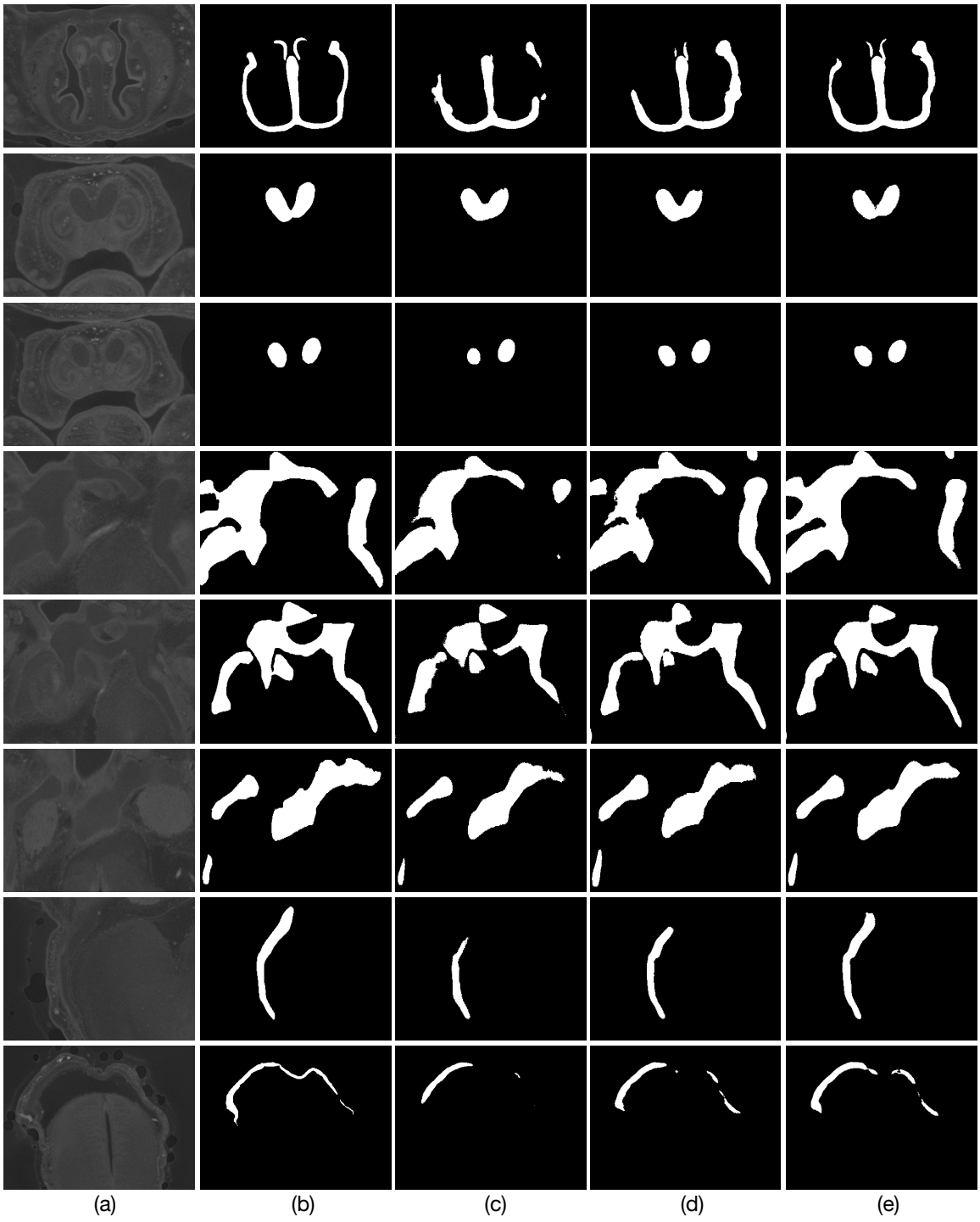

Figure 6.4. Qualitative examples: (a) Raw subregions; (b) ground truth; (c) U-Net* (TL); (d) K-head FCN (TL); (e) K-head FCN-R3-U (TL∪PL). (XX) = (trained using XX).

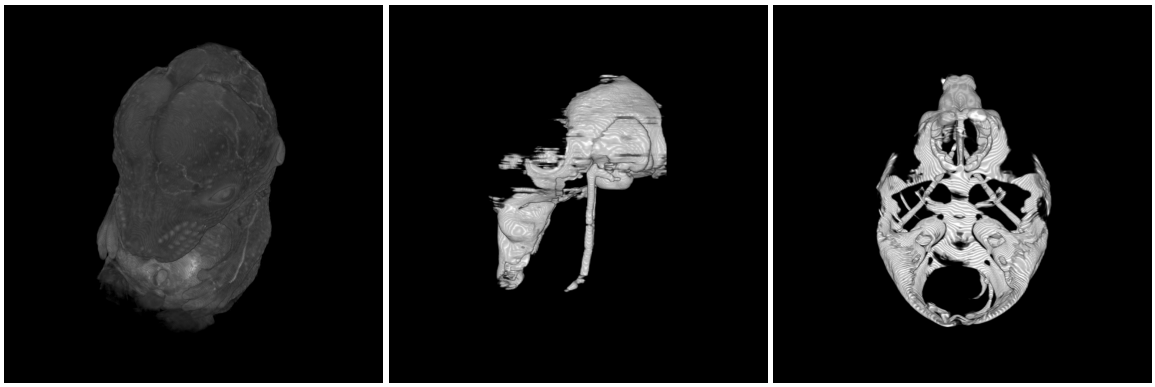

Figure 6.5. Qualitative results. From left to right: A raw image; 3D results of our proposed method (K-head FCN-R3-U (TL∪PL)) from different views.

TABLE 6.2

SEGMENTATION RESULTS OF K-HEAD FCN-R3-U (TL∪PL) WITH DIFFERENT ITERATIONS AND ANNOTATION RATIOS[2]

Data	Iteration		
	1	2	3
$\{\mathcal{A}u_i\}_{i=1}^{L}$	83.19	86.39	87.08
$\{\mathcal{B}_i\}_{i=1}^{U}$	78.98	82.70	83.16

Data	Annotation Ratio		
	1.5%	3.0%	12.0%
$\{\mathcal{A}u_i\}_{i=1}^{L}$	80.12	87.08	89.20
$\{\mathcal{B}_i\}_{i=1}^{U}$	75.73	83.16	85.65

labeling more slices.

Uncertainty Estimation. We visualize the samples along with estimated segmentation results and the corresponding epistemic and aleatoric uncertainties from the test data in Fig. 6.6. It is shown that the model is less confident (i.e., with a

[2]The results are evaluated using DSC (%). TL: true labels; PL: pseudo labels. Left: "K-head FCN-R3-U (TL∪PL)" w/ annotation ratio of 3.0%. Right: "K-head FCN-R3-U (TL∪PL)" w/ different annotation ratios.

Figure 6.6. Visualization of uncertainty. From left to right: a raw image region, ground truth, prediction result, estimated epistemic uncertainty, and estimated aleatoric uncertainty. Brighter white color means higher uncertainty.

higher uncertainty) on the boundaries and hard mimic regions where the epistemic and aleatoric uncertainties are prominent.

6.4 Conclusions

In this chapter, we presented a new framework for cartilage segmentation in high-resolution 3D micro-CT images with very sparse annotation. Our K-head FCN produces segmentation predictions and uncertainty estimation simultaneously, and the iterative uncertainty-guided self-training strategy gradually refines the segmentation results. Comprehensive experiments showed the efficacy of our new method.

CHAPTER 7

HIERARCHICAL SELF-SUPERVISED LEARNING FOR MEDICAL IMAGE SEGMENTATION BASED ON MULTI-DOMAIN DATA AGGREGATION

A paper published in *2021 24th International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)* [184]

7.1 Backgrounds

Although supervised deep learning has achieved great success on medical image segmentation [19, 62, 128, 189], it heavily relies on sufficient good-quality manual annotations which are usually hard to obtain due to expensive acquisition, data privacy, etc. Public medical image datasets are normally smaller than the generic image datasets (see Fig. 7.1(a)), and may hinder improving segmentation performance. Deficiency of annotated data has driven studies to explore alternative solutions. Transfer learning fine-tunes models pre-trained on ImageNet for target tasks [53, 187, 190], but it could be impractical and inefficient due to the pre-defined model architectures [97] and is not as good as transferred from medical images due to image characteristics differences [190]. Semi-supervised learning utilizes unlimited amounts of unlabeled data to boost performance, but it usually assumes that the labeled data sufficiently covers the data distribution, and needs to address consequent non-trivial challenges such as adversarial learning [101, 174] and noisy labels [169, 183]. Active learning selects the most representative samples for annotation [162, 179, 187] but focuses on saving manual effort and does not utilize unannotated data. Considering these limitations and the fact that considerable unlabeled medical images are easy to acquire

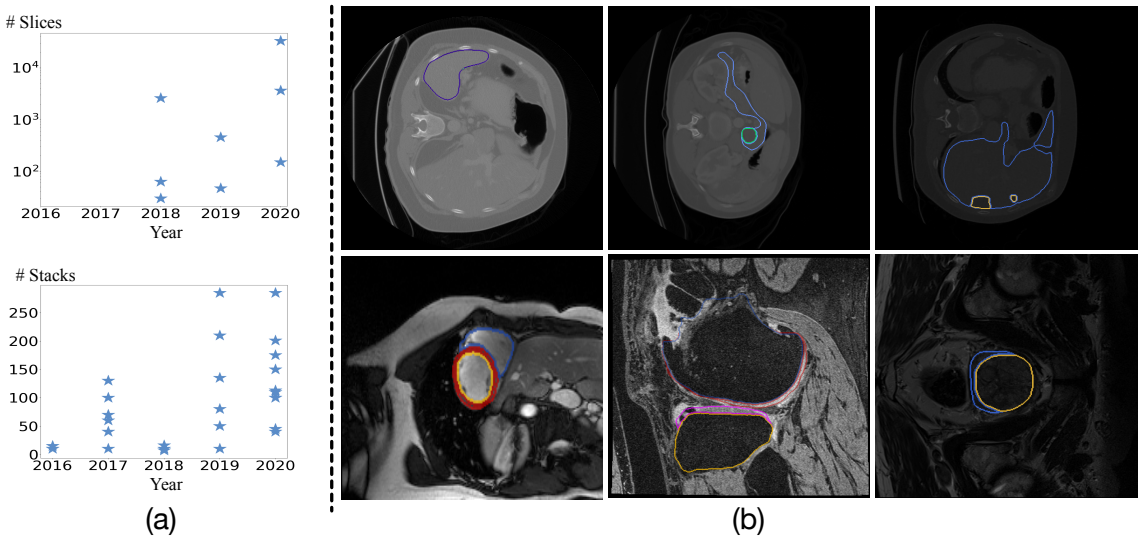

Figure 7.1. (a) The number of images for each medical image segmentation challenge every year since 2016 at MICCAI (top: 2D images; bottom: 3D stacks). (b) Diverse medical image and mask examples (left to right and top to bottom): spleen, pancreas & tumours, liver & tumours, cardiovascular structures, knee bones & cartilages, and prostate.

and free to use, we seek to answer the question: *Can we improve segmentation performance with limited training data by directly exploiting raw data information and representation learning?*

Recently, self-supervised learning (SSL) approaches, which initialize models by constructing and training surrogate tasks with unlabeled data, attracted much attention due to soaring performance on representation learning [35, 44, 47, 56, 80, 109, 112, 120] and downstream tasks [13, 21, 114, 138, 190, 194]. It was shown that the learned representation by *contrastive learning*, a variant of SSL, gradually approaches the effectiveness of representations learned through strong supervision, even under circumstances when only limited data or a small-scale dataset is available [23, 54]. However, three key factors of contrastive learning have not been well explored for medical segmentation tasks: (1) A medical image dataset is often insufficiently large due to the intrusive nature of some imaging techniques or expensive annotations (e.g.,

3D(+T) images), which suppresses self-supervised pre-training and hinders representation learning using a single dataset. (2) The contrastive strategy considers only congenetic image pairs generated by different transformations used in data augmentation, which suppresses the model from learning task-agnostic representations from heterogeneous data collected from different sources (see Fig. 7.1(b)). (3) Most studies focused on extracting high-level representations by pre-training the encoder while neglecting to learn low-level features explicitly and initialize the decoder, which hinders the performance of dense prediction tasks such as semantic segmentation.

To address these challenges, in this chapter, we propose a new *hierarchical self-supervised learning* (HSSL) framework to pre-train on heterogeneous unannotated data and obtain an initialization beneficial for training multiple downstream medical image segmentation tasks with limited annotations. First, we investigate available public challenge datasets on medical image segmentation and propose to aggregate a multi-domain (modalities, organs, or facilities) dataset. In this way, our collected dataset is considerably larger than a task-specific dataset and the pretext model is forced to learn task-agnostic knowledge (e.g., texture, intensity distribution, etc). Second, we construct pretext tasks at multiple abstraction levels to learn hierarchical features and explicitly force the model to learn richer semantic features for segmentation tasks on medical images. Specifically, our HSSL utilizes contrasting and classification strategies to supervise image-, task-, and group-level pretext tasks. We also extract multi-level features from the network encoding path to bridge the gap between low-level texture and high-level semantic representations. Third, we attach a lightweight decoder to the encoder and pre-train the encoder-decoder architecture to obtain a suitable initialization for downstream segmentation tasks.

We experiment on our aggregated dataset composed of eight medical image segmentation tasks and show that our HSSL is effective in utilizing multi-domain data to initialize model parameters for target tasks and achieves considerably better seg-

mentation, especially when only limited annotations are available.

7.2 Methodology

We discuss the necessity and feasibility of aggregating multi-domain image data and show how to construct such a dataset in Sect. 7.2.1, and then introduce our hierarchical self-supervised learning pretext tasks (shown in Fig. 7.2) in Sect. 7.2.2. After pre-training, we fine-tune the trained encoder-decoder network on downstream segmentation tasks with limited annotations.

7.2.1 Multi-Domain Data Aggregation

Necessity. As shown in Fig. 7.1(a), most publicly available medical image segmentation datasets are of relatively small sizes. Yet, recent progresses on contrastive learning empirically showed that training on a larger dataset often learns better representations and brings larger performance improvement in downstream tasks [23, 24, 54]. Similarly, a larger dataset is beneficial for supervised classification tasks and unsupervised image reconstruction tasks, because such a dataset tends to be more diverse and better cover the true image space distribution.

Feasibility. First, there are quite a few medical image dataset archives (e.g., TICA[1]) and public challenges (e.g., Grand Challenge[2]). Typical imaging modalities (CT, MRI, X-ray, etc) of multiple regions-of-interest (ROIs, organs, structures, etc) are covered. Second, common/similar textures or intensity distributions are shared among different datasets (see Fig. 7.1(b)), and their raw images may cover the same physical regions (e.g., abdominal CT for the spleen dataset and liver dataset). Therefore, an aggregated multi-domain dataset can (1) enlarge the data size of a shared

[1] `https://www.cancerimagingarchive.net/`

[2] `https://grand-challenge.org/challenges/`

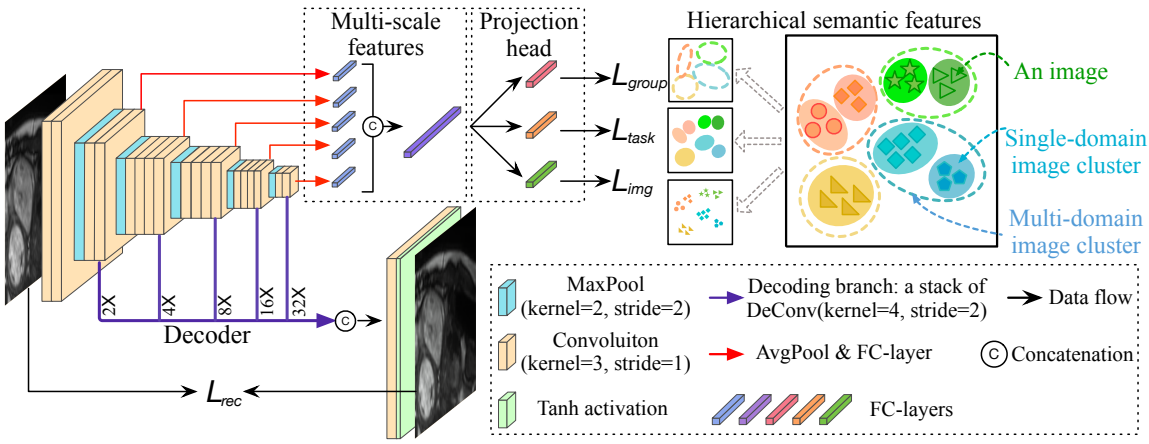

Figure 7.2. An overview of our proposed hierarchical self-supervised learning (HSSL) framework (best viewed in color). The backbone encoder builds a pyramid of multi-scale features from the input image, forming a rich latent vector. Then it is stratified to represent hierarchical semantic features of the aggregated multi-domain data, supervised by different pretext tasks in the hierarchy. Besides, an auxiliary reconstruction pretext task helps initialize the decoder.

image space and (2) force the model to distinguish different contents from the raw images. In this way, task-agnostic knowledge is extracted.

Dataset Aggregation. To ensure the effectiveness of multi-domain data aggregation, three principles should be considered. (1) Representativeness: The datasets considered for aggregation should cover a moderate range of medical imaging techniques/modalities. (2) Relevance: The datasets considered should not drastically differ in content/appearance. Otherwise, it is easy for the model to distinguish them and a less common feature space is shared among them. (3) Diversity: The datasets considered should benefit a range of applications. In this work, we focus on CT and MRI of various ROIs (i.e., heart, liver, prostate, pancreas, knee, and spleen). The details of aggregated dataset are shown in Table 7.1.

7.2.2 Hierarchical Self-Supervised Learning (HSSL)

Having aggregated multiple datasets, $\mathcal{D} = \{D_1, D_2, \ldots, D_N\}$, where D_i is a dataset for a certain segmentation task. A straightforward method to use $\mathcal{D}$ is to directly extend some known pretext tasks (e.g., SimCLR [23]) and conduct joint pre-training. However, such pretext tasks only explicitly force the model to learn a global representation and are not tailored for the target segmentation tasks. Hence, taking imaging techniques and prior knowledge (e.g., appearance, ROIs) into account, we propose to extract richer semantic features from hierarchical abstract levels and devise the network for target segmentation tasks.

We formulate three hierarchical levels (see Fig. 7.3). (1) *Image-level*: Each image I is a learning subject; we want to extract distinguishable features of I w.r.t. another image, regardless of which dataset it originally comes from or what ROIs it contains. Specifically, we follow the state-of-the-art SimCLR [23] and build positive and negative pairs with various data augmentations. (2) *Task-level*: Each D_i is originally imaged for a specific purpose (e.g., CT for spleen). Generally, images belonging to a same dataset are similar inherently. As shown in Fig. 7.4, images of different modalities and ROIs are easier to distinguish. For abdominal CTs of spleen and liver, although the images are similar, their contents are different. Thus, each task's dataset forms a single domain of certain ROI and image types. (3) *Group-level*: Despite the differences among different segmentation tasks, the contents of images may show a different degree of similarity. For example, in the physical space, liver CT scans have overlapping with spleen CT scans; cardiac MRIs scanned for different purposes (e.g., diverse cardiovascular structures) contain the same ROI (i.e., the heart) regardless of the image size and contrast. In this way, we categorize multiple domains of images into a group, which forms a multi-domain cluster in the feature space. Assigned with both task-level and group-level labels, each image constitutes a tuple (I, y^t, y^g), where t and g are task-class and group-class, respectively.

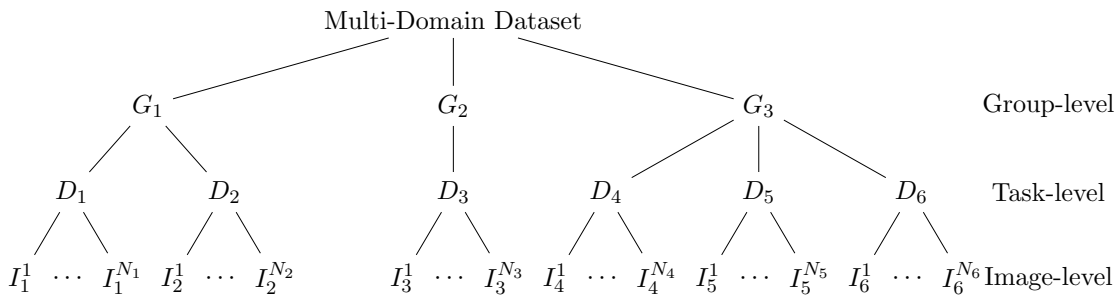

Figure 7.3. An example showing the hierarchical structure of a multi-domain dataset. Each chosen dataset/task D_i forms a domain consisting of a set of images $\{I_i^k\}_{k=1}^{N_i}$, where N_i is the total number of images in D_i. Multiple tasks form a multi-domain cluster called a *group* (G_j).

Further, to better aggregate low- and high-level features from the encoder, we compress multi-scale feature vectors from the feature pyramid and concatenate them together, and then attach three different projection heads to automatically extract hierarchical representations (see Fig. 7.2).

Image-Level Loss. Given an input image I, the contrastive loss is formulated as:

$$l(\tilde{I}, \hat{I}) = -\log \frac{e^{sim(\tilde{z},\hat{z})/\tau}}{e^{sim(\tilde{z},\hat{z})/\tau} + \sum_{\bar{I}\in\Lambda^-} e^{sim(\tilde{z},\bar{z})/\tau}}, \tag{7.1}$$

where $\tilde{z} = P_l(E(\tilde{I}))$, $\hat{z} = P_l(E(\hat{I}))$, $\bar{z} = P_l(E(\bar{I}))$, $P_l(\cdot)$ is the image-level projection head, $E(\cdot)$ is the encoder, $\tilde{I}$ and $\hat{I}$ are two different augmentations of image I (i.e., $\tilde{I} = \tilde{t}(I)$ and $\hat{I} = \hat{t}(I)$), $\bar{I} \in \Lambda^-$ consisting of all negative samples of I, and $\tilde{t}$, $\hat{t} \in \mathcal{T}$ are two augmentations. The augmentations $\mathcal{T}$ include random cropping, resizing, blurring, and adding noise. $sim(\cdot,\cdot)$ is cosine similarity, and τ is a temperature scaling parameter. Given our multi-domain dataset $\mathcal{D}$, the image-level loss is defined

as:

$$\mathcal{L}_{img} = \frac{1}{|\Lambda^+|} \sum_{\forall(\tilde{I},\hat{I})\in\Lambda^+} [l(\tilde{I},\hat{I}) + l(\hat{I},\tilde{I})], \tag{7.2}$$

where Λ^+ is a set of all similar pairs sampled from $\mathcal{D}$. In implementation, positive and negative pairs are constructed in each mini-batch.

Task-Level Loss & Group-Level Loss. Given task-class and group-class, we formulate task- and group-level pretext tasks as classification tasks. The training objectives are:

$$\mathcal{L}_{task} = -\sum_{c=1}^{T} y_c^t \log(p_c^t);\ \mathcal{L}_{group} = -\sum_{c=1}^{G} y_c^g \log(p_c^g), \tag{7.3}$$

where $p_c^t = P_t(E(I))$, $p_c^g = P_g(E(I))$, $P_t(\cdot)$ (or $P_g(\cdot)$) is the task-level (or group-level) projection head, $E(\cdot)$ is the encoder, y_c^t (or y_c^g) is the task-class (or group-class) of input image I, and T (or G) is the number of classes of tasks (or groups).

After pre-training the model, we extract features from the hierarchical abstract levels, denoted by F_{img}, F_{task}, and F_{group}, and project them to 2D planes using t-SNE [100]. For comparison, we build an auto-encoder based on pre-trained VGG-19 [133] and extract the feature vector of its deepest FC-layer as the latent code of the input image. As shown in Fig. 7.4, the hierarchical layout is as expected, implying that our model is capable of extracting richer semantic features at different abstract levels of the input images.

Decoder Initialization. A decoder is also indispensable for semantic segmentation tasks. To find a good initialization for decoder, we devise a multi-scale decoder and combine it with the encoder. We formulate the pretext task as a reconstruction

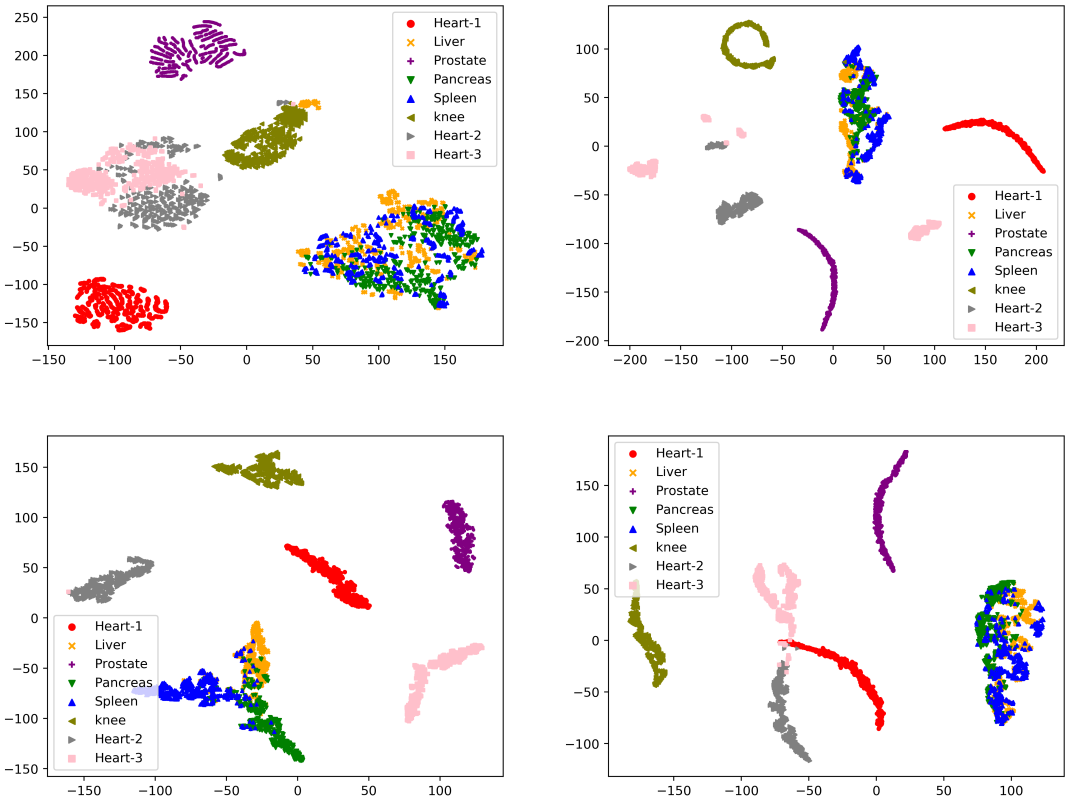

Figure 7.4. Extracted features after t-SNE projection [100] (best viewed in color). Top-left: F_{VGG-19}; top-right: F_{image}; bottom-left: F_{task} (forming single-domain task-level clusters as in Table 7.1); bottom-right: F_{group} (forming multi-domain group-level clusters as in Table 7.1).

task. The loss is defined as:

$$\mathcal{L}_{rec} = \frac{1}{|\mathcal{D}|} \sum_{I \in \mathcal{D}} ||S(E(I)) - I||_2, \tag{7.4}$$

where $E(\cdot)$ is the encoder, $S(\cdot)$ is the decoder, and $||\cdot||_2$ is the L_2 norm.

In summary, we combine the hierarchical self-supervised losses at all the levels and the auxiliary reconstruction loss to jointly optimize the model:

$$\mathcal{L}_{total} = \lambda_1 \mathcal{L}_{img} + \lambda_2 \mathcal{L}_{task} + \lambda_3 \mathcal{L}_{group} + \lambda_4 \mathcal{L}_{rec}, \tag{7.5}$$

Algorithm 4: The HSSL Training Algorithm

Input: Batch size N, temperature τ, encoder $E(\cdot)$, image-, task-, and group-level projection heads $P_l(\cdot)$, $P_t(\cdot)$, and $P_g(\cdot)$, and decoder $S(\cdot)$;

`/* Pre-training stage */`

while *stopping condition not met* **do**
  sample mini-batch of images $\{I_i, y_i^t, y_i^g\}_{i=1}^N$;
  for $i = 1$ *to* N **do**
    Sample two augmentations $t \in \mathcal{T}$ and $t' \in \mathcal{T}$;
    Extract hierarchical representations using encoder $E(\cdot)$ and projection heads $P_l(\cdot)$, $P_t(\cdot)$, $P_g(\cdot)$;
    Compute hierarchical loss;
    Reconstruct image using $E(\cdot)$ and $S(\cdot)$;
    Compute reconstruction loss;
  Update networks $E(\cdot)$, $P_l(\cdot)$, $P_t(\cdot)$, $P_g(\cdot)$, and $S(\cdot)$ by minimizing Eq. (7.5);

`/* Segmentation stage */`

while *stopping condition not met* **do**
  sample mini-batch of image-mask pairs $\{I_i, M_i\}_{i=1}^N$;
  for $i = 1$ *to* N **do**
    Generate segmentation mask: $\tilde{M}_i = S(E(I_i))$;
  Update networks $E(\cdot)$ and $S(\cdot)$ by minimizing pixel-wise cross-entropy loss $\mathcal{L}_{mce}(\tilde{M}_i, M_i)$;

return the encoder $E(\cdot)$ and decoder $S(\cdot)$.

where $\lambda_i (i = 1, 2, 3, 4)$ are the weights to balance loss terms. For simplicity, we let $\lambda_1 = \lambda_2 = \lambda_3 = 1/3, \lambda_4 = 50$.

Segmentation. Once trained, the encoder-decoder can be fine-tuned for downstream multi-domain segmentation tasks. For a give task D_i, we acquire some annotations (e.g., 10%) and optimize the network with cross-entropy loss.

We summarize the training procedure in Algorithm 4.

7.3 Experiments

Datasets. We employ multiple MRI and CT image sets from 8 different data sources with distribution shift (as summarized in Table 7.1). Task-1: the LASC dataset was originally released in STACOM and MICCAI 2013 [140], and includes

TABLE 7.1

DETAILS OF OUR AGGREGATED MULTI-DOMAIN DATASET[3]

Task ID	Group ID	ROI-Type	Segmentation Class	# of Slices	Source
1	1	Heart-MRI	1: Left atrium	1262	LASC [140]
2	2	Liver-CT	1: Liver, 2: Tumor	4342	LiTS [8]
3	3	Prostate-MRI	1: Central gland, 2: Peripheral zone	483	MSD [135]
4	2	Pancreas-CT	1: Pancreas, 2: Tumor	8607	MSD [135]
5	2	Spleen-CT	1: Spleen	1466	MSD [135]
6	4	Knee-MRI	1: Femur bone, 2: Tibia bone, 3: Femur cartilage, 4: Tibia cartilage	8187	Knee [165]
7	1	Heart-MRI	1: Left ventricle, 2: Right ventricle, 3: Myocardium	1891	ACDC [7]
8	1	Heart-MRI		3120	M&Ms [1]

20 MRI images covering the entire heart with expert annotations for left atrium segmentation. Task-2: the LiTS dataset was hosted in ISBI and MICCAI 2017 challenge [8], and consists of 131 CT scans with annotations by radiologists for two categories: liver and tumours. Task-3: the prostate dataset was released in MICCAI 2018 medical segmentation decathlon (MSD) challenge [95, 135], and includes 32 expert-annotated T2-weighted MRIs of the prostate region for peripheral zone and central gland segmentation. Task-4: the pancreas dataset was hosted in MICCAI 2018 MSD challenge [135], and is composed of 281 portal venous phase CT scans with expert annotations for segmenting pancreatic parenchyma and pancreatic mass (cyst or tumour). Task-5: the spleen dataset was provided by [134] and released in MICCAI 2018 MSD challenge [135], and consists of 41 CT scans with expert

[3]Details of data obtained from public sources. The left two columns: their task-classes and group-classes based on our multi-domain data aggregation principles.

annotations for segmenting spleen. Task-6: a public knee dataset [165], composing of 206 MRIs with expert annotations for four structures: femur bone & cartilage, tibia bone & cartilage. Task-7: the ACDC dataset was released in MICCAI 2017 [7], and consists of 100 short-axis cardiac cine-MRI images with left ventricle, myocardium, and right ventricle manually annotated. Task-8: the M&Ms dataset was hosted in MICCAI 2020 challenge [1], and consists of 150 caridac MRIs.

Pre-processing. First, we extract from NIfTI-formatted images and conduct min-max normalization on each 3D stack. To maintain the annotation accuracy, we do not resize 3D stacks. A typical 2D slice is of size $300^2 \sim 512^2$. Second, due to the spatial resolution differences among the tasks along the z-axis, we sample 2D slices from different tasks at different ratios to maintain balance of re-organized data.

Experimental Setup. Each dataset is split into X_{tr}, X_{val}, and X_{te} in the ratios of $7:1:2$. We use all images for the pre-training stage and then fine-tune the pre-trained network with labeled images from X_{tr}. We experiment with different amounts of training data X_{tr}^s, where $s \in \{5\%, 10\%, 100\%\}$ denotes the ratio of $\frac{X_{tr}^s}{X_{tr}}$.

Post-processing & Evaluation. Having obtained the probability maps of different classes, we conduct max-voting for each pixel, and generate the final segmentation map. For Tasks 5$\sim$8, we also remove the small connected components. The segmentation accuracy is measured by the Dice-Sørensen Coefficient (DSC): $\frac{2|\mathcal{Y} \cap \hat{\mathcal{Y}}|}{|\mathcal{Y}|+|\hat{\mathcal{Y}}|}$, where $\mathcal{Y}$ is the ground truth and $\hat{\mathcal{Y}}$ is the predicted segmentation.

Implementation Details. For self-supervised pre-training, we use ResNet-34 [52] as the base encoder network, two FC-layers at each scale of the encoding path to extract multi-scale feature vectors, three 2-layer MLP projection heads at the hierarchical levels to obtain a 512-dimensional latent space, and $K-1$ DeConv layers at scale-K of the encoder followed by two Conv layers in the decoder to reconstruct images. The model is optimized using Adam [74] with linear learning rate scaling and weight decay $= 10^{-6}$ for $1k$ epochs (initial learning rate: $3e^{-4}$).

For segmentation tasks, we optimize the network using Adam with "poly" learning rate policy with the power variable = 0.9 and weight decay = 10^{-10} (initial learning rate: $5e^{-4}$) for $10k$ epochs. Random cropping and rotation are applied for augmentation. In all the experiments, the mini-batch size is 30 and input image size is 192×192. If not initialized by pre-trained parameters, other models are initialized with a Gaussian distribution ($\mu = 0$, $\sigma = 0.01$).

7.3.1 Main Experimental Results

Our approach contributes to the "pre-training + fine-tuning" diagram in two aspects: hierarchical self-supervised learning (HSSL) and multi-domain data aggregation. To validate the effectiveness of our framework, we first compare with state-of-the-art pretext task training methods on downstream segmentation tasks. Then we demonstrate the benefits of joint training on our aggregated multi-domain dataset.

Effectiveness of HSSL. We compare with state-of-the-art pretext task training methods [23, 44, 54, 120] on seven downstream segmentation tasks, and summarize quantitative results of three representative tasks in Table 7.2.

First, our method surpasses training from scratch (TFS) substantially, showing the effectiveness of better model initialization. Second, our approach outperforms known SSL-based methods in almost all the settings, indicating a better capability to extract features for segmentation tasks. Third, our HSSL can more effectively boost performance, especially when extremely limited annotations are available (e.g., +18.60% with 5% annotated data on Task-3), implying potential applicability when abundant images are acquired but few are labeled. Fourth, with more annotations, our method can further improve accuracy and achieve state-of-the-art performance (e.g., +1.84% to +2.57% with 100% annotated data over TFS). All these promising results show that our HSSL is capable of learning richer semantic information from unannotated data for segmentation tasks. Qualitative results are given in Fig. 7.6.

TABLE 7.2

QUANTITATIVE RESULTS ON TASK-1, TASK-3, AND TASK-5[4]

Task-#	Anno.	TFS	Rotation [44]	In-painting [120]	MoCo [54]	SimCLR [23]	HSSL (Ours)
1	5%	71.56	72.83	65.40	75.97	73.45	**81.46**
	10%	79.64	**82.31**	81.99	79.07	81.19	81.79
	100%	85.81	87.43	86.56	87.19	87.06	**87.65**
3	5%	20.65; 47.56 (34.10)	28.74; 67.11 (47.93)	20.13; 52.16 (36.14)	29.55; 64.95 (47.25)	39.67; 68.35 (**54.01**)	35.30; 70.08 (52.69)
	10%	40.10; 66.95 (53.53)	44.15; 70.63 (57.39)	33.81;67.14 (50.48)	40.16; 67.98 (54.07)	46.04; 70.39 (58.22)	46.97; 72.21 (**59.59**)
	100%	50.19; 76.74 (63.47)	55.21; 78.21 (66.71)	53.19 77.97 (65.59)	56.31; 77.59 (66.95)	56.53; 77.86 (67.20)	58.80; 78.35 (**68.58**)
5	5%	48.75	56.74	47.86	54.91	63.40	**67.35**
	10%	67.44	74.68	71.30	68.22	78.25	**80.95**
	100%	85.88	86.96	85.96	85.75	87.76	**88.45**

[4]Task-1: heart, Task-3: prostate, Task-5: spleen. Dice scores for each class are listed and the average scores are in parentheses. TFS: training from scratch. Same network architecture is used for fair comparison in all the experiments. Our HSSL achieves the best performance in most settings (in bold).

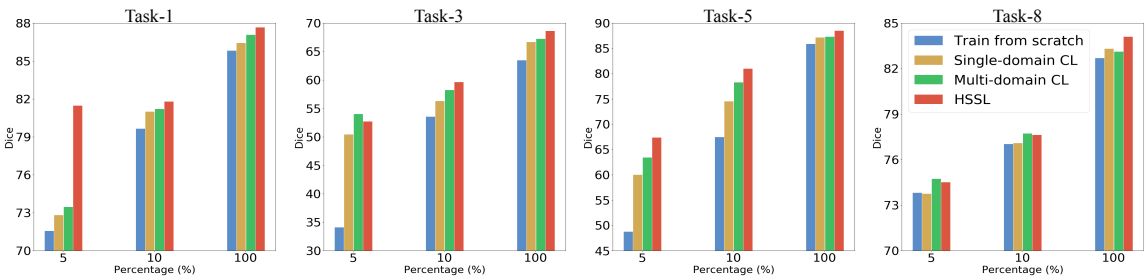

Figure 7.5. Quantitative results of TFS *vs.* single-domain CL *vs.* multi-domain CL *vs.* HSSL for Task-1/-3/-5/-8 with different ratios (5%, 10%, 100%) of labeled data, respectively.

Effectiveness of Multi-Domain Data Aggregation. We conduct pre-training on single-domain and aggregated multi-domain data, and compare the segmentation performances. "Single-domain CL" and "Multi-domain CL" are all based on the state-of-the-art SimCLR [23].

As sketched in Fig. 7.5, one can see that multi-domain data aggregation (i.e., multi-domain CL and HSSL) consistently outperforms single-domain pre-training (sometimes significantly). For instance, with 10% annotated data on Task-5, multi-domain CL and HSSL outperform single-domain CL by 3.74% and 6.41%, respectively. This suggests that more data varieties can provide complementary information and help improve the overall performance.

Meanwhile, we also observe that, in a few occasions, multi-domain CL yields higher performance than HSSL (e.g., Task-8 with 5% annotated data). A possible reason is: Task-8 is for multi-class segmentation that is inherently more difficult than Task-1. Different classes may interfere with one another and the average Dice score can be lower. Especially, when the differences of object sizes and segmentation difficulties are large between two classes, Dice scores of the harder class could influence the average score greatly. Besides, we notice that with more training data, such situations become less severe and HSSL is better.

TABLE 7.3

QUANTITATIVE RESULTS OF DIFFERENT MODELS ON THREE TASKS WITH DIFFERENT AMOUNTS OF ANNOTATED DATA[6]

Method	Param. (M)	Task-1			Task-3			Task-5		
		5%	10%	50%	5%	10%	50%	5%	10%	50%
UNet [128]	39.40	75.43	77.72	86.75	38.19	49.44	62.61	54.71	62.81	81.48
UNet3+ [62]	26.97	78.48	78.81	**87.52**	42.06	50.94	63.50	60.05	64.83	82.74
HSSL (Ours)	22.07	**81.46**	**81.79**	87.02	**52.69**	**59.59**	**66.64**	**67.35**	**80.95**	**85.86**

7.3.2 Comparison with State-of-the-Art Models

To thoroughly evaluate our method, we compare with state-of-the-art models for medical image segmentation tasks (some in challenges[5]). As shown in Table 7.3, our method outperforms the state-of-the-art UNet3+ [62] significantly in almost all the settings. Further, with limited annotated data (e.g., 5%), our method bridges the performance gap significantly with respect to the results obtained by training with more annotated data. Also, our model is most lightweight, and thus efficient as well. Qualitative results are given in Fig. 7.6(c).

7.3.3 Ablation Study

As shown in Table 7.4, each hierarchical loss contributes to representation learning and leads to segmentation improvement.

[5]Note that it is unfair to directly compare our results with the reported results on the leader boards, because: (1) we re-organize the data and split the training data into training, validation, and test sets; (2) our method is based on 2D models; (3) our focus is self-supervised training and we do not design specific network architectures or loss functions.

[6]Task-1: heart, Task-3: prostate, Task-5: spleen. Models are fine-tuned with 5%, 10%, and 50% annotated data, respectively. Our HSSL achieves the best performance in most the settings (highest scores in bold).

TABLE 7.4

ABLATION STUDY OF LOSS FUNCTIONS[7]

L_{rec}	L_{img}	L_{task}	L_{group}	Task-1	Task-5
✓				65.71	46.13
	✓			73.45	63.40
✓	✓			77.26	65.01
✓	✓	✓		79.32	66.67
✓	✓	✓	✓	81.46	67.35

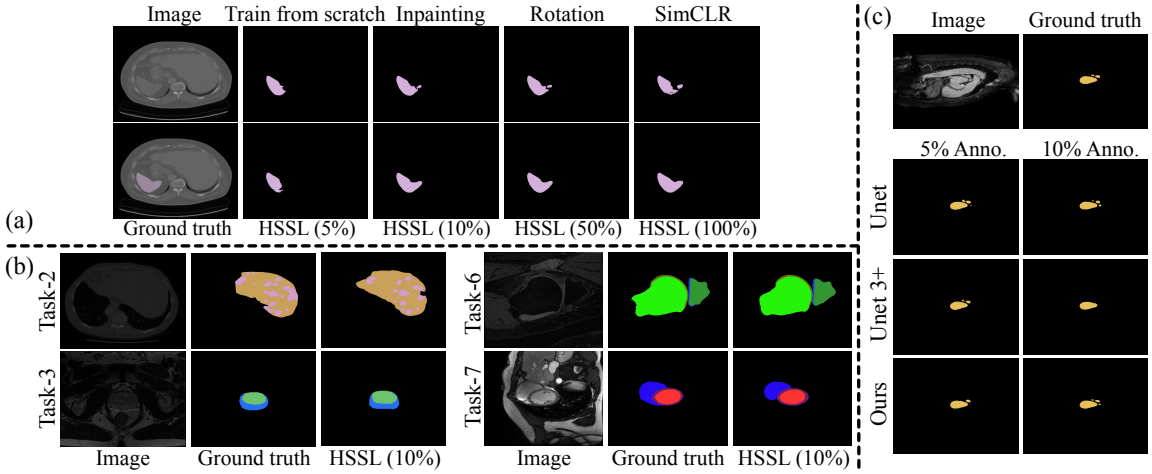

Figure 7.6. Qualitative comparison (best viewed in color). (a) Top: results of different methods on Task-5 (10% annotated data); Bottom: results of our HSSL with different ratios of annotated data. (b) Results of Task-2/-3/-6/-7 (10% annotated data). (c) Results of different models on Task-1 trained with 5% and 10% annotated data, respectively.

[7]Task-1: heart, Task-5: spleen. Models are fine-tuned with 5% annotated data.

7.4 Conclusions

In this chapter, we proposed *hierarchical self-supervised learning*, a novel self-supervised framework that learns hierarchical features from aggregated multi-domain medical image data. Contrastive loss and classification loss at the image-, task-, and group-levels explicitly supervise pre-training, which further distills multi-level semantic features for downstream segmentation tasks. Moreover, multi-level features are aggregated to keep both low-level texture and high-level semantic features for better representation learning. A decoder is attached to form an auxiliary reconstruction task to obtain an effectual initialization. Extensive experiments demonstrate that joint training on multi-domain data by our method outperforms training from scratch and conventional pre-training strategies, especially in limited annotation scenarios.

CHAPTER 8

CONCLUSIONS AND FUTURE WORKS

8.1 Summary of Main Results

In this dissertation, we have presented new deep learning algorithms for achieving efficient and robust biomedical image segmentation. Firstly, we designed new methods to maximize the utilization of available annotations and boost segmentation performance for 3D data. Specifically, we developed a novel deep heterogeneous feature aggregation network where heterogeneous contextual information is extracted in parallel asymmetrical encoding paths and fused in a content-aware manner (Chapter 2 [180]). Furthermore, we introduced a new ensemble learning framework that combines the merits of 2D and 3D fully convolutional networks (FCNs) by introducing stacking [151] into the deep learning regime. We devised algorithms to train the meta-learner and reduce the risk of over-fitting (Chapter 4 [181]). Secondly, we proposed methods to save manual annotation effort but maintain similar performance by selecting the most valuable samples for annotation and utilizing unlabeled data for training. Inspired by the inherency of active learning based methods [162, 188], we directly selected the representatives from unlabeled data in one shot (Chapter 3 [179]). Furthermore, we extended our method to 3D data and introduced a new algorithm to bridge the performance gap of sparse annotation with respect to full annotation (Chapter 5 [183]). Moreover, we proposed a new method to measure the confidence of generated pseudo labels on unseen images and utilize the uncertainty to guide the self-training of models (Chapter 6 [182]). Thirdly, we proposed extracting knowl-

edge from heterogeneous images without manual annotation and demonstrated better model generalizability in large-scale applications. We aggregated a multi-domain dataset and pretrained model parameters by designing sophisticated pretext tasks to force the model to extract rich task-agnostic knowledge from multiple scales and hierarchical abstract levels (Chapter 7 [184]). Our pretrained model can be adapted to various downstream tasks by utilizing a limited amount of labeled data and can achieve remarkable performance.

8.2 Suggested Future Works

Data is the fuel of deep learning technologies and constitutes the core challenge of medical image analysis. To develop annotation-efficient and robust deep learning techniques that can generalize to different medical tasks without requiring intensive manual annotation, we should focus on the following directions: (1) design better model architectures; (2) devise new annotation selection and suggestion methods; (3) enhance model generalizability; (4) develop better model training methods; (5) utilize prior knowledge of the data.

8.2.1 3D Neural Networks

In Chapter 2, we proposed a new 3D neural network architecture to exploit volumetric information from multiple geometric views; in Chapter 4, we proposed a new framework to combine 2D and 3D networks for 3D segmentation, which is a paradigm rather than a specific model. However, more effort is needed to extract richer contextual information for better segmentation performance and to reduce computational efforts for efficient training and inference.

With the development of deep learning, the attention mechanism [158] automatically discovers "where" and "what" to focus on image content for final prediction and achieves superior performance in various tasks [34, 43, 60, 105, 111, 129]. Recently,

the prevalent Transformer model [33, 144] in natural language processing (NLP) has also achieved tremendous success in computer vision tasks by modeling long-range dependencies in the data [11, 12, 36, 50, 63, 98, 141, 149, 157]. Although the transformer is flexible to scale to high parametric complexity, it suffers from the high computational cost of core self-attention operations (i.e., a quadratic increase with the number of patches), which hinders its application to most tasks involving high-resolution images. To overcome this issue, we should further unify the covolutional neural network and the transformer to combine their merits [11, 98, 153, 170]. Specifically, convolution performs better in the early stages [156] and is much cheaper in computation. Therefore, it can be used to model low-level and local contexts, upon which the transformer can build spatial relationships between local neighbourhoods.

Such a unified network may still encounter the bottleneck of computational cost when it comes to volumetric biomedical image segmentation tasks. There are three possible directions to reduce the overall amount of memory in both training and inference: (1) *Lightweight designs*. For example, special covolutions [59, 102, 118, 137, 167, 172] and efficient self-attention computations [6, 77, 139, 145, 148] can be utilized. By reducing the cost of unit operation, we can obtain higher-dimensional, deeper, or wider networks that can achieve better performance under certain computation constraints. (2) *Dynamic patch division*. There is much spatial redundancy in low-texture regions of biomedical images (e.g., the central region of relatively large organs, background regions). Image features of their low-resolution counterparts may maintain similar and sufficient representation [20]. Therefore, we can divide the image into patches/cubes of different granularities to reduce the number of patches/cubes. (3) *Difficulty-aware and region-aware routing*. Since the regions-of-interest (ROIs) have different sizes and levels of difficulty in segmentation, utilizing the same network for the whole image may not be necessary or efficient. For example, in abdominal CT, the liver is large and its boundaries are mostly clear, but the pancreas is small and

its boundaries are usually ambiguous; in knee MRI, the femur/tibia bone is large and obvious, but the femoral/tibial cartilage is small and thin. We can dedicate a powerful lightweight stem network to efficiently segmenting large and easy ROIs and an auxiliary sophisticated head network to focusing on small and difficult ROIs. One possible solution to seamlessly combining these two parts and sharing as much contextual information as possible is the dynamic network design [79, 88, 89, 92, 171].

8.2.2 Continual Representative Annotation

In Chapter 3, sufficient annotated data is hard to acquire for biomedical image segmentation, and we have proposed representative annotation to only select the most diverse and informative samples for manual annotation. However, in real-world applications, data is not always readily available but is collected continually (e.g., at different times or by different imaging protocols). Although the contents of biomedical images are similar (e.g., both knee MRIs, mouse micro-CTs), their appearances vary among each other. Moreover, new requirements of applications may occur, such as increasing targets of interest (e.g., doctors may require segmenting more organs). We need to deal with the *domain-shift* problem and enable deep learning models to adapt to processing new data (or tasks) while maintaining good segmentation performance on historical data (or tasks). In our future work, we aim to exploit the idea of *continual representative annotation* to select representative samples from continually arriving new data and adapt/fine-tune deep learning models efficiently.

Suppose we have a series of biomedical image segmentation datasets, $D_1, D_2, \ldots$, where D_i contains N_i images. For starters, we may utilize the representative annotation [179] (Chapter 3) to extract image features with a VAE, V_1, choose samples from D_1 and request experts to annotate them, and train an FCN, M_1, for segmentation. Then, we need to determine which data are not well represented by current models (V_1 and M_1). There are two choices: (1) Utilize V_1 to extract features of

images from D_1 and D_2 and compare the differences in clustering and distribution. We can find images that are not well covered by the previous labeled images from D_1 and select representatives from D_1 and D_2 for manual annotation. (2) Apply M_1 to D_2, estimate the uncertainty of images (similar to suggestive annotation [162]), and annotate the most uncertain samples. Next, we fine-tune V_1 and M_1 on both D_1 and D_2 and obtain updated models, denoted as V_2 and M_2. In this way, V_i learns better representation of datasets and M_i achieves good segmentation performance on the unified dataset. With more datasets arriving, the process is repeatedly and continuously conducted.

Two challenges remain: (1) Improve the efficiency and effectiveness of representative annotation. The overall dataset size is increasing continuously, so the frequency of updating models and the ratio of annotating new samples should be carefully determined. And we can keep a subset of the whole dataset and use it as a reference for feature matching of new and historical data. Moreover, we may consider combining the two aforementioned selection schemes to extract more comprehensive features and choose representatives for annotation. (2) Domain-shift problem. When updating models, we need to develop a method to accumulate, maintain, and utilize knowledge to learn new data (or tasks) without significant adverse effects on the learned data (or tasks). To this end, domain adaptation techniques [14, 91, 117] and continual learning [25, 40, 41, 103, 115, 178] can bridge the intra-domain gap and alleviate the catastrophic forgetting problem.

8.2.3 Push the Frontier of Annotation-Efficient Learning

As discussed in Chapter 3 and Chapter 5–7, since labeling is time-consuming and labor-intensive, and only experts are able to annotate well, and there exist both inter-user and intra-user labeling inconsistencies, biomedical images are associated with sparse and noisy labels. In our future work, we should push the frontiers of

annotation-efficient learning by efficiently making the most of both labeled and unlabeled images from the following three aspects:

Self-Supervised Learning. Although labeled data is limited, unlabeled data is unlimited and free to use. Once the representation is learned through proxy tasks, it can be fine tuned by using annotated data. First, in addition to prevalent transformation based proxy tasks (e.g., inpainting, rotation, reconstruction, denoising, etc. [44, 120, 138, 190, 194]), we can integrate the following tasks to extract semantic information as well. Compared with natural scene images, biomedical images have rich anatomical information. For example, the relative positions of the regions-of-interest are fixed in CT/MR/X-ray images. Strong supervision signals include cross-case similarity of patches in nearby locations and intra-case dissimilarity of patches at different anatomical locations. Second, the majority of previous work has focused on 2D images; there is no *3D* model that is pre-trained on *large-scale* datasets. We can extend our HSSL framework (in Chapter 7) in two directions: (1) The 3D scenario. Not only can we directly extend 2D proxy tasks to their 3D counterparts, but we also need to integrate 3D information. For example, we can utilize pixel/region consistency in consecutive slices to devise predictive proxy tasks. (2) The large-scale applications. As demonstrated in Chapter 7, a large and heterogeneous multi-domain dataset benefits model pre-training, but we only validate this hypothesis in eight MR and CT datasets. We should expand the range to more types of data, such as pathological data, X-ray, multi-modality MR/CT data, and more applications, such as more organs and diseases. We should conduct systematic experiments to examine how these datasets mutually help or hinder each other. Releasing the results of the large-scale empirical study and pre-trained model will advance the development of the whole biomedical image analysis community.

Semi-Supervised Learning. Pseudo-labels and retraining are two foundations of semi-supervised segmentation. In Chapter 4, we studied multi-view pseudo-labels

and ensemble learning to reduce the overfitting risk; in Chapter 6, we utilized uncertainty quantification to guide the retraining with highly confident pseudo-labels. Some important hyperparameters were determined empirically and were not fully understood by the community. In the future, more investigation is needed to find out what role pseudo-labels play in semi-supervised segmentation and which advanced retraining algorithms are more effective. First, regarding the pseudo-labels, we may focus on two directions: (1) How different amounts of pseudo-labels influence the final performance. Specifically, boundary areas are the most uncertain but only constitute a small portion of the whole image; the backgrounds and interior areas of ROIs occupy a larger proportion of the whole image. We need to figure out what ratio of these two types of pseudo-labels is optimal in the semi-supervised setting. (2) How different types of error in pseudo-labels influence the final performance. Specifically, deep learning models trained with limited annotated data may systematically generate two types of errors on unlabeled data: false-positive and false-negative pixels, so-called "noisy" labels. We need to understand how they bias the model retraining towards final convergence. Second, regarding the retraining methods, we need to focus on two principles: consistency and uncertainty. Consistency represents the differences between unlabeled data compared with labeled data, and uncertainty is the referent of the quality of pseudo-labels. Specifically, we can quantify the quality of generated pseudo-labels and judiciously determine which pseudo-labels should be utilized for retraining and how to utilize them (based on answers to the above two directions in the first point).

Weakly-Supervised Learning. Although pixel-wise annotation is expensive, there are other cheaper options, such as point-wise and bounding-box-wise annotation. We can use mixed weak supervision to train the model. The challenges are two-fold: (1) We need to design a multi-task network to be compatible with mixed supervision signals. It may have a shared stem network and multiple prediction

heads. (2) We need to design a new training strategy to determine which object should be annotated with which format. Specifically, we can start with the random assignment of annotations to a small portion of objects. In addition to the patch-wise uncertainty and diversity based selection criteria (as in Chapter 3 and [162]), we propose to consider the prediction consistency of the multi-task network and make multi-level representativeness suggestions for annotation.

8.2.4 Human-in-the-Loop Medical Image Segmentation

Current deep learning models generate error predictions. This problem is especially severe under limited annotation scenarios and cannot be eliminated with semi-supervised methods. As discussed in Chapter 3, adding more human annotation to less represented regions can help segment more ROIs. But another type of error remains uncorrected because the model shows high confidence in these regions (i.e., false-positive predictions). Therefore, in future work, we aim to solve the *human-in-the-loop annotation and correction* problem. Two questions need to be answered: (1) *How can we identify such typical errors?* (2) *How can we force the deep learning model to correct them efficiently?*

First, we could design a new network that can generate error assessment for model output. Besides the main segmentation network (Main-Net), our network has a small parametric quality assessment module (QA-Net) attached that can learn to predict target losses of inputs. In training, Main-Net is supervised by ground truth annotation, and the auxiliary QA-Net is supervised by the segmentation loss (e.g., cross-entropy loss). In inference, Main-Net and QA-Net output a segmentation map S and a quality map Q, respectively. Next, both S and Q are provided to experts, and Q guides them to interpret S. In the clinic, they finally make a judgment on how well the segmentation network works and identify notable errors. To alleviate the burden of humans, we should devise algorithms to find representative patterns

for experts to choose from (e.g., by extending the representative selection algorithm in Chapter 3).

Second, after obtaining corrections from experts, we could use them to resume training the network. Related works on segmentation with human-machine interaction [147, 175] are mostly semi-automatic methods. It is time-consuming and tedious to correct all similar error regions. Instead of using corrections as ground truth directly, we propose designing an algorithm to propagate correctness information to more similar wrong predictions, so we can expand the training set for such errors. For example, we can use pattern matching to choose similar image patches, train a small network using corrected samples, and propagate the correction to all similar image patches. Then all of these samples are utilized to train the network (Main-Net + QA-Net). Lastly, we can repeat this process iteratively until satisfying results are obtained.

8.2.5 Model Generalizability

Multi-domain datasets are very common, such as images from different facilities and different imaging techniques. Transfer learning [26, 143], domain adaptation [14, 70, 113, 176, 195], and meta-learning [39, 42, 86, 87] methods have achieved significant progress in training a more robust and general model for such heterogeneous data. They either need to collect labeled multi-domain data in training or extra data from the target domain (and some associated annotations) for fine-tuning the model. But target domain data is not always readily available in practice. Hence, we propose to devise new methods to improve model generalizability with no/limited target domain data. Here are two possible directions we can explore:

Homogeneity Learning. We can collect publicly available datasets of a certain kind, such as abdominal CT images (as discussed in Chapter 7). Although they were scanned for different purposes originally (e.g., liver, pancreas, kidney, etc.), they have

shared content space and cover a large variety of domain shifts. We plan to learn homogeneous features by calibrating the feature space of the multi-domain dataset and making the model focus on semantic content rather than low-level and local appearance. Besides image-wise augmentation to mix image styles across multiple domains, we also need to conduct region-wise augmentation to mix region-specific styles guided by different partially annotated multi-domain datasets. When new target images arrive, we can transfer them to the homogeneous feature space and apply the model to them.

Test-Time Adaptation. Without unlabeled target images in training, on-the-fly test-time adaptation estimates more stable batch-norm statistics to achieve robust predictions (e.g., TENT [146]) and shows promising results on classification tasks. We aim to develop test-time adaptation methods for biomedical segmentation tasks. There are several potential issues: (1) Segmentation is a dense prediction task. The loss function in TENT is based on image-level entropy and does not consider the consistency of segmentation results. (2) A better algorithm to estimate accurate batch-norm statistics remains unexplored, such as the optimal batch size of test samples taken into account and the estimation timing and frequency.

8.2.6 Incorporating Prior Knowledge of Medical Images

Compared with natural scene images, biomedical images have more constraints, such as anatomy, topology, geometry, and statistics on object shapes/sizes/locations. Prior knowledge is not explicitly leveraged in deep learning models. Considering annotated data are limited, this extra information is particularly valuable. In Chapter 2, substructures in the human heart have highly similar textures, and many errors in the raw output probability map can be reduced if geometrical interaction and distance prior between different substructures are considered in the post-processing step. Therefore, if such prior knowledge can be imposed in network training, the network

can become more robust. In Chapter 6, the chondrocranium structure varies a lot in developing embryonic mice but has a similar coarse topological structure. After segmenting E14.5 mice, we will proceed to segment the chondrocranium of mice at five ages (i.e., E13.5 $\sim$ E17.5). The most promising method is to transfer the model/knowledge of E14.5 to other ages. Chondrocranial anatomy prior knowledge is the key to regularizing the coarse structure and can be utilized to select representatives for additional manual annotation. Moreover, our ultimate goal is to analyze the relationship between bone and cartilage in development. The bone appearance is very different and the contrast with the background is even lower. We can use the cartilage as an anchor and prior knowledge to localize the possible regions of bones and further segment bones precisely.

BIBLIOGRAPHY

1. Multi-centre, multi-vendor & multi-disease cardiac image segmentation challenge (M&Ms). `https://www.ub.edu/mnms/`. Accessed: 2021-12-1.

2. M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. TensorFlow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, volume 16, pages 265–283, 2016.

3. E. Aljalbout, V. Golkov, Y. Siddiqui, and D. Cremers. Clustering with deep learning: Taxonomy and new methods. *arXiv preprint arXiv:1801.07648*, 2018.

4. F. Ambellan, A. Tack, M. Ehlke, and S. Zachow. Automated segmentation of knee bone and cartilage combining statistical shape knowledge and convolutional neural networks: Data from the osteoarthritis initiative. *Medical Image Analysis*, 52:109–118, 2019.

5. A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei. What's the point: Semantic segmentation with point supervision. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 549–565, 2016.

6. I. Beltagy, M. E. Peters, and A. Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

7. O. Bernard, A. Lalande, C. Zotti, et al. Deep learning techniques for automatic MRI cardiac multi-structures segmentation and diagnosis: Is the problem solved? *IEEE Transactions on Medical Imaging*, 37(11):2514–2525, 2018.

8. P. Bilic, P. F. Christ, E. Vorontsov, G. Chlebus, H. Chen, Q. Dou, C.-W. Fu, X. Han, P.-A. Heng, J. Hesser, et al. The liver tumor segmentation benchmark (LiTS). *arXiv preprint arXiv:1901.04056*, 2019.

9. A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT)*, pages 92–100, 1998.

10. J. F. Brinkley, S. Fisher, M. P. Harris, G. Holmes, J. E. Hooper, E. W. Jabs, K. L. Jones, C. Kesselman, O. D. Klein, R. L. Maas, et al. The facebase consortium: a comprehensive resource for craniofacial researchers. *Development*, 143(14):2677–2688, 2016.

11. H. Cao, Y. Wang, J. Chen, D. Jiang, X. Zhang, Q. Tian, and M. Wang. Swin-Unet: Unet-like pure transformer for medical image segmentation. *arXiv preprint arXiv:2105.05537*, 2021.

12. N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision (ECCV)*, pages 213–229, 2020.

13. K. Chaitanya, E. Erdil, N. Karani, and E. Konukoglu. Contrastive learning of global and local features for medical image segmentation with limited annotations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

14. C. Chen, Q. Dou, H. Chen, J. Qin, and P.-A. Heng. Synergistic image and feature adaptation: Towards cross-modality domain adaptation for medical image segmentation. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, pages 865–872, 2019.

15. H. Chen, X. Qi, L. Yu, and P.-A. Heng. DCAN: Deep contour-aware networks for accurate gland segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2487–2496, 2016.

16. H. Chen, X. J. Qi, J. Z. Cheng, and P. A. Heng. Deep contextual networks for neuronal structure segmentation. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI)*, pages 1167–1173, 2016.

17. H. Chen, Q. Dou, L. Yu, J. Qin, and P.-A. Heng. VoxResNet: Deep voxelwise residual networks for brain segmentation from 3D MR images. *NeuroImage*, 170:446–455, 2018.

18. J. Chen, L. Yang, Y. Zhang, M. Alber, and D. Z. Chen. Combining fully convolutional and recurrent neural networks for 3D biomedical image segmentation. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, pages 3036–3044, 2016.

19. J. Chen, S. Banerjee, A. Grama, W. J. Scheirer, and D. Z. Chen. Neuron segmentation using deep complete bipartite networks. In *Proceedings of the 20th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 21–29, 2017.

20. J. Chen, X. Wang, Z. Guo, X. Zhang, and J. Sun. Dynamic region-aware convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8064–8073, 2021.

21. L. Chen, P. Bentley, K. Mori, K. Misawa, M. Fujiwara, and D. Rueckert. Self-supervised learning for medical image analysis using image context restoration. *Medical Image Analysis*, 58:101539, 2019.

22. T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016.

23. T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, pages 1597–1607. PMLR, 2020.

24. T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. E. Hinton. Big self-supervised models are strong semi-supervised learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

25. Z. Chen and B. Liu. Lifelong machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 12(3):1–207, 2018.

26. B. Cheng, M. Liu, D. Shen, Z. Li, D. Zhang, A. D. N. Initiative, et al. Multi-domain transfer learning for early diagnosis of alzheimer's disease. *Neuroinformatics*, 15(2):115–132, 2017.

27. V. Cheplygina, M. de Bruijne, and J. P. Pluim. Not-so-supervised: A survey of semi-supervised, multi-instance, and transfer learning in medical image analysis. *Medical Image Analysis*, 54:280–296, 2019.

28. Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger. 3D U-Net: Learning dense volumetric segmentation from sparse annotation. In *Proceedings of the 19th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 424–432, 2016.

29. D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, pages 2843–2851, 2012.

30. R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A matlab-like environment for machine learning. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS) Workshop*, 2011.

31. B. D. de Vos, F. F. Berendsen, M. A. Viergever, H. Sokooti, M. Staring, and I. Išgum. A deep learning framework for unsupervised affine and deformable image registration. *Medical image analysis*, 52:128–143, 2019.

32. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. L. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

33. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the*

2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), pages 4171–4186, 2019.

34. X. Ding, Y. Peng, C. Shen, and T. Zeng. Cab u-net: An end-to-end category attention boosting algorithm for segmentation. *Computerized Medical Imaging and Graphics*, 84:101764, 2020.

35. C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1422–1430, 2015.

36. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021.

37. Q. Dou, H. Chen, Y. Jin, L. Yu, J. Qin, and P.-A. Heng. 3D deeply supervised network for automatic liver segmentation from CT volumes. In *Proceedings of the 19th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 149–157, 2016.

38. 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 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 691–697, 2018.

39. Q. Dou, D. Coelho de Castro, K. Kamnitsas, and B. Glocker. Domain generalization via model-agnostic learning of semantic features. *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 32:6450–6461, 2019.

40. A. Douillard, Y. Chen, A. Dapogny, and M. Cord. Plop: Learning without forgetting for continual semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4040–4050, 2021.

41. X. Du, Z. Li, J.-s. Seo, F. Liu, and Y. Cao. Noise-based selection of robust inherited model for accurate continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 244–245, 2020.

42. C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70, pages 1126–1135, 2017.

43. J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3146–3154, 2019.

44. S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations (ICLR)*, 2018.

45. F. Gonda, D. Wei, T. Parag, and H. Pfister. Parallel separable 3D convolution for video and volumetric data understanding. In *British Machine Vision Conference (BMVC)*, page 42, 2018.

46. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, pages 2672–2680, 2014.

47. J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

48. I. H. Guldner, Q. Wang, L. Yang, S. M. Golomb, Z. Zhao, J. A. Lopez, A. Brunory, E. N. Howe, Y. Zhang, B. Palakurthi, et al. Cns-native myeloid cells drive immune suppression in the brain metastatic niche through cxcl10. *Cell*, 183(5):1234–1248, 2020.

49. Z. Guo, L. Zhang, L. Lu, M. Bagheri, R. M. Summers, M. Sonka, and J. Yao. Deep LOGISMOS: deep learning graph-based 3d segmentation of pancreatic tumors on ct scans. In *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*, pages 1230–1233, 2018.

50. A. Hatamizadeh, Y. Tang, V. Nath, D. Yang, A. Myronenko, B. Landman, H. Roth, and D. Xu. Unetr: Transformers for 3D medical image segmentation. *arXiv preprint arXiv:2103.10504*, 2021.

51. K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015.

52. K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

53. K. He, R. Girshick, and P. Dollár. Rethinking ImageNet pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4918–4927, 2019.

54. K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9729–9738, 2020.

55. G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS) Workshop*, 2015.

56. R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations (ICLR)*, 2019.

57. D. S. Hochbaum. Approximating covering and packing problems: Set cover, vertex cover, independent set, and related problems. In *Approximation Algorithms for NP-hard Problems*, pages 94–143. PWS Publishing Co., Boston, MA, USA, 1997.

58. L. Hou, A. Agarwal, D. Samaras, T. M. Kurc, R. R. Gupta, and J. H. Saltz. Robust histopathology image analysis: To label or to synthesize? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8533–8542, 2019.

59. A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

60. J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 7132–7141, 2018.

61. G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4700–4708, 2017.

62. H. Huang, L. Lin, R. Tong, H. Hu, Q. Zhang, Y. Iwamoto, X. Han, Y.-W. Chen, and J. Wu. UNet 3+: A full-scale connected UNet for medical image segmentation. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1055–1059, 2020.

63. Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, pages 603–612, 2019.

64. S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, volume 37, pages 448–456, 2015.

65. J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. Ball, K. Shpanskaya, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 590–597, 2019.

66. J. Islam and Y. Zhang. Early diagnosis of alzheimer's disease: A neuroimaging study with deep learning architectures. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops (CVPRW)*, pages 1881–1883, 2018.

67. S. D. Jain and K. Grauman. Active image segmentation propagation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2864–2873, 2016.

68. S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshop*, pages 1175–1183, 2017.

69. C. Ju, A. Bibaut, and M. van der Laan. The relative performance of ensemble methods with deep convolutional neural networks for image classification. *Journal of Applied Statistics*, 45(15):2800–2818, 2018.

70. K. Kamnitsas, C. Baumgartner, C. Ledig, V. Newcombe, J. Simpson, A. Kane, D. Menon, A. Nori, A. Criminisi, D. Rueckert, et al. Unsupervised domain adaptation in brain lesion segmentation with adversarial networks. In *Proceedings of the 25th International Conference on Information Processing in Medical Imaging (IPMI)*, pages 597–609, 2017.

71. A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5574–5584, 2017.

72. A. Kendall, Y. Gal, and R. Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7482–7491, 2018.

73. A. Khoreva, R. Benenson, J. Hosang, M. Hein, and B. Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 876–885, 2017.

74. D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

75. D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.

76. D. P. Kingma and M. Welling. Auto-encoding variational Bayes. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*, 2014.

77. N. Kitaev, Ł. Kaiser, and A. Levskaya. Reformer: The efficient transformer. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020.

78. F. Kordon, P. Fischer, M. Privalov, B. Swartman, M. Schnetzke, J. Franke, R. Lasowski, A. Maier, and H. Kunze. Multi-task localization and segmentation for X-ray guided planning in knee surgery. In *Proceedings of the 22nd International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pages 622–630, 2019.

79. A. Kouris, S. I. Venieris, S. Laskaridis, and N. D. Lane. Multi-exit semantic segmentation networks. *arXiv preprint arXiv:2106.03527*, 2021.

80. G. Larsson, M. Maire, and G. Shakhnarovich. Learning representations for automatic colorization. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 577–593, 2016.

81. Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

82. C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. In *Artificial Intelligence and Statistics*, pages 562–570, 2015.

83. D.-H. Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, International Conference on Machine Learning (ICML)*, volume 3, page 2, 2013.

84. K. Lee, A. Zlateski, V. Ashwin, and H. S. Seung. Recursive training of 2D-3D convolutional networks for neuronal boundary prediction. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, pages 3573–3581, 2015.

85. K. M. Lesciotto, S. M. Motch Perrine, M. Kawasaki, T. Stecko, T. M. Ryan, K. Kawasaki, and J. T. Richtsmeier. Phosphotungstic acid-enhanced microCT: Optimized protocols for embryonic and early postnatal mice. *Developmental Dynamics*, 249:573–585, 2020. URL `https://doi.org/10.1002/dvdy.136`.

86. D. Li, Y. Yang, Y.-Z. Song, and T. M. Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

87. D. Li, J. Zhang, Y. Yang, C. Liu, Y.-Z. Song, and T. M. Hospedales. Episodic training for domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1446–1455, 2019.

88. F. Li, G. Li, X. He, and J. Cheng. Dynamic dual gating neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5330–5339, 2021.

89. X. Li, Z. Liu, P. Luo, C. Change Loy, and X. Tang. Not all pixels are equal: Difficulty-aware semantic segmentation via deep layer cascade. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 3193–3202, 2017.

90. X. Li, H. Chen, X. Qi, Q. Dou, C.-W. Fu, and P.-A. Heng. H-DenseUNet: Hybrid densely connected UNet for liver and tumor segmentation from CT volumes. *IEEE Transactions on Medical Imaging*, 37(12):2663–2674, 2018.

91. Y. Li, L. Yuan, and N. Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6936–6945, 2019.

92. Y. Li, L. Song, Y. Chen, Z. Li, X. Zhang, X. Wang, and J. Sun. Learning dynamic routing for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8553–8562, 2020.

93. P. Liang, J. Chen, H. Zheng, L. Yang, Y. Zhang, and D. Z. Chen. Cascade decoder: A universal decoding method for biomedical image segmentation. In *Proceedings of the 16th IEEE International Symposium on Biomedical Imaging (ISBI)*, pages 339–342, 2019.

94. D. Lin, J. Dai, J. Jia, K. He, and J. Sun. ScribbleSup: Scribble-supervised convolutional networks for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3159–3167, 2016.

95. G. Litjens, O. Debats, W. van de Ven, N. Karssemeijer, and H. Huisman. A pattern recognition approach to zonal segmentation of the prostate on MRI. In *Proceedings of the 15th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 413–420, 2012.

96. S. Liu, D. Xu, S. K. Zhou, O. Pauly, S. Grbic, T. Mertelmeier, J. Wicklein, A. Jerebko, W. Cai, and D. Comaniciu. 3D anisotropic hybrid network: Transferring convolutional features from 2D images to 3D anisotropic volumes. In *Proceedings of the 21st International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 851–858, 2018.

97. S. Liu, D. Xu, S. K. Zhou, S. Grbic, W. Cai, and D. Comaniciu. Anisotropic hybrid network for cross-dimension transferable feature learning in 3D medical images. In *Deep Learning and Convolutional Neural Networks for Medical Imaging and Clinical Informatics*, Advances in Computer Vision and Pattern Recognition, pages 199–216. Springer, 2019.

98. Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.

99. J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.

100. L. v. d. Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11):2579–2605, 2008.

101. A. Madani, M. Moradi, A. Karargyris, and T. Syeda-Mahmood. Semi-supervised learning with generative adversarial networks for chest X-ray classification with ability of data domain adaptation. In *Proceedings of the 16th IEEE International Symposium on Biomedical Imaging (ISBI)*, pages 1038–1042, 2018.

102. S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, and H. Hajishirzi. Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. In *Proceedings of the european conference on computer vision (ECCV)*, pages 552–568, 2018.

103. U. Michieli and P. Zanuttigh. Continual semantic segmentation via repulsion-attraction of sparse and disentangled latent representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1114–1124, 2021.

104. P. A. Mossey, E. E. Catilla, et al. Global registry and database on craniofacial anomalies: Report of a WHO registry meeting on craniofacial anomalies. 2003.

105. L. Mou, Y. Zhao, L. Chen, J. Cheng, Z. Gu, H. Hao, H. Qi, Y. Zheng, A. Frangi, and J. Liu. Cs-net: channel and spatial attention network for curvilinear structure segmentation. In *Proceedings of the 22nd International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 721–730, 2019.

106. R. Müller, S. Kornblith, and G. E. Hinton. When does label smoothing help? In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, pages 4696–4705, 2019.

107. I. Nigam, C. Huang, and D. Ramanan. Ensemble knowledge transfer for semantic segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1499–1508, 2018.

108. P. Niyogi. Manifold regularization and semi-supervised learning: Some theoretical analyses. *The Journal of Machine Learning Research*, 14(1):1229–1250, 2013.

109. M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 69–84, 2016.

110. M.-h. Oh, P. A. Olsen, and K. N. Ramamurthy. Crowd counting with decomposed uncertainty. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*, pages 11799–11806, 2020.

111. O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, et al. Attention u-net: Learning where to look for the pancreas. *arXiv preprint arXiv:1804.03999*, 2018.

112. A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

113. M. Orbes-Arteaga, T. Varsavsky, C. H. Sudre, Z. Eaton-Rosen, L. J. Haddow, L. Sørensen, M. Nielsen, A. Pai, S. Ourselin, M. Modat, et al. Multi-domain adaptation in brain mri through paired consistency and adversarial learning. In *Proceedings of the 1st Domain Adaptation and Representation Transfer and Medical Image Learning with Less Labels and Imperfect Data Workshop, MICCAI*, pages 54–62, 2019.

114. C. Ouyang, C. Biffi, C. Chen, T. Kart, H. Qiu, and D. Rueckert. Self-supervision with superpixels: Training few-shot medical image segmentation without annotation. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 762–780, 2020.

115. S. Özgün, A.-M. Rickmann, A. G. Roy, and C. Wachinger. Importance driven continual learning for segmentation across domains. In *Proceedings of the 11th International Workshop on Machine Learning in Medical Imaging (MLMI), Held in Conjunction with MICCAI 2020*, pages 423–433, 2020.

116. D. F. Pace, A. V. Dalca, T. Geva, A. J. Powell, M. H. Moghari, and P. Golland. Interactive whole-heart segmentation in congenital heart disease. In *Proceedings of the 18th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 80–88, 2015.

117. F. Pan, I. Shin, F. Rameau, S. Lee, and I. S. Kweon. Unsupervised intra-domain adaptation for semantic segmentation through self-supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3764–3773, 2020.

118. A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello. ENet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.

119. A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. In

Proceedings of Advances in Neural Information Processing Systems (NeurIPS) Workshop, 2017.

120. D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2536–2544, 2016.

121. C. Payer, D. Štern, H. Bischof, and M. Urschler. Multi-label whole heart segmentation using CNNs and anatomical label configurations. In *International Workshop on Statistical Atlases and Computational Models of the Heart*, pages 190–198, 2017.

122. A. Pimkin, G. Makarchuk, V. Kondratenko, M. Pisov, E. Krivov, and M. Belyaev. Ensembling neural networks for digital pathology images classification and segmentation. In *International Conference Image Analysis and Recognition (ICIAR)*, pages 877–886, 2018.

123. A. Prasoon, K. Petersen, C. Igel, F. Lauze, E. Dam, and M. Nielsen. Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network. In *Proceedings of the 16th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 246–253, 2013.

124. Z. Qiu, T. Yao, and T. Mei. Learning spatio-temporal representation with pseudo-3D residual networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5533–5541, 2017.

125. A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *Proceedings of the 4th International Conference on Learning Representations (ICLR)*, 2016.

126. I. Radosavovic, P. Dollár, R. Girshick, G. Gkioxari, and K. He. Data distillation: Towards omni-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4119–4128, 2018.

127. J. T. Richtsmeier, L. L. Baxter, and R. H. Reeves. Parallels of craniofacial maldevelopment in Down syndrome and Ts65Dn mice. *Developmental Dynamics*, 217(2):137–145, 2 2000.

128. O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Proceedings of the 18th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241, 2015.

129. A. G. Roy, N. Navab, and C. Wachinger. Concurrent spatial and channel 'squeeze & excitation'in fully convolutional networks. In *Proceedings of the 21st International conference on medical image computing and computer-assisted intervention (MICCAI)*, pages 421–429, 2018.

130. D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, pages 318–362, 1986.

131. R. Shahzad, S. Gao, Q. Tao, O. Dzyubachyk, and R. van der Geest. Automated cardiovascular segmentation in patients with congenital heart disease from 3D CMR scans: Combining multi-atlases and level-sets. In *the First International Workshops on Reconstruction, Segmentation, and Analysis of Medical Images, RAMBO 2016 and HVSMR 2016, Held in Conjunction with MICCAI 2016*, pages 147–155, 2016.

132. W. Shen, B. Wang, Y. Jiang, Y. Wang, and A. Yuille. Multi-stage multi-recursive-input fully convolutional networks for neuronal boundary detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2391–2400, 2017.

133. K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.

134. A. L. Simpson, J. N. Leal, A. Pugalenthi, P. J. Allen, R. P. DeMatteo, Y. Fong, M. Gönen, W. R. Jarnagin, T. P. Kingham, M. I. Miga, et al. Chemotherapy-induced splenic volume increase is independently associated with major complications after hepatic resection for metastatic colorectal cancer. *Journal of the American College of Surgeons*, 220(3):271–280, 2015.

135. A. L. Simpson, M. Antonelli, S. Bakas, M. Bilello, K. Farahani, B. Van Ginneken, A. Kopp-Schneider, B. A. Landman, G. Litjens, B. Menze, et al. A large annotated medical image dataset for the development and evaluation of segmentation algorithms. *arXiv preprint arXiv:1902.09063*, 2019.

136. K. Sirinukunwattana, J. P. Pluim, H. Chen, X. Qi, P.-A. Heng, Y. B. Guo, L. Y. Wang, B. J. Matuszewski, E. Bruni, U. Sanchez, et al. Gland segmentation in colon histology images: The GlaS challenge contest. *Medical Image Analysis*, 35:489–502, 2017.

137. M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning (ICML)*, pages 6105–6114, 2019.

138. X. Tao, Y. Li, W. Zhou, K. Ma, and Y. Zheng. Revisiting rubik's cube: self-supervised learning with volume-wise transformation for 3D medical image segmentation. In *Proceedings of the 23rd International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 238–248, 2020.

139. Y. Tay, D. Bahri, L. Yang, D. Metzler, and D.-C. Juan. Sparse sinkhorn attention. In *International Conference on Machine Learning (ICML)*, pages 9438–9447, 2020.

140. C. Tobon-Gomez, A. J. Geers, J. Peters, et al. Benchmark for algorithms segmenting the left atrium from 3D CT and MRI datasets. *IEEE Transactions on Medical Imaging*, 34(7):1460–1473, 2015.

141. J. M. J. Valanarasu, P. Oza, I. Hacihaliloglu, and V. M. Patel. Medical transformer: Gated axial-attention for medical image segmentation. In *Proceedings of the 24th Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pages 36–46, 2021.

142. M. J. Van der Laan, E. C. Polley, and A. E. Hubbard. Super learner. *Statistical Applications in Genetics and Molecular Biology*, 6(1), 2007.

143. A. Van Opbroek, M. A. Ikram, M. W. Vernooij, and M. De Bruijne. Transfer learning improves supervised image segmentation across imaging protocols. *IEEE Transactions on Medical Imaging*, 34(5):1018–1030, 2014.

144. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems (NeurIPS)*, pages 5998–6008, 2017.

145. A. Vyas, A. Katharopoulos, and F. Fleuret. Fast transformers with clustered attention. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.

146. D. Wang, E. Shelhamer, S. Liu, B. Olshausen, and T. Darrell. Tent: Fully test-time adaptation by entropy minimization. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, 2021.

147. G. Wang, W. Li, M. A. Zuluaga, R. Pratt, P. A. Patel, M. Aertsen, T. Doel, A. L. David, J. Deprest, S. Ourselin, et al. Interactive medical image segmentation using deep learning with image-specific fine tuning. *IEEE Transactions on Medical Imaging*, 37(7):1562–1573, 2018.

148. S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

149. X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 7794–7803, 2018.

150. Y. Wang, Z. Deng, X. Hu, L. Zhu, X. Yang, X. Xu, P.-A. Heng, and D. Ni. Deep attentional features for prostate segmentation in ultrasound. In *Proceedings of the 21st International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 523–530, 2018.

151. D. H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.

152. J. M. Wolterink, T. Leiner, M. A. Viergever, and I. Išgum. Dilated convolutional neural networks for cardiovascular MR segmentation in congenital heart disease. In *the First International Workshops on Reconstruction, Segmentation, and Analysis of Medical Images, RAMBO 2016 and HVSMR 2016, Held in Conjunction with MICCAI 2016*, pages 95–102, 2016.

153. H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang. Cvt: Introducing convolutions to vision transformers. *arXiv preprint arXiv:2103.15808*, 2021.

154. J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum. Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, pages 82–90, 2016.

155. Y. Xia, L. Xie, F. Liu, Z. Zhu, E. K. Fishman, and A. L. Yuille. Bridging the gap between 2D and 3D organ segmentation. In *Proceedings of the 21st International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, volume 11073, pages 445–453, 2018.

156. T. Xiao, M. Singh, E. Mintun, T. Darrell, P. Dollár, and R. Girshick. Early convolutions help transformers see better. *arXiv preprint arXiv:2106.14881*, 2021.

157. E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo. SegFormer: Simple and efficient design for semantic segmentation with transformers. *arXiv preprint arXiv:2105.15203*, 2021.

158. K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the International conference on machine learning (ICML)*, pages 2048–2057, 2015.

159. Y. Xu, Y. Li, Y. Wang, M. Liu, Y. Fan, M. Lai, I. Eric, and C. Chang. Gland instance segmentation using deep multichannel neural networks. *IEEE Transactions on Biomedical Engineering*, 64(12):2901–2912, 2017.

160. K. Yan, X. Wang, L. Lu, and R. M. Summers. Deeplesion: Automated deep mining, categorization and detection of significant radiology image findings using large-scale clinical lesion annotations. *arXiv preprint arXiv:1710.01766*, 2017.

161. J. Yang, G. Sharp, H. Veeraraghavan, W. van Elmpt, A. Dekker, T. Lustberg, and M. Gooding. Lung CT segmentation challenge 2017 — the cancer imaging archive. http://doi.org/10.7937/k9/tcia.2017.3r3fvz08, 2017.

162. L. Yang, Y. Zhang, J. Chen, S. Zhang, and D. Z. Chen. Suggestive annotation: A deep active learning framework for biomedical image segmentation. In *Proceedings of the 20th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 399–407, 2017.

163. L. Yang, Y. Zhang, Z. Zhao, H. Zheng, P. Liang, M. T. Ying, A. T. Ahuja, and D. Z. Chen. BoxNet: Deep learning based biomedical image segmentation using boxes only annotation. *arXiv preprint arXiv:1806.00593*, 2018.

164. Y. Yang, Z. Zhong, T. Shen, and Z. Lin. Convolutional neural networks with alternately updated clique. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2413–2422, 2018.

165. Y. Yin, X. Zhang, R. Williams, X. Wu, D. D. Anderson, and M. Sonka. LOGISMOS—layered optimal graph image segmentation of multiple objects and surfaces: Cartilage segmentation in the knee joint. *IEEE Transactions on Medical Imaging*, 29(12):2023–2037, 2010.

166. S. You, R. Barkalifa, E. J. Chaney, H. Tu, J. Park, J. E. Sorrells, Y. Sun, Y.-Z. Liu, L. Yang, D. Z. Chen, et al. Label-free visualization and characterization of extracellular vesicles in breast cancer. *Proceedings of the National Academy of Sciences*, 116(48):24012–24018, 2019.

167. C. Yu, B. Xiao, C. Gao, L. Yuan, L. Zhang, N. Sang, and J. Wang. Lite-hrnet: A lightweight high-resolution network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10440–10450, 2021.

168. 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 20th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 287–295, 2017.

169. L. Yu, S. Wang, X. Li, C.-W. Fu, and P.-A. Heng. Uncertainty-aware self-ensembling model for semi-supervised 3D left atrium segmentation. In *Proceedings of the 22nd International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 605–613, 2019.

170. K. Yuan, S. Guo, Z. Liu, A. Zhou, F. Yu, and W. Wu. Incorporating convolution designs into visual transformers. *arXiv preprint arXiv:2103.11816*, 2021.

171. H. Zhang, Y. Liao, H. Yang, G. Yang, and L. Zhang. A local-global dual-stream network for building extraction from very-high-resolution remote sensing images. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

172. X. Zhang, X. Zhou, M. Lin, and J. Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE/CVF*

conference on computer vision and pattern recognition (CVPR), pages 6848–6856, 2018.

173. Y. Zhang, L. Yang, J. Chen, M. Fredericksen, D. P. Hughes, and D. Z. Chen. Deep adversarial networks for biomedical image segmentation utilizing unannotated images. In *Proceedings of the 20th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 408–416, 2017.

174. Y. Zhang, L. Yang, J. Chen, M. Fredericksen, D. P. Hughes, and D. Z. Chen. Deep adversarial networks for biomedical image segmentation utilizing unannotated images. In *Proceedings of the 20th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 408–416, 2017.

175. F. Zhao and X. Xie. An overview of interactive medical image segmentation. *Annals of the British Machine Vision Association*, 2013(7):1–22, 2013.

176. S. Zhao, B. Li, X. Yue, Y. Gu, P. Xu, R. Hu, H. Chai, and K. Keutzer. Multi-source domain adaptation for semantic segmentation. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, pages 7287–7300, 2019.

177. Z. Zhao, L. Yang, H. Zheng, I. H. Guldner, S. Zhang, and D. Z. Chen. Deep learning based instance segmentation in 3D biomedical images using weak annotation. In *Proceedings of the 21st International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 352–360, 2018.

178. E. Zheng, Q. Yu, R. Li, P. Shi, and A. Haake. A continual learning framework for uncertainty-aware interactive image segmentation. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, pages 6030–6038, 2021.

179. H. Zheng, L. Yang, J. Chen, J. Han, Y. Zhang, P. Liang, Z. Zhao, C. Wang, and D. Z. Chen. Biomedical image segmentation via representative annotation. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, pages 5901–5908, 2019.

180. H. Zheng, L. Yang, J. Han, Y. Zhang, P. Liang, Z. Zhao, C. Wang, and D. Z. Chen. HFA-Net: 3D cardiovascular image segmentation with asymmetrical pooling and content-aware fusion. In *Proceedings of the 22nd International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 759–767, 2019.

181. H. Zheng, Y. Zhang, L. Yang, P. Liang, Z. Zhao, C. Wang, and D. Z. Chen. A new ensemble learning framework for 3D biomedical image segmentation.

In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, pages 5909–5916, 2019.

182. H. Zheng, S. M. M. Perrine, M. K. Pitirri, K. Kawasaki, C. Wang, J. T. Richtsmeier, and D. Z. Chen. Cartilage segmentation in high-resolution 3D micro-CT images via uncertainty-guided self-training with very sparse annotation. In *Proceedings of the 23rd International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 802–812, 2020.

183. H. Zheng, Y. Zhang, L. Yang, C. Wang, and D. Z. Chen. An annotation sparsification strategy for 3D medical image segmentation via representative selection and self-training. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pages 6925–6932, 2020.

184. H. Zheng, J. Han, H. Wang, L. Yang, Z. Zhao, C. Wang, and D. Z. Chen. Hierarchical self-supervised learning for medical image segmentation based on multi-domain data aggregation. In *Proceedings of the 24th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 622–632, 2021.

185. S. K. Zhou, H. Greenspan, C. Davatzikos, J. S. Duncan, B. van Ginneken, A. Madabhushi, J. L. Prince, D. Rueckert, and R. M. Summers. A review of deep learning in medical imaging: Image traits, technology trends, case studies with progress highlights, and future promises. *arXiv preprint arXiv:2008.09104*, 2020.

186. Y. Zhou, Y. Wang, P. Tang, S. Bai, W. Shen, E. K. Fishman, and A. L. Yuille. Semi-supervised multi-organ segmentation via deep multi-planar co-training. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 121–140, 2019.

187. Z. Zhou, J. Shin, L. Zhang, S. Gurudu, M. Gotway, and J. Liang. Fine-tuning convolutional neural networks for biomedical image analysis: actively and incrementally. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7340–7351, 2017.

188. Z. Zhou, J. Y. Shin, S. R. Gurudu, M. B. Gotway, and J. Liang. AFT*: Integrating active learning and transfer learning to reduce annotation efforts. *arXiv preprint arXiv:1802.00912*, 2018.

189. Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang. UNet++: Redesigning skip connections to exploit multiscale features in image segmentation. *IEEE Transactions on Medical Imaging*, 39(6):1856–1867, 2019.

190. Z. Zhou, V. Sodha, M. M. R. Siddiquee, R. Feng, N. Tajbakhsh, M. B. Gotway, and J. Liang. Models genesis: Generic autodidactic models for 3D medical

image analysis. In *Proceedings of the 22nd International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 384–393, 2019.

191. Z.-H. Zhou. *Ensemble Methods: Foundations and Algorithms*. CRC press, 2012.

192. L. Zhu, R. Deng, M. Maire, Z. Deng, G. Mori, and P. Tan. Sparsely aggregated convolutional networks. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 186–201, 2018.

193. X. Zhuang and J. Shen. Multi-scale patch and multi-modality atlases for whole heart segmentation of MRI. *Medical Image Analysis*, 31:77–87, 2016.

194. X. Zhuang, Y. Li, Y. Hu, K. Ma, Y. Yang, and Y. Zheng. Self-supervised feature learning for 3D medical images by playing a Rubik's cube. In *Proceedings of the 22nd International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 420–428, 2019.

195. Y. Zou, Z. Yu, B. Vijaya Kumar, and J. Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European conference on computer vision (ECCV)*, pages 289–305, 2018.