TOWARDS ROBUST DEEP LEARNING FOR MEDICAL IMAGE ANALYSIS

by
Yingda Xia

A dissertation submitted to Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy

Baltimore, Maryland
November, 2021

Abstract

Multi-dimensional medical data are rapidly collected to enhance healthcare. With the recent advance in artificial intelligence, deep learning techniques have been widely applied to medical images, constituting a significant proportion of medical data. The techniques of automated medical image analysis have the potential to benefit general clinical procedures, e.g., disease screening, malignancy diagnosis, patient risk prediction, and surgical planning. Although preliminary success takes place, the robustness of these approaches requires to be cautiously validated and sufficiently guaranteed before their application to real-world clinical problems.

In this thesis, we propose different approaches to improve the robustness of deep learning algorithms for automated medical image analysis. (i) In terms of network architecture, we leverage the advantages of both 2D and 3D networks, and propose an alternative 2.5D approach for 3D organ segmentation. (ii) To improve data efficiency and utilize large-scale unlabeled medical data, we propose a unified framework for semi-supervised medical image segmentation and domain adaptation. (iii) For the safety-critical applications, we design a unified approach for failure detection and anomaly segmentation. (iv) We study the problem of Federated Learning, which enables collaborative learning and preserves data privacy, and improve the robustness of the algorithm in the non-i.i.d setting. (v) We incorporate multi-phase information for more accurate pancreatic tumor detection. (vi) Finally, we show our discovery for potential pancreatic cancer screening on non-contrast CT scans which outperform expert radiologists.

Thesis Readers

Dr. Alan L. Yuille (Primary Advisor)
Bloomberg Distinguished Professor
Department of Computer Science
Johns Hopkins University
IEEE Fellow

Dr. Wei Shen
Associate Professor
Artificial Intelligence Institute
Shanghai Jiao Tong University

Dr. Le Lu
Head of Medical AI R&D
Alibaba Group
IEEE Fellow

Dedicated to my wife and my parents for their unending support.

Acknowledgements

First and foremost, I want to express my gratitude to my advisor Prof. Alan Yuille for his insightful guidance and selfless support. Alan is the professor who lead me to my very first computer vision research project in the summer of 2016, when I was an undergraduate student. His incomparable insights and wisdom arouse my great interest of the field, and finally led to the start of my career as a computer vision researcher. Since 2017, I have been working with him on the FELIX project funded by the Lustgarten Foundation, where we aim to detect early-stage pancreatic cancer with AI. Medical AI has become my major research focus since then. There was certainly up-and-downs during my years as a PhD student, but Alan was always there to support me and provide professional advice. Without Alan, I could never reached to where I am today.

Next, I want to thank Dr. Lingxi Xie and Prof. Wei Shen for their generous help on my works especially during my junior PhD years. The collaboration with them has always been a pleasure and the hands-on experience I learned from them accelerated my pace towards an competent researcher of this field. Besides, I want to thank Prof. Wei Shen and Dr. Le Lu for joining my defense committee, and Prof. Greg Hager, Prof. Vishal Patel, Prof. Elliot Fishman, Prof. Mathias Unberath, and Prof. Linda Chu for joining my GBO committee and offered constructive suggestions on my GBO exam. I also want to express special thanks to the FELIX members, including Dr. Elliot Fishman, Dr. Bert Volgestein, Dr. Seyoun Park, Dr. Linda Chu, Dr. Karen Horton, Dr. Ralph Hruban, Dr. Kenneth Kinzler, and Dr. Bert Vogelstein, for their

professional medical expertise and valuable suggestions. Their decades of dedication to the improvement of patient care motivated me to devote my every effort to bring AI technology for the medical field, which could hopefully benefit the interest of patients.

Additionally, I want to thank my mentors in my past internships. I'm fortunate to work with Dr.Holger Roth, Dr. Dong Yang, Dr. Daguang Xu, Dr. Wenqi Li, and Dr. Andriy Myronenko at Nvidia. I also had chance to intern at PAII, where I worked with Dr. Le Lu, Dr. Ling Zhang, and Dr. Adam Harrison. Before my PhD days, I interned at MSRA, where I worked with Dr. Kuiyuan Yang and Dr. Pengfei Xu. They are all amazing mentors and researchers, and working with them was of great pleasure.

The years as a lab member at CCVL are unforgettable. I want to thank all my colleagues and friends, including Yan Wang, Yongyi Lu, Zongwei Zhou, Zhuotun Zhu, Yuyin Zhou, Fengze Liu, Qihang Yu, Jieneng Chen, who are also working on the FELIX project, Adam Kortylewski, Weichao Qiu, Chenxi Liu, Zhishuai Zhang, Siyuan Qiao, Qing Liu, Chenxu Luo, Cihang Xie, Yi Zhang, Huiyu Wang, Qi Chen, Hongru Zhu, Yingwei Li, Jieru Mei, Yixiao Zhang, Zhuowan Li, Zihao Xiao, Chenglin Yang, Yutong Bai, Chen Wei, Angtian Wang, Ju He, Prakhar Kaushik, Qihao Liu, Xiaoding Yuan, Kate Sanders, Runtao Liu, Bowen Li, and many others. Being a member of CCVL at JHU is one of the luckiest thing in my life.

Finally, I want to thank all my course instructors and staff members at JHU. Specially, I would like to thank Lilian Oonyu and Micah McDowell for the organization of the lab affairs, and Kim Franklin and Zachary Burwell for the responsive and professional assistance of the process related to the academic program.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background and Motivation

The recent breakthrough of deep learning has led to tremendous progress in the field of computer vision [1] and natural language processing [2], and has become one of the key techniques of general artificial intelligence (AI). The advance of deep learning also provides opportunities for intelligent healthcare systems. Meanwhile, the healthcare system is experiencing rapid growth in imaging data that are collected to enhance patient care [3]. As a result, deep learning for automated medical image analysis has become a heated topic recently.

During the past years, deep learning has been explored in various aspects throughout the clinical workflow, e.g., screening for disease [4], diagnosis of malignancy [5], prognosis prediction [6], and pathology [7]. However, obstacles remain before we can achieve satisfying outcomes in real-world scenario [3]. This is partly because of the drawbacks of the current deep learning algorithms. Most deep learning approaches require a large amount of labeled data to train on, but the annotation of medical images is expensive and requires expertise. On the other hand, medical images differ from natural images in various aspects. Medical images are sometimes in 3D formats and have multiple phases or modalities. In addition, the necessity of prior medical knowledge are often neglected in the design of deep learning algorithms. Moreover,

healthcare is a safety-critical field where the cost of mistakes could be expensive. These challenges motivate us to design and establish robust AI systems for the purpose of medical image analysis.

To begin with, we argue that a robust automated medical image analysis system should acquire the following functionalities.

- Effectiveness. The system should achieve good performance, in the measure of human expertise. Some recent works have achieved similar performance or already outperformed radiologists in certain tasks, while the performance are still unacceptable in many other tasks.
- Generalizability. The system should generalize well to outside data that come from other hospitals and institutions. This requires the system to be robust to domain change, e.g., difference imaging machines, reconstruction protocols, and population.
- Alarm mechanism. The system should be capable of reporting to humans that certain cases are not suitable for itself. Successful examples should include input-level alarm, such as image quality assessment and out-of-distribution detection, and output-level alarm, such as failure detection.

Given the drawbacks of deep learning algorithms that they require large-scale datasets to train on and are not easily explainable, these requirements of robustness are challenging.

In this thesis, we provide directions and approaches toward a robust medical image analysis system that has the potential to satisfy real-world clinical needs. Here are some examples.

- The utilization of unlabeled data. Due to the expense of annotation for medical data, semi-supervised or unsupervised approaches that leverage large-scale

unlabeled data have the potential to relieve the annotation burden and increase the generalizability of deep learning models.

- Efficient representation learning. The architecture of the deep networks should be suitable for medical images, which differs from natural images, thus inducing efficient and effective representation learning.

- Quality assessment. Both input-level and output-level quality assessment algorithms should be integrated. Since deep networks tend to behave unpredictably when encountering out-of-distribution (OOD) data, e.g., producing high-confidence errors, the mechanism of OOD detection and failure prediction should be developed.

- Federated learning (FL). FL enables collaborative training of multiple institutions without sharing data, which has the potential of boosting generalizability if the model sees multi-site data points.

- The ability to integrate multi-modal information. For precision medicine, clinicians use multi-modal information for diagnosis purposes, such as multi-modal imaging data, medical records, and other test results. A robust AI system should also utilize all available information.

Our approaches are proposed based on the discussed motivations and aim to make efforts toward the listed research directions above.

1.2 Contribution

Firstly, we focus on improving the robustness of network architecture for medical image segmentation, which is a fundamental task and is the prerequisite of medical image analysis systems. Different from the semantic segmentation task in 2D natural images, medical image segmentation often works on 3D images, such as CT scans and MRI

scans. Previous state-of-the-art approaches either train 2D networks [8], [9] on the slices of 3D medical images or directly train 3D networks [10], [11] on the volumes. However, the former fails to incorporate 3D context into the network training, and the latter is less computationally efficient and suffers from the problem of lacking pre-trained models. We propose an alternative approach. We first train three 2D networks with three different slicing directions, corresponding to the radiologists' reformatted views, i.e., coronal, sagittal, and axial view. Then we use a light-weighted 3D network, which takes the input of the original 3D image and the three 2D prediction maps as input, to produce more accurate segmentation results. We validate this approach on NIH pancreas segmentation dataset [12] and the JHU multi-organ dataset. The superior results demonstrate that our method outperforms 2D state-of-the-art approaches and has less computation burden than 3D networks. This approach explores an accurate yet efficient way for medical image segmentation, and is capable of stabilizing and accelerating the training process of vanilla 3D networks, thus improving the robustness in terms of network architecture.

Secondly, we propose to improve the data efficiency and domain robustness. As mentioned in Section 1.1, deep learning algorithms usually require a large amount of labeled data to train on. Unlike the computer vision tasks on natural images, the annotation of medical images is commonly expensive and requires expertise. As a result, semi-supervised or unsupervised methods which utilize unlabeled data are valuable to be explored on medical image analysis tasks. The ability to utilize large-scale unlabeled data could benefit the model by boosting the general performance in the same domain [13], and also improve the robustness when adapted to other domains [14]. We propose uncertainty-aware multi-view co-training, a framework that is capable of utilizing unlabeled 3D medical images for semi-supervised medical image segmentation and domain adaptation. This work is motivated by the success of co-training on 2D images [15]. In order to effectively extend co-training to 3D medical

images, we generate multiple views in co-training by the permutation or rotation of the volume. Our framework can boost the pancreas segmentation on the NIH dataset by 12% with only 10% data labeled and the rest unlabeled. Directly applying our method to unsupervised domain adaptation tasks, we outperform standard self-training and adversarial training methods.

Thirdly, we design a new alarm system to enhance the safety of AI applications. Medical image analysis is a safety-critical scenario for the application of deep learning. Recent research shows that deep networks tend to produce high-confidence errors [16] and are unpredictable when encountering out-of-distribution data as the input [17]. So the ability of failure detection and anomaly detection is crucial for a reliable medical AI system. We hereby propose a new approach that is capable of detecting failures and out-of-distribution data at the same time. We name our approach Synthesize then Compare (SynthCP), which contains a synthesis module and a comparison module. The image synthesis module generates a synthesized image from a segmentation layout map implemented by a conditional generative adversarial network (cGAN) [18], and the comparison module computes the difference between the synthesized image and the input image to predict failures and anomalous objects. This method outperforms probability-based failure detection methods and Bayesian methods on three datasets, which include pancreatic tumor segmentation. Our method sheds light on the potential of self-alarm AI systems with improved reliability and safety than regular deep learning algorithms.

Fourthly, we focus on the task of Federated Learning(FL). Medical data is often privacy sensitive, but deep learning models usually require versatile data sources to generalize well to unseen domains. FL [19] is suitable for collaborative training of machine learning models without sharing data between the participants, providing us opportunities to improve model robustness. The most common technique of FL is Federated Averaging (FedAvg) [19], where the server collects locally trained models

from the clients, averages to obtain a global model, and sends it back to the clients in an iterative fashion. The averaging weights are set to be proportional to the number of data on each client. However, since the data distribution of the clients remains unknown to the server, this prior could hardly be optimal and could lead to unsatisfying results. We propose a data-driven approach, named Auto-FedAvg, to dynamically adjust the averaging weights for better performance and model generalizability. We design an efficient communication algorithm between the server and the clients to iteratively update the global aggregation parameters and local model parameters. Our approach is validated on two multi-institutional medical image analysis tasks, i.e., COVID-19 lesion segmentation in chest CT and pancreas segmentation in abdominal CT, and outperforms the state-of-the-art methods in FL.

Fifthly, We focus on the task of the early detection of pancreatic tumors in CT scans. Previous work has presented approaches for pancreatic tumor segmentation in single-phase contrast-enhanced CT scans [20]. We propose to automated align and segment pancreatic tumors in the arterial and venous phase simultaneously. The major challenge of the goal is that the two phases are usually not aligned due to the inevitable movement of the patient during imaging, and the tumors have heterogeneous appearances across phases. One straightforward strategy is to first align the phases in image space via image registration techniques [21], which we name Early Align. In the alternative, we propose to automatically align and segment pancreatic tumors in the feature space, named Late Align, or gradually align in multiple levels of the deep network, named Late Align. We discover that a simple ensemble of the three alignment strategies can significantly boost the performance of this task. We validate our approach on two PDAC datasets and outperform single-phase algorithms, illustrating our superiority of integrating dual-phase information.

Finally, we reveal our discovery that deep networks can detect pancreatic tumors in non-contrast CT scans (NCCT), a cheaper and safer substituent of the regular

contrast-enhanced CT scans (CECT), which is commonly used for the diagnosis purpose of radiologists. In order to obtain training labels for NCCT, we transfer the segmentation annotations of the radiologists from CECT to NCCT via image registration. We then design an anatomy-aware hybrid transformer to jointly segment the tumor mass and classify the type of abnormality. We collect a large-scale dataset, which contains the images of 1627 patients: 558 PDACs, 474 nonPDACs (including nine subtypes), and 595 normal, confirmed by pathological reports. Our approach achieves a sensitivity of 95.2% and specificity of 95.8% in the test set which contains 306 cases and significantly outperforms the performance of 11 radiologists who participated in the reader study.

1.3 Thesis Outline

This thesis is organized as follows:

In Chapter 2, we focus on the network architecture. We propose a new framework for 3D medical image segmentation, which leverages the benefits of 2D and 3D deep networks.

In Chapter 3, we focus on improving data efficiency. We extend co-training framework to three dimensional data for semi-supervised medical image segmentation and domain adaptation.

In Chapter 4, we integrate an alarm system into semantic segmentation applications to detect failures and out-of-distributional data.

In Chapter 5, we improve the federated learning framework for multi-institutional medical image analysis.

In Chapter 6, we improve the existing pancreatic tumor detection algorithm with multi-phase alignment.

In Chapter 7, we propose to detect pancreatic tumors in non-contrast CT scans, a

safer and cheaper constituent of the widely-used contrast-enhanced CT scans.

In Chapter 8, we conclude this thesis and provide insights on future work.

1.4 Relevant Publications

The following publications constitute or contribute to the context of this dissertation. The "*" indicates equal contribution.

1. **Yingda Xia**, Lingxi Xie, Fengze Liu, Zhuotun Zhu, Elliot Fishman, Alan Yuille. "Bridging the gap between 2d and 3d organ segmentation with volumetric fusion net." In MICCAI 2018.

2. Zhuotun Zhu, **Yingda Xia**, Wei Shen, Elliot Fishman, Alan Yuille. "A 3d coarse-to-fine framework for volumetric medical image segmentation." In 3DV 2018.

3. Yingwei Li*, Zhuotun Zhu*, Yuyin Zhou, **Yingda Xia**, Wei Shen, Elliot Fishman, Alan Yuille. "Volumetric medical image segmentation: a 3D deep coarse-to-fine framework and its adversarial examples." In Deep Learning and Convolutional Neural Networks for Medical Imaging and Clinical Informatics, p69-91.

4. Zhuotun Zhu*, **Yingda Xia***, Lingxi Xie, Elliot Fishman, Alan Yuille. "Multi-scale coarse-to-fine segmentation for screening pancreatic ductal adenocarcinoma." In MICCAI 2019.

5. Fengze Liu, **Yingda Xia**, Dong Yang, Alan Yuille, Daguang Xu. "An Alarm System For Segmentation Algorithm Based On Shape Model." In ICCV 2019.

6. Linda Chu, Seyoun Park, Satomi Kawamoto, Yan Wang, Yuyin Zhou, Wei Shen, Zhuotun Zhu, **Yingda Xia**, Lingxi Xie, Fengze Liu, Qihang Yu, Daniel F Fouladi, Shahab Shayesteh, Eva Zinreich, Jefferson S Graves, Karen M Horton,

Alan Yuille, Ralph H Hruban, Kenneth W Kinzler, Bert Vogelstein, Elliot Fishman. "Application of deep learning to pancreatic cancer detection: lessons learned from our initial experience." In Journal of the American College of Radiology 2019.

7. **Yingda Xia**, Dong Yang, Wenqi Li, Andriy Myronenko, Daguang Xu, Hirofumi Obinata, Hitoshi Mori, Peng An, Stephanie Harmon, Evrim Turkbey, Baris Turkbey, Bradford Wood, Francesca Patella, Elvira Stellato, Gianpaolo Carrafiello, Anna Ierardi, Alan Yuille, Holger Roth. "Auto-FedAvg: Learnable Federated Averaging for Multi-Institutional Medical Image Segmentation." arxiv:2104.10195, 2021.

8. **Yingda Xia**, Fengze Liu, Dong Yang, Jinzheng Cai, Lequan Yu, Zhuotun Zhu, Daguang Xu, Alan Yuille, Holger Roth. "3D Semi-Supervised Learning with Uncertainty-Aware Multi-View Co-Training." In WACV 2020.

9. **Yingda Xia***, Qihang Yu*, Wei Shen, Yuyin Zhou, Elliot Fishman, Alan Yuille. "Detecting Pancreatic Ductal Adenocarcinoma in Multi-phase CT Scans via Alignment Ensemble." In MICCAI 2020.

10. **Yingda Xia**, Dong Yang, Zhiding Yu, Fengze Liu, Jinzheng Cai, Lequan Yu, Zhuotun Zhu, Daguang Xu, Alan Yuille, Holger Roth. "Uncertainty-aware multi-view co-training for semi-supervised medical image segmentation and domain adaptation." In Medical Image Analysis 2020.

11. **Yingda Xia***, Yi Zhang*, Fengze Liu, Wei Shen, Alan Yuille. "Synthesize then Compare: Detecting Failures and Anomalies for Semantic Segmentation." In ECCV 2020.

12. **Yingda Xia**, Jiawen Yao, Le Lu, Lingyun Huang, Guotong Xie, Jing Xiao, Alan Yuille, Kai Cao, Ling Zhang. "Effective Pancreatic Cancer Screening on

Non-contrast CT Scans via Anatomy-Aware Transformers." In MICCAI 2021.

Other publications or preprints that I authored are listed below.

1. Chen Wei, Lingxi Xie, Xutong Ren, **Yingda Xia**, Chi Su, Jiaying Liu, Qi Tian, Alan Yuille. "Iterative Reorganization with Weak Spatial Constraints: Solving Arbitrary Jigsaw Puzzles for Unsupervised Representation Learning." In CVPR 2019.

2. Fengze Liu, Lingxi Xie, **Yingda Xia**, Elliot Fishman, Alan Yuille. "Joint shape representation and classification for detecting PDAC." In MLMI 2019.

3. Qihang Yu, **Yingda Xia**, Lingxi Xie, Elliot Fishman, Alan Yuille. "Thickened 2D networks for efficient 3D medical image segmentation." arXiv:1904.01150, 2019.

4. Jinzheng Cai, **Yingda Xia**, Dong Yang, Daguang Xu, Lin Yang, Holger Roth. "End-to-end adversarial shape learning for abdomen organ deep segmentation." In MLMI 2019.

5. Qihang Yu, **Yingda Xia**, Yutong Bai, Yongyi Lu, Alan Yuille, Wei Shen. "Glance-and-Gaze Vision Transformer."In NeurIPS 2021.

6. Liangqiong Qu, Yuyin Zhou, Paul Pu Liang,**Yingda Xia**, Feifei Wang, Li Fei-Fei, Ehsan Adeli, Daniel Rubin. "Rethinking Architecture Design for Tackling Data Heterogeneity in Federated Learning." arXiv:2106.06047, 2021.

Chapter 2

Bridging the Gap Between 2D and 3D Organ Segmentation with Volumetric Fusion Net

In this chapter, we adopt 3D Convolutional Neural Networks to segment volumetric medical images. Although deep neural networks have been proven to be very effective on many 2D vision tasks, it is still challenging to apply them to 3D tasks due to the limited amount of annotated 3D data and limited computational resources. We propose a novel 3D-based coarse-to-fine framework to ***effectively*** and ***efficiently*** tackle these challenges. The proposed 3D-based framework outperforms the 2D counterpart to a large margin since it can leverage the rich spatial information along all three axes. We conduct experiments on two datasets which include healthy and pathological pancreases respectively, and achieve the current state-of-the-art in terms of Dice-Sørensen Coefficient (DSC). On the NIH pancreas segmentation dataset, we outperform the previous best by an average of over 2%, and the worst case is improved by 7% to reach almost 70%, which indicates the reliability of our framework in clinical applications.

2.1 Introduction

With the increasing requirement of fine-scaled medical care, computer-assisted diagnosis (CAD) has attracted more and more attention in the past decade. An important prerequisite of CAD is an intelligent system to process and analyze medical data, such as CT and MRI scans. In the area of medical imaging analysis, organ segmentation is a traditional and fundamental topic [22]. Researchers often designed a specific system for each organ to capture its properties. In comparison to large organs (*e.g.*, the liver, the kidneys, the stomach, *etc.*), small organs such as the pancreas are more difficult to segment, which is partly caused by their highly variable geometric properties [12].

In recent years, with the arrival of the deep learning era [23], powerful models such as convolutional neural networks [24] have been transferred from natural image segmentation to organ segmentation. But there is a difference. Organ segmentation requires dealing with volumetric data, and two types of solutions have been proposed. The first one trains 2D networks from three orthogonal planes and fusing the segmentation results [12][9][8], and the second one suggests training a 3D network directly [25][26][27]. But 3D networks are more computationally expensive yet less stable when trained from scratch, and it is difficult to find a pre-trained model for medical purposes. In the scenario of limited training data, fine-tuning a pre-trained 2D network [24] is a safer choice [28].

This paper presents an alternative framework, which trains 2D segmentation models and uses a light-weighted 3D network, named **Volumetric Fusion Net** (VFN), in order to fuse 2D segmentation at a late stage. A similar idea is studied before based on either the EM algorithm [29] or pre-defined operations in a 2D scenario [30], but we propose instead to construct generalized linear operations (convolution) and allow them to be learned from training data. Because it is built on top of reasonable 2D segmentation results, VFN is relatively shallow and does not use fully-connected layers

(which contribute a large fraction of network parameters) to improve its discriminative ability. In the training process, we first optimize 2D segmentation networks on different viewpoints individually (this strategy was studied in [31][32][8]), and then use the validation set to train VFN. When the amount of training data is limited, we suggest a *cross-cross-augmentation* strategy to enable reusing the data to train both 2D segmentation and 3D fusion networks.

We first apply our system to a public dataset for pancreas segmentation [12]. Based on the state-of-the-art 2D segmentation approaches [9][8], VFN produces a consistent accuracy gain and outperforms other fusion methods, including majority voting and statistical fusion [29]. In comparison to 3D networks such as [27], our framework achieves comparable segmentation accuracy using fewer computational resources, *e.g.*, using 10% parameters and being 3× faster at the testing stage (it only adds 10% computation beyond the 2D baselines). We also generalize our framework to other small organs such as the adrenal glands and the duodenum, and verify its favorable performance.

2.2 Our Approach

2.2.1 Framework: Fusing 2D Segmentation into a 3D Volume

We denote an input CT volume by $\mathbf{X}$. This is a $W \times H \times L$ volume, where W, H and L are the numbers of voxels along the *coronal*, *sagittal* and *axial* directions, respectively. The i-th voxel of $\mathbf{X}$, x_i, is the intensity (Hounsfield Unit, HU) at the corresponding position, $i = (1, 1, 1), \ldots, (W, H, L)$. The ground-truth segmentation of an organ is denoted by $\mathbf{Y}^\star$, which has the same dimensionality as $\mathbf{X}$. If the i-th voxel belongs to the target organ, we set $y_i^\star = 1$, otherwise $y_i^\star = 0$. The goal of organ segmentation is to design a function $\mathbf{g}(\cdot)$, so that $\mathbf{Y} = \mathbf{g}(\mathbf{X})$, with all $y_i \in \{0, 1\}$, is close to $\mathbf{Y}^\star$. We measure the similarity between $\mathbf{Y}$ and $\mathbf{Y}^\star$ by the Dice-Sørensen coefficient (DSC):

$\text{DSC}(\mathbf{Y}, \mathbf{Y}^\star) = \frac{2\times|\mathcal{Y}\cap\mathcal{Y}^\star|}{|\mathcal{Y}|+|\mathcal{Y}^\star|}$, where $\mathcal{Y}^\star = \{i \mid y_i^\star = 1\}$ and $\mathcal{Y} = \{i \mid y_i = 1\}$ are the sets of foreground voxels.

There are, in general, two ways to design $\mathbf{g}(\cdot)$. The first one trains a 3D model to deal with volumetric data directly [25][26], and the second one works by cutting the 3D volume into slices, and using 2D networks for segmentation. Both 2D and 3D approaches have their advantages and disadvantages. We appreciate the ability of 3D networks to take volumetric cues into consideration (radiologists also exploit 3D information to make decisions), but, as shown in Section 2.3.2, 3D networks are sometimes less stable, arguably because we need to train all weights from scratch, while the 2D networks can be initialized with pre-trained models from the computer vision literature [24]. On the other hand, processing volumetric data (*e.g.*, 3D convolution) often requires heavier computation in both training and testing (*e.g.*, requiring 3× testing time, see Table 2.1).

In mathematical terms, let $\mathbf{X}_l^{\text{A}}$, $l = 1, 2, \ldots, L$ be a 2D slice (of $W \times H$) along the *axial* view, and $\mathbf{Y}_l^{\text{A}} = \mathbf{s}^{\text{A}}\left(\mathbf{X}_l^{\text{A}}\right)$ be the segmentation score map for $\mathbf{X}_l^{\text{A}}$. $\mathbf{s}^{\text{A}}(\cdot)$ can be a 2D segmentation network such as FCN [24], or a multi-stage system such as a coarse-to-fine framework [8]. Stacking all $\mathbf{Y}_l^{\text{A}}$'s yields a 3D volume $\mathbf{Y}^{\text{A}} = \mathbf{s}^{\text{A}}(\mathbf{X})$. This slicing-and-stacking process can be performed along each axis independently. Due to the large image variation in different views, we train three segmentation models, denoted by $\mathbf{s}^{\text{C}}(\cdot)$, $\mathbf{s}^{\text{S}}(\cdot)$ and $\mathbf{s}^{\text{A}}(\cdot)$, respectively. Finally, a fusion function $\mathbf{f}[\cdot]$ integrates them into the final prediction:

$$\mathbf{Y} = \mathbf{f}\left[\mathbf{X}, \mathbf{Y}^{\text{C}}, \mathbf{Y}^{\text{S}}, \mathbf{Y}^{\text{A}}\right] = \mathbf{f}\left[\mathbf{X}, \mathbf{s}^{\text{C}}(\mathbf{X}), \mathbf{s}^{\text{S}}(\mathbf{X}), \mathbf{s}^{\text{A}}(\mathbf{X})\right]. \tag{2.1}$$

Note that we allow the image $\mathbf{X}$ to be incorporated. This is related to the idea known as auto-contexts [33] in computer vision. As we shall see in experiments, adding $\mathbf{X}$ improves the quality of fusion considerably. Our goal is to equip $\mathbf{f}[\cdot]$ with partial abilities of 3D networks, *e.g.*, learning simple, local 3D patterns.

2.2.2 Volumetric Fusion Net

The VFN approach is built upon the 2D segmentation volumes from three orthogonal (*coronal*, *sagittal* and *axial*) planes. Powered by state-of-the-art deep networks, these results are generally accurate (*e.g.*, an average DSC of over 82% [8] on the NIH pancreas segmentation dataset [12]). But, as shown in Figure 2.2, some *local* errors still occur because 2 out of 3 views fail to detect the target. Our assumption is that these errors can be recovered by learning and exploiting the 3D image patterns in its surrounding region.

Regarding other choices, majority voting obviously cannot take image patterns into consideration. The STAPLE algorithm [29], while being effective in multi-atlas registration, does not have a strong ability of fitting image patterns from training data. We shall see in experiments that STAPLE is unable to improve segmentation accuracy over majority voting.

Motivated by the need to learn *local* patterns, we equip VFN with a small input region (64^3) and a shallow structure, so that each neuron has a small receptive field (the largest region seen by an output neuron is 50^3). In comparison, in the 3D network VNet [26], these numbers are 128^3 and 551^3, respectively. This brings twofold benefits. First, we can sample more patches from the training data, and the number of parameters is much less, and so the risk of over-fitting is alleviated. Second, VFN is more computationally efficient than 3D networks, *e.g.*, adding 2D segmentation, it needs only half the testing time of [27].

The architecture of VFN is shown in Figure 2.1. It has three down-sampling stages and three up-sampling stages. Each down-sampling stage is composed of two $3 \times 3 \times 3$ convolutional layers and a $2 \times 2 \times 2$ max-pooling layer with a stride of 2, and each up-sampling stage is implemented by a single $4 \times 4 \times 4$ deconvolutional layer with a stride of 2. Following other 3D networks [26][27], we also build a few residual

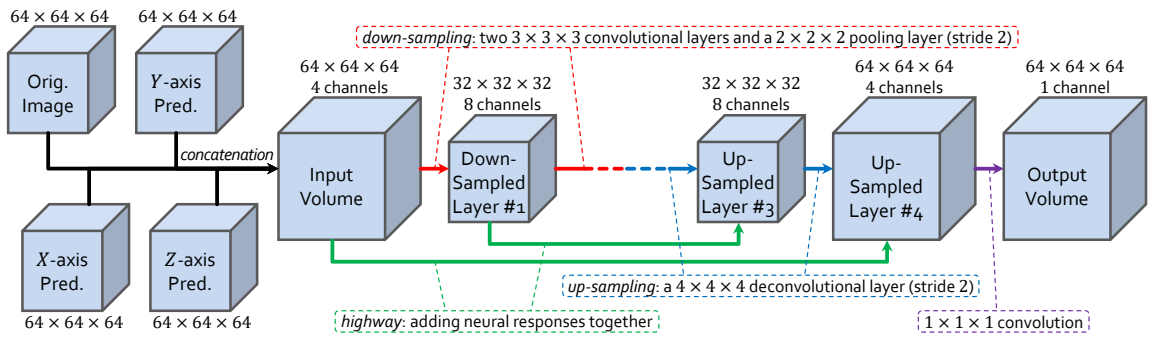


Figure 2.1. The network structure of VFN (best viewed in color). We only display one down-sampling and one up-sampling stages, but there are 3 of each. Each down-sampling stage shrinks the spatial resolution by $1/2$ and doubles the number of channels. We build 3 highway connections (2 are shown). We perform batch normalization and ReLU activation after each convolutional and deconvolutional layer.

connections [1] between hidden layers of the same scale. For our problem, this enables the network to preserve a large fraction of 2D network predictions (which are generally of good quality) and focus on refining them (note that if all weights in convolution are identical, then VFN is approximately equivalent to majority voting). Experiments show that these highway connections lead to faster convergence and higher accuracy. A final convolution of a $1 \times 1 \times 1$ kernel reduces the number of channels to 1.

The input layer of VFN consists of 4 channels, 1 for the original image and 3 for 2D segmentations from different viewpoints. The input values in each channel are normalized into $[0, 1]$. By this we provide equally-weighted information from the original image and 2D multi-view segmentation results, so that VFN can fuse them at an early stage and learn from data automatically. We verify in experiments that image information is important – training a VFN without this input channel shrinks the average accuracy gain by half.

2.2.3 Training and Testing VFN

We train VFN from scratch, *i.e.*, all weights in convolution are initialized as random white noises. Note that setting all weights as 1 mimics majority voting, and we find

that both ways of initialization lead to similar testing performance. All $64 \times 64 \times 64$ volumes are sampled from the region-of-interest (ROI) of each training case, defined as the bounding box covering all foreground voxels padded by 32 pixels in each dimension. We introduce data augmentation by performing random 90°-rotation and flip in 3D space (each cube has 24 variants). We use a Dice loss to avoid background bias (a voxel is more likely to be predicted as background, due to the majority of background voxels in training). We train VFN for 30,000 iterations with a mini-batch size of 16. We start with a learning rate of 0.01, and divide it by 10 after 20,000 and 25,000 iterations, respectively. The entire training process requires approximately 6 hours in a Titan-X-Pascal GPU. In the testing process, we use a sliding window with a stride of 32 in the ROI region (the minimal 3D box covering all foreground voxels of multi-plane 2D segmentation fused by majority voting). For an average pancreas in the NIH dataset [12], testing VFN takes around 5 seconds.

An important issue in optimizing VFN is to construct the training data. Note that we cannot reuse the data used for training segmentation networks to train VFN, because this will result in the input channels contain very accurate segmentation, which limits VFN from learning meaningful local patterns and generalizing to the testing scenarios. So, we further split the training set into two subsets, one for training the 2D segmentation networks and the other for training VFN with the testing segmentation results.

However, under most circumstances, the amount of training data is limited. For example, in the NIH pancreas segmentation dataset, each fold in cross-validation has only 60 training cases. Partitioning it into two subsets harms the accuracy of both 2D segmentation and fusion. To avoid this, we suggest a **cross-cross-augmentation** (CCA) strategy, described as follows. Suppose we split data into K folds for cross-validation, and the k_1-th fold is left for testing. For all $k_2 \neq k_1$, we train 2D segmentation models on the folds in $\{1, 2, \ldots, K\} \setminus \{k_1, k_2\}$, and test on the k_2-th

fold to generate training data for the VFN. In this way, all data are used for training both the segmentation model and the VFN. The price is that a total of $K(K-1)/2$ extra segmentation models need to be trained, which is more costly than training K models in a standard cross-validation. In practice, this strategy improves the average segmentation accuracy by $\sim 1\%$ in each fold. Note that we perform CCA only on the NIH dataset due to the limited amount of data – in our own dataset, we perform standard training/testing split, requiring $< 10\%$ extra training time and ignorable extra testing time.

2.3 Experiments

2.3.1 The NIH Pancreas Segmentation Dataset

We first evaluate our approach on the NIH pancreas segmentation dataset [12] containing 82 abdominal CT volumes. The width and height of each volume are both 512, and the number of slices along the *axial* axis varies from 181 to 466. We split the dataset into 4 folds of approximately the same size, and apply cross-cross-augmentation (see Section 2.2.3) to improve segmentation accuracy.

Results are summarized in Table 2.1. We use two recent 2D segmentation approaches as our baseline, and compare VFN with two other fusion approaches, namely majority voting and non-local STAPLE (NLS) [29]. The latter was verified more effective than its former local version. We measure segmentation accuracy using DSC and report the average accuracy over 82 cases. Based on [8], VFN improves majority voting significantly by an average of 1.69%. The improvement over 82 cases is consistent (the student's t-test reports a p-value of 6.9×10^{-7}), although the standard deviation over 82 cases is relatively large – this is mainly caused by the difference in difficulties from case to case. Figure 2.2 shows an example on which VFN produces a significant accuracy gain. VFN does not improve [9] significantly, arguably because [9]

Approach	Average	Min	1/4-Q	Med	3/4-Q	Max	Time (m)
Roth *et al.* [12]	71.42 ± 10.11	23.99	–	–	–	86.29	6–8
Roth *et al.* [34]	78.01±8.20	34.11	–	–	–	88.65	2–3
Roth *et al.* [35]	81.27±6.27	50.69	–	–	–	88.96	2–3
Cai *et al.* [36]	82.4 ± 6.7	60.0	–	–	–	90.1	N/A
Zhu *et al.* [27]	84.59±4.86	69.62	–	–	–	91.45	4.1
Zhou *et al.* [8]	82.50±6.14	56.33	81.63	84.11	86.28	89.98	0.9
[8] + NLS	82.25±6.57	56.86	81.54	83.96	86.14	89.94	1.1
[8] + VFN	**84.06** ± 5.63	62.93	81.98	85.69	87.62	91.28	1.0
Yu *et al.* [9]	84.48±5.03	62.23	82.50	85.66	87.82	91.17	1.3
[9] + NLS	84.47±5.03	62.22	82.42	85.59	87.78	91.17	1.5
[9] + VFN	**84.63** ± 5.07	61.58	82.42	85.84	88.37	91.57	1.4

Table 2.1. Comparison of segmentation accuracy (DSC, %) and testing time (in minutes) between our approach and the state-of-the-arts on the NIH dataset [12]. Both [8] and [9] are reimplemented by ourselves, and the default fusion is majority voting.

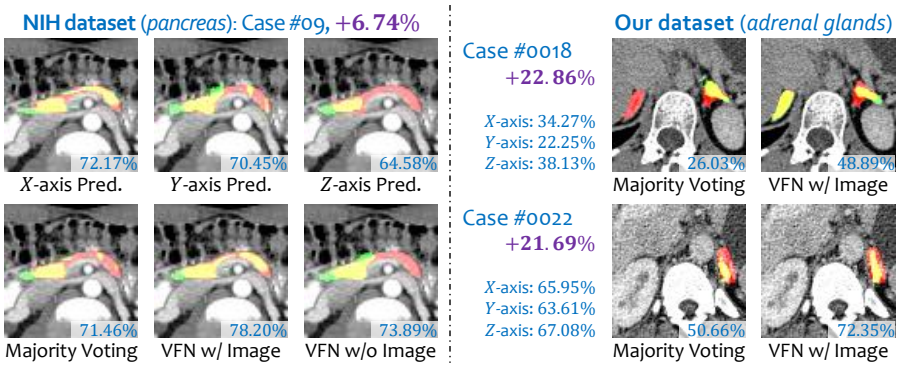


Figure 2.2. Two typical examples, each with the original image, segmentation results from three viewpoints, and different fusion results. In each label map, red, green and yellow indicate ground-truth, prediction and overlap, respectively (best viewed in color).

has almost reached the human-level agreement (we invited a radiologist to segment this dataset individually, and she achieves an average accuracy of $\sim 86\%$). Note that the other approaches without CCA used both the training and validation folds for training, and so all numbers are comparable in Table 2.1.

Due to our analysis in Section 2.2.2, NLS does not produce any accuracy gain over either [8] and [9]. NLS is effective in multi-atlas registration, where the labels come from different images and the annotation is relatively accurate [29]. But in our problem, segmentation results from 2D networks can be noisy, thus recovering these errors requires learning local image patterns from training data, which is what VFN does to outperform NLS.

To reveal the importance of image information, we train a VFN without the image channel in the input layer. Based on [8], this version produces approximately half of the improvement (1.69%) by the full model. We show an example in Figure 2.2, in which the right part of the pancreas is missing in both *sagittal* and *axial* planes, but the high confidence in the *coronal* plane and the continuity of image intensities suggest its presence in the final segmentation.

Approach	*adrenal g.*	*duodenum*	*gallbladder*	*pancreas*
Zhu *et al.* [27]	36.74 ± 25.14	68.80 ± 14.38	42.01 ± 29.47	85.25±6.04
Zhou *et al.* [8]	66.09 ± 18.19	71.65 ± 13.15	90.39±5.30	84.52±6.23
[8] + VFN	**69.24** ± 17.42	**72.77** ± 12.80	**91.40** ± 5.19	**86.39** ± 6.20
Yu *et al.* [9]	71.40 ± 12.87	77.48±8.70	91.81±4.90	87.22±5.90
[9] + VFN	**72.09** ± 13.61	**77.77** ± 8.46	**92.15** ± 5.05	**88.06** ± 5.33

Table 2.2. Comparison of segmentation accuracy (DSC, %) on our multi-organ dataset. The baseline for [8] and [9] is majority voting. The numbers of [9] are different from those in their original paper, because we are using a different dataset.

2.3.2 Our Multi-Organ Dataset

The radiologists in our team collected a dataset with 300 high-resolution CT scans. These scans were performed on some potential renal donors. Four experts in abdominal anatomy annotated 11 abdominal organs, taking 3–4 hours for each scan, and all annotations were verified by an experienced board certified Abdominal Radiologist. Except for the *pancreas*, we choose several challenging targets, including the *adrenal glands*, the *duodenum*, and the *gallbladder* (easy cases such as the *liver* and the *kidneys* are not considered). We use 150 cases for training 2D segmentation models, 100 cases for training VFN, and test on the remaining 50 cases. The data split is random but identical for different organs.

Results are shown in Table 2.2. Again, our approach consistently improves 2D segmentation, which demonstrates the transferability of our methodology. In *pancreas*, based on [9], we obtain a p-value of 2.7×10^{-5} over 50 testing cases. In *adrenal glands*, although the average accuracy gains are not large, the improvement is significant in some badly segmented cases, *e.g.*, Figure 2.2 shows two examples with more than 20% accuracy boosts. Refining bad segmentations makes our segmentation results more

reliable. By contrast, the 3D network [27] produces unstable performance ([27] was designed for pancreas segmentation, thus works reasonably well in *pancreas*), which is mainly caused by the limited training data especially for small organs such as *adrenal glands* and *gallbladder*.

Therefore, we conclude that 2D segmentation followed by 3D fusion is currently a very promising idea to bridge the gap between 2D and 3D segmentation approaches, particularly if there is limited training data.

2.4 Conclusions

In this paper, we discuss an important topic in medical imaging analysis, namely bridging the gap between 2D and 3D organ segmentation approaches. We propose to train more stable 2D segmentation networks, and then use a light-weighted 3D fusion module to fuse their results. In this way, we enjoy the benefits of exploiting 3D information to improve segmentation, as well as avoiding the risk of over-fitting caused by tuning 3D models (which have 10× more parameters) on a limited amount of training data. We verify the effectiveness of our approach on two datasets, one of which contains several challenging organs.

Based on our work, a promising direction is to train the segmentation and fusion modules in a joint manner, so that the 2D networks can incorporate 3D information in the training process by learning from the back-propagated gradients of VFN. Another issue involves training VFN more efficiently, *e.g.*, using hard example mining. These topics are left for future research.

Acknoledgements This work was supported by the Lustgarten foundation for pancreatic cancer research. We thank Prof. Seyoun Park, Prof. Wei Shen, Dr. Yan Wang and Yuyin Zhou for instructive discussions.

Chapter 3

Uncertainty-aware multi-view co-training for semi-supervised medical image segmentation and domain adaptation

Although having achieved great success in medical image segmentation, deep learning-based approaches usually require large amounts of well-annotated data, which can be extremely expensive in the field of medical image analysis. Unlabeled data, on the other hand, is much easier to acquire. Semi-supervised learning and unsupervised domain adaptation both take the advantage of unlabeled data, and they are closely related to each other. In this paper, we propose **uncertainty-aware multi-view co-training** (UMCT), a unified framework that addresses these two tasks for volumetric medical image segmentation. Our framework is capable of efficiently utilizing unlabeled data for better performance. We firstly rotate and permute the 3D volumes into multiple views and train a 3D deep network on each view. We then apply co-training by enforcing multi-view consistency on unlabeled data, where an uncertainty estimation of each view is utilized to achieve accurate labeling. Experiments on the NIH pancreas segmentation dataset and a multi-organ segmentation dataset show state-of-the-art performance of the proposed framework on semi-supervised medical image segmentation. Under unsupervised domain adaptation settings, we validate the effectiveness of this work by

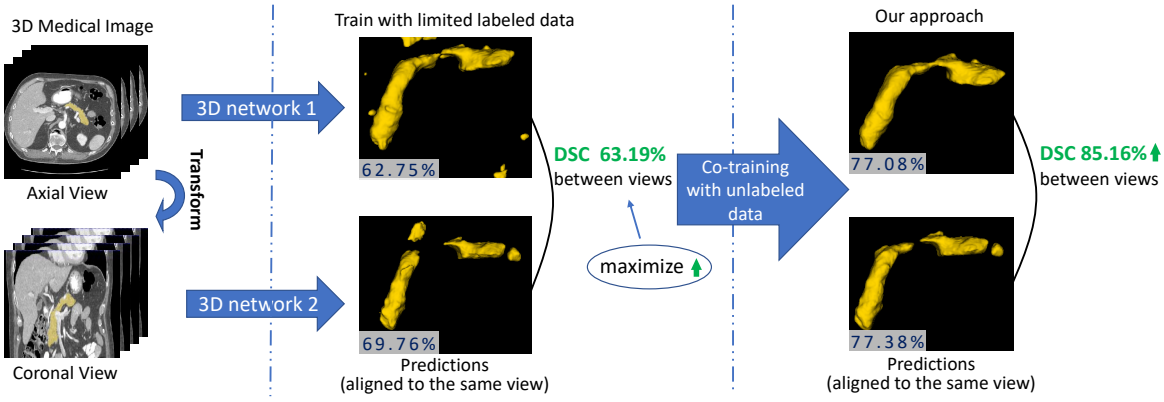


Figure 3.1. An example of our approach for pancreas segmentation (best viewed in color). With limited training data, two 3D networks which are trained on axial and coronal view, respectively, both perform poorly as measured by DSC scores (in dark blue) with ground truth annotations. We observe that the DSC between the two views (in green) is also low, indicating large view differences. With our co-training approach, we minimize the difference between the two predictions on unlabeled data, resulting in significant improvement on each view.

adapting our multi-organ segmentation model to two pathological organs from the Medical Segmentation Decathlon Datasets. Additionally, we show that our UMCT-DA model can even effectively handle the challenging situation where labeled source data is inaccessible, demonstrating strong potentials for real-world applications.

3.1 Introduction

Deep learning has achieved great successes in various computer vision tasks, such as 2D image recognition [1], [23], [37]–[39] and semantic segmentation [24], [40]–[42]. However, deep networks usually rely on large-scale labeled datasets for training. When it comes to medical volumetric data, human labeling can be extremely costly and often requires expert domain knowledge. Medical image segmentation (i.e. the labeling tissues and organs in CTs and MRIs) plays a critical role in biomedical image analysis and surgical planning. Deep learning-based approaches have been widely adopted for this task and have led to state-of-the-art performance [9], [26], [43], [44]. However,

acquiring well-annotated segmentation labels in medical images requires both high-level expertise of radiologists and careful manual labeling of object masks or surface boundaries.

In this paper, we aim to design an approach that can utilize large-scale unlabeled data to improve volumetric medical image segmentation, and is applicable to the scenarios of both semi-supervised learning (SSL) and unsupervised domain adaptation (UDA). SSL and UDA share a common setting by assuming the availability of a labeled training set (denoted as $\mathcal{S}$), as well as an unlabeled one (denoted as $\mathcal{U}$). The difference between the two tasks is that for $\mathcal{S}$ and $\mathcal{U}$ we assume the same distribution in SSL while a larger domain shift is assumed in the UDA setting. Despite such differences, approaches in these two tasks are often closely related. SSL approaches such as self-training [45]–[47], co-training [15], [48] and GAN based methods [49], [50] have been widely applied to UDA [14], [51]–[56], and vice versa.

Inspired by the success of co-training [15] and its application to single 2D images [57], we further extend this idea to 3D volumetric data. Typical co-training requires at least two views (i.e. sources) of the data, of which either should be sufficient to train a classifier on. Co-training minimizes the disagreement by assigning pseudo labels between each view on unlabeled data. [15] further proved that co-training has PAC-like guarantees on semi-supervised learning with an additional assumption that the two views are conditionally independent given the category. Since most computer vision tasks have only one source of data, encouraging view differences is a crucial factor for successful co-training. For example, *deep co-training* [57] trains multiple deep networks to act as different views by utilizing adversarial examples [58] to address this issue. Another aspect of co-training to emphasize is view confidence estimation. In multi-view settings, with growing differences between each view, the quality of each prediction becomes less and less guaranteed and might result in bad pseudo labels that can be harmful if used in the training process. Co-training could

benefit from trusting reliable predictions and degrading the unreliable ones. However, distinguishing reliable and unreliable predictions is challenging for unlabeled data due to lack of ground-truth.

To address the above two important issues, we propose an *uncertainty-aware multi-view co-training* (UMCT) framework, shown in Fig. 3.2. We introduce view differences by exploring multiple viewpoints of 3D data through spatial transformations, such as rotation and permutation. The permutation here is defined as the rearrangements of the coordinate system, such as transpose and flip, and "view" is defined as the transformed input data after permutation. Hence, our multi-view approach naturally applies to analyzing 3D data and can be integrated with the proposed co-training framework. Fig. 3.1 gives an example of the intuition of our approach in two-view scenario. On unlabeled data, we propose to maximize the similarity of the predictions between the two views, resulting in improved segmentation performance on each view. Another key component is the view confidence estimation. We propose to estimate the uncertainty of predictions in each view with Bayesian deep networks by adding dropout in the architectures [59]. A confidence score is computed based on epistemic uncertainty [60], which can act as a weight for each prediction. After propagation through this *uncertainty-weighted label fusion module* (ULF), a set of more accurate pseudo labels can be obtained for each view, which is used as supervision signal for unlabeled data.

UMCT was previously published as a conference paper [61], in which we verified its effectiveness under standard semi-supervised settings on individual organs. In this paper, we extensively validate our approach on more challenging tasks, e.g. multi-organ segmentation. Moreover, we apply our approach to the task of unsupervised domain adaptation, with a labeled source domain and unlabeled target domain. In medical image analysis, this is considered as an important task since we should prefer a model or an approach that has the capability to generalize across datasets from different data

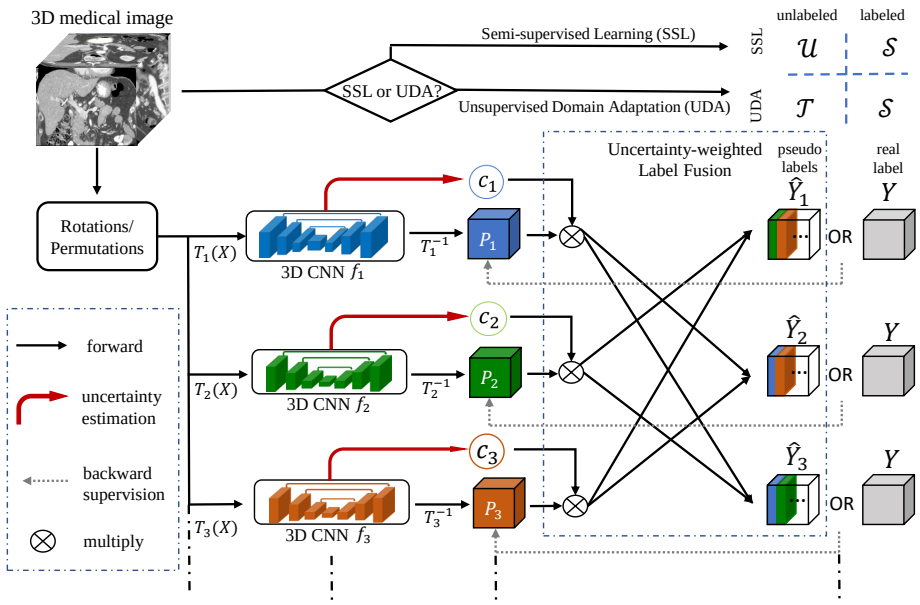


Figure 3.2. Overall framework of **uncertainty-aware multi-view co-training** (UMCT), best viewed in color. UMCT can be applied to either the semi-supervised learning (SSL) task or the unsupervised domain adaptation (UDA) task, both of which include an unlabeled and a labeled subset of data. The overall pipeline is described as follows. The n multi-view inputs of $\mathbf{X}$ are first generated through different transforms $\mathbf{T}$, like rotations and permutations, before being fed into n deep networks with asymmetrical 3D kernels. A confidence score c is computed for each view by uncertainty estimation and acts as the weights to compute the pseudo labels $\hat{Y}$ of other views (Eq. 3.6) after inverse transform $\mathbf{T}^{-1}$ of the predictions. The pseudo labels $\hat{Y}$ for unlabeled data and ground truth Y for labeled data are used as supervisions during training.

sources (e.g. types of machines, acquisition protocols, and characteristics of patients). In addition to the original experiments on NIH pancreas dataset [12], we validate our approach on a multi-organ dataset used in [62] with 8 labeled abdominal organs under semi-supervised settings. We then utilize our co-training approach to adapt the multi-organ model to the Medical Image Decathlon (MSD) [63]) pathological liver and pancreas datasets. We even push our approach one step further by assuming that we only have the source model in the absence of source data. With very simple modifications, our final model UMCT-DA illustrates strong potential on this challenging scenario.

3.2 Related Work

Semi-supervised learning aims at learning models with limited labeled data and a large proportion of unlabeled data [15], [48], [64], [65]. Emerging semi-supervised approaches have been successfully applied to image recognition using deep neural networks [13], [66]–[71]. These algorithms mostly rely on additional regularization terms to train the networks to be resistant to some specific noise. A recent approach [57] extended the co-training strategy to 2D deep networks and multiple views, using adversarial examples to encourage view differences to boost performance.

Semi-supervised medical image analysis. [72] mentioned that current semi-supervised medical analysis methods fall into 3 types - self-training (teacher-student models), co-training (with hand-crafted features) and graph-based approaches (mostly applications of graph-cut based optimization). [47] introduced a deep network based self-training framework with conditional random field (CRF) based iterative refinements for medical image segmentation. [73] trained three 2D networks from three planar slices of the 3D data and fused them in each self-training iteration to get a stronger student model. [74] extended the self-ensemble approach π model [13] with 90-degree rotations making the network rotation-invariant. Generative adversarial network (GAN) based approaches are also popular recently for medical imaging [75]–[77]. Moreover, mixed supervisions [78], [79] combining dense label masks and weak labels like bounding boxes, slice- or image-level labels, etc., is another field of study to alleviate labeling efforts and is related to semi-supervised medical image analysis.

Uncertainty Estimation. Traditional approaches include particle filtering and CRFs [80], [81]. For deep learning, uncertainty is more often measured with Bayesian deep networks [59], [60], [82]. In our work, we emphasize the importance of uncertainty estimation in semi-supervised learning, since most of the training data here is not annotated. We propose to estimate the confidence of each view in our co-training

framework via Bayesian uncertainty estimation.

2D/3D hybrid networks. 2D networks and 3D networks both have advantages and limitations. The former benefits from 2D pre-trained weights and well-studied architectures on natural images, while the latter better explores 3D information utilizing 3D convolution kernels. [83], [84] either uses 2D probability maps or 2D feature maps for building 3D models. [85] proposed a 3D architecture which can be initialized by 2D pre-trained models. Moreover, [8], [86] illustrates the effectiveness of multi-view training on 2D slices, even by simply averaging multi-planar results, indicating complementary latent information exists in the biases of 2D networks. This inspired us to train 3D multi-view networks with 2D initializations jointly using an additional loss function for multi-view networks which encourages each network to learn from one another.

Unsupervised Domain Adaptation. Contrary to semi-supervised learning, domain adaptation problems often contain two datasets that have different distribution. Under unsupervised domain adaptation (UDA) settings, networks are trained from a labeled source domain and an unlabeled target domain. Traditional approaches [87]–[90] align domains with statistical constraints. Recent works [14], [52]–[56], [91], [92] utilizes adversarial training and self-training to adapt feature training between source domain and target domain. In the field of medical image analysis, [93]–[95] have investigated this topic with existing approaches, i.e. adversarial training and self-training.

3.3 Problem Definitions

Before we describe our proposed approach, we firstly discuss the definition and relationships of the three problems, i.e. semi-supervised learning (SSL), unsupervised domain adaptation (UDA) and UDA without data from source domain. Table 3.1

lists the comparison among the three problems.

Semi-supervised learning. Under standard semi-supervised learning (SSL) settings, we denote $\mathcal{S}$ and $\mathcal{U}$ as the labeled and unlabeled dataset, respectively. Let $\mathcal{D} = \mathcal{S} \cup \mathcal{U}$ be the whole available dataset. We denote each labeled data pair as $(\mathbf{X}, \mathbf{Y}) \in \mathcal{S}$ and unlabeled data as $\mathbf{X} \in \mathcal{U}$. We aim to improve performance on a specific task with unlabeled data. When we consider volumetric medical image segmentation, $\mathbf{X}$ is a three-dimensional tensor and the ground truth $\mathbf{Y}$ is a densely-labeled voxel-wise 3D segmentation mask.

Unsupervised domain adaptation (UDA) assumes a labeled source domain dataset $\mathcal{S}$ and an unlabeled target domain dataset $\mathcal{T}$, where distributions of data are different but tasks are identical. Our goal is to achieve relatively high performance of a specific task on the target domain. In medical image analysis, domain gaps can result from differences in imaging modalities (e.g. CT / MRI / PET), qualities or imaging protocols (e.g. various machine types and doses of radiation), types of patients (e.g. healthy or with disease), and combinations thereof. The difference between UDA and SSL only lies in data distributions, so SSL approaches can also be applied to solve UDA problems. In our paper, we illustrate that our proposed approach can effectively handle both problems.

UDA without data from source domain was barely investigated in the literature but is an important challenge to be addressed in the field of medical imaging. Here we assume an available pre-trained model from the source domain and unlabeled data from target domain. Differently from UDA, data from source domain is absent.

In our work, we aim to propose a unified approach that is capable of solving the three tasks described above.

settings	dataset 1	dataset 2	same domain?	pre-trained model?
SSL	labeled	unlabeled	yes	-
UDA	labeled	unlabeled	no	-
UDA w/o $\mathcal{S}$	N/A	unlabeled	no	on dataset 1

Table 3.1. The relationship among the three settings i.e. semi-supervised learning (SSL), unsupervised domain adaptation (UDA) and UDA without source domain (UDA w/o $\mathcal{S}$).

3.4 Uncertainty-aware Multi-view Co-training

In this section, we introduce our framework of *uncertainty-aware multi-view co-training* (UMCT) for semi-supervised segmentation and domain adaptation. UMCT is designed to effectively utilize unlabeled data, which is firstly targeted at semi-supervised segmentation of volumetric medical images. In the following sections, we will explain how they are achieved in our 3D framework: a general mathematical formulation of the approach is shown in Sec 3.4.1; then we demonstrate how to encourage view differences in Sec 3.4.2, and how to compute the confidence of each view by uncertainty estimation in Sec 3.4.3, which are the two factors to boost the performance of co-training. Last but not least, the UMCT-DA model is introduced in Sec 3.4.4 for unsupervised domain adaptation.

3.4.1 Overall Framework

We first consider the task of semi-supervised segmentation for 3D data. Recall that $\mathcal{S}$ and $\mathcal{U}$ are the labeled and unlabeled set, respectively. Each labeled data pair is denoted as $(\mathbf{X}, \mathbf{Y}) \in \mathcal{S}$ and unlabeled data as $\mathbf{X} \in \mathcal{U}$. The ground truth $\mathbf{Y}$ is a voxel-wise segmentation label map which has the same shape as $\mathbf{X}$.

Suppose for each input $\mathbf{X}$, we can generate N different views of 3D data by applying a transformation T_i (rotation or permutation), resulting in multi-view inputs $T_i(X)$, $i = 1, ..., N$. Such operations will introduce a data-level view difference. N models $f_i(\cdot)$, $i = 1, ..., N$ are then trained over each view of data respectively. For $(\mathbf{X}, \mathbf{Y}) \in \mathcal{S}$,

a supervised loss function $\mathcal{L}_{sup}$ is optimized to measure the similarity between the prediction of each view $p_i(\mathbf{X}) = T_i^{-1} \circ f_i \circ T_i(\mathbf{X})$ and $\mathbf{Y}$:

$$\mathcal{L}_{sup}(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^{N} \mathcal{L}(p_i(\mathbf{X}), \mathbf{Y}), \tag{3.1}$$

where $\mathcal{L}$ is a standard loss function for segmentation tasks and $\{p_i(\mathbf{X})\}_{i=1}^{N}$ are the corresponding voxel-wise prediction score maps after inverse rotation or permutation.

For unlabeled data, we make a co-training assumption under a semi-supervised setting. The co-training strategy assumes the predictions on each view should reach a consensus. So the prediction of each model can act as a pseudo label to supervise other views in order to learn from unlabeled data. However, since the prediction of each view is expected to be diverse after encouraging the view differences, the quality of each view's prediction needs to be measured before generating trustworthy pseudo labels. This is accomplished via *uncertainty-weighted label fusion module* (ULF) introduced in Sec 3.4.3. With ULF, the co-training loss for unlabeled data can be formulated as:

$$\mathcal{L}_{cot}(\mathbf{X}, \hat{\mathbf{Y}}_\mathbf{i}) = \sum_{i}^{N} \mathcal{L}(p_i(\mathbf{X}), \hat{\mathbf{Y}}_\mathbf{i}), \tag{3.2}$$

where

$$\hat{\mathbf{Y}}_\mathbf{i} = U_{f_1,..f_n}(p_1(\mathbf{X}), .., p_{i-1}(\mathbf{X}), p_{i+1}(\mathbf{X}), .., p_n(\mathbf{X})) \tag{3.3}$$

is the pseudo label for the i^{th} view, $U_{f_1,..f_n}$ is the ULF computational function, which we will further explain in Sec 3.4.3.

Overall, the combined loss function is:

$$\sum_{(\mathbf{X},\mathbf{Y})\in\mathcal{S}} \mathcal{L}_{sup}(\mathbf{X}, \mathbf{Y}) + \lambda_{cot} \sum_{\mathbf{X}\in\mathcal{U}} \mathcal{L}_{cot}(\mathbf{X}, \hat{\mathbf{Y}}_\mathbf{i}). \tag{3.4}$$

where λ_{cot} is a tunable weight coefficient.

3.4.2 Encouraging View Differences

A successful co-training requires the "views" to be different in order to learn complementary information in the training procedure. In our framework, several techniques

Algorithm 1: Uncertainty-aware Multi-view Co-training

Input:
Labeled dataset $\mathcal{S}$ & Unlabeled dataset $\mathcal{U}$
uncertainty-weighted label fusion module (ULF) $U_{f_1,..f_n}(\cdot)$
Output:
Model of each view $f_1, ..f_n$

while stopping criterion not met **do**
  Sample batch $b_l = (x_l, y_l) \in \mathcal{S}$ and batch $b_u = (x_u) \in \mathcal{U}$
  Generate multi-view inputs $T_i(x_l)$ and $T_i(x_u)$, $i \in \{1, .., N\}$
  for i **in** all views **do**
    Compute predictions for each view and apply inverse rotation or permutation
    $p_i(x_l) \leftarrow T_i^{-1} \circ f_i \circ T_i(x_l)$
    $p_i(x_u) \leftarrow T_i^{-1} \circ f_i \circ T_i(x_u)$
  for i **in** all views **do**
    Compute pseudo labels for x_u with ULF
    $\hat{y}_i \leftarrow U_{f_1,..f_n}(p_1(x_u), .., p_{i-1}(x_u), p_{i+1}(x_u), .., p_n(x_u))$
  $\mathcal{L}_{sup} = \frac{1}{|b_l|} \sum_{(x_l,y_l)\in b_l} [\sum_i^N \mathcal{L}(p_i(x_l), y_l)]$
  $\mathcal{L}_{cot} = \frac{1}{|b_u|} \sum_{(x_u)\in b_u} [\sum_i^N \mathcal{L}(p_i(x_u), \hat{y}_i)]$
  $\mathcal{L} = \mathcal{L}_{sup} + \lambda_{cot}\mathcal{L}_{cot}$
  Compute gradient of loss function $\mathcal{L}$ and update network parameters $\{\theta_i\}$ by back propagation
return $f_1, ..f_n$

are proposed to encourage view differences, both at the data level and the feature level of the neural networks.

3D multi-view generation. As stated above, in order to generate multi-view data, we transpose $\mathbf{X}$ into multiple views by rotations or permutations[1] $\mathbf{T}$. For three-view co-training, these can correspond to the coronal, sagittal and axial views in medical imaging, which matches the multi-planar reformatted views that radiologists typically use to analyze the image. Such operation is a natural way to introduce data-level view difference.

Asymmetric 3D kernels and 2D initialization. The co-training assumption encourages models to make similar predictions on both $\mathcal{S}$ and $\mathcal{U}$, which potentially can lead to collapsed neural networks mentioned in [57], a phenomenon that results in

[1] A permutation rearranges the dimensions of an array in a specific order.

a sudden and significant drop in validation accuracy during training of co-training algorithms. In our multi-view settings, this could also happen when the models from different views only learn the permutation or rotation of the kernels, resulting in exactly the same learned feature representation despite the view-point difference. To address this problem, we further encourage view difference at the feature level by designing a task-specific model. We propose to use asymmetric 3D models initialized with 2D pre-trained weights as the backbone network of each view to encourage diverse features for each view learning. In practice, we modify the symmetric 3D convolutional kernels $n \times n \times n$ into $n \times n \times 1$ for each branch after the permutation to avoid learning symmetrical representations among views. This structure also makes the model convenient to be initialized with 2D pre-trained weights but fine-tuned in a 3D fashion.

3.4.3 Compute Reliable Psuedo Labels for Unlabeled Data with Uncertainty Estimation

Encouraging view difference means enlarging the variance of each view's prediction $var(p_i(\mathbf{X}))$. This raises the question of which view we should trust most on unlabeled data during co-training. Bad predictions from one view may hurt the training procedure of other views through pseudo-label assignments. Meanwhile, encouraging to trust a good prediction as a "strong" label from co-training will boost the performance, and lead to improved performance of overall semi-supervised learning. Instead of assigning a pseudo-label for each view directly from the predictions of other views, we propose an adaptive approach, namely *uncertainty-weighted label fusion module* (ULF), to fuse the outputs of different views. ULF is built up of all the views, takes the predictions of each view as input, and then outputs a set of pseudo labels for each view.

Motivated by uncertainty measurements in Bayesian deep networks, we measure the uncertainty of each view branch for each training sample after turning our model into a

Bayesian deep network by adding dropout layers. Between the two types of uncertainty candidates – aleatoric and epistemic uncertainties, we choose to compute the epistemic uncertainty that is driven by the lack of training data [60]. Such measurement fits the semi-supervised learning goal: to improve the model's generalizability by exploring unlabeled data. Suppose y is the output of a Bayesian deep network, then the epistemic uncertainty can be estimated as:

$$U_e(y) \approx \frac{1}{K}\sum_{k=1}^{K} \hat{y_k}^2 - (\frac{1}{K}\sum_{k=1}^{K} \hat{y_k})^2, \tag{3.5}$$

where $\{\hat{y}_k\}_{k=1}^{K}$ are a set of sampled outputs. These sampled outputs are obtained by feeding the same input volume into the sub-network defined by K different random dropout configurations [60]. The voxel-wise epistemic uncertainty is estimated as the statistical variance of the K predictions. More details are available in Sec 3.4.5.

With a transformation function $\mathbf{h}(\cdot)$, we can transform the uncertainty score into a confidence score $\mathbf{c}(y) = \mathbf{h}(U_e(y))$. In practice, we simply define $\mathbf{h}(U_e(y)) = 1/U_e(y)$. After normalization over all views, the confidence score will act as the weight for each prediction to assign as a pseudo label for other views. The pseudo label $\hat{\mathbf{Y}}_\mathbf{i}$ assigned for a single view i can be formulated as

$$\hat{\mathbf{Y}}_\mathbf{i} = \frac{\sum_{j\neq i}^{N} \mathbf{c}(p_j(\mathbf{X}))p_j(\mathbf{X})}{\sum_{j\neq i}^{N} \mathbf{c}(p_j(\mathbf{X}))}. \tag{3.6}$$

Thus the pseudo label $\hat{\mathbf{Y}}_\mathbf{i}$ for view i is computed from predictions from all the other views.

3.4.4 UMCT-DA model for unsupervised domain adaptation

Standard unsupervised domain adaptation (UDA)

We extensively validate our approach on unsupervised domain adaptation setting, where the labeled source domain $\mathcal{S}$ and the unlabeled target domain $\mathcal{T}$ are available for training. The task is shared between the two domains and the ultimate goal is to

achieve good performance on the target domain test data. Despite the domain shift in labeled and unlabeled data, the overall settings of semi-supervised learning (SSL) and unsupervised domain adaptation (UDA) are the same. Hence, we can directly apply our UMCT to solve this problem. The optimization objective can be modified as the follows :

$$\mathcal{L}_{UDA} = \sum_{(\mathbf{X},\mathbf{Y})\in\mathcal{S}} \mathcal{L}_{sup}(\mathbf{X},\mathbf{Y}) + \lambda_{cot} \sum_{\mathbf{X}\in\mathcal{T}} \mathcal{L}_{cot}(\mathbf{X}). \tag{3.7}$$

where $\mathcal{S}$ is the labeled source domain and $\mathcal{T}$ is the unlabeled target domain.

UDA without source domain data

Standard UDA methods usually require the existence of source domain data to allow joint training while doing adaptation to the target domain. Here we consider a more challenging setting where source domain data is unavailable and only deep network model (denoted as $\mathbf{M}_{\mathcal{S}}$) pre-trained on source domain is available. In our co-training framework, when source data is unavailable, we can still finetune $\mathbf{M}_{\mathcal{S}}$ with $\mathcal{L}_{cot}$ by iteratively refining pseudo labels. The objective function for UDA without source domain data, namely **UMCT-DA**, can be formulated as:

$$\mathcal{L}_{UDA} = \sum_{\mathbf{X}\in\mathcal{T}} \lambda_{cot}\mathcal{L}_{cot}(\mathbf{X}). \tag{3.8}$$

3.4.5 Implementation Details

Network Structure. In practice, we build an encoder-decoder network based on ResNet-18 [1], and modify it into a 3D version. For the encoder part, the first 7×7 convolution layer is extended to $7 \times 7 \times 3$ kernels for low-level 3D feature extraction similar to [85]. All other 3×3 convolution layers are simply changed into $3 \times 3 \times 1$ that can be trained as a 3D convolution layer. In the decoder part, we adopt 3 skip connections from the encoder followed by 3D convolutions to give low-level cues for more accurate boundary prediction needed in segmentation tasks.

Uncertainty-weighted Label Fusion. In terms of view confidence estimation, we modify the network into a Bayesian deep network by adding dropouts. We sample $K = 10$ outputs for each view and compute voxel-wise epistemic uncertainty. Since we are using Dice loss [26], a common loss function for medical image segmentation which is computed on the image level, an image-wise uncertainty estimation is most suitable. We thus sum over the whole volume to estimate the uncertainty for each view. We then simply use the reciprocal for the confidence transformation function $\mathbf{h}= \frac{1}{\mathbf{c}}$ to compute the confidence score. The resulting pseudo label assigned for each view is a weighted average of all predictions of multiple views based on the normalized confidence score.

Loss Function. We extend the Dice loss [26] for multi-class targets as our training objective function:

$$\mathcal{L}_{Dice} = \frac{1}{D}\sum_{d=0}^{D}(1 - \frac{2\sum_{i=1}^{N} y_i^d \hat{y}_i^d}{\sum_{i=1}^{N}(y_i^d)^2 + \sum_{i=1}^{N}(\hat{y}_i^d)^2}), \tag{3.9}$$

Data Pre-Processing. All the training and testing data are firstly re-sampled to an isotropic volume resolution of 1.0 mm for each axis. Data intensities are normalized to have zero mean and unit variance. We adopt patch-based training, and sample training patches of size 96^3 with 1:1 ratio between foreground and background.

Training. Our training algorithm is shown in Algorithm 1. We firstly train the views separately on the labeled data and then conduct our co-training by fine-tuning the weights. The stochastic gradient descent (SGD) optimizer is used in both stages. In the view-wise training stage, a constant learning rate policy at 7×10^{-3}, momentum at 0.9 and weight decay of 4×10^{-5} for 20k iterations is used. In the co-training stage, we adopt a constant learning rate policy at 1×10^{-3} and train for 5k iterations. The parameter $\lambda_{cot} = 0.2$ resulted in the best performance which we report here. The batch size is 20 in co-training, among which 4 images are labeled and 16 are unlabeled, maintaining a ratio of labeled and unlabeled to be 1:4. Our framework is implemented

Method	Backbone	10% lab	20% lab
Supervised	3D ResNet-18	66.75	75.79
DMPCT [73]	2D ResNet-101	63.45	66.75
DCT [57] (2v)	3D ResNet-18	71.43	77.54
TCSE [74]	3D ResNet-18	73.87	76.46
Ours (2 views)	3D ResNet-18	75.63	79.77
Ours (3 views)	3D ResNet-18	77.55	80.14
Ours (6 views)	3D ResNet-18	77.87	80.35
Ours (ensemble)	3D ResNet-18	**78.77**	**81.18**

Table 3.2. Comparison to other semi-supervised approaches on NIH dataset (DSC, %). Note that we use the same backbone network as [74] [57]. Here, "2v" means two views. For our approach, we report the average of all single views' DSC score for a fair comparison (2 views to 6 views), as well as multi-view ensemble results. "10% lab" and "20% lab" mean the percentage of labeled data used for training.

in PyTorch. For 3D ResNet-18 on NIH dataset, the whole co-training procedure takes ∼24 hours on one single NVIDIA Titan RTX GPU with 24 GB memory. In our implementation, training occupies ∼15GB GPU memory in total.

Testing. In the testing phase, there are two choices to finalize the output results: either to choose one single view prediction or to ensemble the predictions of the multi-view outputs with majority voting. We will report both results in subsequent sections for fair comparisons with the baselines since the multiple view networks can be thought of being similar to the ensemble of several single view models. The experimental results show that our model improves the performance in both settings (single view and multi-view ensemble) over all the other approaches. We use sliding-window testing and re-sample our testing results back to the original image resolution to obtain the final results. Testing time for each case ranges from 1 minute to 5 minutes depending on the size of the input volume.

3.5 Experiments

In this section, we first evaluate our framework under semi-supervised settings on the NIH pancreas segmentation dataset [12] with cases from a healthy patient population (e.g. kidney donors[2]); and an multi-organ segmentation dataset [62] with eight abdominal organs [96] with conditions mostly unrelated to the organs of interest (e.g. colorectal cancer or ventral hernia[3]). We will provide detailed experiments, including ablation studies, on the former dataset. Note that the volumes come from different patients in each dataset and were separated at the patient-level for the different training, validation and testing splits. Next, we validate the capability of our approach on the task of unsupervised domain adaptation, which is critical but under-investigated in the field of medical image analysis. The multi-organ segmentation dataset serves as source data. The targets of adaptation include two pathological organ datasets i.e. pancreas and liver datasets in the Medical Segmentation Decathlon (MSD) [63], which both can include tumors in their respective organs. More strictly, we also evaluate our approach under the situation where source data is inaccessible (UDA without source data).

3.5.1 NIH Pancreas Segmentation Dataset

The NIH pancreas segmentation dataset contains 82 abdominal CT volumes. The width and height of each volume are 512, while the axial view slice number can vary from 181 to 466. Under semi-supervised settings, the dataset is randomly split into 20 testing cases and 62 training cases. We report the results of 10% labeled training cases (6 labeled and 56 unlabeled), 20% labeled training cases (12 labeled and 50 unlabeled) and 100% labeled training cases.

[2] https://wiki.cancerimagingarchive.net/display/Public/Pancreas-CT
[3] https://www.synapse.org/#!Synapse:syn3193805/wiki/217789

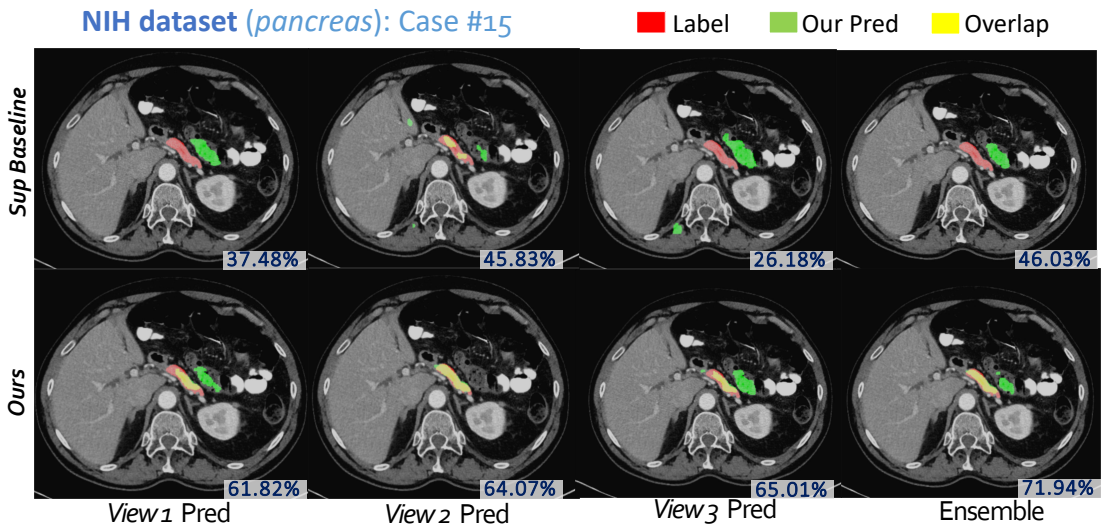


Figure 3.3. 2D visualizations for one example of NIH pancreas segmentation dataset 10% labeled data setting. The first row is the supervised baseline and the second row is the prediction after our 3-view co-training. DSC scores are largely improved. Best viewed in color.

3.5.1.1 Results

In Table 3.2, we first report the average of all single views' DSC score for a fair comparison (2 views to 6 views, last 2-4 rows), which can be viewed as the average performance of one single view model. Then we report the multi-view ensemble results (6 view ensemble, last row), where we align the multi-view prediction maps to the same view (axial) and average the prediction maps at each pixel to make a final prediction. For 2-view co-training, we use the axial and coronal views. For 3-view co-training, we use the axial, coronal and sagittal view. For 6-view co-training, we use the axial, coronal and sagittal view as well as the horizontal flip version of the three views ($3 \times 2 = 6$). The first row is the supervised training results, using only labeled data and trained on the axial view. The segmentation accuracy is evaluated by Dice-Sørensen coefficient (DSC). A large margin improvement over the fully supervised baselines in terms of single view performance can be observed, proving that our approach effectively leverages the unlabeled data. A Wilcoxon signed-rank test comparing to the supervised

baseline's results (20% labeling) shows significant improvements of our approach with a p-value of 0.0022. Fig. 3.3 shows 3 cases in 2D and 3D with ITK-SNAP [97]. In addition, our model is compared with the state-of-the-art semi-supervised approach of deep co-training [57] and recent semi-supervised medical segmentation approaches. In particular, we compare to [74] who extended the π model [13] with transformation consistent constraints; and [73] who extended the self-training procedure by iteratively updating pseudo labels on unlabeled data using a fusion of three 2D networks trained on cross-sectional views. The results reported in Table 3.2 are based on our careful re-implementations in order to allow a fair comparison.

The implementations of [57] and [74] are operated on the axial view of our single view branch with the same backbone structure (our customized 3D ResNet-18 model). Our co-training approach achieve about 4% gain in the 10% labeled and 90% unlabeled settings. We also find that improvements of other approaches are small in the 20% settings (only 1% compared to the baseline), while ours still is capable to achieve a reasonable performance gain with the growing number of labeled data. For [73] with a 2D approach, their experiment is conducted on 50 labeled cases. We modify their backbone network (FCN [24]) into DeepLab v2 [40], in order to fit our stricter settings (6 and 12 labeled cases). This modification leads to an improvement of 3% in 100% fully supervised training (from 73% to 76%). Their approach outputs the result after using an ensemble with majority voting of three slice-wise 2D models obtained from their semi-supervised training approach..

Since the main difference in two-view learning between our approach and [57] is the way of encouraging view differences, the results illustrate the effectiveness of our multi-view analysis combined with asymmetric feature learning on 3D co-training. With more views, our uncertainty-weighted label fusion can further improve co-training performance. We will report ablation studies later in this section.

3.5.1.2 Analysis and ablation studies

Data utilization efficiency

We perform a study on data utilization efficiency of our approach compared to the baseline fully-supervised network (3D ResNet-18). Fig. 3.4 shows the performance change according to labeled data proportion on NIH pancreas segmentation. From the plot, one can see that when labeled data is over 80%, simple supervised training (with 3D ResNet-18) suffices. Note that our approach with 20% labeled data (DSC 80.35%) performs better than 60% supervised training (DSC 78.95%). At such a performance, our approach can save $\sim$ 70% of the labeling efforts.

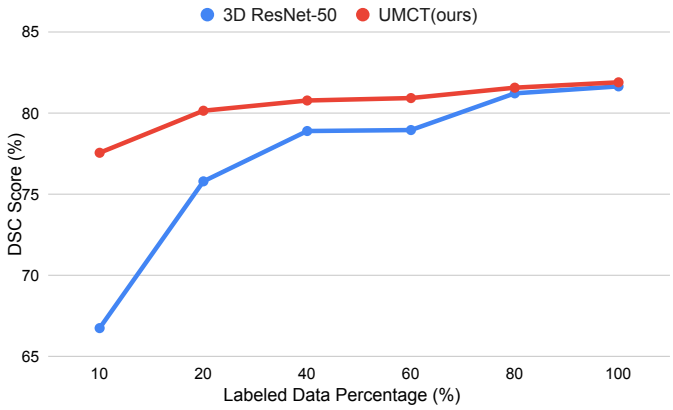


Figure 3.4. Performance plot of our semi-supervised approach over the fully-supervised baseline on different labeled data ratio.

Effect of backbone structure

Our backbone selection (2D-initialized, heavily asymmetric 3D architecture) will introduce 2D biases in the training phase while benefiting from such 2D pre-trained models. We have claimed that we can utilize the complementary information from 3-view networks while exploring the unlabeled data with UMCT. We give an ablation study on the network structure, which contains a V-Net [26], a common 3D segmentation network with all symmetrical kernels in all dimensions. Such network also shares a similar amount of parameters with our customized 3D ResNet-18, see

Table 3.3. The results of V-Net show that our multi-view co-training can be generally and successfully applied to 3D networks. Although the results of fully supervised parts are similar, our ResNet-18 outperforms V-Net by more than 1%, illustrating that our asymmetric design, encouraging view differences, brings advantages over traditional 3D deep networks.

Backbone	Params	MACs	10% Sup	Ours
V-Net	9.44M	41.40G	66.97	76.89
3D ResNet-18	11.79M	17.08G	66.76	77.55
3D ResNet-50	27.09M	23.03G	67.96	**78.74**

Table 3.3. Ablation studies on backbone structures (3 views UMCT). "Params" is short for parameters and "MACs" is short for multiply–accumulate operations. "10% Sup" means supervised training with 10% labeled data. A Wilcoxon signed-rank test reveals significant improvements ($p << 0.01$) of our 3D ResNets over V-Net in the last column, illustrating our asymmetrical design is beneficial for our co-training method.

Uncertainty-weighted label fusion (ULF)

ULF acts as an important role in pruning out bad predictions and keeping good ones as supervision to train other views. Table 3.4 gives the single view results in multiple views experiments. The performance becomes better with more views. For two views, ULF is not applicable since we can only obtain one view prediction as a pseudo label for the other view. For three views and six views, ULF helps boost the performance, illustrating the effectiveness of our proposed approach for view confidence estimation.

Views	DSC(%)
2 views	75.63
3 views	76.49
3 views + ULF	**77.55**
6 views	76.94
6 views + ULF	**77.87**

Table 3.4. On uncertainty-weighted label fusion (ULF) with difference views in training (10% labeled data, 3D ResNet-18).

Experiment setups	spleen	l.kidney	gallbladder	esophagus	liver	stomach	pancreas	duodenum
Supervised (upper bound)	94.20	93.90	71.89	66.74	94.78	88.60	81.46	71.29
10% lab	88.46	90.88	42.77	52.41	91.33	76.50	69.63	52.52
10% lab+90% unlab (ours)	**91.14**	**92.35**	**58.29**	**57.61**	**92.23**	**79.67**	**73.86**	**57.50**
20% lab	92.77	92.29	63.60	61.84	93.95	82.56	75.60	60.26
20% lab+80% unlab (ours)	92.80	**92.99**	**66.29**	**65.01**	93.93	**83.67**	**77.91**	**63.34**

Table 3.5. Experimental results for semi-supervised learning on a multi-organ dataset under four fold cross-validation. "lab" is short for "labeled" and "unlab" is short for "unlabeled". Supervised results (first row) uses 100% labeled training data in the training set, which is the upper bound but requires 100% annotation. 10% lab means we only use 10% training data with annotation for supervised training. 10%lab + 90% unlab (ours) means we use 10% labeled data and 90% unlabeled data for our co-training method. Results are reported via 4-fold cross-validation. Numbers in **bold** indicate significant improvement over supervised counterparts by Wilcoxon signed rank tests ($p << 0.01$).

3.5.2 Multi-organ Segmentation Dataset

Next, we validate our approach on multi-organ datasets. The dataset we use is a multi-organ re-annotated version from [62], combining two public datasets - The Cancer Image Archive (TCIA) Pancreas-CT data set [12] and Beyond the Cranial Vault (BTCV) Abdomen data set [96]. We perform 4-fold cross validation on 90 cases in total. In each fold, we then randomly split the training cases into our labeled set $\mathcal{S}$ and unlabeled set $\mathcal{U}$. We train our models on different labeled data ratio 10%, 20%, which approximately corresponds to (7,81), (13,75) of (labeled, unlabeled) data pairs and validate on ∼22 cases in each fold. Results are shown in Table 3.5 and an example is shown in Fig 5.1.

Our approach improves consistently over almost every organ under every labeled-unlabeled ratio of data. The results illustrate the ability of our approach to handle the situation of complex multi-organ settings.

3.5.3 Unsupervised domain adaptation from multi-organ segmentation to MSD Dataset

We aim to unsupervisedly adapt a model trained on TCIA multi-organ dataset to pancreas and liver cases in Medical Decathlon Challenge [63] that can exhibit

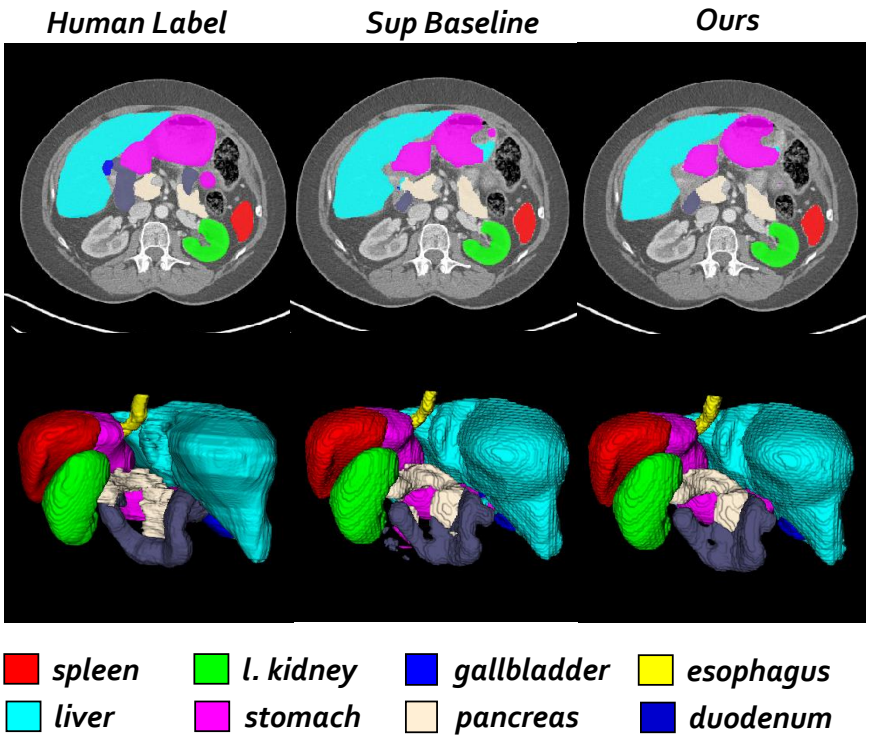


Figure 3.5. An example of semi-supervised multi-organ segmentation.

tumors. The target domains contain a shift from the source domain because of the differences in (i) image quality and contrast, and (ii) textures due to the existence of pancreatic/hepatic tumors.

MSD pancreas dataset contains 282 CT scans in portal venous phase, all of which are pathological cases with pancreas and tumor annotation. We randomly split the whole dataset into 200 cases for training (without label) and 81 cases for validation. Since the source domain (multi-organ dataset) only contains healthy pancreas, we aim at segmenting the whole pancreas region (combining pancreas and tumor together). For MSD liver dataset, we aim at segmenting the whole liver region as well, with a random split of 100 training cases (unlabeled) and 31 cases for validation. 118 out of 131 cases contains hepatic tumor.

Table 3.6 shows the results of unsupervised domain adaptation experiments. The first row is the segmentation performance (in terms of DSC) on pancreas and liver of

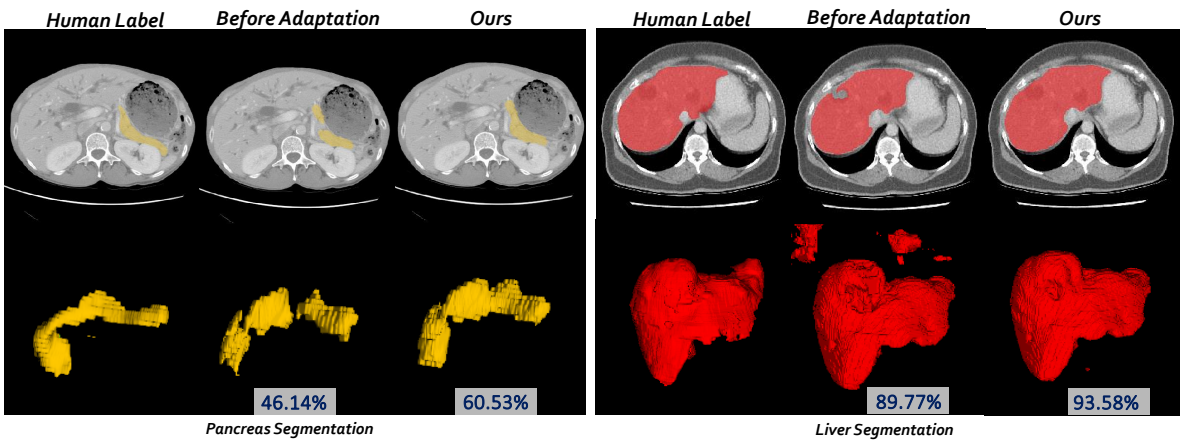


Figure 3.6. 2D and 3D visualizations for unsupervised domain adaptation of pancreas (left) and liver segmentation (right).

the original multi-organ validation set. From the second row to the last, the results are DSC scores on MSD liver / pancreas validation set. In standard UDA settings (UMCT w/ source), significant improvements are achieved with our approach (1.12% in liver and 4.70% in pancreas), compared to source only version (direct Test on MSD). Due to the superior performance of self-training based approaches [55], [56] for unsupervised domain adaptation, we implement a vanilla self-training method under our settings (denoted as "Self-training"). We first test the model on the unlabeled set and then use the prediction as pseudo labels to train on the whole data set. We iterate these two steps every 1k iterations and trains for 5k iterations in total, which is in line with the proposed co-training scheme. We also implement another baseline approach (AdaptSegNet [53], denoted as "Adv training"), which applies adversarial training onto the predicted masks of semantic segmentation. The segmentation network serves as the generator to output segmentation masks with segmentation loss and tries to fool a patch-based discriminator (a 3D version of the discriminator used in AdaptSegNet [53]) with GAN loss [58]. The discriminator is also trained jointly to distinguish between the predicted mask and the ground-truth mask on unlabeled data. Our approach significantly outperforms all of the baselines. We also show one example for each

Train	Test	Method	liver	pancreas
MO (L)	MO	Supervised	95.59	81.69
MO (L)	MSD	Supervised	92.78	70.23
MO (L) + MSD (U)	MSD	Adv training	93.35	71.23
		Self-training	92.67	71.38
		UMCT	**93.90**	**74.93**
MO model + MSD (U)	MSD	UMCT-DA	92.98	74.38

Table 3.6. Experiments of unsuperivsed domain adaptation (UDA). The source domain is Multi-organ dataset (denoted as "MO") and target domains are MSD liver dataset and pancreas dataset. "L" represents this dataset is labeled and "U" means the opposite.

organ in Fig 3.6.

The last row gives the results in the absence of source domain data. Under such condition, only a pre-trained source domain model and unlabeled target domain data are available. Our UMCT-DA model (last row) is able to solve this problem by only using the co-training loss L_{cot} to train on the target dataset, and achieve comparable results with standard UDA settings, even without source domain data.

3.6 Discussions

3.6.1 Impact on large-scale benchmarks

Under fully supervised training, our team NVDLMED was ranked the 3^{rd} place in the first phase and the **2^{nd} place in the final validation phase of Medical Segmentation Decathlon Challenge** [63] (challenge leaderboard available[4]). We applied our 3-view co-training framework taken from axial, coronal and sagittal views to ten medical image segmentation tasks simultaneously. The winning team [98] applied heavy model selection and ensemble by cross-validation on the training set, while we used a fixed framework without complicated data augmentation. Although not originally targeted at improving the performance of fully supervised training, our

[4] http://medicaldecathlon.com/results.html

approach still illustrated the effectiveness and robustness of co-training from multiple views.

3.6.2 Magnitude of domain shift

In this work, domain shift mainly lies in various sources of CT scans and the pathological/healthy status of abdominal organs. We consider it a reasonable domain shift from different CT datasets originating from different hospitals and patient populations. Typically, this means that when directly transferring a model from one to another (TCIA to MSD in our case), the performance drops significantly (liver 95% to 92%, pancreas 81% to 70% average Dice). While this shift is relatively small compared to, for example, cross modality testing (say CT to MRI), it is unacceptable when considering these models for potential clinical applications. Considering the importance of this topic, we shed light on how well our semi-supervised approach performs on UDA tasks, given their similarity (discussed in Sec 3.3). Other types of domain shifts of medical images, though not investigated in this paper, are also of great importance. Investigation of domain adaptation under larger domain shifts such as modality changes (e.g. CT to MRI adaptation), contrast and resolution issues remains an active research topic.

3.7 Summary & Conclusion

In this paper, we presented *uncertainty-aware multi-view co-training* (UMCT), aimed at semi-supervised learning and domain adaptation. We extended dual view co-training and deep co-training into 3D volumetric image data by analysing from different viewpoints, then estimating uncertainty and finally enforcing multi-view consistency on large scale unlabeled data. Our approach was first validated on NIH pancreas dataset, where we outperformed other approaches by a large margin. We further applied our approach to multi-organ datasets and found significant improvements for each

organ. Finally, we adapted the multi-organ dataset to MSD pathological pancreas and liver in an unsupervised manner. Our UMCT-DA model achieved good performance even in the absence of source domain data, illustrating strong potential for real-world applications in medical image segmentation.

In the future, we plan to conduct further research in the following aspects. Currently the views of co-training are fixed and pre-defined, so one feasible idea is to incorporate more views and random views. This could increase the robustness of our model and lead to better performance. For domain adaptation, we will also try to explore co-training based approaches on other types of domain shifts including but not limited to image modality changes and contrast variants. We believe co-training based approaches will make a contribution to large scale medical image analysis with limited human annotations.

Chapter 4

Synthesize then Compare: Detecting Failures and Anomalies for Semantic Segmentation

The ability to detect failures and anomalies are fundamental requirements for building reliable systems for computer vision applications, especially safety-critical applications of semantic segmentation, such as autonomous driving and medical image analysis. In this paper, we systematically study failure and anomaly detection for semantic segmentation and propose a unified framework, consisting of two modules, to address these two related problems. The first module is an image synthesis module, which generates a synthesized image from a segmentation layout map, and the second is a comparison module, which computes the difference between the synthesized image and the input image. We validate our framework on three challenging datasets and improve the state-of-the-arts by large margins, *i.e.*, 6% AUPR-Error on Cityscapes, 7% Pearson correlation on pancreatic tumor segmentation in MSD and 20% AUPR on StreetHazards anomaly segmentation.

4.1 Introduction

Deep neural networks [1], [23], [37], [39] have achieved great success in various computer vision tasks. However, when they come to real world applications, such as autonomous

driving [99], medical diagnoses [100] and nuclear power plant monitoring [101], the safety issue [102] raises tremendous concerns particularly in conditions where failure cases have severe consequences. As a result, it is of enormous value that a machine learning system is capable of detecting the failures, *i.e.*, wrong predictions, as well as identifying the anomalies, *i.e.*, out-of-distribution (OOD) cases, that may cause these failures.

Previous works on failure detection [103]–[105] and anomaly (OOD) detection [17], [106]–[109] mainly focus on classifying small images. Although failure detection and anomaly detection for semantic segmentation have received little attention in the literature so far, they are more closely related to safety-critical applications, *e.g.*, autonomous driving and medical image analysis. The objective of failure detection for semantic segmentation is not only to determine whether there are failures in a segmentation result, but also to locate where the failures are. Anomaly detection for semantic segmentation, *a.k.a* anomaly segmentation, is related to failure detection, and its objective is to segment anomalous objects or regions in a given image.

In this paper, our goal is to build a reliable alarm system to address failure detection for semantic segmentation (Fig. 7.1(i)) and anomaly segmentation (Fig. 7.1(ii)). Unlike image classification outputs only a single image label, semantic segmentation outputs a structured semantic layout. Thus, this requires that the system should be able to provide more detailed analysis than those for image classification, *i.e.*, pixel-level error/confidence maps. Some previous works [60], [105], [110] directly applied the failure/anomaly detection strategies for image classification pixel by pixel to estimate a pixel-level error map, but they lack the consideration of the structured semantic layout of a segmentation result.

We propose a unified framework to address failure detection and anomaly detection for semantic segmentation. This framework consists of two components: an image synthesis module, which synthesizes an image from a segmentation result to reconstruct

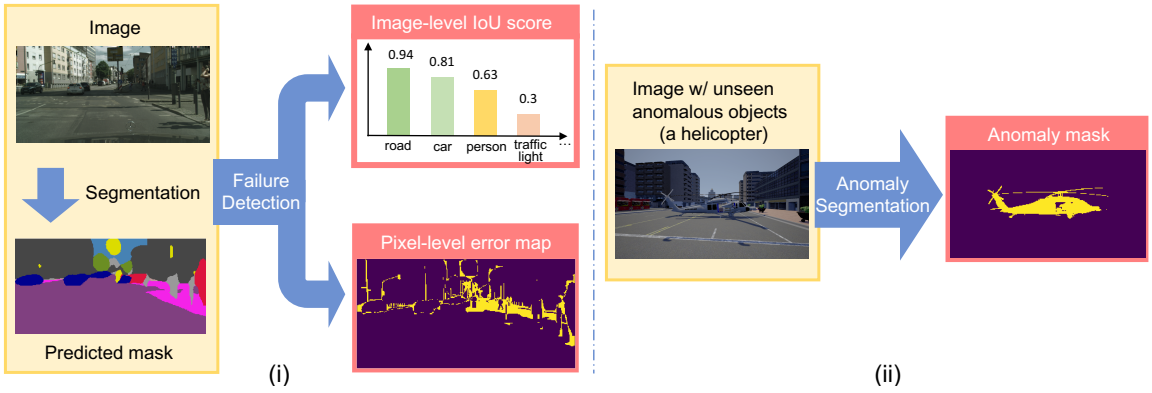


Figure 4.1. We aim at addressing two tasks: (i) failure detection, *i.e.*, image-level per-class IoU prediction (top left) and pixel-level error map prediction (bottom left) (ii) anomaly segmentation i.e. segmenting anomalous objects (right middle).

its input image, *i.e.*, a reverse procedure of semantic segmentation, and a comparison module which computes the difference between the reconstructed image and the input image. Our framework is motivated by the fact that the quality of semantic image synthesis [18], [111], [112] can be evaluated by the performance of segmentation network. Presumably the converse is also true, the better is the segmentation result, the closer a synthesized image generated from the segmentation result is to the input image. If a failure occurs during segmentation, for example, if a person is mis-segmented as a pole, the synthesized image generated from the segmentation result does not look like a person and an obvious difference between the synthesized image and the input image should occur. Similarly, when an anomalous (OOD) object occurs in a test image, it would be classified as any possible in-distribution objects in a segmentation result, and then appear as in-distribution objects in the synthesized image generated from the segmentation result. Consequently, the anomalous object can be identified by finding the differences between the test image and the synthesized image. We refer to our framework as SynthCP, for "synthesize then compare".

We model this synthesis procedure by a semantic-to-image conditional GAN (cGAN) [18], which is capable of modeling the mapping from the segmentation layout

space to the image space. This cGAN is trained on label-image pairs. Given the segmentation result of an input image obtained by an semantic segmentation model, we apply the trained cGAN to the segmentation result to generate a reconstructed image. Then, the reconstructed image and the input image are fed into the comparison module to identify the failures/anomalies. The comparison module is designed task-specifically: For failure detection, the comparison module is modeled by a Siamese network, outputting both image-level confidences and pixel-level confidences; For anomaly segmentation, the comparison module is realized by computing the distance defined on the intermediate features extracted by the semantic segmentation model.

We validate SynthCP on the Cityscapes street scene dataset, a pancreatic tumor segmentation dataset in the Medical Segmentation Decathlon (MSD) challenge and the StreetHazards dataset, and show its superiority to other failure detection and anomaly segmentation methods. Specifically, we achieved improvements over the state-of-the-arts by approximately 6% AUPR-Error on Cityscapes pixel-level error prediction, 7% Pearson correlation on pancreatic tumor DSC prediction and 20% AUPR on StreetHazards anomaly segmentation.

We summarize our contribution as follows:

- To the best of our knowledge, we are the first to systematically study failure detection and anomaly detection for semantic segmentation
- We propose a unified framework, SynthCP, which enjoys the benefits of a semantic-to-image conditional GAN, to address both of the two tasks.
- SynthCP achieves state-of-the-art failure detection and anomaly segmentation results on three challenging datasets.

4.2 Related Work

In this section, we first review the topics closely related to failure detection and anomaly segmentation, such as uncertainty/confidence estimation, quality assessment and out-of-distribution (OOD) detection. Then, we review generative adversarial networks (GANs), which serves a key module in our framework.

Uncertainty estimation or confidence estimation has been a hot topic in the field of machine learning for years, and can be directly applied to the task of **failure detection**. Standard baselines was established in [103] for detecting failures in classification where maximum softmax probability (MSP) provides reasonable results. However, the main drawback of using MSP for confidence estimation is that deep networks tend to produce high confidence predictions [105]. Geifman *et al.* [113] controled the user specified risk-level by setting up thresholds on a pre-defined confidence function (e.g. MSP). Jiang *et al.* [104] measured the agreement between the classifier and a modified nearest-neighbor classifier on the test examples as a confidence score. A recent approach [105] proposed to direct regress "true class probability" which improved over MSP for failure detection. Additionally, Bayesian approaches have drawn attention in this field of study. Dropout based approaches [59], [60] used Monte Carlo Dropout (MCDropout) for Bayesian approximation. Computing statistics such as entropy or variance is capable of indicating uncertainty. However, all these approaches mainly focus on small image classification tasks. When applied to semantic segmentation, they lack the information of semantic structures and contexts.

Segmentation quality assessment aims at estimating the overall quality of segmentation, without using ground-truth label, which is suitable to make alarms when model fails. Some approaches [114], [115] utilize Bayesian CNNs to predict the segmentation quality of medical images. [116], [117] regressed the segmentation quality from deep features computed from a pair of an image and its segmentation result. [118], [119]

plugged an extra IoU regression head into object detection or instance segmentation. [120], [121] used unsupervised learning methods to estimate the segmentation quality using geometrical features. Recently, Liu *et al.* [122] proposed to use VAE [123] to capture a shape prior for segmentation quality assessment on 3D medical image. However, it is hardly applicable to natural images considering the complexity and large shape variance in 2D scenes and objects. Segmentation quality assessment will be referred to as image-level failure detection in the rest of the paper.

OOD detection aims at detecting out-of-distribution examples in testing data. Since the baseline MSP method [103] was brought up, many approaches have improved OOD detection from various aspects [17], [106]–[109]. While these approaches mainly focus on image level OOD detection, *i.e.*, to determine whether an image is an OOD example, *e.g.*, [124], [125] targeted at detecting hazardous scenes in the Wilddash dataset [126]. On the contrary, we focus on **anomaly segmentation**, *i.e.*, a pixel-level OOD detection task that aims at segmenting anomalous regions from an image. Pixel-wise reconstruction loss [127], [128] with auto-encoders(AE) are the main stream approaches for anomaly segmentation. However, they can hardly model the complex street scenes in natural images and AEs can not guarantee to generate an in-distribution image from OOD regions. Recently, it was found that MSP surprisingly outperform AE and Bayesian network based approaches on a newly built larger scale street scene dataset StreetHazards [110] - with 250 types of anomalous objects and more than 6k high resolution images. Lis *et al.* [129] proposed to re-synthesize an image from the predicted semantic map to detect OOD objects in street scenes, which is the **pioneer** work for synthesis-based anomaly detection for semantic segmentation. SynthCP also follows this spirit, but we use a simple yet effective feature distance measure rather than a discrepancy network to find anomalies. In addition, we extend this idea to do a systematic study of failure detection.

Generative adversarial networks [130] generate realistic images by playing a "min-

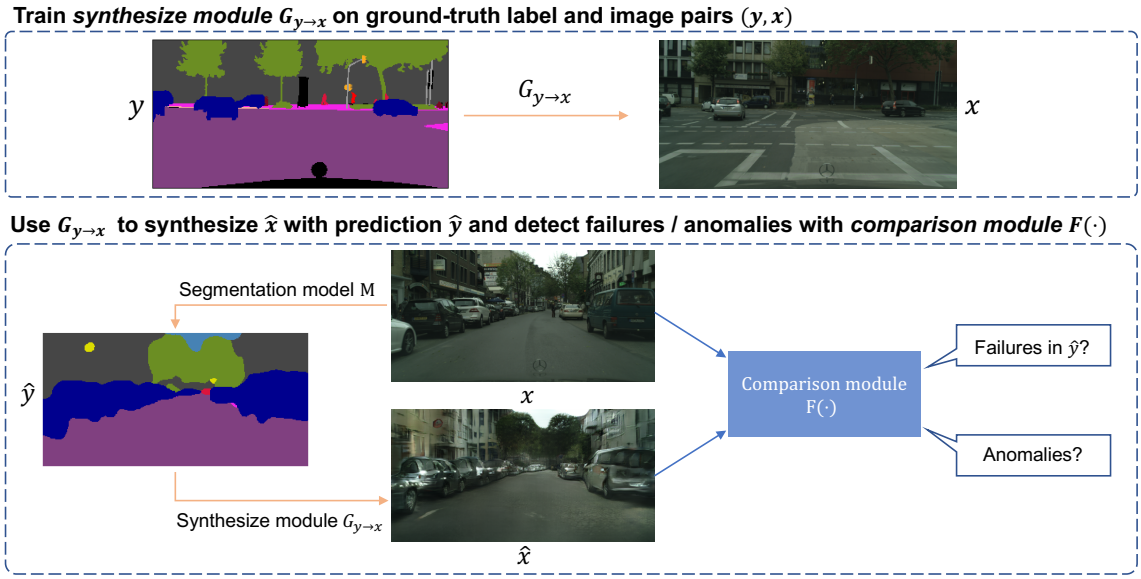


Figure 4.2. We first train the synthesis module $G_{y\to x}$ on label-image pairs and then use this module to synthesize the image conditioning on the predicted segmentation mask $\hat{y}$. By comparing x and $\hat{x}$ with a comparison module $F(\cdot)$, we can detect failures as well as segment anomalous objects. $F(\cdot)$ is instantiated in Sec 4.3.2.2 and Sec 4.3.3.2.

max" game between a generator and a discriminator. GANs effectively minimize a Jensen-Shannon divergence, thus generating in-distribution images. SynthCP utilizes conditional GANs [131] (cGANs) for image translation [111], *a.k.a* pixel-to-pixel translation. Approaches designed for semantic image synthesis [18], [112], [132] improves pixel-to-pixel translation in synthesizing real images from semantic masks, which is the reverse procedure of semantic segmentation. Since semantic image synthesis is commonly evaluated by the performance of a segmentation model, reversely, we are motivated to use a semantic-to-image generator for failure detection for semantic segmentation.

4.3 Methodology

In this section, we introduce our framework, SynthCP, for failure detection and anomaly detection for semantic segmentation. SynthCP consists of two modules, an image

synthesis module and a comparison module. We first introduce the general framework (shown in Fig. 7.2), then describe the details of the modules for failure detection and anomaly detection in Sec 4.3.2 and Sec 4.3.3, respectively. Unless otherwise specified, the notations in this paper follow this criterion: We use a lowercase letter, *e.g.*, x, to represent a tensor variable, such as a 1D array or a 2D map, and denote its i-th element as $x^{(i)}$; We use a capital letter, *e.g.*, F, to represent a function.

4.3.1 General Framework

Let x be an image with size of $w \times h$ and $\mathbb{L} = \{1, 2, \ldots, L\}$ be a set of integers representing the semantic labels. By feeding image x to a segmentation model M, we obtain its segmentation result, *i.e.*, a pixel-wise semantic label map $\hat{y} = M(x) \in \mathbb{L}^{w \times h}$. Our goal is to identify and locate the failures in $\hat{y}$ or detect anomalies in x based on $\hat{y}$.

4.3.1.1 Image Synthesis Module

We model this image synthesize module by a pixel-to-pixel translation conditional GAN (cGAN) [131], which is known for its excellent ability for semantic-to-image mapping. It consists of a generator G and a discriminator D.

Training. We train this translation conditional GAN on label-image pairs: (y, x), where y is a grouth-truth pixel-wise semantic label map and x is its corresponding image. The objective of the generator G is to translate semantic label maps to realistic-looking images, while the discriminator D aims to distinguish real images from the synthesized ones. This cGAN minimizes the conditional distribution of real images via the following min-max game:

$$\min_G \max_D \mathcal{L}_{GAN}(G, D), \tag{4.1}$$

where the objective function $\mathcal{L}_{GAN}(G, D)$ is defined as:

$$\mathbb{E}_{(y,x)}[\log D(y, x)] + \mathbb{E}_y[\log(1 - D(y, G(y))]. \tag{4.2}$$

Testing. After training, we fix the generator G. Given an image x and a segmentation model M, we feed the predicted segmentation mask $\hat{y} = M(x)$ into G, and obtain a synthesized (*i.e.*, reconstructed) image $\hat{x}$:

$$\hat{x} = G(\hat{y}). \tag{4.3}$$

$\hat{x}$ and x are then served as the input for the comparison module.

4.3.1.2 Comparison Module

We detect failures and anomalies in $\hat{y}$ by comparing $\hat{x}$ with x. Our assumption is that, if $\hat{x}$ is more similar to x, then $\hat{y}$ is more similar to y. However, since the optimization of G does not guarantee that the synthesized image $\hat{x}$ has the same style as the original image x, simple similarity measurements such as ℓ_1 distance between x and $\hat{x}$ is not accurate. In order to address this issue, we model the comparison module by a task-specific function F which estimates a trustworthy task-specific confidence measure $\hat{c}$ between x and $\hat{x}$:

$$\hat{c} = F(x, \hat{x}) = F(x, G(\hat{y})). \tag{4.4}$$

For the task of failure detection, the confidence measure $\hat{c} = (\hat{c}_{iu}, \hat{c}_m)$ includes an image-level per-class intersection over union (IoU) array $\hat{c}_{iu} \in [0,1]^{|\mathbb{L}|}$ and a pixel-level error map $\hat{c}_m \in [0,1]^{w \times h}$; For the task of anomaly segmentation, the confidence measure $\hat{c}$ is a pixel-level confidence map $\hat{c}_n \in [0,1]^{w \times h}$ for anomalous objects.

4.3.2 Failure Detection

4.3.2.1 Problem Definition

Our failure detection contains two tasks: 1) a per-class IoU prediction $\hat{c}_{iu} \in [0,1]^{|\mathbb{L}|}$, which is useful to indicate whether there are failures in the segmentation result $\hat{y}$, and 2) to locate the failures in $\hat{y}$, which needs to compute a pixel-level error map $\hat{c}_m \in [0,1]^{w \times h}$.

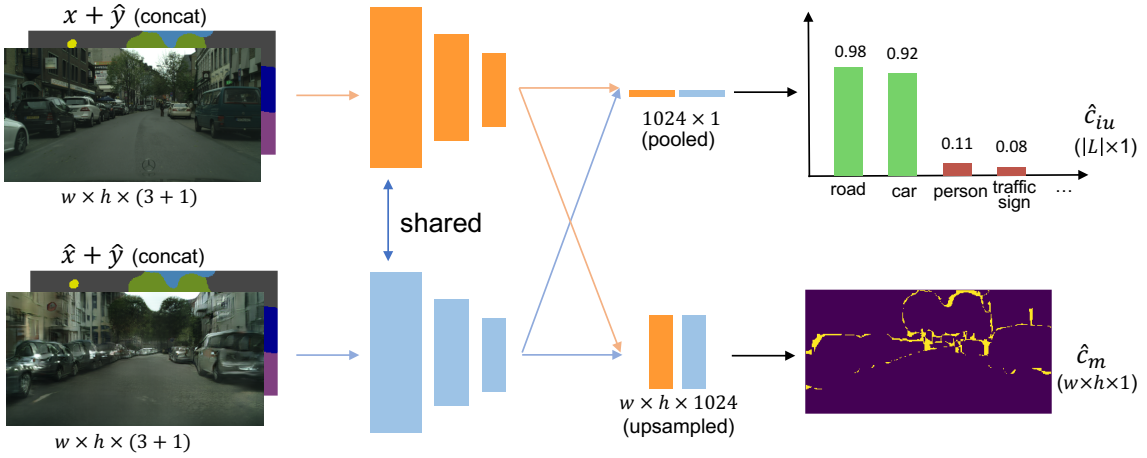


Figure 4.3. We instantiate $F(\cdot)$ as a light-weighted siamese network $F(x, \hat{x}, \hat{y}; \theta)$ for joint image-level per-class IoU prediction and pixel-level error map prediction.

4.3.2.2 Instantiation of Comparison Module

We instantiate the comparison module $F(\cdot)$ as a light-weighted deep network. In practice, we use ResNet-18 [1] as the base network and follow a siamese-style design for learning the relationship between x and $\hat{x}$. As illustrated in Fig. 6.3, x and $\hat{x}$ are first concatenated with $\hat{y}$ and then separately encoded by a shared-weight siamese encoder. Then two heads are built upon the siamese encoder and output the image-level per-class IoU array $\hat{c}_{iu} \in [0,1]^{|\mathbb{L}|}$ and pixel-level error map $\hat{c}_m \in [0,1]^{w \times h}$, respectively. We rewrite the function F for failure detection as below:

$$\hat{c}_{iu}, \hat{c}_m = F(x, \hat{x}, \hat{y}; \theta) \tag{4.5}$$

where θ represents the network parameters.

In the training stage, the supervision of network training is obtained by computing the ground-truth confidence measure c from y and $\hat{y}$. For the ground-truth image-level per-class IoU array c_{iu}, we compute it by

$$c_{iu}^{(l)} = \frac{|\{i|\hat{y}^{(i)} = l\} \cap \{i|y^{(i)} = l\}|}{|\{i|\hat{y}^{(i)} = l\} \cup \{i|y^{(i)} = l\}|}, \tag{4.6}$$

where l is the l-th semantic class in label set $\mathbb{L}$. The ℓ_1 loss function $\mathcal{L}_{\ell_1}(c_{iu}^{(l)}, \hat{c}_{iu}^{(l)})$ is

applied to learning this image-level per-class IoU prediction head. For the ground-truth pixel-level error map, we compute it by

$$c_m^{(i)} = \begin{cases} 1 & \text{if } y^{(i)} \neq \hat{y}^{(i)} \\ 0 & \text{if } y^{(i)} = \hat{y}^{(i)} \end{cases}. \tag{4.7}$$

The binary cross-entropy loss $\mathcal{L}_{ce}(c_m^{(i)}, \hat{c}_m^{(i)})$ is applied to learning this pixel-level error map prediction head. The overall loss function of failure detection $\mathcal{L}$ is the sum of the above two:

$$\mathcal{L} = \frac{1}{|\mathbb{L}|} \sum_{l}^{|\mathbb{L}|} \mathcal{L}_{\ell_1}(c_{iu}^{(l)}, \hat{c}_{iu}^{(l)}) + \frac{1}{wh} \sum_{i}^{wh} \mathcal{L}_{ce}(c_m^{(i)}, \hat{c}_m^{(i)}). \tag{4.8}$$

4.3.3 Anomaly Segmentation

4.3.3.1 Problem Definition

The goal of anomaly segmentation is segmenting anomalous objects in a test image which are unseen in the training images. Formally, given a test image x, an anomaly segmentation method should output a confidence score map $\hat{c}_n \in [0,1]^{w \times h}$ for the regions of the anomalous objects in the image, *i.e.*, $\hat{c}_n^{(i)} = 1$ and $\hat{c}_n^{(i)} = 0$ indicate the i^{th} pixel belongs to an anomalous object and an in-distribution object (the object is seen in the training images), respectively.

4.3.3.2 Instantiation of Comparison Module

As the same as failure detection, we first train a cGAN generator G on the training images, which maps the in-distribution object labels to realistic images. Given a semantic segmentation model M, we feed its prediction $\hat{y} = M(x)$ into G and obtain $\hat{x} = G(\hat{y})$. Since $\hat{y}$ only contains in-distribution object labels, $\hat{x}$ also only contain in-distribution objects. Thus, we can compare x with $\hat{x}$ to find the anomalies. The pixel-wise semantic difference of x and $\hat{x}$ is a strong indicator of anomalous objects.

Here, we simply instantiate the comparison function $F(\cdot)$ as the cosine distance defined on the intermediate features extracted by the segmentation model M:

$$\hat{c}_n^{(i)} = F(x, \hat{x}; M) = 1 - \langle \frac{\mathbf{f}_M^i(x)}{\|\mathbf{f}_M^i(x)\|_2}, \frac{\mathbf{f}_M^i(\hat{x})}{\|\mathbf{f}_M^i(\hat{x})\|_2} \rangle \tag{4.9}$$

where $\mathbf{f}_M^i$ is the feature vector at the i^{th} pixel position outputted by the last layer of segmentation model M and $\langle \cdot, \cdot \rangle$ is the inner product of the two vectors.

Post-processing with MSP. Due to the artifacts and generalized errors of GANs, our approach may mis-classify an in-distribution object into an anomalous object (false positives). We use a simple post-processing to address this issue. We refine the result by maximum softmax probability (MSP) [103], which is known as an effective uncertainty estimation strategy: $\hat{c}_n^{(i)} \leftarrow \hat{c}_n^{(i)} \cdot \mathbb{1}\{p^{(i)} \leq t\} + (1 - p^{(i)}) \cdot \mathbb{1}\{p^{(i)} > t\}$, where $p^{(i)}$ is the maximum soft-max probability at the i-th pixel outputted by the segmentation model M, $t \in [0, 1]$ is a threshold and $\mathbb{1}\{\cdot\}$ is the indicator function.

4.3.4 Conceptual Explanation

We give conceptual explanations of SynthCP in Fig. 4.4, where $\mathcal{X}$ and $\mathcal{Y}$ correspond to image space and label space. $M_{x \to y}$ is the segmentation model and $G_{y \to x}$ is a

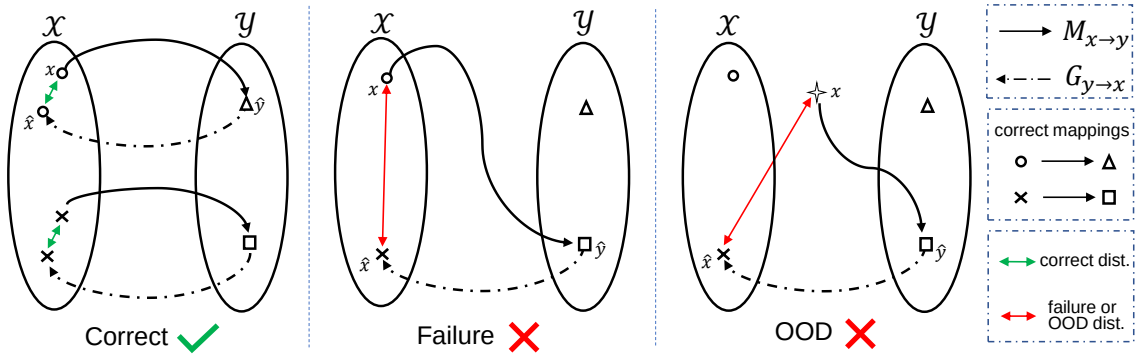


Figure 4.4. An analysis of SynthCP. Left: $M_{x \to y}$ correctly maps x to $\hat{y}$, resulting in small distance between x and the synthesized $\hat{x}$. However, when there are failures in $\hat{y}$ (middle) or there are OOD examples in x (right), the distance between x and $\hat{x}$ is larger, given a reliable reverse mapping $G_{y \to x}$.

semantic-to-image generator. The left image shows when $M_{x\to y}$ correctly maps an image to its corresponding segmentation mask, the synthesized image generated from $G_{y\to x}$ is close to the original image. However, when $M_{x\to y}$ makes a failure (middle) or encounters an OOD case (right), the synthesized image should be far away from the original image. As a result, the synthesized image serves as a strong indicator for either failure detection or OOD detection.

4.4 Experiments

4.4.1 Failure Detection

4.4.1.1 Evaluation Metrics

Following [122], we evaluate the performance of image-level failure detection, *i.e.*, per-class IoU prediction, by four metrics: **MAE**, **STD**, **P.C** and **S.C**. MAE (mean absolute error) and its STD measure the average error between predicted IoUs and ground-truth IoUs. P.C (Pearson correlation) and S.C.(Spearman correlation) measures their correlation coefficients. For pixel-level failure detection, *i.e.*, pixel-level error map prediction, we use the metrics in literature [103], [105]: **AUPR-Error**, **AUPR-Success**, **FPR at 95% TPR** and **AUROC**. Following [105], AUPR-Error is our main metric, which computes the area under the Precision-Recall curve using errors as the positive class.

4.4.1.2 The Cityscapes Dataset

We validate SynthCP on the Cityscapes dataset [133], which contains 2975 high-resolution training images and 500 validation images. As far as we know, it is the largest one for failure detection for semantic segmentation.

Baselines. We compare SynthCP to MCDropout [59], VAE alarm [122], MSP [103], TCP [105] and "Direct Prediction". MCDropout, MSP and TCP output pixel-level confidence maps, serving as standard baselines for pixel-level failure prediction. VAE

alarm [122] is the state-of-the-art in image-level failure prediction method. Following [122], we also use MCDropout to predict image-level failures. Direct Prediction is a method that directly uses a network to predict both image-level and pixel-level failures, by taking an image and its segmentation result as input. Note that, Direct Prediction shares the same experimental settings (backbone and training strategies) with SynthCP, which can be seen as an ablation study on the effectiveness of the synthesized image $\hat{x}$.

Implementation details. We use the state-of-the-art semantic-to-image cGAN - SPADE [18] in SynthCP. We re-trained SPADE from scratch following the same hyper-parameters as in [18] with only semantic segmentation maps as the input (without the instance maps). The backbone of our comparison module is ResNet-18 [1] pretrained from Image-Net. We use ImageNet pre-trained model and train the network for 20k iterations using Adam optimizer [134] with initial learning 0.01 and $\beta = (0.9, 0.999)$, which takes about 6 hours on one single Nvidia Titan Xp GPU. Since we use a network to predict failures, we need to generate training data for this network. A straightforward strategy is to divide the original training set into a training subset and a validation subset, then train the segmentation model on the training subset and test it on the

Table 4.1. Experiments on the Cityscapes dataset. We detect failures in the segmentation results of FCN-8 and Deeplab-v2. "SynthCP-separate" and "SynthCP-joint" mean training the image-level and pixel-level failure detection heads in our network separately and jointly, respectively.

	FCN-8				Deeplab-v2			
image-level	MAE↓	STD↓	P.C.↑	S.C.↑	MAE↓	STD↓	P.C.↑	S.C.↑
MCDropout [59]	17.28	13.33	3.62	5.97	19.31	12.86	4.55	1.37
VAE alarm [122]	16.28	11.88	21.82	18.26	16.78	12.21	17.92	19.63
Direct Prediction	13.25	11.96	58.34	59.74	14.45	12.20	60.94	62.01
SynthCP-separate	**11.58**	11.50	**64.63**	**65.63**	**13.60**	12.32	62.51	63.41
SynthCP-joint	12.69	**11.29**	62.52	61.23	13.68	**11.60**	**64.05**	**65.42**
	FCN-8				Deeplab-v2			
pixel-level	AP-Err↑	AP-Suc↑	AUC↑	FPR95↓	AP-Err↑	AP-Suc↑	AUC↑	FPR95↓
MSP [103]	50.31	99.02	91.54	25.34	48.46	99.24	92.26	24.41
MCDropout [59]	49.23	99.02	91.47	25.16	47.85	99.23	92.19	24.68
TCP [105]	48.54	98.82	90.29	32.21	45.57	98.84	89.14	36.98
Direct Prediction	52.17	99.15	92.55	**22.34**	48.76	99.34	92.94	**21.56**
SynthCP-separate	54.14	99.15	92.70	22.47	48.79	99.31	92.74	22.15
SynthCP-joint	**55.53**	**99.18**	**92.92**	22.47	**49.99**	**99.34**	**92.98**	21.69

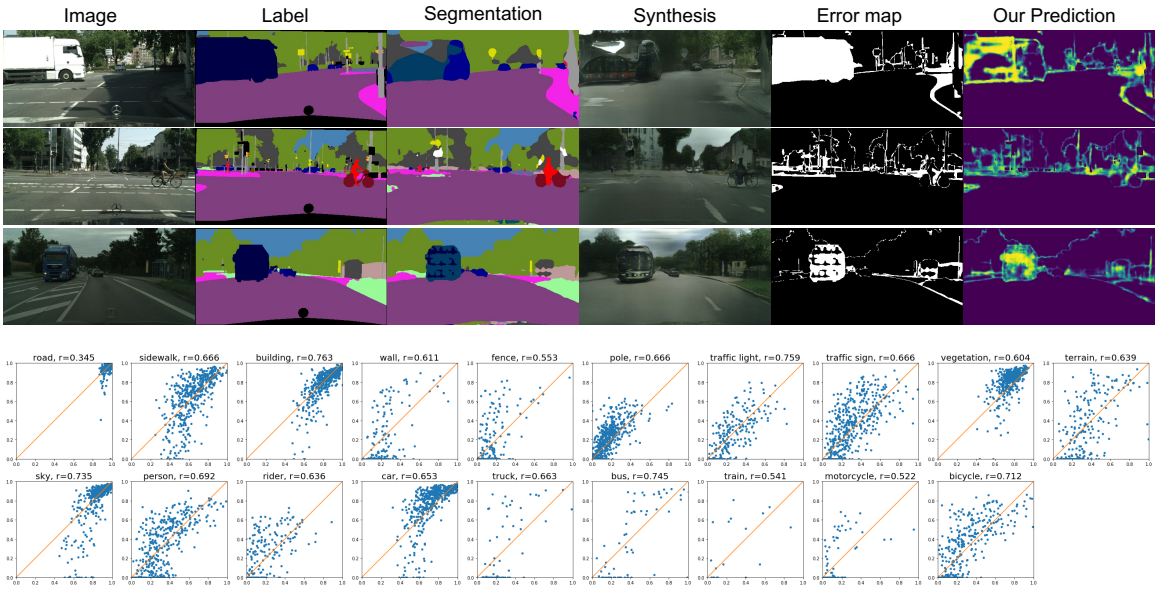


Figure 4.5. Visualization on the Cityscapes dataset for pixel-level error map prediction (top) and image-level per-class IoU prediction (bottom). For each example from left to right (top), we show the original image, ground-truth label map, segmentation prediction, synthesized image conditioned on the segmentation prediction, (ground-truth) errors in the segmentation prediction and our pixel-level error prediction. The plots (bottom) show significant correlations between the ground-truth IoU and our predicted IoU on most of the classes.

validation subset. The testing results on the validation subset can be used to train the failure predictor. We extend this strategy by doing 4-fold cross validation on the training set. Since the cross-validated results cover all samples in the original training set, we are able to generate sufficient training data to train our failure prediction network.

Results. Experimental results are shown in Table 4.1 and visualizations are shown in Fig. 4.5. We use the well-known FCN8 [24] and Deeplab-v2 [40] as the segmentation models. For image-level failure detection, our approach consistently outperforms other methods on all metrics. Results are averaged over 19 classes for all four metrics (detailed results in supplementary). We find that VAE alarm does not perform well 2D images of street scenes, since small objects are easily missed in the VAE reconstruction.

Without the synthesized images from the segmentation results, Direct Prediction performs worse than ours despite achieving better performance than the others.

For pixel-level failure detection, our approach achieves the state-of-the-art performance as well, especially for AP-Error metric where our approach outperforms other methods by a considerable margin. The comparison to Direct Prediction demonstrates that the improvements come from the image synthesis module in our framework. We hypothesis that TCP performs not as well because it is mainly designed for classification and it might be hard to fit the true class probability for dense predictions on large images in our settings. We find that our method produces slightly more false positives than "Direct Prediction" baseline (FPR95 is lower). We think the reason might be some correctly segmented regions are not synthesized well by the generative model.

We conducted another experiment to validate the **generalizability** on unseen segmentation models. We directly test our failure detection model, which is trained on Deeplab-v2 masks, on the segmentation masks produced by FCN8. We achieve an AUPR-Error of 53.12 for pixel-level error detection and MAE of 12.91 for image-level failure prediction. Full results are available in the supplementary material. The results are comparable to those obtained by our model trained on FCN8 segmentation model, as shown in table 4.1.

Table 4.2. Failure detection results on the pancreatic tumor segmentation dataset in MSD [63]

	tumor DSC prediction			
Method	MAE ↓	STD ↓	P.C. ↑	S.C. ↑
Direct Prediction	23.20	29.81	45.50	45.36
Jungo *et al.* [115]	26.57	29.78	-23.87	-20.23
Kwon *et al.* [114]	26.14	29.24	14.61	14.70
VAE alarm [122]	20.21	23.60	60.24	63.30
VAE (our imple.)	18.60	13.73	63.42	58.47
SynthCP	18.13	13.77	61.11	62.66
SynthCP + VAE	**15.19**	**13.37**	**67.97**	**71.35**

4.4.1.3 The Pancreatic Tumor Segmentation Dataset

We also validate SynthCP on medical images. Following VAE alarm [122], we applied SynthCP to the challenging pancreatic tumor segmentation task of Medical Segmentation Decathlon [63], where we randomly split the 281 cases into 200 training and 81 testing. The VAE alarm system [122] is the main competitor on this dataset. Since their approach explored shape prior for accurate quality assessment and tumor shapes have large variance, we expect SynthCP can outperform shape-based models or be complementary to the VAE-based alarm model. We only compare image-level failure detection in this dataset, because the VAE alarm system [122] is targeted to this task and sets up standard baselines.

We use the state-of-the-art network 3D AH-Net [85] as the segmentation model. Instead of IoU, the segmentation performance is measured by Dice coefficient (DSC), a standard evaluation metric used for medical image segmentation. Moving into 3D is challenging for SynthCP, since training 3D GANs is extremely hard, considering the limited GPU memory and high computational costs. In practice, we modify SPADE into 3D. Results and visualizations are shown in Table 4.2 and Fig. 4.6 respectively. In terms of baselines, we re-implement the VAE alarm system for a fair comparison in our settings, while the results of other methods are quoted from [122]. SynthCP achieved comparable performances as VAE alarm system. When combined with VAE alarm (a simple ensemble of the predicted DSC), all of the four metric improves significantly (P.C. and S.C correlation coefficient both improves by approximately 7% and 10% respectively), illustrating SynthCP which captures label-to-image information is complementary to the shape-based VAE approach.

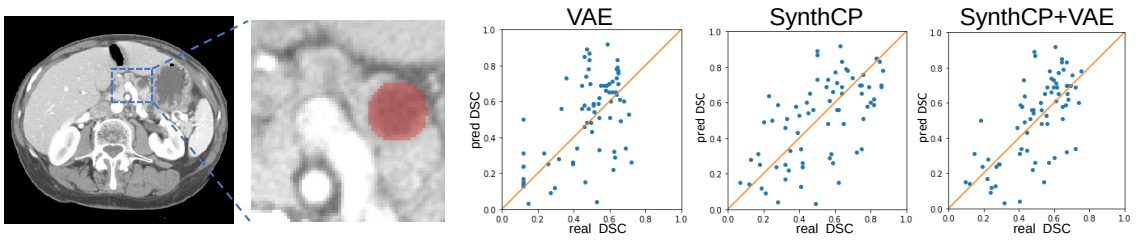


Figure 4.6. Left two: an example of pancreatic tumor segmentation (in red). Right three: plots for tumor segmentation DSC score prediction by VAE alarm [122], SynthCP and the ensemble of SynthCP and VAE alarm.

4.4.2 Anomaly Segmentation

4.4.2.1 Evaluation metrics.

We use the standard metrics for OOD detection and anomaly segmentation: area under the ROC curve (AUROC), false positive rate at 95% recall (FPR95), and area under the precision recall curve (AUPR).

4.4.2.2 The StreetHazards Dataset

We validate SynthCP on the StreetHazards dataset of CAOS Benchmark [110]. This dataset contains 5125 training images, 1000 validation images and 1500 test images. 250 types of anomaly objects appears only in the testing images.

Baselines. Baseline approaches include MSP [103], MSP+CRF [110], Dropout [59] and an auto-encoder (AE) based approach [127]. Except for AE, all the other three approaches require a segmentation model to provide either softmax probability or uncertainty estimation. AE is the only approach that requires extra training of an

Table 4.3. Anomaly segmentation results on StreetHazards dataset [110]

Method	FPR95↓	AUROC↑	AUPR↑
AE [127]	91.7	66.1	2.2
Dropout [59]	79.4	69.9	7.5
MSP [103]	33.7	87.7	6.6
MSP + CRF [110]	29.9	88.1	6.5
SynthCP	**28.4**	**88.5**	**9.3**

auto-encoder for the images and computes pixel-wise ℓ_1 loss for anomaly segmentation.

Implementation details. Following [110], we use two network backbones as the segmentation models: ResNet-101 [1] and PSPNet [41]. The cGAN is also SPADE [18] trained with the same training strategy as in Sec 4.3.2. The post-processing threshold $t = 0.999$ is chosen for better AUPR and is discussed in detail in the following paragraph.

Results Experimental results are shown in Table 4.3. SynthCP improves the previous state-of-the-art approach MSP+CRF from 6.5% to 9.3% in terms of AUPR. Fig. 4.7 shows some anomaly segmentation examples.

To study how much MSP post-processing contributes to SynthCP, we conduct experiments on different thresholds of t for post-processing. As shown in Table 4.4, without post-processing ($t = 1.0$), SynthCP achieves higher AUPR, but also produces more false positives, resulting in degrading FPR95 and AUROC. After pruning out false positives at high MSP positions ($p^{(i)} > 0.999$), we achieved the state-of-the-art performances under all three metrics.

Table 4.4. Performance change by varying post-processing threshold t

t	0.8	0.9	0.99	0.999	1.0
FPR95 ↓	**28.6**	28.5	28.2	28.4	46.0
AUROC ↑	88.3	88.4	**88.6**	88.5	81.9
AUPR ↑	7.4	7.7	8.8	**9.3**	8.1

4.5 Discussions

Why does our approach work better? Current approaches, such as MSP, TCP and MCDropout, mainly focus on improving failure detection with self-estimated statistics. However, deep networks tend to yield high confidence prediction [16], [103], thus self-estimated statistics are not trustable. The approaches that leverage extra data [109] or alternating training strategies [16] can alleviate this problem. We propose

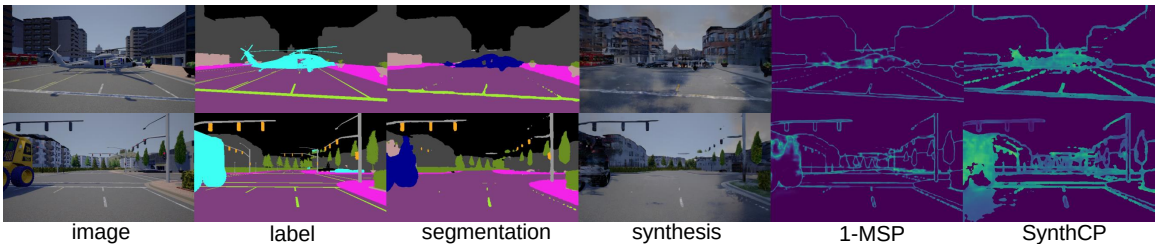


Figure 4.7. Visualizations on the StreetHazards dataset. For each example, from left to right, we show the original image, ground-truth label map, segmentation prediction, synthesized image conditioned on segmentation prediction, MSP anomaly segmentation prediction and our anomaly segmentation prediction.

to solve this problem from another prospective - **analyzing the performance of deep discriminative models by generative models**, the reverse procedure that models the conditional data distribution prior $P(x|y)$. Our method models $P(x|y)$ with a cGAN, which is proved to be beneficial to both failure and OOD detection of segmentation models.

Extra computational cost. There are two steps that requires extra computation besides the original segmentation network (M, latency T) in our approach - GAN reconstruction (G) and the comparison function computation for failure detection/anomaly segmentation. Since M and G are mutually inverse procedures, the inference time should be in the same magnitude. Compared to M or G, the inference time of the failure detection network and distance computation for anomalies are insignificant. So the overall extra computational cost for our framework is the T. Compared to other approaches, MSP based approaches [103], [109] are the most efficient. VAE alarm [122] and the AE-based approach [127] both need a separate network, basically have the same latency as ours. Dropout based approaches [59], [114] require multiple sampling of a segmentation network, which typically consumes more than time of $10T$.

Failure detection on predictions of unseen model M. We evaluate the generalizability of our failure detection system. We directly test our failure detection model, which is trained on Deeplab-v2 masks, on the segmentation masks produced by FCN8.

We achieve an AUPR-Error of 53.12 on pixel-level error detection and MAE of 12.91 on image-level failure prediction. Full results are available in supplementary material. The results are comparable with our model trained on FCN8 segmentation model in table 4.1, which illustrates the **generalizability** of our failure detection system.

GAN types. We assume a stronger GAN would yield better synthesis, and thus choose the state-of-the-art SPADE model [18] for all the main experiments. We also tried a weaker generator - pix2pixHD [112]. It turns out that the synthesis quality is far from satisfactory when the generator takes the prediction $\hat{y}$ as input (shown in supplementary materials). Under the same settings (FCN8 and pixel-level failure detection), AUPR of pix2pixHD model is only 51.31, which is close to the baseline "direct prediction" (AUPR 52.17). We thus conclude that a stronger generator benefits our failure detection scheme.

Adding image style encoder. Since the generator G does not guarantee the same style between x and $\hat{x}$ which increases the difficulty of the comparison module, we try to mitigate the effect by using an image encoder version of SPADE [18]. We hope the encoder can encode the style and generate images condition on segmentation map with the same style. However, the performance is not satisfactory (AUPR-Error experiences a subtle drop from 55.53 to 55.22). We hypothesize that the style encoder may also encode content (semantic) information and "cheat" to synthesize image without the segmentation mask, thus make the generator "less conditional".

4.6 Conclusions

We present a unified framework, SynthCP, to detect failures and anomalies for semantic segmentation, which consists of an image synthesize module and a comparison module. We model the image synthesize module with a semantic-to-image conditional GAN (cGAN) and train it on label-image pairs. We then use it to reconstruct the image

based on the predicted segmentation mask. The synthesized image and the original image are fed forward to the comparison module and output either failure detection (both image-level and pixel-level) or the mask of anomalous objects, depending on the specific task. SynthCP achieved the state-of-the-art performances on three challenging datasets.

Chapter 5

Auto-FedAvg: Learnable Federated Averaging for Multi-Institutional Medical Image Segmentation

Federated learning (FL) enables collaborative model training while preserving each participant's privacy, which is particularly beneficial to the medical field. FedAvg is a standard algorithm that uses fixed weights, often originating from the dataset sizes at each client, to aggregate the distributed learned models on a server during the FL process. However, non-identical data distribution across clients, known as the non-i.i.d problem in FL, could make this assumption for setting fixed aggregation weights sub-optimal. In this work, we design a new data-driven approach, namely **Auto-FedAvg**, where aggregation weights are dynamically adjusted, depending on data distributions across data silos and the current training progress of the models. We disentangle the parameter set into two parts, local model parameters and global aggregation parameters, and update them iteratively with a communication-efficient algorithm. We first show the validity of our approach by outperforming state-of-the-art FL methods for image recognition on a heterogeneous data split of CIFAR-10. Furthermore, we demonstrate our algorithm's effectiveness on two multi-institutional medical image analysis tasks, i.e., COVID-19 lesion segmentation in chest CT and pancreas segmentation in abdominal CT.

5.1 Introduction

Federated Learning (FL) [19], [135], [136] is a machine learning paradigm where clients collaboratively train a model without exchanging the underlying raw data. Compared to traditional centralized training, FL aims to benefit each participant while mitigating the potential for violating data privacy. FL was initially designed for mobile and edge devices [19] involving thousands of clients with often interrupted connectivity and only relatively small data each. However, recent studies involving only a small number of relatively reliable clients, e.g., medical institutions, have raised interest in utilizing FL for healthcare applications [137]. The latter scenario is referred to as "cross-silo" FL in Kairouz et al. [138] and is the focus of this paper.

Federated averaging (FedAvg) [19] is a simple yet effective algorithm for federated learning, following a server-client setup with two repeated stages: (i) the clients train their models locally on their data, and (ii) the server collects and aggregates the models to obtain a global model by weighted averaging. The aggregation weight of FedAvg is usually determined by the number of data samples on each client. This design choice assumes that data is uniformly distributed on the clients, and a stochastic gradient descent (SGD) optimizer is enforced. However, this setting can hardly be optimal and even detrimental because the clients' underlying data distributions remain unknown and are most likely non-independent and identically distributed (non-i.i.d). Domain shifts in the data are expected among different clients in real-world scenarios.

In this paper, we aim to improve FedAvg by automatically learning how to aggregate different client models more optimally. Our approach, namely Auto-FedAvg, is data-driven and differentiable while keeping the privacy-preserving aspects of FL. Recall that FedAvg involves two iterative steps. Our approach introduces a third step. After the clients finish training their local models, we learn a set of global aggregation weights in a data driven fashion, which the server later uses in the weighted average

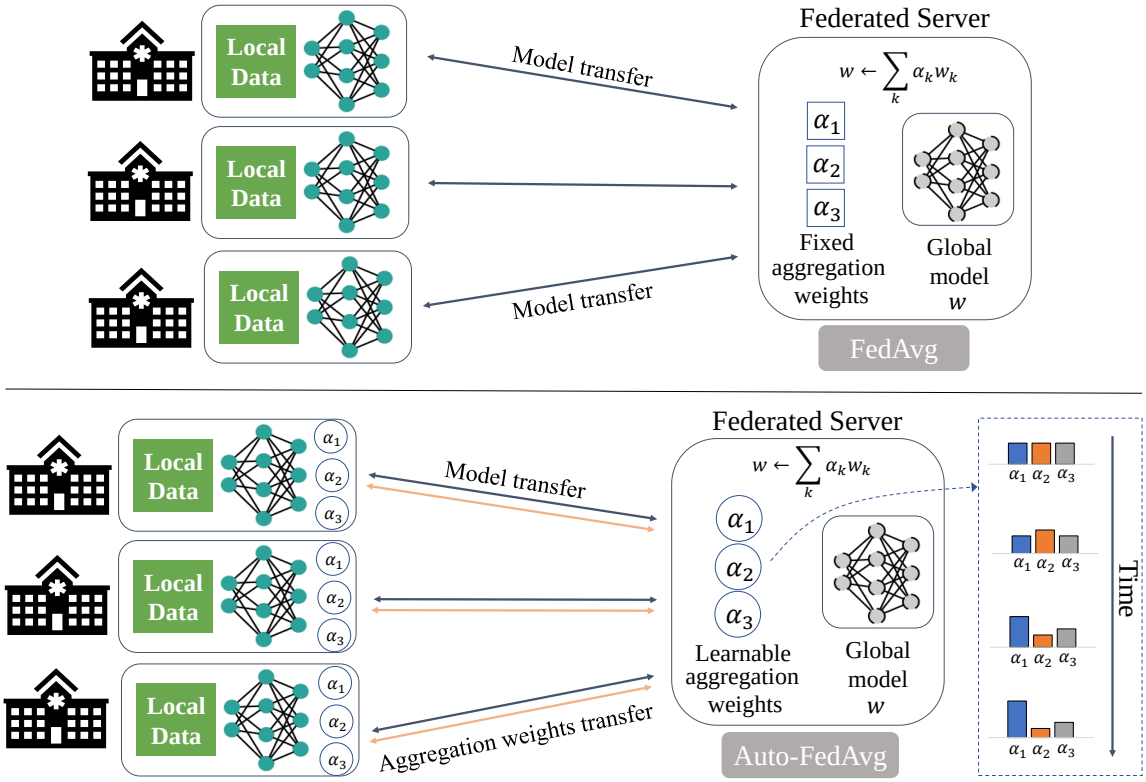


Figure 5.1. An illustration of FedAvg (top) and Auto-FedAvg (bottom). In FedAvg, the server collects locally trained models from each client and obtains a global model by weighted averaging with fixed aggregation weights. In contrast, in Auto-FedAvg, the aggregation weights are learned on the clients and dynamically adjusted throughout the training process when communicating with the server.

for computing the global model. Learning the global aggregation weights is beneficial in two aspects: (i) Since the convergence rate is likely to be different across the clients, dynamically adjusting aggregation weights can accelerate the training process. (ii) Better performance and generalizability can be achieved because the global model is more robust when applied to all the client's test data since we directly optimize the local loss to update the aggregation weights by modelling them as a stochastic process utilizing the Dirichlet distribution. We also designed a communication-efficient algorithm to achieve this goal without violating the data privacy constraint of FL.

We first validate the effectiveness of our approach on the CIFAR-10 dataset, where we outperform the state-of-the-art method, FedMA [139] by 1.45% using the same

heterogeneous data partitioning. Moreover, we outperform the FedAvg algorithm on two medical image segmentation tasks, *i.e.*, multi-institutional and multi-national COVID-19 lesion segmentation and pancreas segmentation, showing its real-world potential.

Our contributions are summarized as follows:

- We propose to directly learn the model aggregation weights in FL from data with gradient descent using a Dirichlet distribution, which is adaptive to the underlying data and learning progress.
- We design a new communication algorithm to fulfill the proposed goal with limited extra communication cost in cross-silo FL and without violating the data privacy constraints of FL.
- We outperform state-of-the-art approaches on a heterogeneous data split of CIFAR-10. Furthermore, we extensively analyze the proposed algorithm on two multi-institutional medical imaging studies with real-world datasets.

5.2 Related Work

Federated Learning. Here, we introduce some common algorithms for FL. Federated Averaging (FedAvg) [19] is a standard algorithm, where parameters of local models are averaged with fixed weights to obtain a global model. The aggregation weight of each client is usually set to be proportional to the size of client's dataset. FedMA [139] refined the aggregation process by matching and averaging hidden elements with similar feature signatures. The idea of integrating knowledge distillation into FL has also been explored [140], [141].

Recently, the issue of FL on non-i.i.d data draws emerging attentions. Several works have been proposed to address data heterogeneity in FL settings [142]–[146],

among which one direction is to optimize the process of model aggregation that we also consider in this paper. For example, Wang et al. [147] proposed a normalized averaging method that eliminates objective inconsistency while preserving fast convergence for heterogeneous data clients. Chen et al. [146] analyzed median-based FL algorithms. Agnostic Federated Learning [148] proposed to optimize a centralized model for any target distribution formed by a mixture of the client distributions. FedBE [149] learns a Bayesian ensemble from the distribution of the models. These works explore statistics or underlying distribution of the models to adjust aggregation strategies. In contrast, we propose to directly learn the aggregation weights by gradient-based optimization on the clients' data. Other recent works also discuss the possibility for model personalization [150]–[152]. Most recent works demonstrate good theoretical analysis but are only evaluated on manually created toy examples. It is not clear if the approaches would generalize well to real-world medical imaging datasets such as those studied in this work.

Multi-institutional Medical Image Analysis. Due to its privacy-preserving attributes, FL is particularly attractive for the medical domain. Rieke et al. [137] discussed the potential of FL in digital health. Meanwhile, multiple real-world investigations of FL have been applied to medical image analysis, which is itself a well-explored field with deep learning [25], [26], [43]. Examples of FL in medical imaging include multi-institutional brain tumor segmentation [153], [154], breast density classification [155] and fMRI analysis [156]. In addition to FL settings, Chang et al. [157] synthesized medical images with a GAN [130] without sharing data between institutions. On top of privacy concerns, Liu et al. [158], Dou et al. [159] and Xia et al. [160] emphasized the challenge of domain shift for multi-institutional medical data and developed algorithms to solve domain adaptation and generalization problems in prostate segmentation, brain tissue segmentation and liver segmentation from multi-site medical images, respectively. However, these non-i.i.d. challenges have not

been resolved in FL for medical imaging [137].

Automated Machine Learning. This paper introduces an automated approach to find the best aggregation weights for federated learning. Our approach is inspired by recent advances of automated machine learning (AutoML), including hyper-parameter search [161]–[163], neural architecture search (NAS) with reinforcement learning [164], [165], evolution algorithm [166], [167] and differentiable approaches [168], [169]. A recent approach [170] improves NAS by modeling the architecture mixing weight using a Dirichlet distribution, a mathematical formulation that we also utilize in this work. In the broad sense of AutoML, our approach can also be categorized as a differentiable hyper-parameter search algorithm in the continuous search space of FedAvg aggregation weights.

5.3 Auto-FedAvg

In this section, we first describe the general notations of federated learning and revisit FedAvg [19]. We then introduce our optimization objective, where we will also introduce how we parameterize the aggregation weights to follow certain constraints, as well as variants of the aggregation strategies, *i.e.*, network-wise and layer-wise. Finally, we describe our full algorithm in detail and analyze the communication cost of the proposed Auto-FedAvg approach.

5.3.1 Revisiting FedAvg

Suppose K clients collaboratively train a global model with parameter w in a standard FL setting. In particular, the aim is to minimize:

$$\min_{w} \sum_{k=1}^{K} \alpha_k \mathcal{L}_k(w), \tag{5.1}$$

where $\mathcal{L}_k(w)$ is the local loss function of client k, $\alpha_k \geq 0$ and $\sum_k \alpha_k = 1$. Suppose

there are n_k data samples on client k, then we usually set $\alpha_k = \frac{n_k}{n}$, where $n = \sum_k n_k$ is the total number of data samples used in the FL setting.

To relieve the communication burden, FedAvg [19] allows the clients to update their local models for a certain period of time with the stochastic gradient descent (SGD) optimizer. We denote the local loss function given a data sample x and the current model weight w as $l(w, x)$. The server then collects C models ($C \leq K$), aggregates them with weighted averaging to update the global model, and sends the new global model back to the clients for re-initialization of next round of FL training. The aggregation weights $\alpha \in \mathbb{R}^K$ are set to be proportional to the number of data samples on each client ($\alpha_k = \frac{n_k}{n}$) as mentioned before. We pick $C = K$ for simplicity and the update of the global model w in each communication round as $w \leftarrow \sum_k \frac{n_k}{n} w_k$ where w_k is the current model of client k.

The aggregation weights chosen by vanilla FedAvg is based on the assumption that data follows a uniform distribution across clients and are computed based on the number of SGD steps performed on each client. However, since the data distribution at each client is unknown and could possibly be non-i.i.d or involve domain shifts, this assumption is not guaranteed and can result in sub-optimal or even detrimental effects.

5.3.2 Optimization Objectives

To counteract the limitations of FedAvg, we propose our differentiable approach to directly learn the aggregation weights α from data at the clients. Denote by $\mathcal{L}$ the loss function. We propose a constrained objective function in:

$$\min_{\alpha} \quad \sum_{k=1}^{K} \mathcal{L}_k(\sum_{k=1}^{K} \alpha_k w_k)$$
$$\text{s. t.} \quad \sum_{k=1}^{K} \alpha_k = 1 \text{ and } \alpha_k > 0, \tag{5.2}$$

where $w_k = \arg\min_{w} \mathcal{L}_k(w)$ is the local model updated on the training set of client k. The motivation of the proposed objective is that we directly learn the aggregation weight by gradient descent from data in a differentiable way, while keeping the local models fixed after completing their local training. Since there is no data sharing between clients, we will introduce a communication algorithm to achieve the learning objective later in the next "Algorithm" subsection. We first discuss the variants of the constraints of Eq. 6.1 as follows.

5.3.2.1 Constraints of the aggregation weights.

Here, we provide two assumptions for the optimization constraints in Eq. 6.1. To achieve these constraints, we introduce a new set of variable $\boldsymbol{\beta} = [\beta_1, .., \beta_K]$, which is a vector with the same dimension as $\boldsymbol{\alpha} = [\alpha_1, .., \alpha_K]$. We define a function γ to transform $\boldsymbol{\beta}$ to $\boldsymbol{\alpha}$:

$$\boldsymbol{\alpha} = \gamma(\boldsymbol{\beta}) \tag{5.3}$$

Softmax function. One obvious choice to satisfy the constraint of $\boldsymbol{\alpha}$ is to apply a *softmax* function to $\boldsymbol{\beta}$

$$\alpha_k = \frac{\exp(\beta_k)}{\sum_{i=1}^{K} \exp(\beta_i)} \tag{5.4}$$

Thus, the loss function becomes $l(\sum_{k=1}^{K} \alpha_k w_k, x) = \mathcal{L}(\boldsymbol{\beta}, x)$, which only depends on β and x, since we keep the model weights w_k fixed in the aggregation weight learning process (Eq. 6.1). In practice, we can compute the gradient of each β_k and directly update them based on a client's local data with gradient descent.

Dirichlet distribution. A better choice is to treat the aggregation weight $\boldsymbol{\alpha}$ as random variables, modeled by the Dirichlet distribution parameterized by the concentration $\boldsymbol{\beta}$: $\boldsymbol{\alpha} \sim \text{Dir}(\boldsymbol{\beta})$. This formulation induces stochasticity that naturally encourages exploration in the search space during the sampling process in training. The probability density function is formed as:

$$\text{Dir}(\boldsymbol{\alpha}|\boldsymbol{\beta}) = \frac{1}{B(\boldsymbol{\beta})} \prod_{k=1}^{K} \alpha_k^{\beta_k - 1}, \tag{5.5}$$

where $B(\boldsymbol{\beta}) = \frac{\prod_{k=1}^{K} \Gamma(\beta_k)}{\Gamma(\sum_{k=1}^{K} \beta_k)}$ and $\Gamma(z) = \int_0^\infty x^{z-1} e^{-x} dx$ is the gamma function. The Dirichlet distribution is the conjugate prior of a multinomial distribution with a simplex. Each sample will already satisfy our constraint of the aggregation weights in Eq. 6.1. Thus we find the Dirichlet distribution to be a natural formulation to model the aggregation weights during FL while utilizing its properties for gradient-based optimization [170], [171]. It is also worth mentioning that the uniform distribution is a special case of the Dirichlet distribution when $\alpha_1 = \alpha_2 = ... = \alpha_K = 1$.

In the training phase, given a data sample x, we sample $\boldsymbol{\alpha}$ from the Dirichlet distribution with concentration $\boldsymbol{\beta}$, approximate the gradient of $\boldsymbol{\beta}$ given the loss function $\mathcal{L}(\boldsymbol{\beta}, x)$ using implicit reparameterization [172] and update the concentration $\boldsymbol{\beta}$. During inference, we compute the mode of the distribution, which represents the values with maximum probability.

$$\alpha_k = \frac{\beta_k - 1}{\sum_{i=1}^{K} \beta_i - K} \tag{5.6}$$

5.3.2.2 Aggregation strategies.

In the process of model aggregation, our approach introduces more flexibility in terms of the design of the aggregation weights than FedAvg, because we are able to learn the parameterized aggregation weights in a differentiable way from data. Here, we

describe two natural variants.

Network-wise aggregation weights. In this scenario, each aggregation weight α_k in $\boldsymbol{\alpha}$ is a scalar. The aggregation process is the same as described previously: $w \leftarrow \sum_k \alpha_k w_k$.

Layer-wise aggregation weights. Our approach allows an easy extension to network-wise aggregation, namely layer-wise aggregation. Suppose the deep network model we are training has P layers. We denote $w_{k,p}$ as the p-th layer parameter of the model of client k. Then $\alpha_k = [\alpha_{k,1}, .., \alpha_{k,P}]$ is a P-dimensional vector. Thus we are able to obtain the p-th layer weight w_p by $w_p \leftarrow \sum_{k=1}^{K} \alpha_{k,p} w_{k,p}$.

As for the constraints discussed previously, $\beta_k = [\beta_{k,1}, .., \beta_{k,P}]$ is now a P-dimensional vector as well. Then, $\alpha_{k,p} = \frac{\exp(\beta_{k,p})}{\sum_{k=1}^{K} \exp(\beta_{k,p})}$ is the equation when using softmax, and $\boldsymbol{\alpha}_p \sim \text{Dir}(\boldsymbol{\beta}_p)$ when applying the Dirichlet distribution.

5.3.3 Algorithm

Optimizing the objective function in Eq. 6.1 is not trivial under the FL setting, since (i) we can only rely on the local data on each client which is inaccessible to the server, and (ii) we would like to maintain a relatively low communication cost. We describe the algorithm of Auto-FedAvg in Algorithm 2. In each communication round t, the server first sends out the global model to all the clients. When the clients finish updating the local models in parallel, the server gathers them and aggregates the models with a set of learnable weights $\boldsymbol{\alpha}^t = [\alpha_1^t, .., \alpha_K^t]$ by weighted averaging to obtain an updated global model w^t. $\boldsymbol{\alpha}^t$ is parameterized by $\boldsymbol{\beta}^t$ using function γ and the actual instantiation of γ in Eq. 5.3 is determined by whether we use softmax (Eq. 5.4) or the Dirichlet distribution (Eq. 5.6) as the method to parameterize α. The learning process of $\boldsymbol{\beta}^t$ is described in `LearnAggWeight` of Algorithm 2.

In `LearnAggWeight`, each client receives a copy of all the model weights $w_1, .., w_K$ and keeps them fixed during this process. In each local iteration s, each client samples

a mini-batch x from their own local data, and computes the current $\boldsymbol{\alpha}$ from $\boldsymbol{\beta}^{s-1}$ depending on the softmax or Dirichlet assumption we apply to the aggregation weights, before forwarding x into the local model with weight $\sum_{k=1}^{K} \alpha_k w_k$. Then the client will compute the loss function $\mathcal{L}(\boldsymbol{\beta}^{s-1}, x)$ and update $\boldsymbol{\beta}^{s,k}$ based on the computation (softmax) or estimation (Dirichlet distribution) of the gradient [172], as mentioned in Sec 5.3.2. The server will gather $\boldsymbol{\beta}^{s,k}$ from every client k in every iteration s and average them to obtain a new global $\boldsymbol{\beta}^{s}$.

5.3.3.1 Communication efficiency analysis.

The communication of $\boldsymbol{\beta}$ is very efficient because $\boldsymbol{\beta}$ is merely a set of K scalars or K low dimensional vectors (of size P) in either "network-wise aggregation weight" or "layer-wise aggregation weight" strategy, which is negligible compared to communicating the full network parameters as in a standard FedAvg round. The major extra communication burden of aggregation weight learning is introduced when the server sends all local models to each client in the very first step. As a result, we only do the aggregation weight learning process every t_0 rounds to further relieve the additional communication burden compared to FedAvg. The extra communication cost ratio (extra cost divided by FedAvg communication cost) is $\frac{K-1}{2t_0}$. A detailed derivation of which can be found in the supplementary material. This is more acceptable in cross-silo federated learning setting, which typically contains only a small number of clients with relatively reliable internet connectivity [138]. For example, in our COVID-19 lesion segmentation experiments, $K = 3$ and $t_0 = 10$, results in an extra 10% communication cost compared to FedAvg.

Table 5.1. CIFAR-10 classification with heterogeneous partition.

Method	final accuracy(%)
FedAvg	86.29
FedProx [173]	85.32
Ensemble	75.29
FedBN [174]	80.77
FedMA [139]	87.53
FedMA [139] (our impl.)	87.47
Auto-FedAvg-L-Softmax*	88.64
Auto-FedAvg-L-Dirichlet*	88.37
Auto-FedAvg-N-Softmax*	88.60
Auto-FedAvg-N-Drichlet*	**88.98**

* With the interval of aggregation weight learning $t_0 = 10$.

5.4 Experiments

5.4.1 CIFAR-10

We first validate our approach on the CIFAR-10 dataset. To compare our approach with the state-of-the-art FL methods such as FedProx [173] and FedMA [139] on the benchmark dataset, we use the same heterogeneous data partition of FedMA [139] on the CIFAR-10 dataset that simulates an environment where the number of data points and class proportions are unbalanced using their publicly available code[1]. In this way, we can directly compare with the results in the paper, which are shown in Table 5.1. The baseline numbers except for FedBN [174] are from [139] on the same data split. We train the baseline and our experiments for 99 rounds with 16 clients before we test on the test set, where the same network architecture of VGG-9 is adopted. The re-implementation of FedMA achieves 87.47% accuracy, which is very close to the reported performance 87.53% [139], indicating the correctness of our experimental setup. We also implement FedBN [174] under the same setup. FedBN explores batch statistics of the batch normalization (BN) layers in the scenario of FL and we hypothesize the unsatisfying results here is because BN is not suitable for

[1] https://github.com/IBM/FedMA

the scenario of numerous clients with limited data. The later experimental results in the other two medical datasets will also demonstrate this hypothesis. For our Auto-FedAvg algorithm, we experiment with different design choices described in the previous section, *i.e.* layer-wise ("L") or network-wise ("N") aggregation strategy and softmax ("Softmax") or Dirichlet assumption ("Dirichlet") over the constraints of the aggregation weights. Based on the metric of final accuracy, all our experimental variants outperform the baselines and our "Auto-FedAvg-N-Dirichlet" achieved the best final accuracy of 88.98%, outperforming the published FedMA result by 1.45%.

5.4.2 Multi-national COVID-19 lesion segmentation

5.4.2.1 Experimental results

The study with first real-world data of our federated learning algorithm is COVID-19 diagnosis, which has caused a world-wide pandemic in the year of 2020 and 2021. Machine learning based algorithms have been developed to quickly diagnose the disease and study the imaging characteristics [175]–[177]. In this study, we focus on the critical task of COVID-19 lesion segmentation on multi-national COVID-19 datasets. Due to page limits, the detailed implementation details are introduced in supplementary material.

Dataset description. This study contains CT scans of SARS-CoV-2 infected patients collected from three international medical centers, including (i) 671 scans from [anonymized hospitals] in China (denoted as Dataset I), (ii) 88 scans from [anonymized hospitals] in Japan (denoted as Dataset II), and (iii) 186 scans from [anonymized hospitals] in Italy (denoted as Dataset III). Two expert radiologists annotated these CT scans assigning a foreground (COVID-19 lesion) and background label for each voxel. For each dataset, we randomly split the annotated cases into training/validation/testing, resulting in splits of 447/112/112 for Dataset I, 30/29/29 for Dataset II, and 124/31/31 for Dataset III. We visualize examples in Fig 5.2 and show the intrinsic

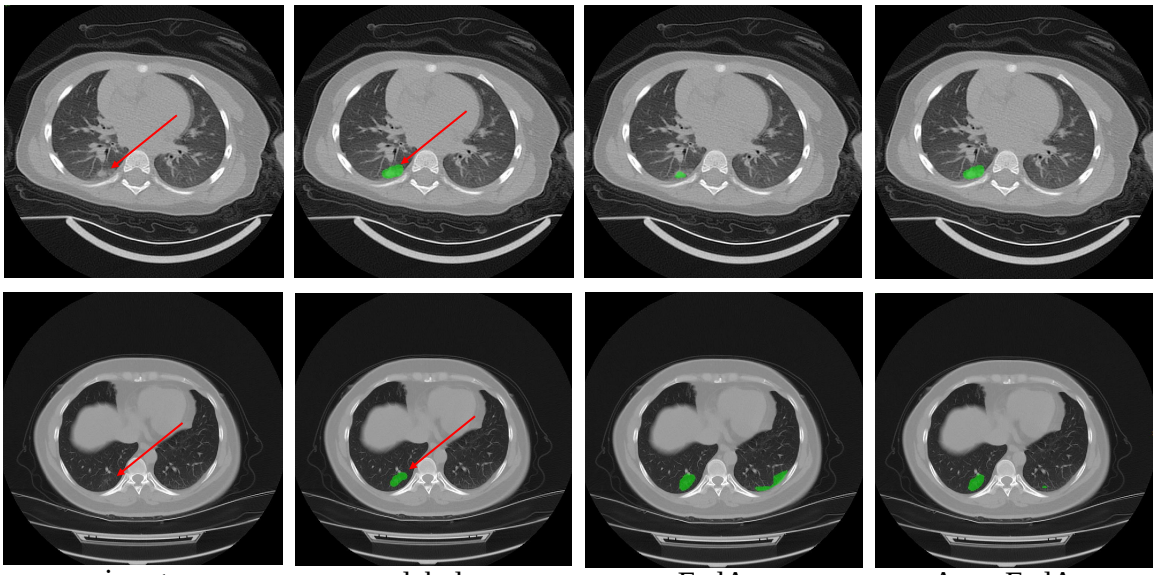


Figure 5.2. Examples of COVID-19 lesion segmentation of patients from China (top) and Italy (bottom). From left to right: original CT scan, human label (in green), FedAvg segmentation results, and our segmentation results. Our Auto-FedAvg mitigates the issue of under-segmentation (top) and reduces false-positive prediction (bottom) in these two examples, respectively.

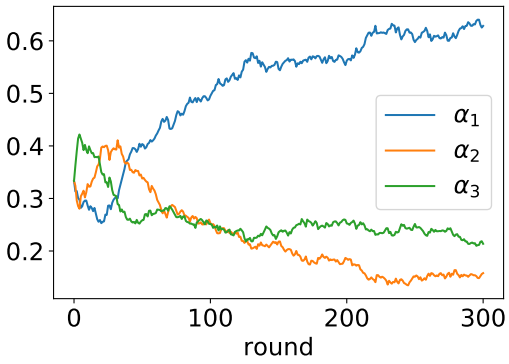
Table 5.2. Multi-national COVID-19 lesion segmentation. "Global test avg" is the major metric to measure the generalizability of the FL global model. n specifies the total dataset size at the client.

Method	I (n=671)	II (n=88)	III (n=186)	global test avg	local avg
Local only - I	59.82	61.82	51.80		
Local only - II	41.92	59.95	50.18	50.68	61.87
Local only - III	34.50	52.54	**65.85**		
FedAvg	59.93	63.79	60.52	61.41	62.47
FedAvg - even	56.73	64.31	64.98	62.01	62.24
FedProx [173]	60.33	64.98	60.45	61.92	61.99
FedBN [174]	**63.24**	63.25	63.91	**63.47**	63.32
Auto-FedAvg-L-Softmax	59.03	64.96†	61.66†	61.89	63.17
Auto-FedAvg-L-Dirichlet	58.59	64.95†	64.96†	62.83	63.08
Auto-FedAvg-N-Softmax	59.58	64.50†	63.35†	62.48	63.42
Auto-FedAvg-N-Dirichlet	60.37	**65.28**†	64.76†	**63.47**†	**64.04**
Auto-FedAvg-N-Dirichlet*	60.42	64.86†	64.07†	63.11†	63.74

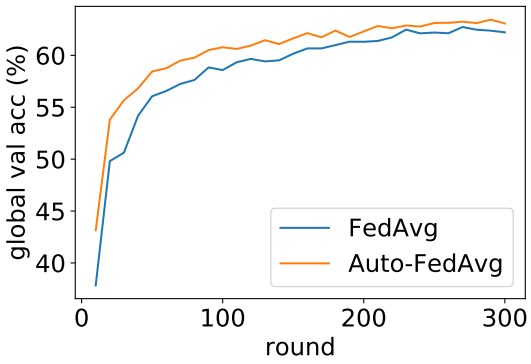
* With the interval of aggregation weight learning $t_0 = 10$.
† Significance of the global model over FedAvg.

domain shift between datasets (e.g., caused by resolution and contrast).

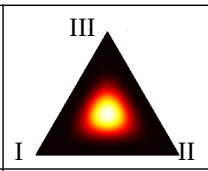
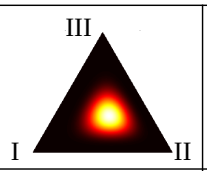
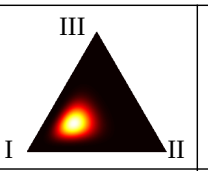
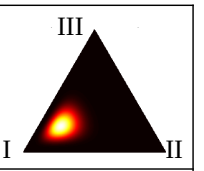
Evaluation metrics. We measure the performance of the segmentation models by



(a) The learning curve of α.



(b) Validation accuracy growth.

probability density (low – high)				
round	0	30	150	300
β	(6.0, 6.0, 6.0)	(6.4, 8.0, 6.4)	(13.4,6.0,6.4)	(18.3, 5.3, 6.9)
α	(0.33,0.33,0.33)	(0.30,0.39,0.31)	(0.54,0.22,0.24)	(0.63,0.16,0.21)

(c) Visualizations of Dirichlet distribution.

Figure 5.3. Analysis of the learning process during "Auto-FedAvg-N-Dichlet".

Dice similarity coefficient (DSC), a standard evaluation metric used for medical image segmentation. For all the FL experiments, we test the performance of the best global model, selected by highest average validation accuracy of all three clients, on the test data of each client, corresponding to the first three columns (I/II/III) of Table 5.2. We compute the average of the three test accuracies to measure the average performance of the model on three datasets, corresponding to the forth column "global test avg". This metric represents a measure for the generalizability of the global model, and serves as the major metric for performance evaluation. Moreover, we test the best local models on all clients, selected by the highest local validation score. The average performance of the locally best models is denoted as "local avg" in column five.

Results. We display the quantitative results in Table 5.2 and two examples for qualitative analysis in Fig 5.2. We first train the models locally without communication

to obtain the baselines of the local models, shown in the first three rows in the table. Unsurprisingly, all three local models have relatively low generalization performance when tested on other clients, indicating domain shifts across the three datasets. For the FedAvg baseline, we experiment with two different sets of aggregation weights, *i.e.*, normalized dataset size and uniform weights, denoted by "FedAvg" and "FedAvg-even", respectively. We also implement FedProx [173] with the empirically best $\mu = 0.001$. For our Auto-FedAvg algorithm, we experiment with different design choices, *i.e.* layer-wise ("L") or network-wise ("N") aggregation strategy and softmax ("Softmax") or Dirichlet assumption ("Dirichlet") over the constraints of the aggregation weights. We find that "Auto-FedAvg-N-Dirichlet" gives the best results, outperforming "FedAvg" by 2.06% on general global model performance (column "global test avg"), and by 1.57% on average local model performance (column "local avg"). We furthermore performed a Wilcoxon signed rank test on the test set (first four columns), where the significant improvements ($p \ll 0.05$) over FedAvg are marked with superscript †.

Generally speaking, the Dirichlet distribution performs better at modeling the aggregation weights than softmax. Interestingly, the performance of the layer-wise aggregation strategy is worse than the network-wise aggregation strategy. The gradient of network-wise aggregation weights can be viewed as a summation of all gradients of layer-wise aggregation weights. In this sense, we suspect that network-wise aggregation acts as a regularization of layer-wise weights. We also conduct diagnosis experiments and provided them in supplementary materials, where we display the patterns of the learned layer-wise weights and suggest a layer-wise smoothing loss can improve the results of the layer-wise aggregation strategy. The improvement of the layer-wise smoothing loss for the layer-wise aggregation strategy further serves as evidence that the network-wise aggregation may act as regularization over the layer-wise one.

5.4.2.2 Analyze the learning process.

Here, we aim to analyze the learning process of Auto-FedAvg. The learning curve of the aggregation weights α, validation accuracy growth, and the visualization of the Dirichlet distribution are displayed in Fig. 5.3. The sub-figures correspond to our best performing model "Auto-FedAvg-N-Dichlet" in Table 5.2. As shown in Fig. 5.3a, in the first 30 rounds, α_2 and α_3 rise moderately, indicating the global model could benefit from increasing the weight of the models from client II and client III in the early stage. This matches our expectation that client II and client III converge faster than client I because client II and client III own significantly less data than client I. Giving them more weight in the aggregation process accelerates the training process. As shown in Fig 5.3b, our approach has a faster growth in validation score than FedAvg. After approximately 40 rounds, we observe a rise of α_1 and drops of α_2 and α_3, indicating that assigning higher weights to client I benefits the global model eventually, making it more generalizable across different clients.

In terms of the latent Dirichlet distribution of α (shown in Fig 5.3c), we plot the different states of α as well as the latent variable β in round 0, 30, 50, and 300. Interestingly, the distribution becomes more concentrated with a smaller variance in round 300 compared to that of round 0. We interpret it as a higher certainty of the aggregation weights in the end of the training process than that in the beginning (starting from an initialization with $\beta = (6.0, 6.0, 6.0)$).

Other analysis experiments, *i.e.*, the impact of the interval t_0 for the aggregation weight learning, and the effect of the re-initialization for aggregation weights before learning each round, are studied in the supplementary materials.

5.4.3 Multi-institutional Pancreas Segmentation

Dataset description. In this experiment, we study pancreas segmentation from CT scans, which is an important pre-requisite of pancreatic tumor detection and surgical planning [178]. We use the provided annotations from three public datasets, *i.e.*, (i) the pancreas subset of Medical Segmentation Decathlon [63] which contains 281 cases (denoted as Dataset I), (ii) the Cancer Image Archive Pancreas-CT dataset [12] which contains 82 cases (denoted as Dataset II), and (iii) Beyond the Cranial Vault Abdomen dataset [96] which contains 30 cases (denoted as Dataset III). All the data include manual per voxel annotations of the pancreas from radiologists. For each dataset, we randomly split the annotated cases into training/validation/test sets, which are 95/93/93 for Dataset I, 28/27/27 for Dataset II, and 10/10/10 for Dataset III. Due to page limits, the implementation details are introduced in supplementary material.

Results. We keep the same notation of our experiments as in the previous COVID-19 experiments. We found the conclusions are the same: our Auto-FedAvg outperforms FedAvg in all metrics and "Auto-FedAvg-N-Dirichlet" is the best in both local performance and generalizability, indicating that the network-wise aggregation and using the Dirichlet distribution to model aggregation weights produce the best results. The conclusion is the same as of COVID dataset that network-wise formulation is better than layer-wise formulation and Dirichlet models aggregation weights better than the softmax. FedBN [174] is unstable under this setup. Interestingly, we find that with interval $t_0 = 5$, as denoted as "Auto-FedAvg-N-Dirichlet*", the performance is even better than its $t_0 = 1$ counterpart (Auto-FedAvg-N-Dirichlet). This could result from the benefit of stabilization when the server keeps the aggregation weights fixed during the interval.

Table 5.3. Multi-institutional pancreas segmentation. "Global test avg" is the major metric to measure the generalizability of the FL global model. n specifies the total dataset size at the client.

Method	global test avg	local avg
Local only - I (n=281)	68.20	
Local only - II (n=82)	59.39	65.32
Local only - III (n=30)	51.34	
FedAvg	73.11	72.84
FedAvg - even	72.97	73.49
FedProx [173]	73.29	73.66
FedBN [174]	54.46	57.61
Auto-FedAvg-L-Softmax	73.54	73.92
Auto-FedAvg-L-Dirichlet	73.74	74.17
Auto-FedAvg-N-Softmax	73.22	74.02
Auto-FedAvg-N-Dirichlet	73.90	74.25
Auto-FedAvg-N-Dirichlet*	**74.21**	**74.33**

* With the interval of aggregation weight learning $t_0 = 5$.

5.5 Conclusions, Limitations, and Future Work

In this paper, we introduced Auto-FedAvg, which improves the standard federated learning (FL) algorithm, FedAvg, by automatically and dynamically learning the aggregation weights instead of keeping them fixed. To achieve that, we proposed a communication-efficient algorithm that alternates between updating the local model weights and the global aggregation weight. We further explored different constraints over the aggregation weights and variants of aggregation strategies. Experiments on two multi-institutional medical image segmentation datasets illustrated the effectiveness of our approach on real-world data. We outperformed other state-of-the-art FL algorithms on a heterogeneous partitioning of the CIFAR-10 dataset [139]. Our approach is also more robust than FedBN [174], especially on heterogeneous clients with limited data each.

One limitation of our algorithm is that relatively stable connections between the server and each client are necessary. This is feasible in our "cross-silo" situation but could be problematic in "cross-device" scenarios where new edge devices regularly

drop in or out [138]. As a result, decreasing the communication frequency and integrating mechanisms for tolerating regular disconnections are two directions to improve the scalability of the current design. Our algorithm also introduced a general and flexible means to boost the performance of FL by updating a small number of global parameters and could be combined with differential privacy techniques for added protection against potential inversion attacks [153], [179]. We only explored the network-wise and layer-wise learning of aggregation weights in this work. However, more options are worth exploring, such as more complex aggregation operations and additional parameters to allow further personalization for addressing non-i.i.d issues in FL.

Algorithm 2: Auto-FedAvg. We denote the total number of rounds as T, the interval to learn aggregation weights as t_0, local training iterations for client k as M_k, and the aggregation weight learning iterations as S.

Server executes:

Define $\boldsymbol{\alpha}^t = [\alpha_1^t, .., \alpha_K^t]$, $\boldsymbol{\beta}^t = [\beta_1^t, .., \beta_K^t]$.
Initialize w^0 and $\boldsymbol{\beta}^0$. $\boldsymbol{\alpha}^0 = \gamma(\boldsymbol{\beta}^0)$
for $t \leftarrow 1, ..., T$ **do**
  for $k \leftarrow 1, ..., K$ **in parallel do**
    $w_k^t \leftarrow \texttt{LocalTrain}(k, w^{t-1})$
  if $t \bmod t_0 = 0$ **then**
    $\boldsymbol{\beta}^t \leftarrow \texttt{LearnAggWeight}(w_1^t, .., w_K^t, \boldsymbol{\beta}^{t-1})$
    $\boldsymbol{\alpha}^t \leftarrow \gamma(\boldsymbol{\beta}^t)$
  else
    $\boldsymbol{\alpha}^t \leftarrow \boldsymbol{\alpha}^{t-1}$
  $w^t \leftarrow \sum_{k=1}^K \alpha_k^t w_k^t$
return w^T

`LocalTrain`(k, w):

for $t \leftarrow 1, .., M_k$ **do**
  Sample batch x from client k's training data
  Compute loss $l(w; x)$
  Compute gradient of w and update w
return w

`LearnAggWeight`$(w_1, .., w_K, \boldsymbol{\beta}^0)$:

for $k \leftarrow 1, ..., K$ **do**
  Server send $w_1, .., w_{k-1}, w_{k+1}, .., w_K$ to client k
for $s \leftarrow 1, .., S$ **do**
  for $k \leftarrow 1, ..., K$ **in parallel do**
    Sever send $\boldsymbol{\beta}^{s-1}$ to client k
    Sample batch x from client k's local data
    Compute loss $\mathcal{L}(\boldsymbol{\beta}^{s-1}; x)$
    Compute/estimate gradient and update $\boldsymbol{\beta}^{s-1}$ as $\boldsymbol{\beta}^{s,k}$
    Send $\boldsymbol{\beta}^{s,k}$ back to the server
  $\boldsymbol{\beta}^s \leftarrow \frac{1}{K} \sum_{k=1}^K \boldsymbol{\beta}^{s,k}$
return $\boldsymbol{\beta}^S$

Chapter 6

Detecting Pancreatic Ductal Adenocarcinoma in Multi-phase CT Scans via Alignment Ensemble

Pancreatic ductal adenocarcinoma (PDAC) is one of the most lethal cancers among the population. Screening for PDACs in dynamic contrast-enhanced CT is beneficial for early diagnosis. In this paper, we investigate the problem of automated detecting PDACs in multi-phase (arterial and venous) CT scans. Multiple phases provide more information than single phase, but they are unaligned and inhomogeneous in texture, making it difficult to combine cross-phase information seamlessly. We study multiple phase alignment strategies, *i.e.*, early alignment (image registration), late alignment (high-level feature registration), and slow alignment (multi-level feature registration), and suggest an ensemble of all these alignments as a promising way to boost the performance of PDAC detection. We provide an extensive empirical evaluation on two PDAC datasets and show that the proposed alignment ensemble significantly outperforms previous state-of-the-art approaches, illustrating the strong potential for clinical use.

6.1 Introduction

Pancreatic ductal adenocarcinoma (PDAC) is the third most common cause of cancer death in the US with a dismal five-year survival of merely 9% [180]. Computed tomography (CT) is the most widely used imaging modality for the initial evaluation of suspected PDAC. However, due to the subtle early signs of PDACs in CTs, they are easily missed by even experienced radiologists.

Recently, automated PDAC detection in CT scans based on deep learning has received increasing attention [20], [181]–[183], which offers great opportunities in assisting radiologists to diagnosis early-stage PDACs. But, most of these methods only unitize one phase of CT scans, and thus fail to achieve satisfying results.

In this paper, we aim to develop a deep learning based PDAC detection system taking multiple phases, *i.e.*, arterial and venous, of CT scans into account. This system consists of multiple encoders, each of which encodes information for one phase, and a segmentation decoder, which outputs PDAC detection results. Intuitively, multiple phases provide more information than a single phase, which certainly benefits PDAC detection. Nevertheless, how to combine this cross-phase information seamlessly is non-trivial. The challenges lie in two folds: 1) Tumor texture changes are subtle and appear differently across phases; 2) Image contents are not aligned across phases because of inevitable movements of patients during capturing multiple phases of CT scans. Consequently, a sophisticated phase alignment strategy is indispensable for detecting PDAC in multi-phase CT scans. An visual illustration is shown in Fig. 7.1.

We investigate several alignment strategies to combine the information across multiple phases. (1) **Early alignment** (EA): the alignment can be done in image space by performing image registration between multiple phases; (2) **Late alignment** (LA): it can be done late in feature space by performing spatial transformation between the encoded high-level features of multiple phases; (3) **Slow alignment**

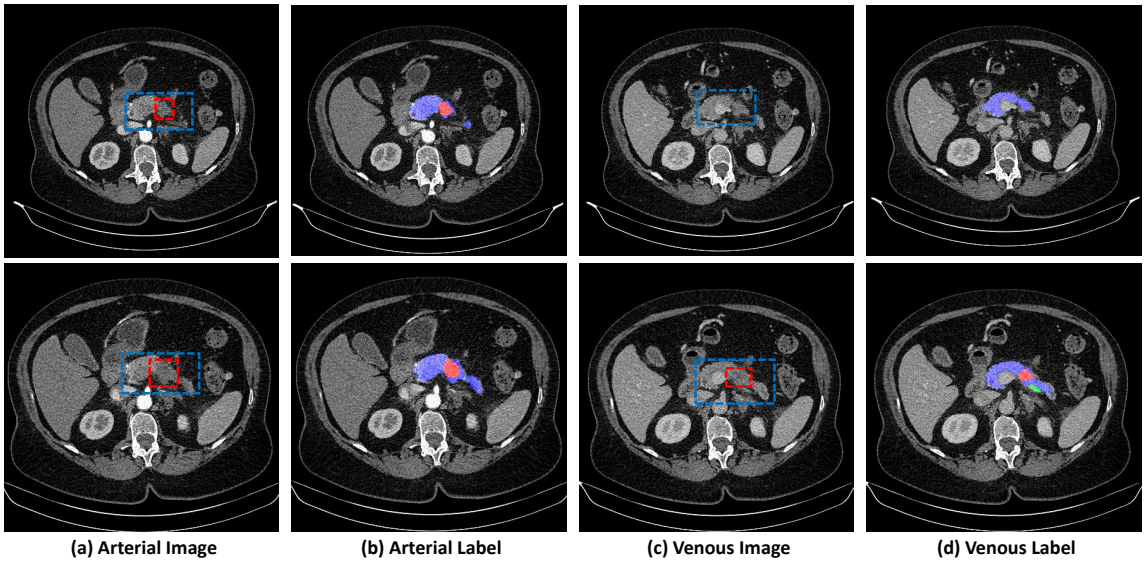


Figure 6.1. Visual illustration of opportunity (top row) and challenge (bottom row) for PDAC detection in multi-phase CT scans (normal pancreas tissue - blue, pancreatic duct - green, PDAC mass - red). Top: tumor is barely visible in venous phase alone but more obvious in arterial phase. Bottom: there exist misalignment for images in these two phases given different organ size/shape and image contrast.

(SA): it can be also done step-wise in feature space by aggregating multi-level feature transformations between multiple phases. Based on an extensive empirical evaluation on two PDAC datasets [20], [183], we observe that 1) All alignment strategies are beneficial for PDAC detection, 2) alignments in feature space leads to better PDAC (tumor) segmentation performance than image registration, and (3) different alignment strategies are complementary to each other, *i.e.*, an ensemble of them (**Alignment Ensemble**) significantly boosts the results, *e.g.*, approximately 4% tumor DSC score improvements over our best alignment model.

Our contributions can be summarized as follows:

- We propose late and slow alignments as two novel solutions for detecting PDACs in multi-phase CT scans and provide extensive experimental evaluation of different phase alignment strategies.
- We highlight early, late and slow alignments are complementary and a simple

ensemble of them is a promising way to boost performance of PDAC detection.

- We validate our approach on two PDAC datasets [20], [183] and achieve state-of-the-art performances on both of them.

6.2 Related Work

6.2.1 Automated Pancreas and Pancreatic Tumor Segmentation

With the recent advances of deep learning, automated pancreas segmentation has achieved tremendous improvements [8]–[10], [12], [34], [83], [184], [185], which is an essential prerequisite for pancreatic tumor detection. Meanwhile, researchers are pacing towards automated detection of pancreatic adenocarcinoma (PDAC), the most common type of pancreatic tumor (85%) [186]. Zhu *et al.* [20] investigated using deep networks to detect PDAC in CT scans but only segmented PDAC masses in venous phase. Zhou *et al.* [183] developed the a deep learning based approach for segmenting PDACs in multi-phase CT scans, *i.e.* arterial and venous phase. They used a traditional image registration [187] approach for pre-alignment and then applied a deep network that took both phases as input. Different to their method, we also investigate how to register multiple phases in feature space.

6.2.2 Multi-modal Image Registration and Segmentation

Multi-modal image registration [21], [187]–[189] is a fundamental task in medical image analysis. Recently, several deep learning based approaches, motivated by Spatial Transformer Networks [190], are proposed to address this task [191]–[193]. In terms of multi-modal segmentation, most of the previous works [183], [194], [195] perform segmentation on pre-registered multi-modal images. We also study these strategies for multi-modal segmentation, but we explore more, such as variants of end-to-end

frameworks that jointly align multiple phases and segment target organs/tissues.

6.3 Methodology

6.3.1 Problem Statement

We aim at detecting PDACs from unaligned two-phase CT scans, *i.e.*, the venous phase and the arterial phase. Following previous works [20], [183], venous phase is our fixed phase and arterial phase is the moving one. For each patient, we have an image $\mathbf{X}$ and its corresponding label $\mathbf{Y}$ in the venous phase, as well as an arterial phase image $\mathbf{X}'$ without label. The whole dataset is denoted as $S = \{(\mathbf{X_i}, \mathbf{X'_i}, \mathbf{Y_i})|i = 1, 2, ..M\}$, where $\mathbf{X_i} \in \mathbb{R}^{H_i \times W_i \times D_i}$, $\mathbf{X'_i} \in \mathbb{R}^{H'_i \times W'_i \times D'_i}$ are 3D volumes representing the two-phase CT scans of the i-th patient. $\mathbf{Y_i} \in \mathcal{L}$ is a voxel-wise annotated label map, which have the same (H_i, W_i, D_i) three dimensional size as $\mathbf{X_i}$. Here, $\mathcal{L} = \{0, 1, 2, 3\}$ represents our segmentation targets, *i.e.*, background, healthy pancreas tissue, pancreatic duct (crucial for PDAC clinical diagnoses) and PDAC mass, following previous literature [20], [183]. Our goal is to find a mapping function $\mathcal{M}$ whose inputs and outputs are a pair of two-phase images $\mathbf{X}, \mathbf{X}'$ and segmentation results $\mathbf{P}$, respectively: $\mathbf{P} = \mathcal{M}(\mathbf{X}, \mathbf{X}')$. The key problem here is how to align $\mathbf{X}$ and $\mathbf{X}'$, either in image space or feature space.

6.3.2 Cross-phase Alignment and Segmentation

As shown in Fig 7.2, we propose and explore three types of alignment strategies, *i.e.*, early alignment, late alignment and slow alignment, for accurate segmentation.

6.3.2.1 Early (image) alignment

Early alignment, or image alignment strategy is adopted in [183] and some other multi-modal segmentation tasks such as BraTS challenge [194], where multiple phases (modalities) are first aligned by image registration algorithms and then fed forward into deep networks for segmentation. Here, we utilize a well-known registration algorithm,

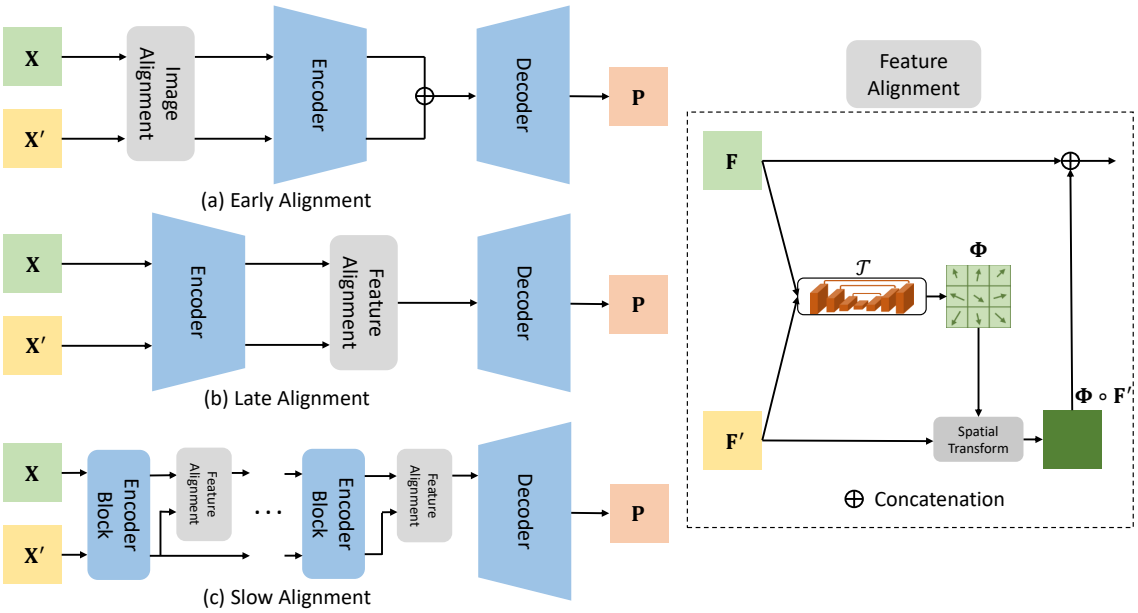


Figure 6.2. An illustration of (a) early alignment (image registration) (b) late alignment and (c) slow alignment. Right: feature alignment block.

DEEDS [21], to estimate the registration field $\mathbf{\Phi}$ from an arterial image $\mathbf{X}'$ to its corresponding venous image $\mathbf{X}$. After registration, we use a network, consisting of two separtae encoders $\mathcal{F}$, $\mathcal{F}'$ and a decoder $\mathcal{G}$, to realize the mapping function $\mathcal{M}$:

$$\mathbf{P} = \mathcal{M}(\mathbf{X}, \mathbf{X}') = \mathcal{G}(\mathcal{F}(\mathbf{X}) \oplus \mathcal{F}'(\mathbf{\Phi} \circ \mathbf{X}')), \tag{6.1}$$

where $\oplus$ and $\circ$ denote the concatenation of two tensors and the element-wise deformation operations on a tensor, respectively.

This strategy relies on the accuracy of image registration algorithms for information alignment. If such algorithms produce errors, especially possible on subtle texture changes of PDACs, these errors will propagate and there will be no way to rescue (since alignment is only done on image level). Also, it remains a question that how much performance gain a segmentation algorithm will achieve through this separate registration procedure.

6.3.2.2 Late alignment

An alternative way is late alignment, *i.e.*, alignment in feature space. We first encode the pair of unaligned images $(\mathbf{X}, \mathbf{X}')$ with two phase-specific encoders $(\mathcal{F}, \mathcal{F}')$, respectively. The encoded features of the two images, *i.e.*, $\mathbf{F} = \mathcal{F}(\mathbf{X})$ and $\mathbf{F}' = \mathcal{F}'(\mathbf{X}')$, are presumablely in a shared feature space. We then use a network $\mathcal{T}$ to estimate the deformable transformation field $\mathbf{\Phi}$ from arterial (moving) to venous (fixed) in the feature space by $\mathbf{\Phi} = \mathcal{T}(\mathbf{F}, \mathbf{F}')$. We apply the estimated transformation field $\mathbf{\Phi}$ to feature map $\mathbf{F}'$, then concatenate this transformed feature map $\mathbf{\Phi} \circ \mathbf{F}'$ to $\mathbf{F}$. The segmentation result $\mathbf{P}$ is obtained by feeding the concatenation to a decoder $\mathcal{G}$:

$$\mathbf{P} = \mathcal{M}(\mathbf{X}, \mathbf{X}') = \mathcal{G}(\mathbf{F} \oplus \mathbf{\Phi} \circ \mathbf{F}') = \mathcal{G}(\mathcal{F}(\mathbf{X}) \oplus \mathcal{T}(\mathbf{F}, \mathbf{F}') \circ \mathcal{F}'(\mathbf{X}')). \tag{6.2}$$

We name such operation as "late alignment" since the alignment is performed at the last block of feature encoders.

6.3.2.3 Slow alignment

Late alignment performs one-off registration between two phases by only using high level features. However, it is known that the low level features of the deep network contain more image details, which motivates us to gradually align and propagate the features from multiple levels of the deep network. Following this spirit, we propose slow alignment, which leverages a stack of convolutional encoders and feature alignment blocks to iteratively align feature maps of two phases.

Let k be an integer which is not less than 1 and $(\mathbf{F}_{k-1}, \mathbf{F}'_{k-1})$ are the fused (aligned to the venous phase) feature map and the arterial feature map outputted by the $(k-1)^{th}$ convolutional encoder, respectively. First, they are encoded by a pair of convolutional encoders $(\mathcal{F}_k, \mathcal{F}'_k)$, respectively, which results in the venous feature map $\mathbf{F}_k = \mathcal{F}_k(\mathbf{F}_{k-1})$ and the arterial feature map $\mathbf{F}'_k = \mathcal{F}'_k(\mathbf{F}'_{k-1})$ at the k-th layer. Then a

feature alignment block estimates a transformation field from the arterial (moving) phase to the venous (fixed) phase by

$$\mathbf{\Phi}_k = \mathcal{T}_k(\mathcal{F}_k(\mathbf{F}_{k-1}), \mathcal{F}'_k(\mathbf{F}'_{k-1})), \tag{6.3}$$

where $\mathcal{T}_k$ is a small U-Net. We apply the transformation field $\mathbf{\Phi}_k$ to the arterial (moving) phase, resulting in transformed arterial feature map $\mathbf{\Phi}_k \circ \mathcal{F}'_k(\mathbf{F}'_{k-1})$. Finally, the transformed arterial feature map is concatenated with the venous feature map $\mathcal{F}_k(\mathbf{F}_{k-1})$, resulting in the fused feature map at the k^{th} layer:

$$\mathbf{F}_k = \mathcal{F}_k(\mathbf{F}_{k-1}) \oplus \mathbf{\Phi}_k \circ \mathcal{F}'_k(\mathbf{F}'_{k-1}). \tag{6.4}$$

Let us rewrite the above process by a function $\mathcal{R}_k$: $\mathbf{F}_k = \mathcal{R}_k(\mathbf{F}_{k-1}, \mathbf{F}'_{k-1})$ and define $\mathbf{F}_0 = \mathbf{X}$ and $\mathbf{F}'_0 = \mathbf{X}'$, then we can iteratively derive the fused feature map at n-th convolutional encoder:

$$\mathbf{F}_n = \mathcal{R}_n\Big(\mathcal{R}_{n-1}\Big(\cdots\Big(\mathcal{R}_1(\mathbf{F}_0, \mathbf{F}'_0), \mathbf{F}'_1\Big), \cdots\Big), \mathbf{F}'_{n-1}\Big), \tag{6.5}$$

where $\mathbf{F}'_{n-1} = \mathcal{F}'_{n-1}(\mathcal{F}'_{n-2}(\cdots(\mathcal{F}'_1(\mathbf{F}'_0))))$. The final fused feature map $\mathbf{F}_n$ is fed to the decoder $\mathcal{G}$ to compute the segmentation result $\mathbf{P}$:

$$\mathbf{P} = \mathcal{M}(\mathbf{X}, \mathbf{X}') = \mathcal{G}(\mathbf{F}_n). \tag{6.6}$$

6.3.2.4 Alignment Ensemble

We ensemble the three proposed alignment variants by simple majority voting of the predictions. The goal of the ensemble are in two folds, where the first is to improve overall performance and the second is to see whether these three alignment methods are complementary. Usually, an ensemble of complementary approaches can lead to large improvements.

6.4 Experiments and discussion

6.4.1 Dataset and evaluation

We evaluate our approach on two PDAC datasets, proposed in [20] and [183] respectively. For the ease of presentation, we regard the former as PDAC dataset I and the latter as PDAC dataset II. PDAC dataset I contains 439 CT scans in total, in which 136 cases are diagnosed with PDAC and 303 cases are normal. Annotation contains voxel-wise labeled pancreas and PDAC mass. Evaluation is done by 4 fold cross-validation on these cases following [20]. PDAC dataset II contains 239 CT scans, all from PDAC patients, with pancreas, pancreatic duct (crucial for PDAC detection) and PDAC mass annotated. Evaluation are done by 3 fold cross-validation following [183].

All cases contain two phases: arterial phase and venous phase, with a spacing of 0.5mm in axial view and all annotations are verified by experienced board certified radiologists. The segmentation accuracy is evaluated using the Dice-Sørensen coefficient (DSC): DSC $(\mathcal{Y}, \mathcal{Z}) = \frac{2\times|\mathcal{Y}\cap\mathcal{Z}|}{|\mathcal{Y}|+|\mathcal{Z}|}$, which has a range of $[0, 1]$ with 1 implying a perfect prediction for each class. On dataset I, we also evaluate classification accuracy by sensitivity and specificity following a "segmentation for classification" strategy proposed in [20].

6.4.2 Implementation details

We implemented our network with PyTorch. The CT scans are first truncated within a range of HU value [-100, 240] and normalized with zero mean and unit variance. In training stage, we randomly crop a patch size of 96^3 in roughly the same position from both arterial and venous phases. The optimization objective is Dice loss [26]. We use SGD optimizer with initial learning 0.005 and a cosine learning rate schedule for 40k iterations. For all our experiments, we implement the encoder and decoder

Method	N.Pancreas	A.Pancreas	Tumor	Misses	Sens.	Spec.
U-Net [43]	86.9±8.6	81.0±10.8	57.3±28.1	10/136	92.7	**99.0**
V-Net [26]	87.0±8.4	81.6±10.2	57.6±27.8	11/136	91.9	99.0
MS C2F [20]	84.5± 11.1	78.6 ± 13.3	56.5± 27.2	8/136	94.1	98.5
Baseline - NA	85.8±8.0	79.5±11.2	58.4±27.4	11/136	91.9	96.0
Ours - EA	86.7±9.7	81.8±10.0	60.9±26.5	4/136	97.1	94.5
Ours - LA	87.5±7.6	82.0±10.3	62.0±27.0	7/136	94.9	96.0
Ours - SA	87.0±7.8	82.8±9.4	60.4±27.4	4/136	97.1	96.5
Ours - Ensemble	**87.6±7.8**	**83.3±8.2**	**64.4±25.6**	**4/136**	**97.1**	96.0

Table 6.1. Results on PDAC dataset I with both healthy and pathological cases. We compare our variants of alignment methods with the state-of-the-art method [20] as well as our baseline - no align (NA) version. "Misses" represents the number of cases failed in tumor detection. We also report healthy vs. pathological case classification (sensitivity and specificity) based on segmentation results. The last row is the ensemble of the three alignments.

architecture as U-Net [43] with 4 downsampling layers, making a total alignments of $n = 4$ in Eq 6.6. The transformation fields are estimated by light-weighted U-Nets in late alignment and slow alignment, each is $\sim 8\times$ smaller than the large U-Net for segmentation, since the inputs of the small U-Nets are already the compact encoded features. The computation of EA/LA/SA is approximately 1.5/1.7/1.9 times of the computation of a single-phase U-Net. The image registration algorithm for our early alignment is DEEDS [21].

6.4.3 Results

Results on dataset I and II are summarized in Table 6.1 and Table 6.2 respectively, where our approach achieves the state-of-the-art performance on both datasets. Based on the results, we have three observations which leads to three findings.

Dual-phase alignments are beneficial for detecting PDACs in multi-phase CT scans. On both datasets, our approaches, *i.e.* early alignment, late alignment and slow alignment, outperform single phase algorithms, i.e. U-Net [43], V-Net [26], ResDSN [10] and MS C2F [20], as well as our non-alignment dual-phase version (Baseline-NA).

Method	A.Pancreas	Tumor	Panc. duct	Misses
U-Net [43]	79.61±10.47	53.08±27.06	40.25±27.89	11/239
ResDSN [10]	84.92±7.70	56.86±26.67	49.81±26.23	11/239
HPN-U-Net [183]	82.45±9.98	54.36±26.34	43.27±26.33	-/239
HPN-ResDSN [183]	85.79±8.86	60.87±24.95	54.18±24.74	7/239
Ours - EA	83.65±9.22	60.87±22.15	55.38±29.47	**5/239**
Ours - LA	86.82±6.13	62.02±24.53	64.35±29.94	9/239
Ours - SA	87.13±5.85	61.24±24.26	64.19±29.46	8/239
Ours - Ensemble	**87.37±5.67**	**64.14±21.16**	**64.38±29.67**	6/239

Table 6.2. Results on PDAC dataset II with pathological cases only. We compare our variants of alignment methods with the state-of-the-art method [183]. "Misses" represents the number of cases failed in tumor detection. The last row is the ensemble of the three alignments.

Feature space alignments have larger improvements on segmentation performances than early alignments. Generally speaking for both datasets, our feature space alignment models (LA, SA) outperform image registration based approaches, i.e. HPN, Ours-EA, in terms of segmentation performance. Since early alignment methods apply image registration in advance, they do not guarantee a final improvement on segmentation performance. In contrast, feature space alignment methods jointly align and segment the targets in an end-to-end fashion by optimizing the final segmentation objective function, which leads to a larger improvements compared with single phase or naive dual phase methods without alignment. However,

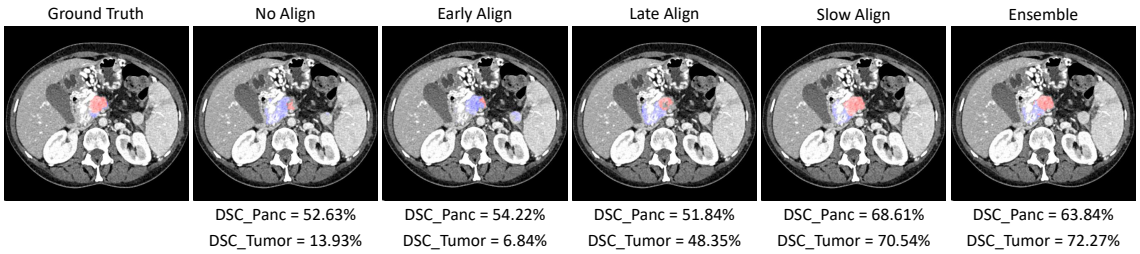


Figure 6.3. An example of PDAC dataset I on venous phase. From left to right, we display ground-truth, prediction of our baseline without alignment, prediction of our early align, late align, slow align and alignment ensemble. Our feature space alignments (LA, SA) outperform no-align baseline and image registration (EA). Ensemble of the three alignment predictions also improves tumor segmentation DSC score.

we indeed observe that early alignment leads to relatively less false negatives (misses).

An ensemble of the three alignment strategies significantly improve the performances. For both dataset, Ours-Ensemble achieves the best performances, illustrating that the three alignment strategies are **complementary** to each other. An ensemble leads to significant performance gain (relatively 4% improvements on tumor segmentation DSC score compared to the best alignment model from 62.0% to 64.4%) and achieves the state-of-the-art performances on both datasets. A qualitative analysis is also shown in Fig 6.3.

Last but not least, our alignment approaches also improve the sensitivity of healthy vs. pathological classification. In dataset I, we adopt the same "segmentation for classification" strategy as in [183], which classifies a case as pathological if we are able to detect any tumor mass larger than 50 voxels. Our approach can improve the overall sensitivity from 94.1% to 97.1% by reducing misses from 8 to 4, which is beneficial for the early detection of PDAC. Our approach thus has valuable potential of winning precious time for early treatments for patients.

6.5 Conclusion

In this paper, we study three types of alignment approaches for detecting pancreatic adenocarcinoma (PDACs) in multi-phase CT scans. Early alignment first applies registration in image space and then segment with a deep network. Late alignment and slow alignment jointly align and segment with an end-to-end deep network. The former aligns in the final encoded feature space while the latter aligns multi-stage features and propagate slowly. An ensemble of the three approaches improve the performances significantly illustrating these alignment variants are complementary to each other. We achieve the state-of-the-art performances on two PDAC datasets.

Chapter 7

Effective Pancreatic Cancer Screening on Non-contrast CT Scans via Anatomy-Aware Transformers

Pancreatic cancer is a relatively uncommon but most deadly cancer. Screening the general asymptomatic population is not recommended due to the risk that a significant number of false positive individuals may undergo unnecessary imaging tests (e.g., multi-phase contrast-enhanced CT scans) and follow-ups, adding health care costs greatly and no clear patient benefits. In this work, we investigate the feasibility of using a single-phase non-contrast CT scan, a cheaper, simpler, and safer substituent, to detect resectable pancreatic mass and classify the detection as pancreatic ductal adenocarcinoma (PDAC) or other abnormalities (nonPDAC) or normal pancreas. This task is usually poorly performed by general radiologists or even pancreatic specialists. With pathology-confirmed mass types and knowledge transfer from contrast-enhanced CT to non-contrast CT scans as supervision, we propose a novel deep classification model with an anatomy-guided transformer. After training on a large-scale dataset including 1321 patients: 450 PDACs, 394 nonPDACs, and 477 normal, our model achieves a sensitivity of 95.2% and a specificity of 95.8% for the detection of abnormalities on the holdout testing set with 306 patients. The

mean sensitivity and specificity of 11 radiologists are 79.7% and 87.6%. For the 3-class classification task, our model outperforms the mean radiologists by absolute margins of 25%, 22%, and 8% for PDAC, nonPDAC, and normal, respectively. Our work sheds light on a potential new tool for large-scale (opportunistic or designed) pancreatic cancer screening, with significantly improved accuracy, lower test risk, and cost savings.

7.1 Introduction

Pancreatic cancer is the third leading cause of death among all cancers in the United States, with a 5-year overall survival rate of ∼10% [196]. Surgical resection by now remains the only treatment that offers curative potential [197], but more than 80% of patients with pancreatic cancer have already lost the opportunity of surgery at the first diagnosis. Thus, screening pancreatic cancer is very important to provide early diagnosis and patient risk monitoring. The most widely used imaging modality for the initial evaluation of suspected pancreatic cancer is the contrast-enhanced CT scan (CECT). The benefit of CECT for early pancreatic cancer detection includes high sensitivity and specificity, general standardization and availability and relatively easy interpretation [198]. However, CECT exposes patients to radiation and requires iodine contrast, which can cause reaction and potential risks in patients [198], making it hard to be recognized as a general protocol to screen for pancreatic cancer.

In this work, we investigate the possibility of using non-contrast CT scans (NCCT) to screen for pancreatic cancer with deep learning. Compared to CECT, NCCT is cheaper and safer, because it does not require iodine contrast and exposes patients to less radiation. NCCT has been generally applied in screening for lung nodules [199] which can possibly be reused for opportunistic pancreatic cancer screening as well. Nevertheless, due to the low contrast in NCCT pancreatic region, the difficulty of tumor detection rises significantly for radiologists without contrast enhancement.

Deep networks, on the other hand, are particularly good at discovering local texture and shape geometry changes, which give us an opportunity to detect pancreatic cancer even without contrast enhancement on NCCT. Those miss detections by human eyes due to low visual contrast do not necessarily become the false negatives by deep learning (DL) detectors.

One major challenge of training deep learning models on NCCT is the difficulty of obtaining expert annotations. Even experienced radiologists could miss masses due to the low contrast on NCCT. This problem is tackled in the process of data collection from the following two aspects. (1) We obtain the pathology-confirmed mass type as classification ground-truth for patients with pancreatic ductal adenocarcinoma (PDAC) or non-PDAC. (2) For the pixel-level labeling of pancreatic tumor mass, the radiologist first annotates on the contrast-enhanced CT; we then transfer the segmentation mask from contrast-enhanced CT to non-contrast CT by performing volumetric registration on the same patient. The combined classification ground-truth labels and segmentation masks serve as the supervisions of our deep learning model with the input of non-contrast CT scans only. The pathology-confirmed mass type and knowledge transfer from CECT to NCCT are the two important pre-requisites for our model to surpass the human expert performance on detecting pancreatic cancer via NCCT.

In terms of the design of deep models, we extend the previous "Segmentation for Classification" [20] paradigm by building a deep classification on top of a segmentation model with transformers [2]. Given the fact that local texture could be insufficient to detect tumors in NCCT, we adopt Transformers to model the pancreas anatomy structure for better classification, which can capture the global context with multi-head attention. This is also in line with the practical diagnosis experience of the radiologists, where sometimes abnormality is discovered by the secondary-sign, such as swelling pancreas head/tail or pancreatic duct dilation, without actually seeing the tumor.

To validate the feasibility of the proposed solution, we collect a large-scale dataset, which covers 1627 patients: 558 PDACs, 474 nonPDACs, and 595 normal. Our model achieves a sensitivity of 95.2% and a specificity of 95.8% on the holdout test set, in terms of abnormality detection. In contrast, the average performance of 11 expert radiologists is 79.7% and 87.6%. This result illustrates the superiority of our designed deep learning-based framework in this specialized task of detecting pancreatic cancer in NCCT. This work sheds light on a potential viable and safe protocol to screen pancreatic cancers on general population.

The main contributions of this paper are summarized as follows.

- For the first time, non-contrast CT (NCCT) is proposed and validated as an effective imaging modality for full-spectrum taxonomy of pancreatic mass/disease screening using deep learning. This sheds light on new computing tools for large-scale opportunistic or designated pancreatic cancer screening of improved accuracy, lower test risk, and cost savings.
- We utilize the pathology-confirmed mass labels, and transfer the imaging information and knowledge from CECT to NCCT as supervision, which is a prerequisite to surpass human expert performance in this task.
- We propose a new framework, named Anatomy-aware Hybrid Transformers, outperforming the mainstream "Segmentation for Classification" paradigm.
- We achieve a sensitivity of 95.2% and a specificity of 95.8% on a large-scale dataset with 1627 patients, demonstrating the good potential of using more convenient non-contrast CT scans for pancreatic cancer screening.

7.1.1 Related Work

Automated pancreatic tumor detection. Recent advances in deep learning have lead to tremendous improvement in pancreas segmentation [8]–[10], [12], [34], [83], [184],

[185], an important pre-requisite step for pancreatic tumor detection. Researchers have started to explore the task of automated pancreatic tumor detection using contrast-enhanced CT scans with deep networks [20], [181], [200]–[202] and radiomics [182]; as well as the task of cancer prognosis prediction [203]. Different from previous work, our framework is designed for non-contrast CT scans, which is beneficial for general asymptomatic patients yet much more challenging.

Vision transformers. Transformer [2] utilizes attention mechanism originally designed for language tasks. It has recently been applied into vision task, *e.g.*, object detection [204], image recognition [2] and semantic segmentation [205], and achieved comparable or better performance than CNN based approaches.

7.2 Methodology

Problem statement. We formulate the task of pancreatic cancer detection in non-contrast CT scans as a three-class classification problem. We denote $\mathcal{L} = \{0, 1, 2\}$ for the three patient classes, *i.e.*, normal, PDAC and non-PDAC. The reasons of having these three classes are: (1) PDAC is a unique group with the most dismal prognosis; (2) any pancreas CT findings with a influence on patient management options. Screening for pancreatic cancer is much more difficult than lung nodules or mammography screening due to the challenge and visual ambiguity of soft-tissue tumor detection without CT contrast enhancement. A key part in our processing pipeline is the availability of knowledge transfer from contrast-enhanced CT by incorporating pathology-confirmed mass type as classification labels and segmentation labels (tumor/pancreas) used for intermediate supervision, as shown in Fig.7.1. Denote the training set by $S = \{(\mathbf{X_i}, \mathbf{Y_i}, \mathbf{Z_i})|i = 1, 2, ..M\}$, where $\mathbf{X_i} \in \mathbb{R}^{H_i \times W_i \times D_i}$, is the 3D volume representing the non-contrast CT scans of the i-th patient. $\mathbf{Y_i}$ is the voxel-wise annotated label map with the same spatial size as $\mathbf{X_i}$. $\mathbf{Z_i} \in \mathcal{L}$ is the class label of the image, confirmed by pathology, radiology, or clinical records. In the testing phase,

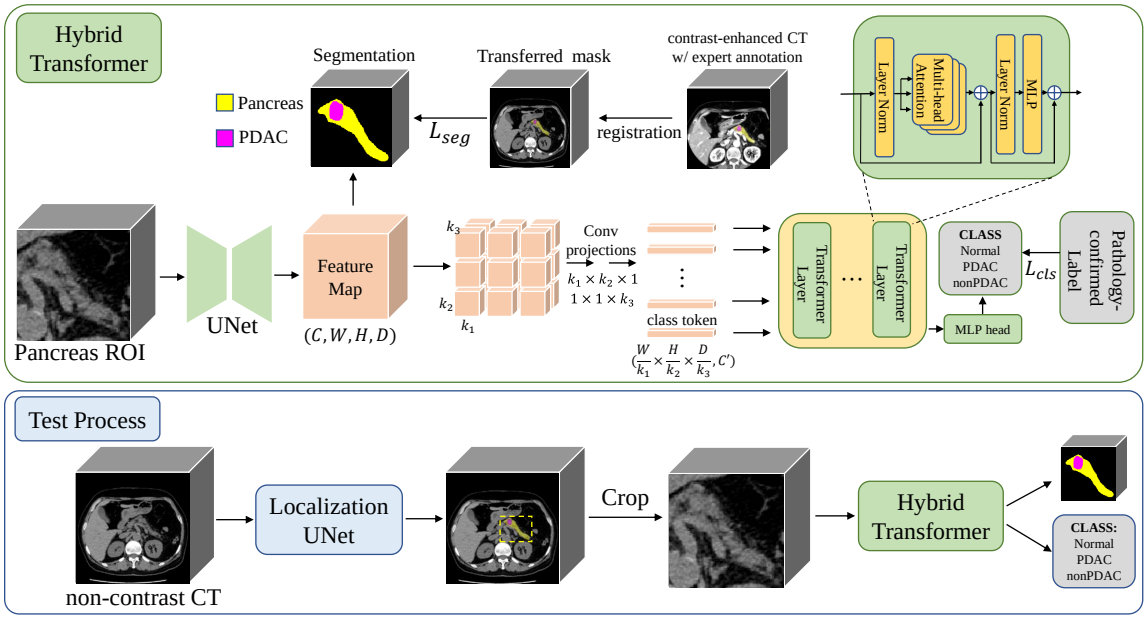


Figure 7.1. A visual illustration of our whole framework. Top: we train our hybrid Vision Transformer on non-contrast CT via two supervisions: (i) class label of normal/PDAC/non-PDAC obtained by pathology-confirmed mass type, and (ii) coarse tumor segmentation label transferred from contrast-enhanced CT by registration. Bottom: in the testing phase, we first crop out the pancreas ROI with a localization UNet (separately trained) and output the class and segmentation prediction with the hybrid transformer given non-contrast CT scans.

only $\mathbf{X_i}$ is given, and our goal is to predict a class label for $\mathbf{X_i}$.

Knowledge transfer from contrast-enhanced to non-contrast CT. Considering the difficulties of mass annotation on non-contrast CT scans (e.g., tumors are barely visible), radiologists first annotate the voxel-wise mass mask on the contrast-enhanced CT scan with the same patient. We then perform image registration using DEEDS [21] from CECT to NCCT and apply the registration field on the manually segmented mass mask. In this way, we can obtain a relatively coarse, but the most reliable mass mask $\mathbf{Y_i}$ on the NCCT image.

7.2.1 Anatomy-aware Classification with Transformers

Segmentation for classification is the most straightforward and adopted representation of the task of pancreatic tumor detection. We train a localization UNet [25] to segment pancreas and mass supervised by the transferred masks generated as above. This localization UNet is also used for cropping out the pancreas ROI region as shown in the test process in Fig 7.1.

Given the superiority of the attention mechanism in modelling the global context, we build a hybrid Vision Transformer [206] on top of the UNet segmentation model (see Fig 7.1). Since the transformer takes the input of the feature map of a segmentation network, we term it as Anatomy-ware Hybrid Transformer. We denote the pancreas ROI region by $\mathbf{X}$, and $\mathbf{X} \in \mathbb{R}^{H \times W \times D}$. We then forward the image $\mathbf{X}$ into a UNet, which consists of a feature extractor $\mathcal{F}$ and an output layer $\mathcal{G}$. This UNet has an intermediate supervision of the mask transferred from contrast-enhanced CT scan of the same patient where the human annotation is available. Therefore, the intermediate output segmentation $\mathbf{P}_s$ can be obtained by $\mathbf{P}_s = \mathcal{G}(\mathcal{F}(\mathbf{X}))$.

The input of the Vision Transformer $\mathcal{H}$ is the final feature map of the UNet right before the output layer, denoted as $\mathcal{F}(\mathbf{X})$, which has a spatial dimension of (C, W, H, D) and C is the number of channels of the feature map. We first use two consecutive 3D convolution layers with a kernel size of $k_1 \times k_2 \times 1$ and $1 \times 1 \times k_3$ to extract $\frac{W}{k_1} \times \frac{H}{k_2} \times \frac{D}{k_3}$ feature patches with C' dimensions each, where C' is also the dimension of the input sequences of the Transformer. Note that previous work [206] directly use one single convolution layer to extract patch features, while we decompose it into two layers to reduce the number of parameters in our 3D settings. Learnable positional embeddings are then added to each patch. These patch features are forwarded through multiple transformer blocks with multi-head attention. Following ViT [206], we also use a class token for classification. The output embedding of the class token is used as the

classification prediction after a MLP (multilayer perceptron). Our overall training objective is formulated as follows:

$$\mathcal{L} = L_{seg}(\mathbf{P}_s, \mathbf{Y}) + L_{cls}(\mathbf{P}_c, \mathbf{Z}), \tag{7.1}$$

where $\mathbf{P}_s = \mathcal{G}(\mathcal{F}(\mathbf{X}))$ and $\mathbf{P}_c = \mathcal{H}(\mathcal{F}(X))$ are the output segmentation of UNet and the final classification prediction of the Transformer, respectively. The loss function for classification is cross-entropy loss.

7.3 Experiments

Dataset and ground truth. Our dataset of CT scans of 1627 patients, is consecutively collected in the years of 2016∼2018 from a high-volume pancreatic cancer institution. PDAC is of the highest priority among all pancreatic abnormalities with a 5-year survival rate of approximately 10% and is the most common type (about 90% of all pancreatic cancers). This is the main reason that we group all abnormalities into two classes of PDAC and nonPDAC (including nine subtypes [5], [202]).The dataset is randomly split into a training and a testing dataset. The training set includes 450 PDACs, 394 nonPDACs, and 477 normal pancreases. The testing set includes 108 PDACs, 80 nonPDACs, 118 normal pancreases. Both PDAC and nonPDAC cases are confirmed by their pathology reports and normal cases by radiology reports and 2-year follow-up. Each patient has multi-phase CT scans. The median imaging spacing is 0.68×0.68×3.0mm in [X,Y,Z]. The manual annotations of masses are performed by an experienced radiologist (with 14 years of specialized experience in pancreatic imaging) on either arterial/pancreatic phase or venous phase with better mass visibility. The annotations of the pancreas are performed automatically by a segmentation model, which is trained on three datasets, including the single-phase pancreas CTs [207] and abdominal CTs [62] as well as our multi-phase CT dataset, by following a self-training

strategy [201], [208].

Reader study. Eleven radiologists (four are board-certified pancreatic imaging specialists) from four high-volume pancreatic cancer institutions read the 306 non-contrast CTs in the testing dataset without time constraint (WOTC), with a three-class decision by each reader: PDAC, nonPDAC, or normal.

Implementation details. Each CT volume is firstly resampled into 0.68 $\times$ 0.68 $\times$ 3.0 mm spacing and normalized into zero mean and unit variance. In the training phase, we crop the foreground 3D bounding box of the pancreas region, randomly pad a small margin on each dimension, and resize the bounding box into a volume of shape (256, 256, 64). The input of the Vision Transformer is the final feature map of the UNet right before the output layer, which has a shape of (32, 256, 256, 64). The two consecutive 3D convolution layers have the kernel size of 32$\times$32$\times$1 and 1$\times$1$\times$8 which leads to 512 feature patches with 256 dimensions each. The transformer contains 12 consecutive 8-head attention blocks. We train our hybrid model in an end-to-end fashion with SGD optimizer. The initial learning rate is set to 1×10^{-3} and decays with a cosine learning rate schedule. In addition to the hybrid UNet-Transformer model, we also trained a standard UNet on the whole image for the localization of pancreas. In the inference phase, we first localize the bounding box of the pancreas region with aforementioned UNet, resize the pancreas region into (256, 256, 64) volume and then classify the pancreas region with our hybrid UNet-Transformer model.

Evaluation methods and metrics. We randomly split the training dataset into a training set (80% data) and a validation set (20% data). Since the primary goal of non-contrast CT screening is to distinguish between abnormal (PDAC+nonPDAC) and normal, a cutoff point (i.e., threshold) is used to dichotomize model's output probabilities into binary predictions. The cutoff point is predefined on the validation set by maximizing the value of (sensitivity + specificity) before model evaluation on the testing set. To further classify the abnormal as PDAC or nonPDAC, the one

	2-class			3-class		
Method	AUC	Sens.	Spec.	PDAC	nonPDAC	Normal
S4C with UNet [20]	95.98	91.48	95.76	75.00	73.75	95.76
Hybrid CNN	98.25	94.68	94.91	76.85	80.00	94.91
Hybrid Transformer	**98.37**	**95.21**	**95.76**	**78.70**	**80.00**	**95.76**
Mean radiologists WOTC	-	79.66	87.58	53.63	57.96	87.58

Table 7.1. Results on two-class classification (PDAC+nonPDAC vs. normal) and three-class classification (PDAC vs. nonPDAC vs. normal). WOTC: without time constraint.

with a larger output probability is selected as the prediction. We first report the result of the 2-class classification (PDAC + nonPDAC vs. normal). The evaluation metrics include AUC (area under the ROC curve), sensitivity ($\frac{TP}{TP+FN}$) and specificity ($\frac{TN}{TN+FP}$). We also report the result of the 3-class classification (PDAC vs. nonPDAC vs. normal), measured by class accuracy. In addition, the mass detection rate by our model is assessed. A detection is considered successful if the intersection (between the ground truth and segmentation mask) over the ground truth is $> 0\%$ – a coarse localization of mass would be useful in this application scenario.

Compared methods. We compare our method to two baseline approaches. One is "segmentation for classification" [20] full-filled by a standard UNet where we classify a case as positive if the detected PDAC or nonPDAC tumor volume is larger than a certain threshold, which maximize the value of (sensitivity+specificity) on the validation set. The other is a hybrid CNN classifier built on the UNet feature map trained in an end-to-end fashion. Specifically, we integrate a classification head into the segmentation model. We extract multiple level of the UNet feature map, apply global max pooling on each feature map, concatenate them and forward into a single-layer perceptron for classification. Quantitative results are shown in Table 7.1. The 2-class ROC curve and a case study are shown in Fig 7.2.

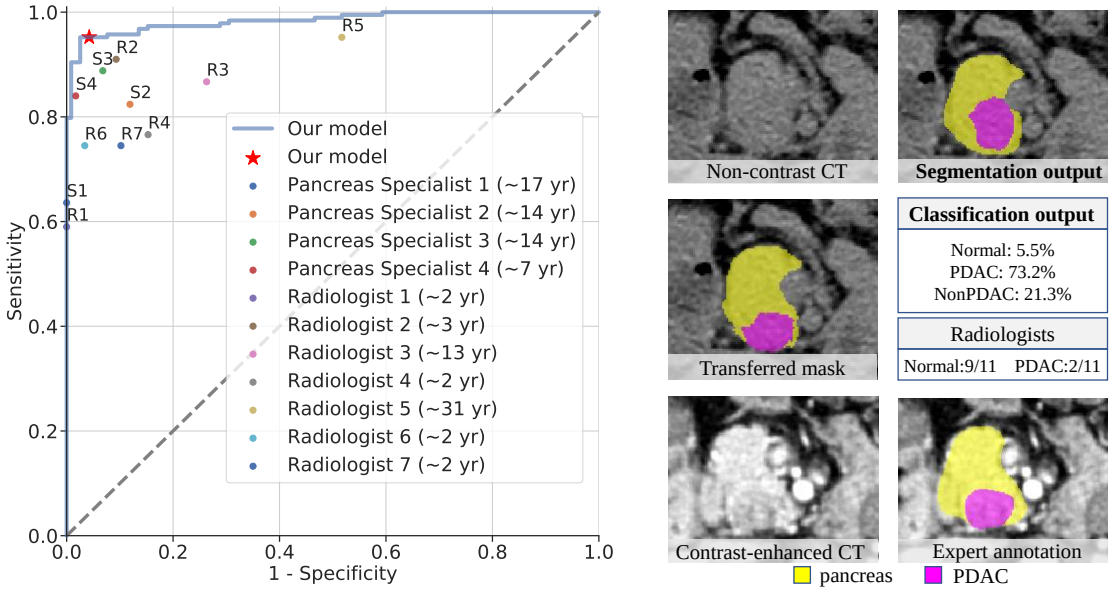
Anatomy-aware transformer outperforms baselines. Compared to two baselines, *i.e.*, segmentation for classification (S4C) and hybrid CNN classifier, our hybrid transformer shows the best performance in all metrics (Table 7.1), with a

relative low STD (about 1%) on sensitivity and specificity. Most medical segmentation models focus on local texture changes and lacks the ability to model the global context. In contrast, our anatomy-aware Hybrid Transformer is built on the locally discriminative features of the UNet, and captures the structural relationship over the whole pancreas region with multi-head attention. Our model is capable of improving the global decision process.

AI models outperforms expert radiologists on non-contrast CT scan. The performance of all 11 radiologists (WOTC) is below our model's ROC curve (Fig 7.2). Our model has a mass detection rate of 87.76%, and its sensitivity in abnormality prediction (95%) outperforms the mean human performance (80%) by a large margin and also surpasses the best performing radiologist (R2: 91%) and specialist (S3: 89%), which is the main goal of pancreatic cancer screening using non-contrast CTs. More surprisingly, for the 3-class classification task (Table 7.1), our model outperforms the mean radiologists (WOTC) by absolute margins of 25%, 22%, and 8% for PDAC, nonPDAC, and normal, respectively.

Human vision system requires adequate visual intensity contrast to distinguish mass from pancreas tissue, which is why contrast-enhanced CT scans are necessary for the diagnosis purpose. Given the surprising performance of DL models on non-contrast CT, we empirically hypothesize that machine vision is better at magnifying the local contrast changes to locate masses. Another crucial reason why computerized model substantially outperforming human performance on non-contrast CT is that we transfer the expert findings from contrast-enhanced CT. Most models are restricted to the performance upper bound of the human annotators. With annotations transfer from CECT (a more "doctor-friendly" modality), and pathology-confirmed labels, DL models are equipped with the essential information/knowledge to break the limit of human observers.

Impact and future work. From the reference of computerized performance



(a) ROC curve **(b)** A case study

Figure 7.2. (a) ROC diagram for our model result versus all other experts' referrals on the test set of n=306 patients for 2-class classification. The asterisk denotes the performance of our model. Filled markers denote 11 experts' performances using the same non-contrast CT only. S1: Pancreas Specialist 1, R1: Radiologist 1. (b) A case study in the test set. This PDAC case is extremely challenging for radiologists (only 2/11 are correct) given the limited intensity contrast in non-contrast CT scans whereas our model can successfully locate the mass and predicts the class label.

using contrast-enhanced CT, (sensitivity, specificity) of PDAC vs. Normal is recorded as (92.7%, 99%) [20] and (97.1%, 96%) [200]. This work involves dealing with nonPDAC masses and is generally harder for deep learning [202]. Our performance on non-contrast CT scans (95.2%, 95.8%) is approaching those methods using contrast-enhanced CT. This finding sheds light on the opportunity to use automated methods to screen pancreatic cancers via non-contrast CT imaging. This may be very beneficial for patients, because non-contrast CT is much cheaper, simpler, and safer than its contrast-enhanced counterpart. We plan to conduct multi-institutional studies to validate the generalizability of our system.

7.4 Conclusion

In this paper, we explore detecting pancreatic cancer from non-contrast CT scans, as a relatively cheap, convenient, simple and safe imaging modality. We propose a hybrid transformer model which is trained by the supervision of pathology-confirmed mass types and the segmentation masks transferred from contrast-enhanced CT scans. We achieve high sensitivity and specificity on a large-scale dataset and outperform the mean radiologists by large margins. Our work suggests the good feasibility of using non-contrast CT scans as a promising clinical tool for large-scale pancreatic cancer screening.

Chapter 8

Conclusion and Future Work

8.1 Summary

In this dissertation, we focus on the topic of improving the robustness of deep learning models for automated medical image analysis. Our studies include but are not limited to improving network architecture, data efficiency, failure detection, domain robustness, and multi-phase learning. In particular, we introduce a new 2.5D framework that leverages the benefits from both 2D and 3D networks for effective and efficient medical image segmentation in Chapter 2. In Chapter 3, we improve data efficiency by introducing a semi-supervised learning framework for 3D medical images. This approach is also validated for the purpose of domain adaptation. In Chapter 4, we design a failure and anomaly detection algorithm for segmentation models in this safety-critical area. Federated learning offers the opportunity for multiple institutions to train a generalizable model while preserving data privacy, and we propose to improve federated learning by dynamically adjusting the aggregation weight of the commonly used FedAvg algorithm in Chapter 5. In Chapter 6, we incorporate the multi-phase information and explore joint alignment and segmentation algorithms for more accurate pancreatic tumor segmentation in contrast-enhanced CT scan. In Chapter 7, we successfully transfer the knowledge from contrast-enhanced CT scan to non-contrast CT scan for multi-type pancreatic abnormality detection and outperform

experienced radiologists on a large-scale dataset.

8.2 Future work

Despite the initial endeavor of deep learning research for healthcare, there still exists a large gap before AI can be successfully migrated into various clinical procedures. We will discuss potential research directions to accelerate the process. First, the construction of high quality, well-annotated, and the large-scale medical dataset is necessary. The training set of AI models should cover as many edge cases as possible to deal with the unpredictability in real-world applications. Second, the model should have multiple sets of knowledge for comprehensive diagnosis. Current models usually target only one task, but the human body functions under the collaboration of various organs and tissues. For example, in the abdominal region, the diagnosis of tumor metastasis requires the detection of numerous organs and vessels. Multi-task learning is worth exploring for an inclusive understanding of the human anatomy before making a decision. Third, the ability to interpret is also crucial in AI system design. If the diagnosis process of the AI system is explainable, the results will be much easier for clinicians to interpret and better benefit clinical decisions.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[3] K.-H. Yu, A. L. Beam, and I. S. Kohane, "Artificial intelligence in healthcare," *Nature biomedical engineering*, vol. 2, no. 10, pp. 719–731, 2018.

[4] S. M. McKinney, M. Sieniek, V. Godbole, J. Godwin, N. Antropova, H. Ashrafian, T. Back, M. Chesus, G. S. Corrado, A. Darzi, *et al.*, "International evaluation of an ai system for breast cancer screening," *Nature*, vol. 577, no. 7788, pp. 89–94, 2020.

[5] S. Springer, D. L. Masica, M. Dal Molin, C. Douville, C. J. Thoburn, B. Afsari, L. Li, J. D. Cohen, E. Thompson, P. J. Allen, *et al.*, "A multimodality test to guide the management of patients with a pancreatic cyst," *Science Translational Medicine*, vol. 11, no. 501, 2019.

[6] N.-M. Cheng, J. Yao, J. Cai, X. Ye, S. Zhao, K. Zhao, W. Zhou, I. Nogues, Y. Huo, C.-T. Liao, *et al.*, "Deep learning for fully automated prediction of overall survival in patients with oropharyngeal cancer using fdg-pet imaging," *Clinical Cancer Research*, 2021.

[7] M. Y. Lu, T. Y. Chen, D. F. Williamson, M. Zhao, M. Shady, J. Lipkova, and F. Mahmood, "Ai-based pathology predicts origins for cancers of unknown primary," *Nature*, vol. 594, no. 7861, pp. 106–110, 2021.

[8] Y. Zhou, L. Xie, W. Shen, Y. Wang, E. K. Fishman, and A. L. Yuille, "A fixed-point model for pancreas segmentation in abdominal ct scans," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2017, pp. 693–701.

[9] Q. Yu, L. Xie, Y. Wang, Y. Zhou, E. K. Fishman, and A. L. Yuille, "Recurrent saliency transformation network: Incorporating multi-stage visual cues for small organ segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8280–8289.

[10] Z. Zhu, Y. Xia, W. Shen, E. Fishman, and A. Yuille, "A 3d coarse-to-fine framework for volumetric medical image segmentation," in *2018 International Conference on 3D Vision (3DV)*, IEEE, 2018, pp. 682–690.

[11] Y. Li, Z. Zhu, Y. Zhou, Y. Xia, W. Shen, E. K. Fishman, and A. L. Yuille, "Volumetric medical image segmentation: A 3d deep coarse-to-fine framework and its adversarial examples," in *Deep Learning and Convolutional Neural Networks for Medical Imaging and Clinical Informatics*, Springer, 2019, pp. 69–91.

[12] H. R. Roth, L. Lu, A. Farag, H.-C. Shin, J. Liu, E. B. Turkbey, and R. M. Summers, "Deeporgan: Multi-level deep convolutional networks for automated pancreas segmentation," in *International conference on medical image computing and computer-assisted intervention*, Springer, 2015, pp. 556–564.

[13] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," *International Conference on Learning Representations (ICLR)*, 2017.

[14] G. French, M. Mackiewicz, and M. Fisher, "Self-ensembling for visual domain adaptation," 2018.

[15] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the eleventh annual conference on Computational learning theory*, ACM, 1998, pp. 92–100.

[16] M. Hein, M. Andriushchenko, and J. Bitterwolf, "Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2019.

[17] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," *International Conference on Learning Representations, ICLR*, 2018.

[18] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2019.

[19] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*, PMLR, 2017, pp. 1273–1282.

[20] Z. Zhu, Y. Xia, L. Xie, E. K. Fishman, and A. L. Yuille, "Multi-scale coarse-to-fine segmentation for screening pancreatic ductal adenocarcinoma," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2019, pp. 3–12.

[21] M. P. Heinrich, M. Jenkinson, M. Brady, and J. A. Schnabel, "Mrf-based deformable registration and ventilation estimation of lung ct," *IEEE transactions on medical imaging*, vol. 32, no. 7, pp. 1239–1248, 2013.

[22] Y. Boykov and M. P. Jolly, "Interactive organ segmentation using graph cuts," in *MICCAI*, 2000.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[24] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[25] O. Cicek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: Learning dense volumetric segmentation from sparse annotation," in *MICCAI*, 2016.

[26] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *2016 Fourth International Conference on 3D Vision (3DV)*, IEEE, 2016, pp. 565–571.

[27] Z. Zhu, Y. Xia, W. Shen, E. K. Fishman, and A. L. Yuille, "A 3d coarse-to-fine framework for automatic pancreas segmentation," *arXiv:1712.00201*, 2017.

[28] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, and J. Liang, "Convolutional neural networks for medical image analysis: Full training or fine tuning?" *IEEE TMI*, vol. 35, no. 5, pp. 1299–1312, 2016.

[29] A. J. Asman and B. A. Landman, "Non-local statistical label fusion for multi-atlas segmentation," *Medical Image Analysis*, vol. 17, no. 2, pp. 194–208, 2013.

[30] H. Yang, J. Sun, H. Li, L. Wang, and Z. Xu, "Deep fusion net for multi-atlas segmentation: Application to cardiac mr images," in *MICCAI*, 2016.

[31] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 945–953.

[32] A. A. A. Setio, F. Ciompi, G. Litjens, P. Gerke, C. Jacobs, S. J. van Riel, M. M. W. Wille, M. Naqibullah, C. I. Sánchez, and B. van Ginneken, "Pulmonary nodule detection in ct images: False positive reduction using multi-view convolutional networks," *IEEE TMI*, vol. 35, no. 5, pp. 1160–1169, 2016.

[33] Z. Tu and X. Bai, "Auto-context and its application to high-level vision tasks and 3d brain image segmentation," *IEEE TPAMI*, vol. 32, no. 10, pp. 1744–1757, 2010.

[34] H. R. Roth, L. Lu, A. Farag, A. Sohn, and R. M. Summers, "Spatial aggregation of holistically-nested networks for automated pancreas segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2016, pp. 451–459.

[35] H. R. Roth, L. Lu, N. Lay, A. P. Harrison, A. Farag, A. Sohn, and R. M. Summers, "Spatial aggregation of holistically-nested convolutional neural networks for automated pancreas localization and segmentation," *arXiv:1702.00045*, 2017.

[36] J. Cai, L. Lu, Y. Xie, F. Xing, and L. Yang, "Pancreas segmentation in mri using graph-based decision fusion on convolutional neural networks," in *MICCAI*, 2017.

[37] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[38] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[39] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[40] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2018.

[41] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2881–2890.

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

[43] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.

[44] L. Yu, X. Yang, H. Chen, J. Qin, and P.-A. Heng, "Volumetric convnets with mixed residual connections for automated prostate segmentation from 3D MR images," in *AAAI*, 2017.

[45] C. Rosenberg, M. Hebert, and H. Schneiderman, "Semi-supervised self-training of object detection models.," *WACV/MOTION*, vol. 2, 2005.

[46] Y. Li, C. Guan, H. Li, and Z. Chin, "A self-training semi-supervised svm algorithm and its application in an eeg-based brain computer interface speller system," *Pattern Recognition Letters*, vol. 29, no. 9, pp. 1285–1294, 2008.

[47] W. Bai, O. Oktay, M. Sinclair, H. Suzuki, M. Rajchl, G. Tarroni, B. Glocker, A. King, P. M. Matthews, and D. Rueckert, "Semi-supervised learning for network-based cardiac mr image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2017, pp. 253–260.

[48] Z.-H. Zhou and M. Li, "Semi-supervised regression with co-training.," in *IJCAI*, vol. 5, 2005, pp. 908–913.

[49] Z. Dai, Z. Yang, F. Yang, W. W. Cohen, and R. R. Salakhutdinov, "Good semi-supervised learning that requires a bad gan," in *Advances in neural information processing systems*, 2017, pp. 6510–6520.

[50] A. Kumar, P. Sattigeri, and T. Fletcher, "Semi-supervised learning with gans: Manifold invariance with improved inference," in *Advances in Neural Information Processing Systems*, 2017, pp. 5534–5544.

[51] M. Chen, K. Q. Weinberger, and J. Blitzer, "Co-training for domain adaptation," in *Advances in neural information processing systems*, 2011, pp. 2456–2464.

[52] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell, "Cycada: Cycle-consistent adversarial domain adaptation," *arXiv preprint arXiv:1711.03213*, 2017.

[53] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker, "Learning to adapt structured output space for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7472–7481.

[54] R. Shu, H. H. Bui, H. Narui, and S. Ermon, "A dirt-t approach to unsupervised domain adaptation," 2018.

[55] 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)*, 2018, pp. 289–305.

[56] Y. Zou, Z. Yu, X. Liu, B. Kumar, and J. Wang, "Confidence regularized self-training," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 5982–5991.

[57] S. Qiao, W. Shen, Z. Zhang, B. Wang, and A. Yuille, "Deep co-training for semi-supervised image recognition," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 135–152.

[58] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, 2015.

[59] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, 2016, pp. 1050–1059.

[60] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" In *Advances in neural information processing systems*, 2017, pp. 5574–5584.

[61] Y. Xia, F. Liu, D. Yang, J. Cai, L. Yu, Z. Zhu, D. Xu, A. Yuille, and H. Roth, "3d semi-supervised learning with uncertainty-aware multi-view co-training," in *The IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 3646–3655.

[62] E. Gibson, F. Giganti, Y. Hu, E. Bonmati, S. Bandula, K. Gurusamy, B. Davidson, S. P. Pereira, M. J. Clarkson, and D. C. Barratt, "Automatic multi-organ segmentation on abdominal ct with dense v-networks," *IEEE transactions on medical imaging*, vol. 37, no. 8, pp. 1822–1834, 2018.

[63] M. Antonelli, A. Reinke, S. Bakas, K. Farahani, B. A. Landman, G. Litjens, B. Menze, O. Ronneberger, R. M. Summers, B. van Ginneken, *et al.*, "The medical segmentation decathlon," *arXiv preprint arXiv:2106.05735*, 2021.

[64] Z.-H. Zhou and M. Li, "Tri-training: Exploiting unlabeled data using three classifiers," *IEEE Transactions on knowledge and Data Engineering*, vol. 17, no. 11, pp. 1529–1541, 2005.

[65] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *Journal of machine learning research*, vol. 7, no. Nov, pp. 2399–2434, 2006.

[66] P. Bachman, O. Alsharif, and D. Precup, "Learning with pseudo-ensembles," in *Advances in Neural Information Processing Systems*, 2014, pp. 3365–3373.

[67] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko, "Semi-supervised learning with ladder networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 3546–3554.

[68] M. Sajjadi, M. Javanmardi, and T. Tasdizen, "Regularization with stochastic transformations and perturbations for deep semi-supervised learning," in *Advances in Neural Information Processing Systems*, 2016, pp. 1163–1171.

[69] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in neural information processing systems*, 2017, pp. 1195–1204.

[70] T. Miyato, S.-i. Maeda, S. Ishii, and M. Koyama, "Virtual adversarial training: A regularization method for supervised and semi-supervised learning," *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[71] D.-D. Chen, W. Wang, W. Gao, and Z.-H. Zhou, "Tri-net for semi-supervised deep learning," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, AAAI Press, 2018, pp. 2014–2020.

[72] 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," *arXiv preprint arXiv:1804.06353*, 2018.

[73] Y. Zhou, Y. Wang, P. Tang, W. Shen, E. K. Fishman, and A. L. Yuille, "Semi-supervised multi-organ segmentation via multi-planar co-training," *WACV*, 2019.

[74] X. Li, L. Yu, H. Chen, C.-W. Fu, and P.-A. Heng, "Semi-supervised skin lesion segmentation via transformation consistent self-ensembling model," *BMVC*, 2018.

[75] N. Dong, M. Kampffmeyer, X. Liang, Z. Wang, W. Dai, and E. Xing, "Unsupervised domain adaptation for automatic estimation of cardiothoracic ratio," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2018, pp. 544–552.

[76] J. Jiang, Y.-C. Hu, N. Tyagi, P. Zhang, A. Rimner, G. S. Mageras, J. O. Deasy, and H. Veeraraghavan, "Tumor-aware, adversarial domain adaptation from ct to mri for lung cancer segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2018, pp. 777–785.

[77] D. Nie, Y. Gao, L. Wang, and D. Shen, "Asdnet: Attention based semi-supervised deep networks for medical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2018, pp. 370–378.

[78] M. P. Shah, S. Merchant, and S. P. Awate, "Ms-net: Mixed-supervision fully-convolutional networks for full-resolution segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2018, pp. 379–387.

[79] P. Mlynarski, H. Delingette, A. Criminisi, and N. Ayache, "Deep learning with mixed supervision for brain tumor segmentation," *Journal of Medical Imaging*, vol. 6, no. 3, p. 034 002, 2019.

[80] A. Blake, R. Curwen, and A. Zisserman, "A framework for spatiotemporal control in the tracking of visual contours," *International Journal of Computer Vision*, vol. 11, no. 2, pp. 127–145, 1993.

[81] X. He, R. S. Zemel, and M. Á. Carreira-Perpiñán, "Multiscale conditional random fields for image labeling," in *Computer vision and pattern recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE computer society conference on*, IEEE, vol. 2, 2004, pp. II–II.

[82] Y. Gal, "Uncertainty in deep learning," *University of Cambridge*, 2016.

[83] Y. Xia, L. Xie, F. Liu, Z. Zhu, E. K. Fishman, and A. L. Yuille, "Bridging the gap between 2d and 3d organ segmentation with volumetric fusion net," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2018, pp. 445–453.

[84] X. Li, H. Chen, X. Qi, Q. Dou, C.-W. Fu, and P. A. Heng, "H-denseunet: Hybrid densely connected unet for liver and liver tumor segmentation from ct volumes," *IEEE Transactions on Medical Imaging*, 2017.

[85] 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 *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2018, pp. 851–858.

[86] H. R. Roth, L. Lu, N. Lay, A. P. Harrison, A. Farag, A. Sohn, and R. M. Summers, "Spatial aggregation of holistically-nested convolutional neural networks for automated pancreas localization and segmentation," *Medical image analysis*, vol. 45, pp. 94–107, 2018.

[87] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *European conference on computer vision*, Springer, 2010, pp. 213–226.

[88] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2012, pp. 2066–2073.

[89] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2010.

[90] B. Sun and K. Saenko, "Subspace distribution alignment for unsupervised domain adaptation.," in *BMVC*, vol. 4, 2015, pp. 24–1.

[91] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[92] J. Hoffman, D. Wang, F. Yu, and T. Darrell, "Fcns in the wild: Pixel-level adversarial and constraint-based adaptation," *arXiv preprint arXiv:1612.02649*, 2016.

[93] 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 *International conference on information processing in medical imaging*, Springer, 2017, pp. 597–609.

[94] 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," *arXiv preprint arXiv:1804.10916*, 2018.

[95] C. S. Perone, P. Ballester, R. C. Barros, and J. Cohen-Adad, "Unsupervised domain adaptation for medical imaging segmentation with self-ensembling," *NeuroImage*, vol. 194, pp. 1–11, 2019.

[96] B. Landman, Z. Xu, J. Eugenio Igelsias, M. Styner, T. Langerak, and A. Klein, *Miccai multi-atlas labeling beyond the cranial vault–workshop and challenge*, 2015.

[97] P. A. Yushkevich, J. Piven, H. Cody Hazlett, R. Gimpel Smith, S. Ho, J. C. Gee, and G. Gerig, "User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability," *Neuroimage*, vol. 31, no. 3, pp. 1116–1128, 2006.

[98] F. Isensee, J. Petersen, A. Klein, D. Zimmerer, P. F. Jaeger, S. Kohl, J. Wasserthal, G. Koehler, T. Norajitra, S. Wirkert, *et al.*, "Nnu-net: Self-adapting framework for u-net-based medical image segmentation," *arXiv preprint arXiv:1809.10486*, 2018.

[99] J. Janai, F. Güney, A. Behl, and A. Geiger, "Computer vision for autonomous vehicles: Problems, datasets and state-of-the-art," *arXiv preprint arXiv:1704.05519*, 2017.

[100] R. Challen, J. Denny, M. Pitt, L. Gompels, T. Edwards, and K. Tsaneva-Atanasova, "Artificial intelligence, bias and clinical safety," *BMJ Quality and Safety*, 2019.

[101] O. Linda, T. Vollmer, and M. Manic, "Neural network based intrusion detection system for critical infrastructures," in *International Joint Conference on Neural Networks, IJCNN*, 2009.

[102] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete problems in ai safety," *arXiv preprint arXiv:1606.06565*, 2016.

[103] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," *International Conference on Learning Representations, ICLR*, 2017.

[104] H. Jiang, B. Kim, M. Guan, and M. Gupta, "To trust or not to trust a classifier," in *Advances in Neural Information Processing Systems*, 2018.

[105] C. Corbiere, N. Thome, A. Bar-Hen, M. Cord, and P. Pérez, "Addressing failure prediction by learning model confidence," in *Advances in Neural Information Processing Systems*, 2019.

[106] K. Lee, H. Lee, K. Lee, and J. Shin, "Training confidence-calibrated classifiers for detecting out-of-distribution samples," *International Conference on Learning Representations, ICLR*, 2018.

[107] T. DeVries and G. W. Taylor, "Learning confidence for out-of-distribution detection in neural networks," *arXiv preprint arXiv:1802.04865*, 2018.

[108] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," in *Advances in Neural Information Processing Systems*, 2018.

[109] D. Hendrycks, M. Mazeika, and T. Dietterich, "Deep anomaly detection with outlier exposure," *International Conference on Learning Representations, ICLR*, 2019.

[110] D. Hendrycks, S. Basart, M. Mazeika, M. Mostajabi, J. Steinhardt, and D. Song, "A benchmark for anomaly segmentation," *arXiv preprint arXiv:1911.11132*, 2019.

[111] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2017.

[112] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2018.

[113] Y. Geifman and R. El-Yaniv, "Selective classification for deep neural networks," in *Advances in Neural Information Processing Systems*, 2017.

[114] Y. Kwon, J.-H. Won, B. J. Kim, and M. C. Paik, "Uncertainty quantification using bayesian neural networks in classification: Application to biomedical image segmentation," *Computational Statistics and Data Analysis*, 2020.

[115] A. Jungo, R. Meier, E. Ermis, E. Herrmann, and M. Reyes, "Uncertainty-driven sanity check: Application to postoperative brain tumor cavity segmentation," *arXiv preprint arXiv:1806.03106*, 2018.

[116] R. Robinson, O. Oktay, W. Bai, V. V. Valindria, M. M. Sanghvi, N. Aung, J. M. Paiva, F. Zemrak, K. Fung, E. Lukaschuk, *et al.*, "Real-time prediction of segmentation quality," in *International Conference on Medical Image Computing and Computer-Assisted Intervention, MICCAI*, 2018.

[117] T. Kohlberger, V. Singh, C. Alvino, C. Bahlmann, and L. Grady, "Evaluating segmentation error without ground truth," in *International Conference on Medical Image Computing and Computer-Assisted Intervention, MICCAI*, 2012.

[118] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang, "Acquisition of localization confidence for accurate object detection," in *Proceedings of the European Conference on Computer Vision, ECCV*, 2018.

[119] Z. Huang, L. Huang, Y. Gong, C. Huang, and X. Wang, "Mask scoring r-cnn," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2019.

[120] S. Chabrier, B. Emile, C. Rosenberger, and H. Laurent, "Unsupervised performance evaluation of image segmentation," *EURASIP Journal on Applied Signal Processing*, 2006.

[121] H. Gao, Y. Tang, L. Jing, H. Li, and H. Ding, "A novel unsupervised segmentation quality evaluation method for remote sensing images," *Sensors*, 2017.

[122] F. Liu, Y. Xia, D. Yang, A. L. Yuille, and D. Xu, "An alarm system for segmentation algorithm based on shape model," in *Proceedings of the IEEE International Conference on Computer Vision, ICCV*, 2019.

[123] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *International Conference on Learning Representations, ICLR*, 2014.

[124] I. Krešo, M. Oršić, P. Bevandić, and S. Šegvić, "Robust semantic segmentation with ladder-densenet models," *arXiv preprint arXiv:1806.03465*, 2018.

[125] P. Bevandić, I. Krešo, M. Oršić, and S. Šegvić, "Discriminative out-of-distribution detection for semantic segmentation," *arXiv preprint arXiv:1808.07703*, 2018.

[126] O. Zendel, K. Honauer, M. Murschitz, D. Steininger, and G. Fernandez Dominguez, "Wilddash-creating hazard-aware benchmarks," in *Proceedings of the European Conference on Computer Vision, ECCV*, 2018.

[127] C. Baur, B. Wiestler, S. Albarqouni, and N. Navab, "Deep autoencoding models for unsupervised anomaly segmentation in brain mr images," in *MICCAI Brainlesion Workshop*, 2018.

[128] M. Haselmann, D. P. Gruber, and P. Tabatabai, "Anomaly detection using deep learning based image completion," in *International Conference on Machine Learning and Applications, ICMLA*, IEEE, 2018.

[129] K. Lis, K. Nakka, P. Fua, and M. Salzmann, "Detecting the unexpected via image resynthesis," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 2152–2161.

[130] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014.

[131] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[132] X. Liu, G. Yin, J. Shao, X. Wang, *et al.*, "Learning to predict layout-to-image conditional convolutions for semantic image synthesis," in *Advances in Neural Information Processing Systems*, 2019.

[133] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2016.

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

[135] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[136] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[137] N. Rieke, J. Hancox, W. Li, F. Milletari, H. Roth, S. Albarqouni, S. Bakas, M. N. Galtier, B. Landman, K. Maier-Hein, *et al.*, "The future of digital health with federated learning," *npj Digit. Med.*, vol. 3, p. 119, 2020.

[138] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.

[139] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, "Federated learning with matched averaging," *ICLR*, 2020.

[140] E. Jeong, S. Oh, H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data," *arXiv preprint arXiv:1811.11479*, 2018.

[141] D. Li and J. Wang, "Fedmd: Heterogenous federated learning via model distillation," *arXiv preprint arXiv:1910.03581*, 2019.

[142] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018.

[143] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek, "Robust and communication-efficient federated learning from non-iid data," *IEEE transactions on neural networks and learning systems*, 2019.

[144] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," *arXiv preprint arXiv:1907.02189*, 2019.

[145] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," *ICML*, 2020.

[146] X. Chen, T. Chen, H. Sun, Z. S. Wu, and M. Hong, "Distributed training with heterogeneous data: Bridging median- and mean-based algorithms," *NeurIPS*, 2020.

[147] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," *NeurIPS*, 2020.

[148] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic federated learning," *ICML*, 2019.

[149] H.-Y. Chen and W.-L. Chao, "Fedbe: Making bayesian model ensemble applicable to federated learning," *arXiv preprint arXiv:2009.01974*, 2020.

[150] F. Hanzely, S. Hanzely, S. Horváth, and P. Richtárik, "Lower bounds and optimal algorithms for personalized federated learning," *NeurIPS*, 2020.

[151] C. T. Dinh, N. H. Tran, and T. D. Nguyen, "Personalized federated learning with moreau envelopes," *NeurIPS*, 2020.

[152] Y. Deng, M. M. Kamani, and M. Mahdavi, "Adaptive personalized federated learning," *arXiv preprint arXiv:2003.13461*, 2020.

[153] W. Li, F. Milletari, D. Xu, N. Rieke, J. Hancox, W. Zhu, M. Baust, Y. Cheng, S. Ourselin, M. J. Cardoso, *et al.*, "Privacy-preserving federated brain tumour segmentation," in *International Workshop on Machine Learning in Medical Imaging*, Springer, 2019, pp. 133–141.

[154] M. J. Sheller, B. Edwards, G. A. Reina, J. Martin, S. Pati, A. Kotrotsou, M. Milchenko, W. Xu, D. Marcus, R. R. Colen, and S. Bakas, "Federated learning in medicine: Facilitating multi-institutional collaborations without sharing patient data," *Scientific reports*, vol. 10, no. 1, pp. 1–12, 2020.

[155] H. R. Roth, K. Chang, P. Singh, N. Neumark, W. Li, V. Gupta, S. Gupta, L. Qu, A. Ihsani, B. C. Bizzo, *et al.*, "Federated learning for breast density classification: A real-world implementation," in *Domain Adaptation and Representation Transfer, and Distributed and Collaborative Learning*, Springer, 2020, pp. 181–191.

[156] X. Li, Y. Gu, N. Dvornek, L. Staib, P. Ventola, and J. S. Duncan, "Multi-site fmri analysis using privacy-preserving federated learning and domain adaptation: Abide results," *arXiv preprint arXiv:2001.05647*, 2020.

[157] Q. Chang, H. Qu, Y. Zhang, M. Sabuncu, C. Chen, T. Zhang, and D. N. Metaxas, "Synthetic learning: Learn from distributed asynchronized discriminator gan without sharing medical image data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 856–13 866.

[158] Q. Liu, Q. Dou, L. Yu, and P. A. Heng, "Ms-net: Multi-site network for improving prostate segmentation with heterogeneous mri data," *IEEE Transactions on Medical Imaging*, 2020.

[159] Q. Dou, D. C. de Castro, K. Kamnitsas, and B. Glocker, "Domain generalization via model-agnostic learning of semantic features," in *Advances in Neural Information Processing Systems*, 2019, pp. 6450–6461.

[160] Y. Xia, D. Yang, Z. Yu, F. Liu, J. Cai, L. Yu, Z. Zhu, D. Xu, A. Yuille, and H. Roth, "Uncertainty-aware multi-view co-training for semi-supervised medical image segmentation and domain adaptation," *Medical Image Analysis*, vol. 65, p. 101 766, 2020.

[161] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas, "Learning to learn by gradient descent by gradient descent," in *Advances in neural information processing systems*, 2016, pp. 3981–3989.

[162] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation policies from data," *arXiv preprint arXiv:1805.09501*, 2018.

[163] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 702–703.

[164] B. Baker, O. Gupta, N. Naik, and R. Raskar, "Designing neural network architectures using reinforcement learning," *arXiv preprint arXiv:1611.02167*, 2016.

[165] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *ICLR*, 2017.

[166] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in *Proceedings of the aaai conference on artificial intelligence*, vol. 33, 2019, pp. 4780–4789.

[167] L. Xie and A. Yuille, "Genetic cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1379–1388.

[168] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," *ICLR*, 2019.

[169] X. Chen, L. Xie, J. Wu, and Q. Tian, "Progressive differentiable architecture search: Bridging the depth gap between search and evaluation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1294–1303.

[170] X. Chen, R. Wang, M. Cheng, X. Tang, and C.-J. Hsieh, "Drnas: Dirichlet neural architecture search," *arXiv preprint arXiv:2006.10355*, 2020.

[171] M. Jankowiak and F. Obermeyer, "Pathwise derivatives beyond the reparameterization trick," in *International conference on machine learning*, PMLR, 2018, pp. 2235–2244.

[172] M. Figurnov, S. Mohamed, and A. Mnih, "Implicit reparameterization gradients," in *Advances in Neural Information Processing Systems*, 2018, pp. 441–452.

[173] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *arXiv preprint arXiv:1812.06127*, 2018.

[174] X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, "Fedbn: Federated learning on non-iid features via local batch normalization," *arXiv preprint arXiv:2102.07623*, 2021.

[175] S. A. Harmon, T. H. Sanford, S. Xu, E. B. Turkbey, H. Roth, Z. Xu, D. Yang, A. Myronenko, V. Anderson, A. Amalou, *et al.*, "Artificial intelligence for the detection of covid-19 pneumonia on chest ct using multinational datasets," *Nature communications*, vol. 11, no. 1, pp. 1–7, 2020.

[176] L. Li, L. Qin, Z. Xu, Y. Yin, X. Wang, B. Kong, J. Bai, Y. Lu, Z. Fang, Q. Song, *et al.*, "Artificial intelligence distinguishes covid-19 from community acquired pneumonia on chest ct," *Radiology*, 2020.

[177] F. Shi, J. Wang, J. Shi, Z. Wu, Q. Wang, Z. Tang, K. He, Y. Shi, and D. Shen, "Review of artificial intelligence techniques in imaging data acquisition, segmentation and diagnosis for covid-19," *IEEE reviews in biomedical engineering*, 2020.

[178] M. E. Lowe, D. K. Andersen, R. M. Caprioli, J. Choudhary, Z. Cruz-Monserrate, A. K. Dasyam, C. E. Forsmark, F. S. Gorelick, J. W. Gray, M. Haupt, *et al.*, "Precision medicine in pancreatic disease—knowledge gaps and research opportunities: Summary of a national institute of diabetes and digestive and kidney diseases workshop," *Pancreas*, vol. 48, no. 10, p. 1250, 2019.

[179] G. A. Kaissis, M. R. Makowski, D. Rückert, and R. F. Braren, "Secure, privacy-preserving and federated machine learning in medical imaging," *Nature Machine Intelligence*, pp. 1–7, 2020.

[180] A. L. Lucas and F. Kastrinos, "Screening for pancreatic cancer," *Jama*, vol. 322, no. 5, pp. 407–408, 2019.

[181] L. C. Chu, S. Park, S. Kawamoto, Y. Wang, Y. Zhou, W. Shen, Z. Zhu, Y. Xia, L. Xie, F. Liu, *et al.*, "Application of deep learning to pancreatic cancer detection: Lessons learned from our initial experience," *Journal of the American College of Radiology*, vol. 16, no. 9, pp. 1338–1342, 2019.

[182] L. C. Chu, S. Park, S. Kawamoto, D. F. Fouladi, S. Shayesteh, E. S. Zinreich, J. S. Graves, K. M. Horton, R. H. Hruban, A. L. Yuille, *et al.*, "Utility of ct radiomics features in differentiation of pancreatic ductal adenocarcinoma from normal pancreatic tissue," *American Journal of Roentgenology*, vol. 213, no. 2, pp. 349–357, 2019.

[183] Y. Zhou, Y. Li, Z. Zhang, Y. Wang, A. Wang, E. K. Fishman, A. L. Yuille, and S. Park, "Hyper-pairing network for multi-phase pancreatic ductal adenocarcinoma segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2019, pp. 155–163.

[184] J. Cai, L. Lu, Z. Zhang, F. Xing, L. Yang, and Q. Yin, "Pancreas segmentation in mri using graph-based decision fusion on convolutional neural networks," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2016, pp. 442–450.

[185] Y. Man, Y. Huang, J. Feng, X. Li, and F. Wu, "Deep q learning driven ct pancreas segmentation with geometry-aware u-net," *IEEE transactions on medical imaging*, vol. 38, no. 8, pp. 1971–1980, 2019.

[186] D. P. Ryan, T. S. Hong, and N. Bardeesy, "Pancreatic adenocarcinoma," *New England Journal of Medicine*, vol. 371, no. 11, pp. 1039–1049, 2014.

[187] T. Vercauteren, X. Pennec, A. Perchant, and N. Ayache, "Diffeomorphic demons: Efficient non-parametric image registration," *NeuroImage*, vol. 45, no. 1, S61–S72, 2009.

[188] A. Roche, G. Malandain, X. Pennec, and N. Ayache, "The correlation ratio as a new similarity measure for multimodal image registration," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 1998, pp. 1115–1124.

[189] T. Gaens, F. Maes, D. Vandermeulen, and P. Suetens, "Non-rigid multimodal image registration using mutual information," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 1998, pp. 1099–1106.

[190] M. Jaderberg, K. Simonyan, A. Zisserman, *et al.*, "Spatial transformer networks," in *Advances in neural information processing systems*, 2015, pp. 2017–2025.

[191] G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag, and A. V. Dalca, "Voxelmorph: A learning framework for deformable medical image registration," *IEEE transactions on medical imaging*, 2019.

[192] C. Qin, B. Shi, R. Liao, T. Mansi, D. Rueckert, and A. Kamen, "Unsupervised deformable registration for multi-modal images via disentangled representations," in *International Conference on Information Processing in Medical Imaging*, Springer, 2019, pp. 249–261.

[193] W. Zhu, A. Myronenko, Z. Xu, W. Li, H. Roth, Y. Huang, F. Milletari, and D. Xu, "Neurreg: Neural registration and its application to image segmentation," in *The IEEE Winter Conference on Applications of Computer Vision*, 2020, pp. 3617–3626.

[194] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest, *et al.*, "The multimodal brain tumor image segmentation benchmark (brats)," *IEEE transactions on medical imaging*, vol. 34, no. 10, pp. 1993–2024, 2014.

[195] J. Dolz, K. Gopinath, J. Yuan, H. Lombaert, C. Desrosiers, and I. B. Ayed, "Hyperdense-net: A hyper-densely connected cnn for multi-modal image segmentation," *IEEE transactions on medical imaging*, vol. 38, no. 5, pp. 1116–1126, 2018.

[196] R. L. Siegel, K. D. Miller, and A. Jemal, "Cancer statistics, 2020," *CA: A Cancer Journal for Clinicians*, vol. 70, no. 1, pp. 7–30, 2020. DOI: 10.3322/caac.21590.

[197] J. D. Mizrahi, R. Surana, J. W. Valle, and R. T. Shroff, "Pancreatic cancer," *The Lancet*, vol. 395, no. 10242, pp. 2008–2020, 2020.

[198] A. D. Singhi, E. J. Koay, S. T. Chari, and A. Maitra, "Early detection of pancreatic cancer: Opportunities and challenges," *Gastroenterology*, vol. 156, no. 7, pp. 2024–2040, 2019.

[199] M. Oudkerk, S. Liu, M. A. Heuvelmans, J. E. Walter, and J. K. Field, "Lung cancer ldct screening and mortality reduction—evidence, pitfalls and future perspectives," *Nature Reviews Clinical Oncology*, pp. 1–17, 2020.

[200] Y. Xia, Q. Yu, W. Shen, Y. Zhou, E. K. Fishman, and A. L. Yuille, "Detecting pancreatic ductal adenocarcinoma in multi-phase ct scans via alignment ensemble," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2020, pp. 285–295.

[201] L. Zhang, Y. Shi, J. Yao, Y. Bian, K. Cao, D. Jin, J. Xiao, and L. Lu, "Robust pancreatic ductal adenocarcinoma segmentation with multi-institutional multi-phase partially-annotated ct scans," in *MICCAI*, Springer, 2020, pp. 491–500.

[202] T. Zhao, K. Cao, J. Yao, I. Nogues, L. Lu, L. Huang, J. Xiao, Z. Yin, and L. Zhang, "3D graph anatomy geometry-integrated network for pancreatic mass segmentation, diagnosis, and quantitative patient management," *arXiv preprint arXiv:2012.04701*, 2020.

[203] J. Yao, Y. Shi, L. Lu, J. Xiao, and L. Zhang, "Deepprognosis: Preoperative prediction of pancreatic cancer survival and surgical margin via contrast-enhanced ct imaging," in *MICCAI*, Springer, 2020, pp. 272–282.

[204] 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*, Springer, 2020, pp. 213–229.

[205] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr, *et al.*, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," *arXiv preprint arXiv:2012.15840*, 2020.

[206] 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," *ICLR*, 2021.

[207] 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.

[208] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 687–10 698.

Vita

Yingda Xia is completing his Ph.D. degree in Computer Science at the Johns Hopkins University, supervised by Bloomberg Distinguished Professor Alan L. Yuille. He obtained his B.S. degree from School of Software, Tsinghua University in 2017. His research interest lies in computer vision and medical image analysis, especially in developing automated medical AI systems for real-world clinical problems. One of his major project during Ph.D. is about early pancreatic cancer detection, which is founded by the Lustgarten Foundation. He also interned in Nvidia, PAII, and MSRA.