

2019

Optimization Algorithms for Deep Learning Based Medical Image Segmentations

Aliasghar Mortazi
University of Central Florida



Part of the [Computer Sciences Commons](#)

Find similar works at: <https://stars.library.ucf.edu/etd>

University of Central Florida Libraries <http://library.ucf.edu>

STARS Citation

Mortazi, Aliasghar, "Optimization Algorithms for Deep Learning Based Medical Image Segmentations" (2019). *Electronic Theses and Dissertations*. 6715.

<https://stars.library.ucf.edu/etd/6715>

This Doctoral Dissertation (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of STARS. For more information, please contact lee.dotson@ucf.edu.



OPTIMIZATION ALGORITHMS FOR DEEP LEARNING BASED MEDICAL IMAGE
SEGMENTATIONS

by

ALIASGHAR MORTAZI
M.S. Sharif University of Technology, 2014

A dissertation submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy
in the Department of Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Fall Term
2019

Major Professor: Ulas Bagci

© 2019 Aliasghar Mortazi

ABSTRACT

Medical image segmentation is one of the fundamental processes to understand and assess the functionality of different organs and tissues as well as quantifying diseases and helping treatment planning. With ever increasing number of medical scans, the automated, accurate, and efficient medical image segmentation is as unmet need for improving healthcare. Recently, deep learning has emerged as one the most powerful methods for almost all image analysis tasks such as segmentation, detection, and classification and so in medical imaging. In this regard, this dissertation introduces new algorithms to perform medical image segmentation for different (a) imaging modalities, (b) number of objects, (c) dimensionality of images, and (d) under varying labeling conditions. First, we study dimensionality problem by introducing a new 2.5D segmentation engine that can be used in single and multi-object settings. We propose new fusion strategies and loss functions for deep neural networks to generate improved delineations. Later, we expand the proposed idea into 3D and 4D medical images and develop a "budget (computational) friendly" architecture search algorithm to make this process self-contained and fully automated without scarifying accuracy. Instead of manual architecture design, which is often based on plug-in and out and expert experience, the new algorithm provides an automated search of successful segmentation architecture within a short period of time. Finally, we study further optimization algorithms on label noise issue and improve overall segmentation problem by incorporating prior information about label noise and object shape information. We conclude the thesis work by studying different network and hyperparameter optimization settings that are fine-tuned for varying conditions for medical images. Applications are chosen from cardiac scans (images) and efficacy of the proposed algorithms are demonstrated on several data sets publicly available, and independently validated by blind evaluations.

EXTENDED ABSTRACT

Image segmentation plays a vital role in understanding and analysis images, specifically in medical imaging and radiology, image segmentation can help quantifying diseases, measuring the structures volumes, and analyzing the organ morphology. For instance ejection fraction (EF) is one of the vital metric to assess function of the heart from cardiac magnetic resonance imaging (MRI). In addition, atrial fibrillation (AF) is a cardiac arrhythmia caused by abnormal electrical discharges in the atrium, often beginning with structural changes in the left atrium (LA). The LA also has an important role in patients with ventricular dysfunction as a booster pump to augment ventricular volume. Annotation of cardiac and its substructures from volumetric medical images play an important role in cardiac assessment. Radiologists need to measure volume of the heart chambers in different point time (end-systole (ES) and end-diastole (ED)) in order to calculate EF. Also, physicians need to delineate the LA to get its morphology, essential in detecting and diagnosis AF. All these delineation and annotation tasks are tedious and prone to intra- and inter-observer errors.

Extensive research and clinical applications have shown that both computed tomography (CT) and MRI have vital roles in non-invasive assessment of cardiovascular diseases (CVDs). In this thesis, we focus on this vital clinical problems. In this regard, CT is used more frequently than MRI due to its fast acquisition and cheaper cost. On the other hand, MRI has an excellent soft tissue contrast and no ionizing radiation. To propose a generic method appreciate for both modalities, we propose a series of supervised deep-learning-based segmentation methods for cardiac functionality assessment. Deep learning has emerged as one of the state-of-the-art methods in different image processing tasks such as detection and segmentation. When it comes to design a deep-learning-based method for medical image analysis, some unique challenges and problems are raised due to different nature of medical images including image dimensionality, lack of enough data, lack of enough expert annotations, which are both expensive and time consuming.

The rest of this dissertation is organized as follows. In Chapter 1 and Chapter 2, the motivation of the dissertation, the summary of proposed methods and also the related works in the literature are given. One of the challenges in medical image analysis is dimensionality on these images. In Chapter 3, a multi-view 2D (2.5D) convolutional neural network (CNN) with a fusion method (called *CardiacNet*) is introduced to segment 3D LA and proximal pulmonary veins (PPVs). We address this unmet clinical need by exploring a new deep learning-based segmentation strategy for quantification of LA and PPVs with high accuracy and heightened efficiency. Our approach is based on a multi-view CNN with an adaptive fusion strategy and a new loss function that allows fast and more accurate convergence of the backpropagation based optimization. As the application, the anatomical and biophysical modeling of LA and PPVs are chosen because of their importance for clinical management of several cardiac diseases. MRI allows qualitative assessment of LA and PPVs through visualization. However, there is a strong need for an advanced image segmentation method to be applied to cardiac MRI for quantitative analysis of LA and PPVs. After training our network from scratch by using more than 60K 2D MRI images (slices), we have evaluated our segmentation strategy to the STACOM 2013 cardiac segmentation challenge benchmark. Qualitative and quantitative evaluations, obtained from the segmentation challenge, indicate that the proposed method achieved the state-of-the-art sensitivity (90%), specificity (99%), precision (94%), and efficiency levels (10 seconds in GPU, and 7.5 minutes in CPU).

Another challenge in medical images is having a general method to do segmentation from different imaging modalities for multiple objects. In Chapter 4, we introduce a multi-planar 2D (2.5D) CNN method with a fusion method for segmenting seven 3D substructures from both MR and CT images for the first time. For CT and MRI, we have separately designed three CNNs (the same architectural configuration) for three planes, and have trained the networks from scratch for voxel-wise labeling for the following cardiac structures: myocardium of left ventricle (Myo), left atrium (LA), left ventricle (LV), right atrium (RA), right ventricle (RV), ascending aorta (Ao), and main pulmonary

artery (PA). We have evaluated the proposed method with 4-fold-cross-validation on the multi-modality whole heart segmentation challenge (MM-WHS 2017) dataset. A precision and dice index of 0.93 and 0.90, and 0.87 and 0.85 were achieved for CT and MR images, respectively. Cardiac CT volume was segmented in about 50 seconds, with cardiac MRI segmentation requiring around 17 seconds with multi-GPU/CUDA implementation.

In the Chapter 5, we introduce a method to automatically design the CNN architecture to segment cardiac substructure from cine-MRI (4D images). Deep neural network architectures have traditionally been designed and explored with human expertise in a long-lasting trial-and-error process. This process requires huge amount of time, expertise, and resources. To address this tedious problem, we propose a novel algorithm to optimally find hyperparameters of a deep network architecture automatically. Our proposed method is based on a policy gradient reinforcement learning for which the reward function is assigned a segmentation evaluation utility (i.e., dice index). We show the efficacy of the proposed method with its low computational cost in comparison with the state-of-the-art medical image segmentation networks. We also present a new architecture design, *a densely connected encoder-decoder CNN*, as a strong baseline architecture to apply the proposed hyperparameter search algorithm. We apply the proposed algorithm to each layer of the baseline architectures. As an application, we train the proposed system on cine cardiac MR images from Automated Cardiac Diagnosis Challenge (ACDC) MICCAI 2017. Starting from a baseline segmentation architecture, the resulting network architecture have obtained the state-of-the-art results in accuracy without performing any trial-and-error based architecture design approaches or close supervision of the hyperparameters changes.

In the Chapter 6, a new method based on learning a geodesic map is introduced to make deep-learning segmentor learn from a noisy inexperienced annotation. The performance of the state-of-the-art image segmentation methods heavily relies on the high-quality annotations, which are not easily affordable, particularly for medical imaging data. To alleviate this limitation, we propose a weakly

supervised image segmentation method based on a *deep geodesic prior*. We hypothesize that integration of this prior information can reduce the adverse effects of weak labels in segmentation accuracy. Our proposed algorithm is based on a prior information, extracted from an auto-encoder, trained to map objects' geodesic maps to their corresponding binary maps. The obtained information is then used as an extra term in the loss function of the segmentor. In order to show efficacy of the proposed strategy, we have experimented segmentation of cardiac substructures with clean and two levels of noisy labels (L1, L2). Our experiments showed that the proposed algorithm boosted the performance of baseline deep learning-based segmentation for both clean and noisy labels by 4.4% (without noise), 4.6%(L1 noise), and 6.3%(L2 noise) in dice score, respectively. We also showed that the proposed method is more robust in the presence of high-level noise due to the existence of shape priors.

Finally, in Chapter 7, the different optimization method in deep-learning for medical image segmentation are investigated. Based on a cyclic and momentum optimization concept, a new optimization method is introduced in order to decrease computational cost while increasing or maintaining the accuracy of deep segmentation networks.

ACKNOWLEDGMENTS

First, I would like to express my greatest appreciation to my advisor, Dr. Ulas Bagci, for his exceptional guidance throughout my Ph.D. journey. It has been truly remarkable to have the opportunity work with him.

I am particularly grateful for the assistance from my committee-members, Dr. Mubarak Shah, Dr. Abhijit Mahalanobis, and Dr. Marianna Pensky, for generously giving their time and thoughtful comments to the development of my work.

I would also like to thank Dr. Sila Kurugol and the Computational Radiology Lab in Harvard Medical School and Boston Children Hospital for supporting me in collaboration during the time I spent at Harvard Medical School as a visiting scholar.

I would like to extend my appreciation to Dr. Jayaram Udupa and Dr. Drew Torigian from the Medical Image Processing Group at University of Pennsylvania for their exceptional support and collaboration.

I would like to thank our collaborators Dr. Rashed Karim and Dr. Kawal Rhode from King's College London for their assistance with the collection and preparation of the data set.

Furthermore, I would like to offer special thanks to my colleagues and friends Dr. Naji Khosravan, Elizabeth Mamourian, Dr. Navid Kardan, Rodney Lalonde, Amir Mazaheri, and Dr. Sarfaraz Hussein for their friendship and supports.

Last, but most important, I want to appreciate and acknowledge my parents and sisters for their infinite support and encourage me in this joyful experience.

TABLE OF CONTENTS

LIST OF FIGURES	xiii
LIST OF TABLES	xvii
CHAPTER 1: INTRODUCTION	1
1.1 2D and 2.5D Segmentation of Medical Images	2
1.2 Multi-object Segmentation of Multi-Modal Medical Images	4
1.3 Automatically Designing CNN Architectures Under Limited Computational Sources	5
1.4 Incorporating Prior Knowledge and Label Noise in Medical Image Segmentation	5
1.5 Optimization of Segmentation Networks	6
CHAPTER 2: RELATED WORKS	8
CHAPTER 3: 2D AND 2.5D SEGMENTATION OF MEDICAL IMAGES	14
3.1 Overview	14
3.2 Multi-View CNN	15
3.3 z-Loss Function	16
3.4 Training <i>CardiacNet</i>	16

3.5	Multi-View Information Fusion	17
3.6	Experimental Results	18
3.7	Discussions and Concluding Remarks	22
CHAPTER 4: MULTI-OBJECT SEGMENTATION OF MULTI-MODAL MEDICAL IM- AGES		24
4.1	Proposed CNN Architecture	24
4.2	Multi-Object Adaptive Fusion	27
4.3	Experimental Results	28
4.4	Discussions and Concluding Remarks	30
CHAPTER 5: AUTOMATICALLY DESIGNING CNN ARCHITECTURES UNDER LIM- ITED COMPUTATIONAL SOURCES		33
5.1	Overview	33
5.2	Policy Gradient	33
5.3	Proposed Backbone Architecture for Image Segmentation	36
5.4	Learnable Hyperparameters	37
5.5	Experiments and Results	38
5.6	Discussions and Conclusion Remarks	41

CHAPTER 6: INCORPORATING PRIOR KNOWLEDGE AND LABEL NOISE IN MEDICAL IMAGE SEGMENTATION	43
6.1 Overview	43
6.2 Proposed Weakly-Supervised Approach	43
6.3 Network Architecture for Segmentation	46
6.4 Learning a Geodesic Prior	47
6.5 Experiments and Results	48
6.6 Discussions and Conclusion Remarks	51
CHAPTER 7: CHOOSING THE RIGHT OPTIMIZERS FOR MEDICAL IMAGE SEGMENTATION	52
7.1 Overview	52
7.2 Introduction	53
7.3 Background	55
7.3.1 Current Optimization Methods in Deep Learning	55
7.3.1.1 Optimizers with fixed LR/MR	56
7.3.1.2 Optimizers with adaptive LR and MR	57
7.4 Methods	58
7.4.1 CNN Architectures	59

7.4.2	Dense Block	59
7.4.3	Cyclic Learning/Momentum Rate Optimizer(CLMR)	62
7.5	Experiments and results	65
7.5.1	Data	65
7.5.2	Implementation details	66
7.5.3	Results	66
7.6	Discussions and Conclusion Remarks	69
CHAPTER 8: CONCLUSION AND FUTURE WORKS		76
8.1	Conclusion	76
8.2	Future Work	77
LIST OF REFERENCES		79

LIST OF FIGURES

2.1	Examples of cardiac-MRI and cardiac-CT images with the corresponding contours for different substructures.	8
2.2	The parameters which need to be measured for quantifying heart. Specifically, left ventricle plays vital role in cardiac assessment. (Source of the figure from [1])	9
3.1	High-level overview of the proposed multi-view CNN architecture.	15
3.2	Details of the CNN architecture. Note that image size is not necessarily fixed for each view's CNN.	17
3.3	Connected components obtained from each view were computed and the residual volume (T-NT) was used to determine the strength for fusion with the other views.	17
3.4	First row shows sample MRI slices from S, C, and A views (red contour is ground-truth and green one is output of proposed method). Second-to-fifth rows: 3D surface visualization for the ground-truth and the output generated by the proposed method w.r.t simple fusion (F), adaptive fusion (AF), and the new loss function (SP).	21
3.5	Box plots for different metrics (sensitivity, precision, and Dice index) for state-of-the-art methods (LTSI_VRG, UCL_1C, UCL_4C, OBS_2) and proposed methods (F_CNN, AF_CNN, AF_CNN_SP) on the LA segmentation benchmark	22

4.1	Overview of Mo-MP-CNN with adaptive fusion.	25
4.2	Details of the CNN architecture. Note that image size is not necessarily fixed for each plane's CNN.	26
4.3	Connected components obtained from each plane were computed and the residual volume (T-NT) was used to determine the strength for fusion with the other planes.	28
4.4	First two rows show axial, sagittal, and coronal planes of the CT (first three columns) and MR images (last three columns), annotated cardiac structures, and their corresponding surface renditions (last two rows). Red arrows indicate some of mis-segmentations.	29
4.5	Box plots for sensitivity, precision, and Dice index for each structure and WHS. Top figure is for CT dataset and bottom figure is for MR dataset . . .	32
5.1	Overview of proposed method. First, the policy is initialized randomly and then P perturbation are generated. The network is trained with each perturbation and reward from each perturbation is calculated. The policy will be updated accordingly and the process will be repeated until no significant changes in the reward. Reward is simply set as dice coefficient for evaluating how good the segmentation is.	34
5.2	Details of the baseline architecture. We combine encoder-decoder based segmentation network with densely connected architecture as a novel segmentation network, which has less parameters to tune and more accurate. Concat: concatenation, BN: batch normalization, and conv: convolution.	37

5.3	An example for cine-MR (4D) cardiac image from ACDC data set ([2]).	39
5.4	Details of the optimally learned architecture by the proposed method. Note that connections among layers inside of each block are same as dense layers.	41
6.1	The proposed framework has two main components: (1) segmentation network: assigns a class label to each pixel, and, (2) <i>AE</i> which learns a prior from geodesic maps. Green arrows show the flow of training of the segmentor and red arrows show the flow of training of GAE. Note that in test phase only segmentor is used. Also, the in our method the noisy annotations are used for whole training process.	44
6.2	Dense Block (DB) content. <i>C</i> : concatenation operation, <i>BN</i> : Batch normalization, <i>LReLU</i> : Leaky ReLU activation, and <i>3D conv</i> : 3D convolution with a $3 \times 3 \times 3$ filter.	46
6.3	Generating noisy labels. Binary images went through a two-step process of adding pepper noise and filling inside object which makes the boundaries inaccurate.	49
7.1	CNN Architecture is used for pixel-wise segmentation. The architecture with CNN blocks and without red skip connections is Encoder-Decoder architecture. The architecture with red skip connection and CNN blocks (Figure 7.2a) is U-Net and with Dense block (Figure 7.2b) is Tiramisu	61
7.2	(a) CNN block used in Enc-Dec and UNet architectures, (b) Dense block used in Tiramisu architectures	62

7.3	CLR and MLR functions. (a) Learning rate triangle function for different C_{lr} values with $min_{lr} = 0.0005$ and $max_{lr} = 0.05$. (b) Momentum rate triangle function for different C_{mr} values with $min_{mr} = 0.85$ and $max_{mr} = 0.95$	63
7.4	Validation loss and dice index for four different architectures with ADAM and CLMR optimizers	70
7.5	Validation loss and dice index for DenseNet_2 architecture with different values of C_{lr} and C_{mr}	71
7.6	Validation loss and dice index for UNet architecture with different values of C_{lr} and C_{mr}	72
7.7	Box plots for DI in test data set for RV, Myo., LV and also average of them (Ave.).	73
7.8	Qualitative results for ground-truth and different methods for same subject in end-diastole from Apex to Base for four slices (from right to left). Green, yellow, and brown contours are showing RV, myo, and LV, respectively.	74
7.9	Qualitative results for ground-truth and different methods for same subject in end-systole from Apex to Base for four slices (from right to left). Green, yellow, and brown contours are showing RV, myo, and LV, respectively.	75

LIST OF TABLES

3.1	Data augmentation parameters and number of training images	19
3.2	The evaluation metrics for state-of-the-art and proposed methods. **: the running time on CPU *: the running time on NVIDIA TitanX GPU	20
4.1	Data augmentation parameters.	25
4.2	Quantitative evaluations of the proposed segmentation method for both CT and MRI are summarized.	30
5.1	Data augmentation	40
5.2	DI and HD for all methods and substructures.	42
6.1	DI and HD are reported for both expert and inexpert labels.	50
7.1	Number of layers in each block of different architectures and number of parameters	60
7.2	DI in the test data set with online evaluation.	68

CHAPTER 1: INTRODUCTION

There have been dramatic increase in the number of medical images which are taken for treatment planning, diagnosis, and other clinical purposes [3]. In the current clinical standards, these measurements (annotations) from medical scans are done by expert physicians. These procedures are tedious, and prone to intra- and inter-observer errors, which can readily affect the diagnosis and treatment procedure [4]. The need for using expert-level automated methods (i.e. segmentation) and software with high efficacy, which can help and ease these tasks and resolve above mentioned problems, is essential.

Deep learning algorithms, specifically convolutional neural networks (CNN) based methods, have been proven themselves as powerful methods in image analysis tasks such as detection, classification, and segmentation [5, 6, 7, 8]. Like in any other fields, deep learning has been used in medical image analysis vastly in last few years and it has been a dominant technical method in medical image processing. However, using deep learning for medical image analysis has its own challenges and limitations too:

- **First**, the dimensionality of medical images (3D or 4D) demands huge number of parameters in CNN, causing the need of huge amount of computational resources that are not affordable with current hardware technology. Standard CNN methods are mostly developed for 2D images.
- **Second**, handling different modalities in medical images is a must in many important tasks, increasing the difficulty of using CNN.
- **Third**, since variation and unexpected information is quite common in medical images, there is a need to incorporate expert information into medical decisions to generate optimal

algorithm.

- **Fourth**, designing an algorithm to search the optimum CNN architecture is another challenge with mentioned limitation in medical images.
- **Fifth**, the optimizers as an engine of deep learning methods need to be investigated and tuned for medical image segmentation purposes.

Clinical problem/motivation: In this thesis, we focus on Cardiovascular diseases (CVDs). CVDs are the first cause of death globally according to the World Health Organization. About 17.7 million people died from CVDs in 2015. For evaluating healthiness of heart, usually radiologists should delineate the different substructures in order to measure the volume of them or to look at their morphology. Some of these substructures are left ventricle (LV), myocardium of left ventricle (myo), right ventricle (RV), left atrium (LA) and its pulmonary veins (PVs) and etc. Also, according to the World Health Organization [9], cardiovascular diseases (CVDs) are the first cause of death global which make it essential to have an automated method to help radiologists for cardiac assessment. In this dissertation, cardiac has been chosen as a target organ for analysis; however, the proposed methods can be applied to any other organs and image modalities. Following are the summary of proposed methods:

1.1 2D and 2.5D Segmentation of Medical Images

Background problem definition: Atrial fibrillation (AF) is a cardiac arrhythmia caused by abnormal electrical discharges in the atrium, often beginning with hemodynamic and/or structural changes in the left atrium (LA) [10]. AF is clinically associated with LA strain, and MRI is shown to be a promising imaging method for assessing the disease state and predicting adverse clinical outcomes. The LA also has an important role in patients with ventricular dysfunction as a booster

pump to augment ventricular volume [11]. CT imaging of the heart is frequently performed when managing AF and prior to pulmonary vein ablation (isolation) therapy due to its rapid processing time. In recent years, there is an increasing interest in shifting towards cardiac MRI due to its excellent soft tissue contrast properties and lack of radiation exposure. For pulmonary vein ablation therapy planning in AF, precise segmentation of the LA and PPVs is essential. However, this task is non-trivial because of multiple anatomical variations of LA and PPV.

To alleviate the problem defined above and accomplish the segmentation of LA and PPVs from 3D cardiac MRI with high *accuracy* and *efficiency*, we propose a new deep CNN. Our proposed method is fully automated, and largely different from previous methods of LA and PPVs segmentation. The summary of these differences and key novelties of the proposed network architecture, named as *CardiacNet*, are listed as follows:

- The proposed method is inspired by SegNet architecture, one of the first segmentor in the literature , but it is unable to handle unique difficulties of medial scans. For instance, training CNN from scratch for 3D cardiac MRI is not feasible with insufficient 3D training data (with ground truth) and limited computer memory. Instead, we parsed 3D data into 2D components (axial (A), sagittal (S), and coronal (C)), and utilized a separate deep learning architecture for each component. The proposed *CardiacNet* was trained using more than 60K 2D slices of cardiac MR images without relying on a pre-training network of non-medical data, which was the common method of choice at the time this work was proposed.
- We have combined three CNN networks through an *adaptive fusion* mechanism where complementary information of each CNN (view) was utilized to improve segmentation results. The proposed adaptive fusion mechanism is based on a new strategy; called *robust region*, definition which measures (roughly) the reliability of segmentation results without the need for ground truth.

- We devised a new loss function in the proposed network, based on a *modified z-loss*, to providing fast convergence of network parameters than others. This does not only improved segmentation results due to reliable allocation of network parameters, but it also provides a significant acceleration of the segmentation process. The overall segmentation process for a given 3D cardiac MRI takes at most 10 seconds in GPU, and 7.5 minutes in CPU on a normal workstation, fastest ever segmentation method at the time of relevant publication, utmost desirable property for clinical adoption of segmentation software.

1.2 Multi-object Segmentation of Multi-Modal Medical Images

Extensive research and clinical applications have shown that both CT and MRI have vital roles in non-invasive assessment of cardiovascular diseases. CT is used more frequently than MRI due to its fast acquisition and cheaper cost. On the other hand, MRI has excellent soft tissue contrast and no ionizing radiation. However, most commercially available image analysis methods have been either tuned for CT or MRI only. Furthermore, many studies are focused on only one substructure of the heart (for instance, the left ventricle or left atrium). Surprisingly, there is very little published research on segmenting all substructures of the heart despite the fact that clinically established markers rely on shape, volumetric, and tissue characterization of all the cardiac substructures. Our study is concerned with this open problem from a machine learning perspective. We have investigated architectural designs of deep learning networks to solve multi-label and multi-modality image segmentation challenges within the scope of limited GPU processing power and limited imaging data.

Contribution: Building up on our proposed *CardiacNet* architecture, we have extended this segmentation engine in several different ways as follows. (1) A deeper CNN has been utilized as compared to *CardiacNet* [12]. (2) We have used both CT and MRI to test and evaluate the pro-

posed system while *CardiacNet* uses only MRI [12]. (3) We have extended the binary segmentation problem into a multi-label segmentation problem. (4) We have devised a rank based adaptive fusion method to assess effective information from different imaging planes for all delineated objects and select the best fusion strategies for highly accurate and efficient delineation results.

1.3 Automatically Designing CNN Architectures Under Limited Computational Sources

Designing highly accurate and efficient deep segmentation networks is not trivial. It is because manual exploration of high-performance deep networks requires extensive research by close supervision of human expert (from several months to several years) and huge amount of time and resources due to training time of networks [8, 12, 13, 14] . Considering that the choice of architecture and hyperparameters affects the segmentation results, it is extremely important to select the optimal hyperparameters. In this part of the thesis, we address this pressing problem by developing a proof of concept architecture search algorithm, specifically for medical image segmentation problems and under budgetary conditions. Our proposed method is generic and can be applied to any medical image segmentation problem. As a proof concept study, we demonstrate its efficacy by automatically segmenting heart structures from cardiac MRI scans. The algorithm is based on policy search and reinforcement learning by utilizing segmentation evaluation metrics as reward function.

1.4 Incorporating Prior Knowledge and Label Noise in Medical Image Segmentation

Since manual measurements are very expensive, time consuming, and prone to inter- and intra-observer variations, having an automated, accurate, and efficient segmentation tool is the ultimate goal in quantitative radiology. In the deep learning era, numerous works have been published,

showing feasibility of deep learning in segmentation of radiology images. However, most of these works focus on new network architectures adopted to the medical problem, and they rarely consider the fundamental challenge of deep segmentation method: availability of precisely annotated data. This is a huge problem and annotations are often noisy and hard to find large amount of data.

In this part of the thesis, we propose a weakly-supervised segmentation method coupled with a deep geodesic prior to solve 3D medical image segmentation problem in a robust manner. This prior is mainly introduced to improve the performance of segmentation networks, more specifically when the annotations are noisy (i.e., not excellent). We argue that our proposed method is a significant step toward using inexpert and noisy annotations to train deep models for image segmentation without sacrificing the accuracy. The *deep geodesic prior* is specifically designed to put more attention in constructing accurate edges from weak labels.

1.5 Optimization of Segmentation Networks

Optimization methods, as an engine in deep learning algorithms, play a crucial role in this field. For past few years, adaptive optimizers such as *ADAGrad* [15] and *ADAM* [16] dominated on the field of deep learning due to their fast convergence. The general idea behind these optimizers is to consider the history of gradients for updating parameters in the next iteration. As recently discussed in some studies, adaptive optimizers may converge to different minima than classical SGD optimizers. This minima may be better in training set, but not necessarily generalizing better to unseen data as it is discussed by [17]. Some papers tried to tackle this problem by changing learning rate manually or cyclic in the classical SGD optimizers such as [18] and [19]. These methods will be discussed in details in the next section.

By getting motivation from the cyclic optimizers, we introduced new optimizer, is called Cyclic

Learning/Momentum Rate (CLMR), for which both learning rate and momentum rate are changing cyclic during training. This optimizer has two advantages over adaptive optimizers: First, it is computationally cheaper than adaptive ones. Second, it generalizes better than adaptive optimizers (or even similar SGD optimizer).

CHAPTER 2: RELATED WORKS

There have been large number of literature in recent years trying to solve the segmentation problem in both natural and medical images area [8, 20, 13, 12, 14, 21]. In practical radiology settings, segmentation plays a vital role since it is necessary to extract information from region of interest, measure the volume of specific structures, and analyze morphology of the target/tissue organ in radiological images. In this chapter, we summarize related literature in medical image segmentation both in pre-deep learning and deep learning eras.

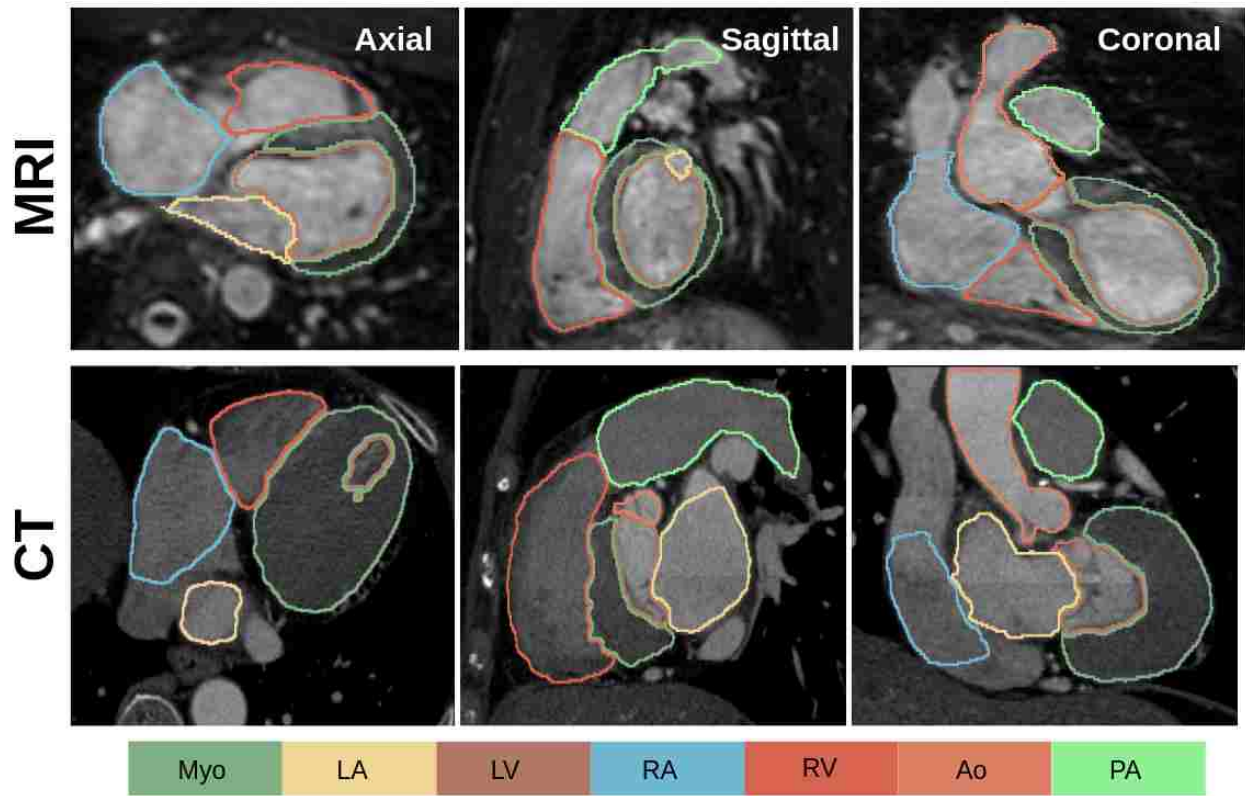


Figure 2.1: Examples of cardiac-MRI and cardiac-CT images with the corresponding contours for different substructures.

In the pre-deep learning era, statistical shape and atlas-based methods have been the state-of-the-art

cardiac segmentation approaches due to their ability to handle large shape/appearance variations. Even now many commercially available tools are based on the methods from this era. One significant challenge for such approaches is their limited efficiency: an average of 50 minutes processing time per volume on a regular workstation [22]. Statistical shape models are faster than atlas-based methods but a high degree uncertainties in the accuracy of such models is inevitable [23]. Among pre-deep learning era works, atlas-based methods have been quite popular and favored for many years. For instance, multi-atlas based whole-heart segmentation using MRI and CT by [24] and atlas propagation based method using prior information by [25] are a few key examples. Despite their accuracy, those methods often lack efficiency due to heavy computations on the registration algorithms (e.g., from 13 minutes to 11 hours of computations reported in the literature). Interested readers can find a survey paper on cardiac image segmentation methods in [26] for a full list of methods and their comparative evaluations.

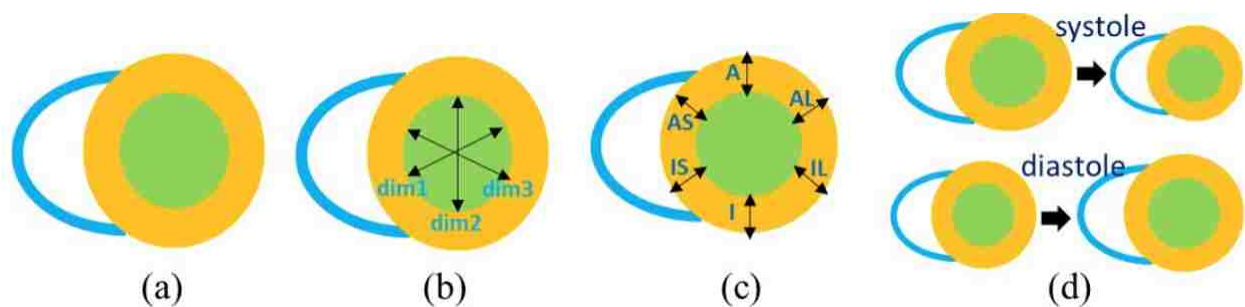


Figure 2.2: The parameters which need to be measured for quantifying heart. Specifically, left ventricle plays vital role in cardiac assessment. (Source of the figure from [1])

After deep learning spring in 2014, CNN based approaches are replacing the conventional methods in medical image segmentation fields in general, and cardiac image analysis field in particular. For instance, in [12], a multi-planar deep learning has been utilized to segment LA and pulmonary veins from MR images. A recurrent fully convolutional neural network has been proposed to segment LV from MRI in [27]. In a similar fashion, a deep learning algorithm combined with a

deformable-model approach was used to segment LV from MRI [28]. In [29], RV segmentation has been accomplished through a joint localization and segmentation algorithm within a deep learning framework. To date, the majority of deep learning methods have segmented only one or two structures of the heart and constrained to only one modality. In natural images some deep learning-based methods such as *SegNet* [8] and *DeepLab* [13] have shown significant improvement over other methods. *SegNet* [8] has proposed an encoder-decoder architecture and the information in the max-pooling layer (in encoder part) are used in decoder part to do unpooling and resizing. *DeepLab* [13] has introduced a new convolution kernel, named atrous convolution, which can capture more informative features from images when combining with features obtained from regular convolution kernel. Then, U-Net [14] was introduced based on the encoder-decoder architecture by adding skip connections from the encoder to decoder part in architecture to make flow of information easier. After that, *Tiramisu* [21] was proposed based on the *DenseNet* [30] by adding local skip connections inside of each block in encoder and decoder parts in addition to global connections from decoder to encoder. *PAN* was introduced to segment pancreas from CT images by using generative adversarial network to learn high-level consistencies in images, in addition to pixel-wise information [31].

The state-of-the-art CNN-based segmentation methods have very similar fixed network architectures and they all have been designed with a trial-and-error basis. *SegNet* [8], *CardiacNet* [12, 32, 33], and *U-Net* [14] are some of the notable approaches in the literature. To design such segmentation networks, experts have often large number of choices involved in design decisions, and manual search process is significantly guided by intuition. To address this issue, there is a considerable interest recently for designing the network architecture automatically. Reinforcement Learning (RL) [34] and evolutionary based algorithms [35] are proposed to search the optimum network hyperparameters. Such methods are computationally expensive and require a large number of processors (as low as 800 GPUs in Google's network search algorithm [34]) and may not be

doable for a widespread and more general use. Instead, in this dissertation, we propose a conceptually simple and very efficient network optimization search algorithm based on a policy gradient (PG) algorithm. PG is one of the successful algorithms in robotics field [36, 37] for learning system design parameters. Another example is by Zoph and Le [34] where authors used LSTM (long short term memory) to learn the hyperparameters of the CNN and the PG was used to learn the parameters of the LSTM. Learning parameters of LSTM need considerable amount of resources as it is discussed in [34]. Unlike that indirect parameter estimation, in this dissertation we propose a PG algorithm to directly learn network hyperparameters. Our proposed approach to learn CNN architecture is inspired by [36] and it has been adapted to deep network architecture design for performing image segmentation tasks with high accuracy. In this study, to make the whole system economical to implement for wide range of applications, search space is significantly restricted [38].

In the another area to do segmentation, the shape prior was used in the segmentation engine. The literature for integrating shape priors into image segmentation is vast, mostly from pre-deep learning era. A mainstream approach is to construct a shape prior from a set of training samples represented implicitly by signed distance functions *citekurugo,levelset*. In the deep learning era, Zotti et al. [39] used image registration to align shape priors and created atlas(es) to guide segmentation. Simply, authors have used this atlas for adding an extra loss term to the segmentation network. Modeling a prior (in shape or appearance) from medical images is still a challenging task due to highly diverse appearance, shape, and size of the anatomical objects. The first attempt to model shape prior with deep features was done by training an auto-encoder (AE) for creating features from the labels [40]. The AE was trained to reconstruct the binary input images in its output with a fully-connected layer as a bottleneck to capture the shape features. Then, these shape features were integrated into the segmentation network through an appropriate loss term. While the work in [40] is promising, it is not entirely clear whether the local anatomical variations are

captured in detail. We hypothesize that, if modeled correctly, prior information can lead to a more robust segmentation even when the labels are noisy (i.e., labels annotated by non-experts). To test this hypothesis, we propose a novel method for learning the prior from the geodesic maps of multiple objects. Then, an AE-like network is used to generate the original binary images from their corresponding geodesic maps. Finally, the features from the trained AE are used as a prior to be integrated into the segmentor for better guidance and performance improvement [41].

Optimizing parameters of neural networks have been challenging from beginning due to huge number of parameters need to be trained. Generally, the current optimizers can be categorized regarding a fixed *learning rate* (LR) and *momentum rate* (MR) or having adaptive LR and MR. *Gradient Descent* (GD), *Stochastic Gradient Descent* (SG) and Mini-batch gradient descent were first optimizers used for training neural networks. The updating rule for GD, SGD, and mini-batch GD by can be considered same by choosing X and Y as whole samples in dataset, a single sample, and a batch of samples respectively. The *Momentum* optimizer [42] speed up the convergence of optimization by considering the value of all past iterations with a rate which called *momentum*. In *Momentum* optimizer the past iterations don't play any role in cost function and cost function is only calculated regarding current iteration. *Nesterov accelerated gradient* [43] (NAG) is the first optimizer which accelerate the convergence by including information of previous iterations in calculating gradient of cost function.

One of the major drawbacks of optimizer with fixed LR and MR was lacking of considering the information of gradient of last iterations in changing (adapting) the LR and MR. i.e that's good idea to increase the LR in dimension with low slope in order to increase speed of convergence; or decrease LR in dimension with high slope to prevent fluctuation around minimum point.

ADAGrad [15] is one of fist adaptive LR optimizer which used in deep learning area and it adapts the learning rate for each parameter in network by dividing gradient of each parameter by its sum

of the squares of gradient One of the drawback of *Adagrad* is gradient vanishing due to accumulation of all past square gradient. *ADADelta* and *RMSProp* solved this problem by considering a limited window for summing past square gradient (instead of all of them). The most popular optimizer which is used in most of the deep learning application is ADaptive Momentum optimizer [16] (ADAM). ADAM optimizer updating rule used past squared gradient (as scale) and also like momentum, it keeps exponentially decaying average of past gradient.

One of the disadvantages of adaptive learning methods is computational cost. Since, they are required to calculate and keep all the past gradients and their squares to update next parameters. Also, as it is discussed by [17] the adaptive learning optimizer converge to different minima point in comparison with fixed learning rate optimizers which is worse in generalization. Cyclic Learning Rate [18] (CLR) is a new method to change the learning rate during training which need no computational cost and it is based on original idea of fixed learning optimizer. The idea behind CLR is changing global LR in cyclic manner in an interval can be beneficial. i.e as it is suggested by [18].

CHAPTER 3: 2D AND 2.5D SEGMENTATION OF MEDICAL IMAGES

The proposed algorithms in this chapter and their results are published in the following papers:

- **Aliasghar Mortazi**, Rashed Karim, Kawal Rhode, Jeremy Burt, Ulas Bagci (2017) *CardiacNet: Segmentation of Left Atrium and Proximal Pulmonary Veins from MRI Using Multi-view CNN*. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI 2017)*. Lecture Notes in Computer Science, vol 10434. Springer, Cham.
- **Aliasghar Mortazi**, Jeremy Burt, Ulas Bagci, (2017) *Deep Learning for Cardiac MRI: Automatically Segmenting Left Atrium and Proximal Veins with Human Level Performance*, in *Annual Meeting of the Radiological Society of North America (RSNA) 2017*.
- **Aliasghar Mortazi**, Jeremy Burt, and Ulas Bagci. (2017) *Machine Learning for Cardiac MRI: Automated Mapping of Left Atrium and Pulmonary Veins with Human Level Performance*, in *North American Society for Cardiovascular Imaging (NASCI) 2017*. (American Heart Association(AHA) and CVRI Young Investigator Award finalist).

3.1 Overview

The proposed pipeline (called *CardiacNet*) for deep learning based segmentation of the LA and PPVs is summarized in Figure 3.1. We used the same CNN architecture for each view of the 3D cardiac MRI after parsing them into axial, sagittal, and coronal views. The rationale behind this

decision is based on the limitation of computer memory and insufficient 3D data for training on 3D cardiac MRI from scratch, common problems in radiology. Instead, we reduce the computational burden of the CNN training by constraining the problem into a 2D domain. The resulting pixel-wise segmentations from each CNN are combined through an adaptive fusion strategy. The fusion operation is designed to maximize the information content from different views. The details of the pipeline are given in the following subsections.

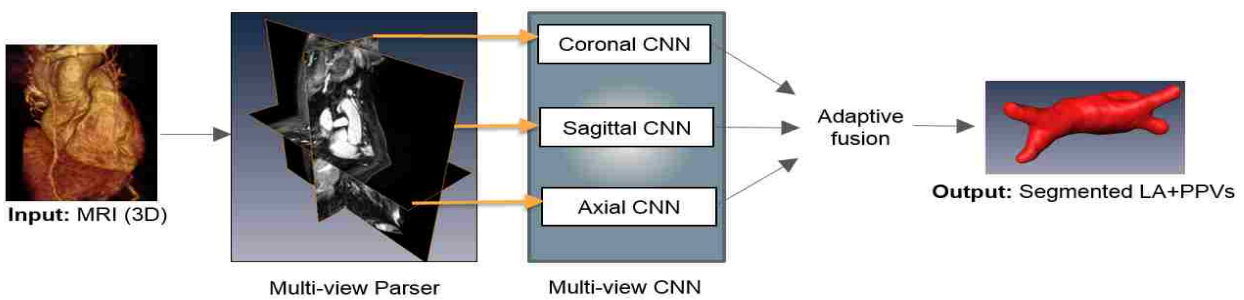


Figure 3.1: High-level overview of the proposed multi-view CNN architecture.

3.2 Multi-View CNN

We constructed an encoder-decoder CNN architecture, similar to that of Noh et al. [44]. The network includes 23 layers (11 in encoder, 12 in decoder units). Two max-pooling layers in encoder units reduce the image dimensions by half, and a total of 19 convolutional (9 in encoder, 10 in decoder), 18 batch normalization, and 18 ReLU (rectified linear unit) layers are used. Specific to the decoder unit, two upsampling layers are used to convert the images back into original sizes. Also, the kernel size of all filters are considered as 3×3 . The final layer of the network includes a softmax function (logistic) for generating a probability score for each pixel. Details of these layers, and associated filter size and numbers are given in Figure 3.2.

3.3 z-Loss Function

We used a new loss function that can estimate the parameters of the proposed network at a much faster rate, a critical necessity in medical field. We trained, the architecture in an end-to-end mapping with a loss function $L(\mathbf{o}, c) = \text{softplus}(a(b - z_c))/a$, called z-loss [45], where \mathbf{o} denotes output of the network, c denotes the ground truth label, and z_c indicate z-normalized label, obtained as $z_c = (o_c - \mu)/\sigma$ where mean (μ) and standard deviation σ are obtained from \mathbf{o} . z-loss is simply obtained with the reparametrization of *soft-plus* (SP) function (i.e., $SP(x) = \ln(1 + e^x)$) through two hyperparameters: a and b . Herein, we kept these hyperparameters fixed, and trained the network with a reduced z-loss function. The rationale behind this choice is the following: the z-loss function provides an efficient training performance as it belongs to spherical loss family, and it is invariant to scale and shift changes in the output, avoiding output parameters to deviate from extreme values.

3.4 Training *CardiacNet*

3D cardiac MRI images along with its corresponding expert annotated ground truths were used to train the CNN after the images are parsed into three views (A, S, C). Data augmentation has been conducted on the training dataset with translation and rotation operation as indicated in Table 3.1. Obtained 3D images were parsed into A, S, and C views, and more than 60K 2D images were obtained to feed training of the CNN (approximately 30K for A and C views, around 11K for S view). As a preprocessing step, all images have undergone anisotropic smoothing filtering and histogram matching.

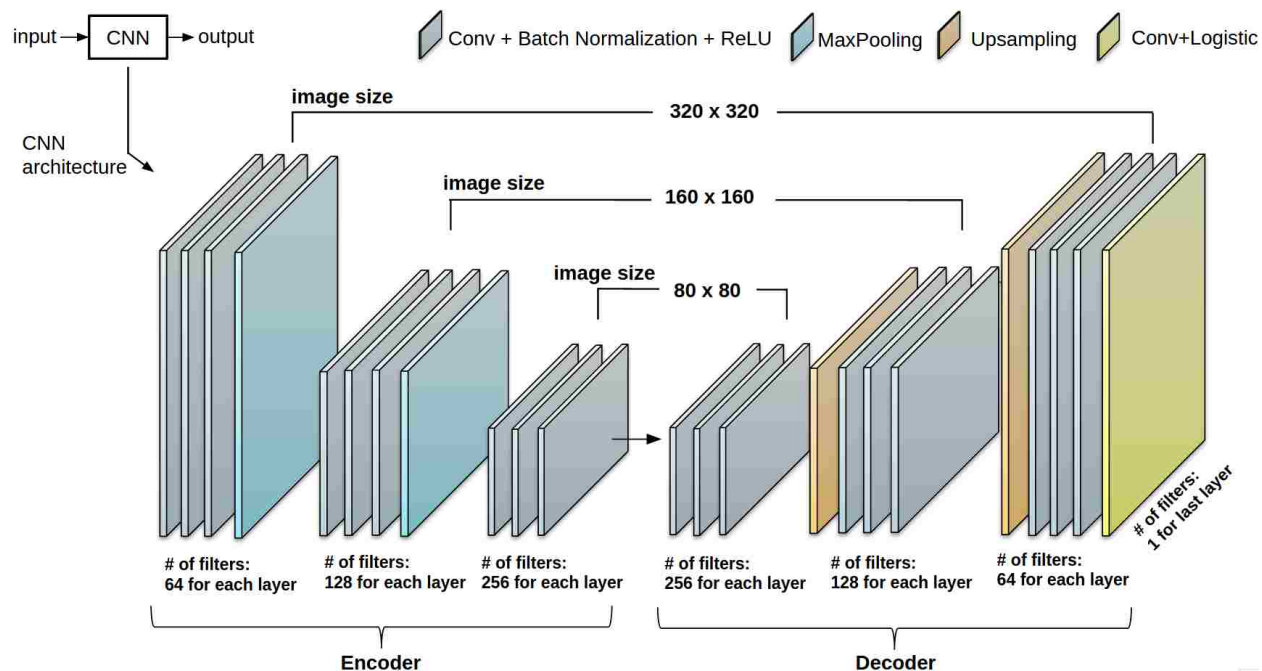


Figure 3.2: Details of the CNN architecture. Note that image size is not necessarily fixed for each view's CNN.

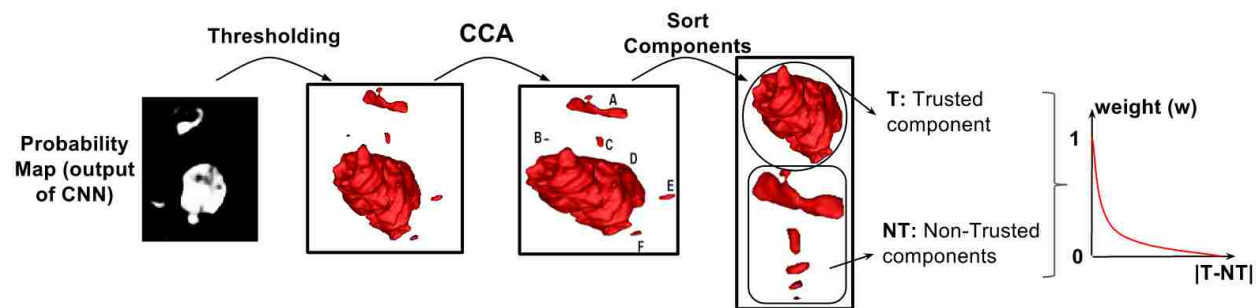


Figure 3.3: Connected components obtained from each view were computed and the residual volume (T-NT) was used to determine the strength for fusion with the other views.

3.5 Multi-View Information Fusion

Since cardiac MRI is often not reconstructed with isotropic resolution, we expect varying segmentation accuracy in different views. A scan from axial, sagittal, and coronal view are shown in first

row of Figure 3.4. In order to alleviate potential adverse effects caused by non-isotropic spatial resolutions of a particular view, it is desirable to reduce the contribution of that view into final segmentation. We have achieved this with the adaptive fusion strategy as described next. For a given MRI volume \mathbf{I} , and its corresponding segmentation \mathbf{o} , we proposed a new strategy, called *robust region*, that roughly determined the reliability of the output segmentation \mathbf{o} by assessing its object distribution. To achieve this, we hypothesize that the output should include only one connected object when the segmentation is successful, and if there is more than a single connected object available, these can be considered as false positives. Accordingly, respective performance of segmentation performance in A, S, and C views can be compared and weighted. To this end, we utilized connected component analysis (CCA) to rank output segmentations and reduced the contribution of CNN for a particular view when false positive findings (non-trusted objects/components) were large and true positive findings (trusted object/component) were small. Figure 3.3 describes the adaptive fusion strategy as $CCA(\mathbf{o}) = \{o_1, \dots, o_n | \cup o_i = \mathbf{o}, \text{ and } \cap o_i = \phi\}$. Thus, the contribution of each view's CNN is computed based on a weighting $w = \max_i\{|o_i|\} / \sum_i |o_i|$, indicating that higher weights are assigned when the component with largest volume dominated the whole output volume. Note that this block has been used only in the test phase. Complementary to this strategy, we also use simple linear fusion of each views for comparison (See Experimental Results section).

3.6 Experimental Results

Data sets: Thirty cardiac MRI data sets were provided by the STACOM 2013 challenge organizers [22]. Ten training data included ground truth labels, and the remaining twenty were provided as a test set. It is important to note that not the complete PVs are considered in the segmentation challenge, but only the proximal segments of the PVs up to the first branching vessel or after 10 mm from the vein ostium were included in the segmentation. MR images were obtained from

a 1.5T Achieva (Philips Healthcare, The Neatherlands) scanner with an ECG-gated 3D balanced steady-state free precession acquisition [22] with TR/TE= 4.4/2.4 ms, and Flip-angle=90°. Typical acquisition time for the cardiac volume imaging was 10 minutes. In-plane resolution was recorded as $1.25 \times 1.25 \text{ mm}^2$, slice thickness was measured as 2.7 mm. Further details on the data acquisition, and image properties can be found in [22].

Table 3.1: Data augmentation parameters and number of training images

<i>Data augmentation</i>		
Methods	Parameters	
Translations	$(x + trans, y = 0), trans \in [-20, 20]$	
	$(x = 0, y + trans), trans \in [-20, 20]$	
Rotation	$k \times 45, k \in [-2, -1, 1, 2]$	
<i>Training images</i>		
CNN	# of images	Image size
Sagittal	10,800	320×320
Axial	28,800	110×320
Coronal	28,800	110×320

Evaluations. For evaluation and comparison with other state-of-the-art method, we have used the same evaluation metrics, provided by the STACOM 2013 challenge: Dice index and surface-to-surface (S2S) metrics. In addition, we calculated Dice index and S2S for the LA and PPVs separately. To provide a comprehensive evaluation and comparisons, sensitivity (true positive rate), specificity (true negative rate), precision (positive prediction value), and Dice index values for the combined LA and PPVs were included too. Table 3.2 summarizes all these evaluation metrics along with efficiency comparisons where we tested our algorithm both in GPU and CPU. LTSI-VRG, UCL-1C, and UCL-4C are three atlas-based method which their output were published publicly as a part of STACOM 2013 challenge. Also, OBS-2 is the result from human observer which its output was available as a part of STACOM 2013 challenge. 8 out of 10 subjects were used for training purpose and 2 of them were used for validation. Also, 20 subjects were used in

the test phase for evaluation. In almost all evaluation metrics in the test set, the proposed method out-performed the state-of-the-art approaches by large margins. Table 3.2 indicates the results of varying combinations baselines methods such as single CNN in particular view (i.e., *S_CNN* for sagittal view), with simple linear fusion F-CNN, adaptive fusion AF-CNN, and with the new loss function AF-CNN-SP. In AF-CNN, the loss function was cross-entropy. The best method in the challenge data set was reported to have a Dice index of 0.94 for LA and 0.65 for PPVs (combined LA and PPVs was less than 0.9). In our proposed method, the Dice index for combined LA and PPVs was well above 0.90. For efficiency comparison, our approach only takes at most 10 seconds on a Nvidia TitanX GPU and 7.5 minutes in a CPU with Octa-core processor (2.4 GHz) configuration. The method in [46] required 30-45 minutes of processing times (with Quad-core processor (2.13 GHz)). For qualitative evaluation, we have used surface rendering of output segmentations compared to ground truth in Figure 3.4. Sample axial, sagittal, and coronal MRI slices are given in the same figure with ground truth annotations overlaid with the segmented LA and PPVs.

Table 3.2: The evaluation metrics for state-of-the-art and proposed methods. **: the running time on CPU *: the running time on NVIDIA TitanX GPU

Methods	LTSI_VRG	UCL_1C	UCL_4C	OBS_2	A_CNN	C_CNN	S_CNN	F-CNN	AF-CNN	AF-CNN-SP
Dice(LA)	0.910	0.938	0.859	0.908	0.903	0.804	0.787	0.873	0.928	0.951
Dice(PPVs)	0.653	0.609	0.646	0.751	0.561	0.478	0.398	0.506	0.616	0.685
S2S(LA) in mm	1.640	1.086	2.136	1.538	1.592	2.679	2.853	1.771	1.359	1.045
S2S(PPVs) in mm	1.994	1.623	2.375	1.594	1.928	2.878	3.581	2.121	1.718	1.427
Sensitivity	0.926	0.828	0.832	0.894	0.806	0.658	0.663	0.743	0.883	0.895
Specificity	0.998	0.999	0.999	0.997	0.996	0.994	0.997	0.997	0.999	0.999
Precision	0.815	0.957	0.814	0.936	0.905	0.774	0.880	0.953	0.936	0.938
Dice (all)	0.862	0.886	0.819	0.911	0.845	0.695	0.734	0.820	0.887	0.905
Running Time (sec)	3100**	1200**	1200**	-	170**	170**	155**	450**	450**	450**
	-	-	-	-	3.5*	3.5*	3*	10*	10*	10*

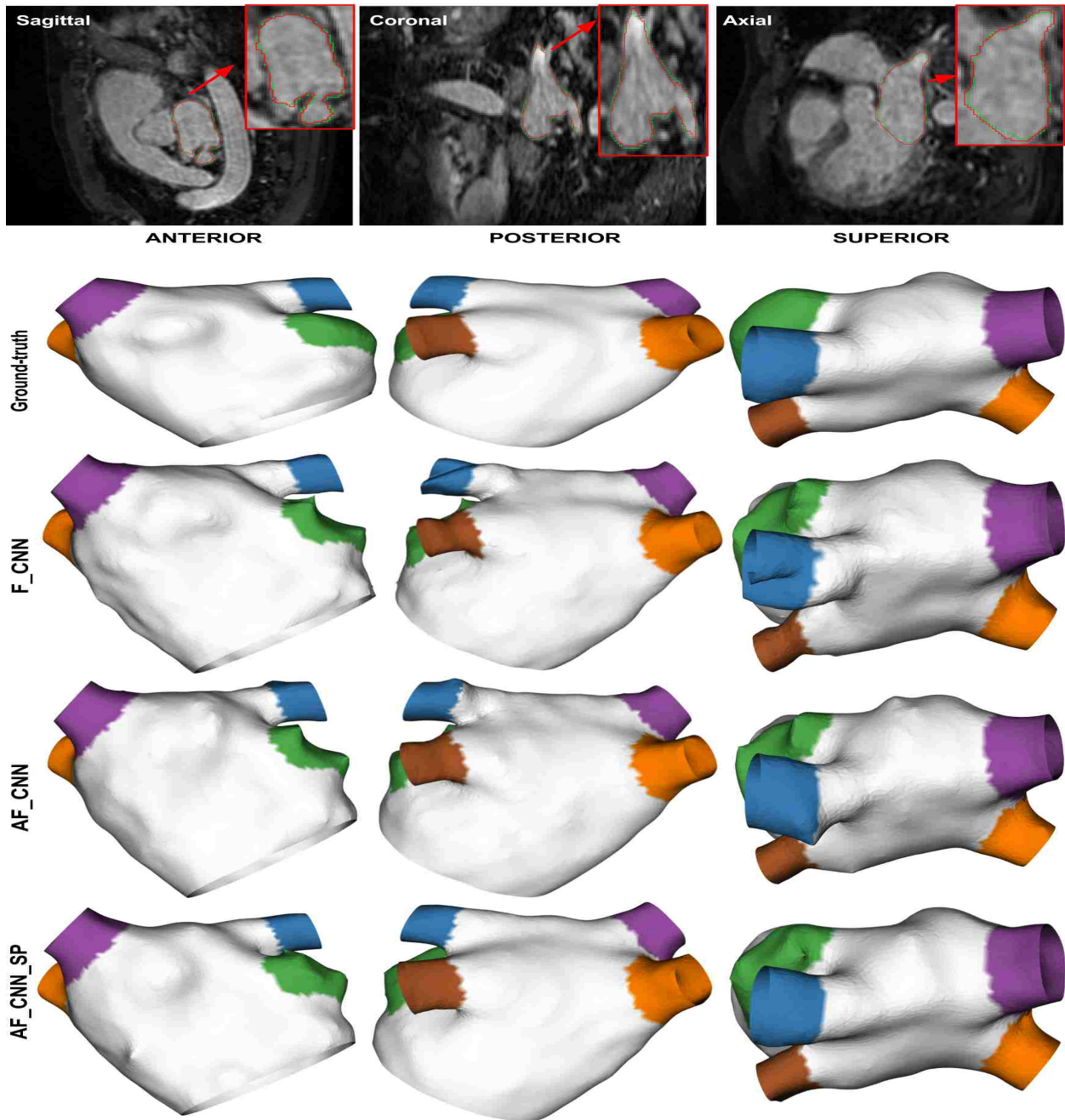


Figure 3.4: First row shows sample MRI slices from S, C, and A views (red contour is ground-truth and green one is output of proposed method). Second-to-fifth rows: 3D surface visualization for the ground-truth and the output generated by the proposed method w.r.t simple fusion (F), adaptive fusion (AF), and the new loss function (SP).

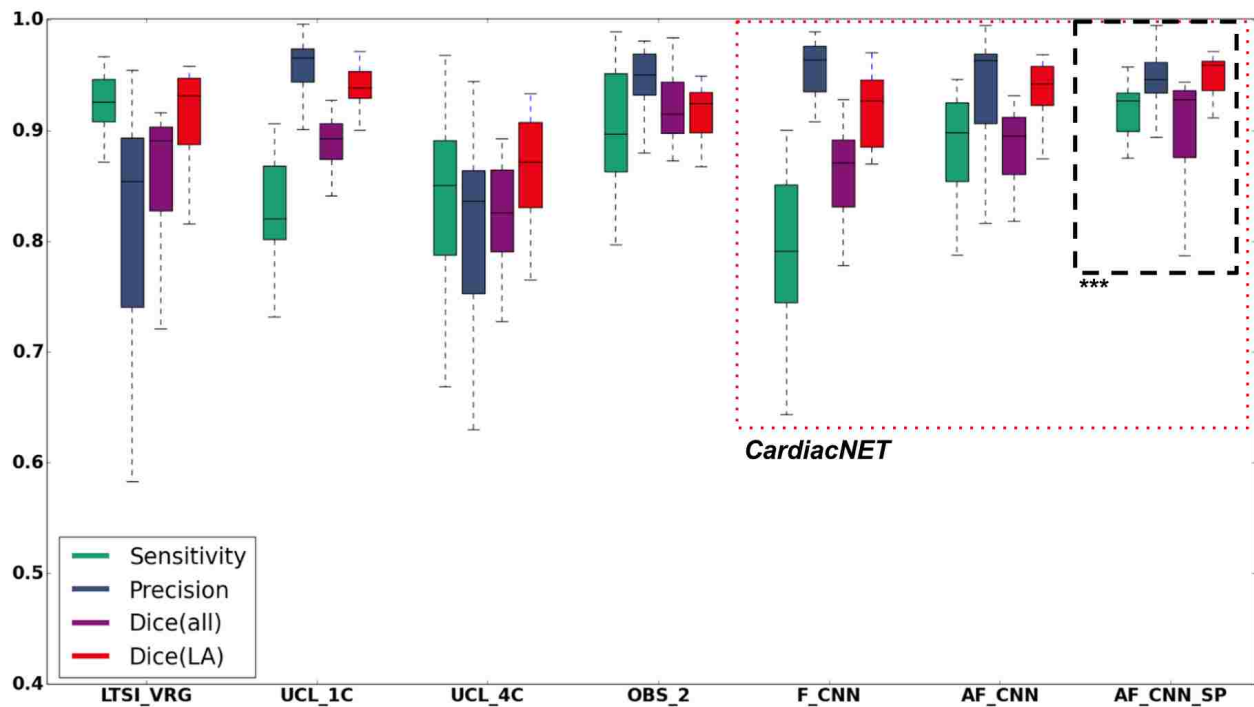


Figure 3.5: Box plots for different metrics (sensitivity, precision, and Dice index) for state-of-the-art methods (LTSI_VRG, UCL_1C, UCL_4C, OBS_2) and proposed methods (F_CNN, AF_CNN, AF_CNN_SP) on the LA segmentation benchmark

3.7 Discussions and Concluding Remarks

The advantage of *CardiacNet* is accurate and efficient method for both LA and PPVs segmentation in atrial fibrillation patients: combined segmentation of the LA and PPVs. Precise segmentation of the LA and PPVs is needed for ablation therapy planning and clinical guidance in AF patients. PPVs have a greater number of anatomical variations than the LA-body, leading to challenges with accurate segmentation. Joint segmentation the LA and PPVs is even more challenging compared to sole LA-body segmentation. Nevertheless, with all available quantitative metrics, the proposed method has been shown to greatly improve the segmentation accuracy on the existing benchmark for LA and PPVs segmentation. The benchmark evaluation has also allowed the method and its

variations to be cross-compared on the same dataset with other existing methods in literature.

Despite the efficacy of the proposed method, there are several possibilities that our work can be improved. Firstly, the new method should be tested, evaluated, and validated our in more diverse data sets from several independent cohorts, and at the different imaging resolution and noise levels, and even across different scanner vendors. Secondly, extending our framework into 4D (i.e motion) analysis of cardiac images can be possible by extending our parsing strategy. Thirdly, we aim to explore the feasibility of training completely 3D cardiac MRI based on the availability of multiple GPUs, or developing sparse CNNs to alleviate the segmentation problem. Fourthly, with low-dose cardiac CT technology on the rise; it is desirable to have similar network structure trained on CT scans. This notable efficacy of the deep learning strategies presented in this work promises a similar performance on CT scans.

In conclusion, the proposed method has utilized the strength of deeply trained CNN to segment LA and PPVs from cardiac MRI. We have shown combining information from different views of MRI by using an adaptive fusion strategy and a new loss function improves segmentation accuracy and efficiency significantly.

CHAPTER 4: MULTI-OBJECT SEGMENTATION OF MULTI-MODAL MEDICAL IMAGES

The proposed algorithms in this chapter and their results are published in the following papers:

- **Aliasghar Mortazi**, Jeremy Burt, Ulas Bagci (2018) Multi-Planar Deep Segmentation Networks for Cardiac Substructures from MRI and CT. In: Pop M. et al. (eds) Statistical Atlases and Computational Models of the Heart. ACDC and MMWHS Challenges. STACOM 2017. Lecture Notes in Computer Science, vol 10663. Springer, Cham (challenge).
- Xiahai Zhuang, Lei Li, Christian Payer, Darko Štern, Martin Urschler, Mattias P. Heinrich, Julien Oster, Chunliang Wang, Örjan Smedby, Cheng Bian, Xin Yang, Pheng-Ann Heng, **Aliasghar Mortazi**, Ulas Bagci, and etc. “Evaluation of algorithms for multi-modality whole heart segmentation: An open-access grand challenge,” Medical Image Analysis, vol. 58, pp. 101537, 2019.

4.1 Proposed CNN Architecture

Our proposed method is called multi-object 2.5D (multi-planar) convolutional neural networks (MO-MP-CNN), and its modules are illustrated in Figure 4.1. MO-MP-CNN takes 3D CT or MR scans as an input and parses them into three perpendicular planes: Axial (A), Coronal (C), and Sagittal (S). For each plane (and modality), a 2D CNN is trained to label pixels. CNNs have been trained from scratch to adapt into CT and MRI context. After training each of the 2D CNNs separately, adaptive fusion strategy is utilized by combining the probability maps of each of the

CNNs. The details of the CNN and adaptive fusion method are explained in the following.

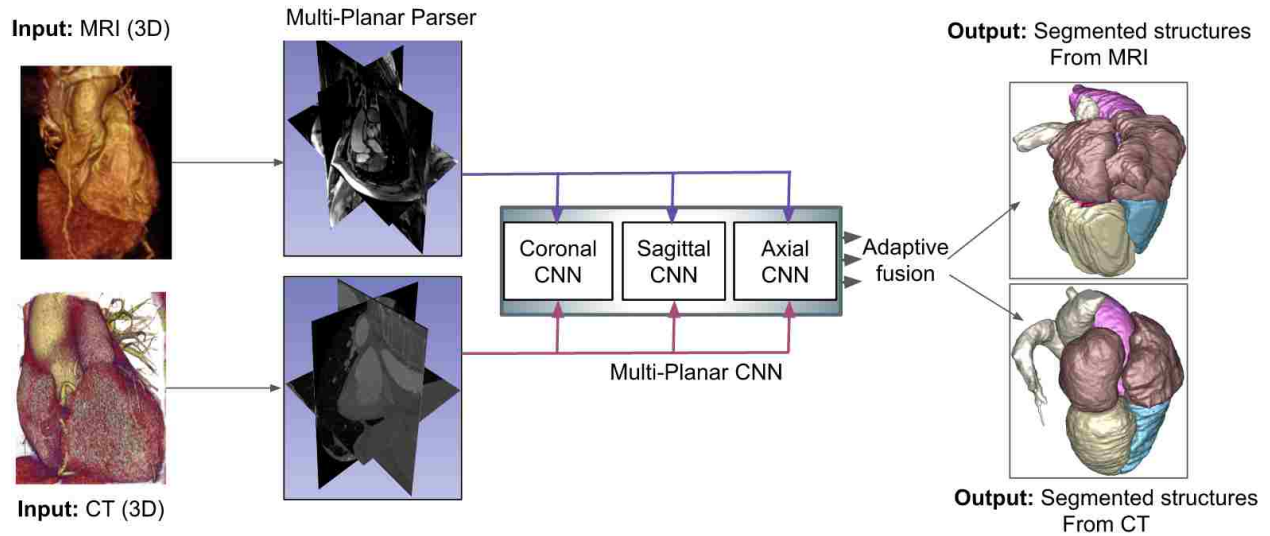


Figure 4.1: Overview of Mo-MP-CNN with adaptive fusion.

Table 4.1: Data augmentation parameters.

<i>Data augmentation</i>		
Methods	parameters	
Zoom in	Scale $\epsilon[1.1, 1.3]$	
Rotation	$k \times 45, k \epsilon[-1, 1]$	
<i>Training images (CT)</i>		
CNN	# of images	Image size
Sagittal	40,960	350×350
Axial	21,417	350×350
Coronal	40,960	350×350
<i>Training images (MRI)</i>		
CNN	# of images	Image Size
Sagittal	20,074	288×288
Axial	29,860	288×160
Coronal	19,404	288×288

The proposed encoder-decoder based network architecture is illustrated in Figure 4.2. Twelve *convolution* layers have been used in encoder and decoder separately. In the encoder part, two

max-pooling layers have been used to reduce the dimension of the image by half and in decoder part two *upsampling* layers (bilinear interpolation) have been used to get the image back to its original size. The size of all filters were set as 3×3 . Each convolution layer is followed by a *batch normalization* and *Rectified Linear Unit (ReLU)* as an activation function. The number of filters in the last convolution layer is equal to the number of classes (i.e., 8 (background+7 objects)) and is followed by a *softmax* function to make a final probability map for each object. Similar to [12], the simplified z-loss [45] function has been used to train the network. To provide a sufficient number of training images for the networks, data augmentation has been applied to the training images by rotation and zoom-in operations. The details of the augmentation and the number of data for each CNN are summarized in Table 4.1.

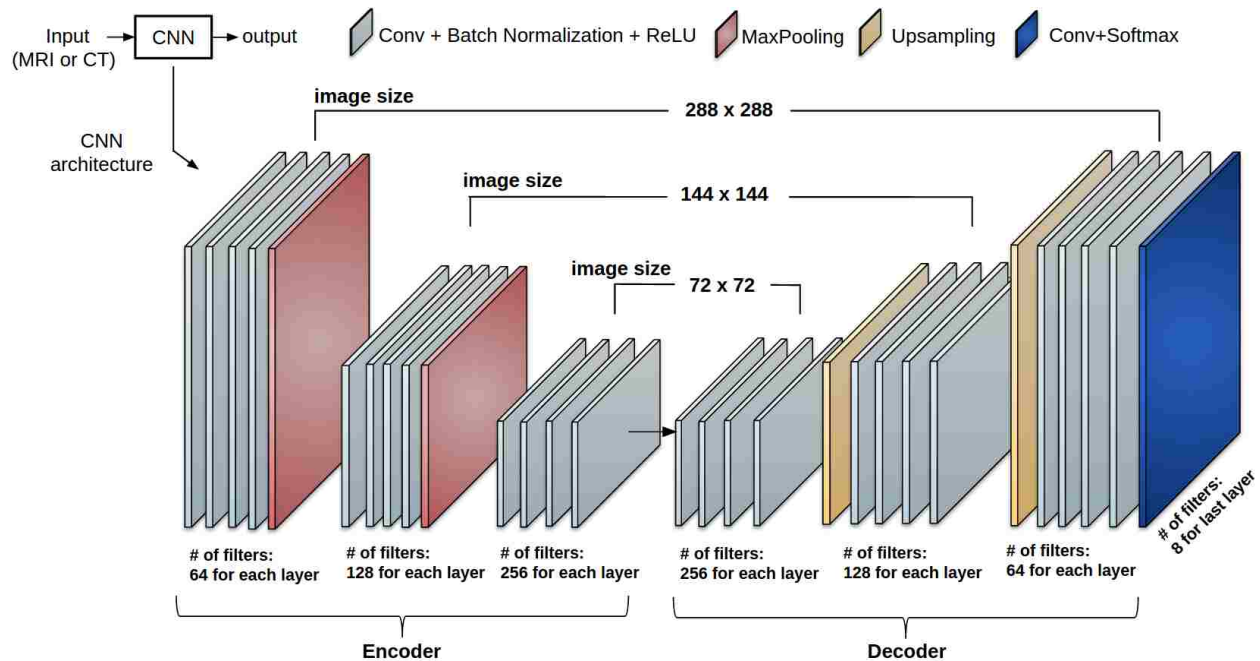


Figure 4.2: Details of the CNN architecture. Note that image size is not necessarily fixed for each plane's CNN.

4.2 Multi-Object Adaptive Fusion

An adaptive fusion strategy has been extended in the way that it can be applied to multi-object segmentation instead of binary segmentation. Let \mathbf{I} and \mathbf{P} denote an input and output image pair, where output is the probability map of the CNN. Also, let the final segmentation be denoted as \mathbf{o} . As shown in Figure 4.3, \mathbf{o} is obtained from the probability map \mathbf{P} by taking the maximum probability of each pixel in all classes (labels). Then, a connected component analysis (CCA) is applied to \mathbf{o} to select reliable and unreliable regions, where unreliable regions are considered to come from false positive findings. Although this approach gives a “rough” estimation of the object, this information can well be used for assessing the quality of segmentations from different planes. If it is assumed that \mathbf{n} is the number of classes (structures) in the images and \mathbf{m} is the number of components in each class, then connected component analysis can be performed as follows: $CCA(\mathbf{o}) = \{o_{11}, \dots, o_{nm} | \cup o_{ij} = \mathbf{o}, \text{ and } o_{11}, \dots, o_{nm} | \cap o_{ij} = \phi\}$. For each class \mathbf{n} , we can now assign reliability parameters (weights) to increase the influence of planes that have more reliable (trusted) segmentations as follows: $w = \sum_i \{max_j \{|o_j|\}\} / \sum_{ij} |o_{ij}|$, where w indicates a weight parameter. In our interpretation of the CCA, the difference between trusted and non-trusted regions have been used to guide the reliability of the segmentation process: the higher the difference, the more reliable the segmentation (See Figure 4.3, weight distribution w.r.t the difference). In test phase, we have simply used those predetermined weights from the training stage.

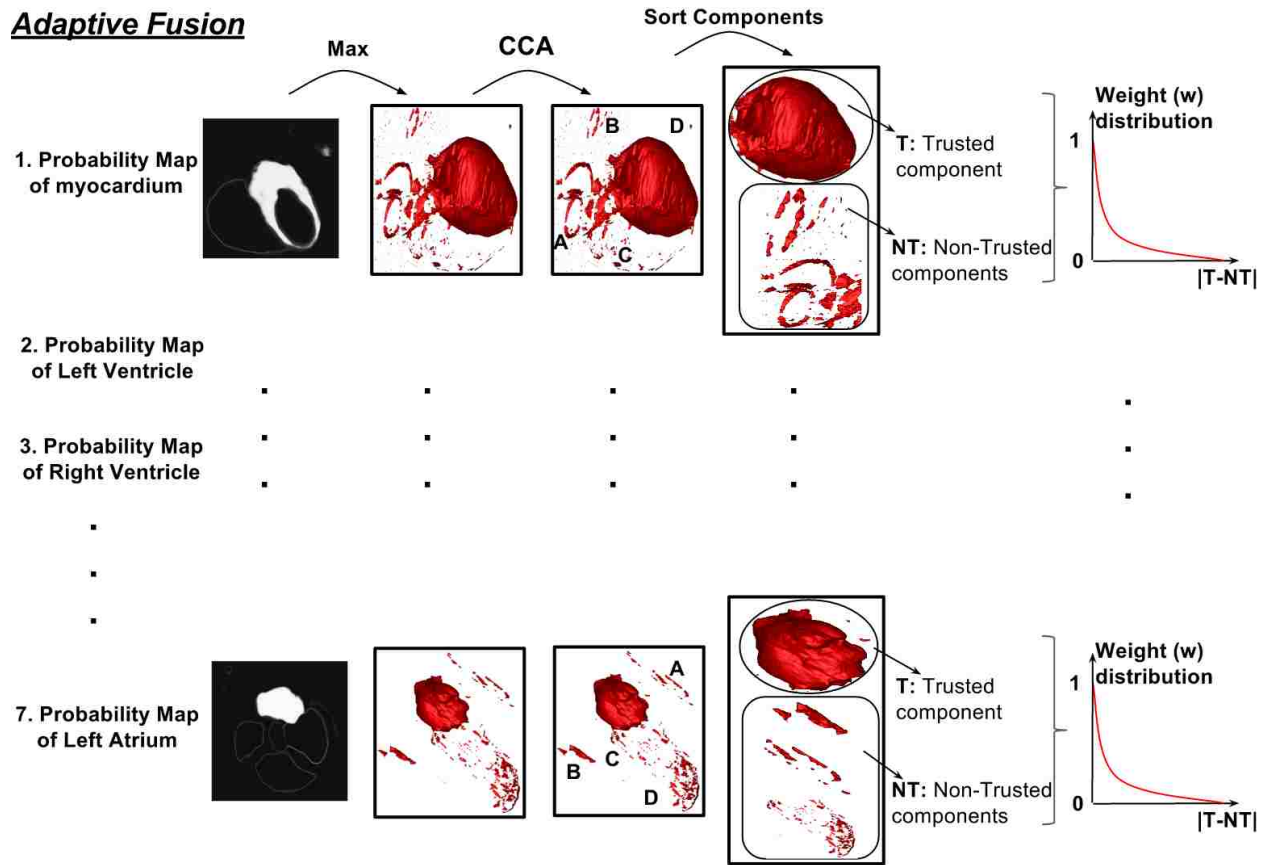


Figure 4.3: Connected components obtained from each plane were computed and the residual volume (T-NT) was used to determine the strength for fusion with the other planes.

4.3 Experimental Results

Dataset and preprocessing: For the experiments and evaluations of the proposed method, we used the STACOM 2017 for whole heart segmentation challenge dataset, containing 20 MR and 20 CT images for training (with ground-truth) and 40 test images without ground-truth for each modality [47]. We performed a 4 fold cross-validation on the dataset such that 15 subjects were used for training and 5 subjects have been chosen for validation for each fold. The CT images were obtained from routine cardiac CT angiography and to cover the whole heart, extending from the

upper abdomen to the aortic arch. Axial in-plane resolution was 0.78×0.78 mm and slice thickness was 1.6 mm. The MR images were acquired by using 3D balanced steady state free precession (b-SSFP) sequences, with about 2 mm acquisition resolution in each direction. In preprocessing step, anisotropic smoothing filtering was applied to both CT and MR images prior to segmentation. In addition, histogram matching was used for MR images to alleviate intensity non-standardness issues.

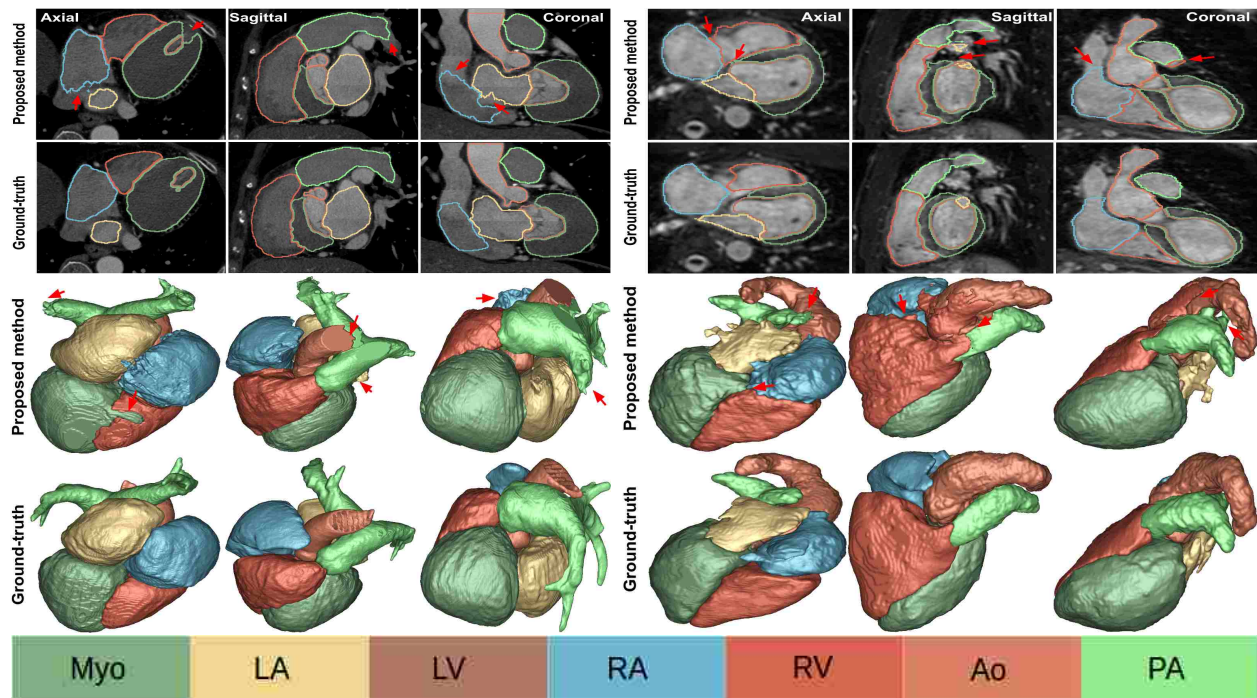


Figure 4.4: First two rows show axial, sagittal, and coronal planes of the CT (first three columns) and MR images (last three columns), annotated cardiac structures, and their corresponding surface renditions (last two rows). Red arrows indicate some of mis-segmentations.

Evaluation: Five metrics were assessed: sensitivity, specificity, precision, dice index (DI), and surface to surface (S2S) distance. A summary of the findings for each structure and also for the whole heart are reported in Table 4.2. The WHS is the average of all structures. The box-plot for sensitivity, precision, and DI for both CT and MRI and for all structures are shown in Figure 4.5. The qualitative results (including difficult cases for segmentation) for CT and MR modalities

are illustrated in Figure 4.4. Algorithms were implemented on the Nvidia TitanXp GPUs using Tensorflow [48]. The average time for segmenting the whole heart from the cardiac CT volume using three TitanXp GPUs was about 50 seconds. Segmenting using the cardiac MR volume took about 17 seconds. For comparison, the time on the Intel Xeon Processor E5-2620 with 8 cores for CT images was about 30 minutes and for MR images was about 8 minutes.

Table 4.2: Quantitative evaluations of the proposed segmentation method for both CT and MRI are summarized.

Structures:	MRI								CT							
	<i>Myo</i>	<i>LA</i>	<i>LV</i>	<i>RA</i>	<i>RV</i>	<i>Aorta</i>	<i>PA</i>	<i>WHS</i>	<i>Myo</i>	<i>LA</i>	<i>LV</i>	<i>RA</i>	<i>RV</i>	<i>Aorta</i>	<i>PA</i>	<i>WHS</i>
Sensitivity	0.816	0.856	0.928	0.827	0.878	0.728	0.782	0.831	0.888	0.903	0.918	0.835	0.872	0.86	0.783	0.866
Specificity	0.999	0.999	0.999	0.998	0.999	0.999	0.998	0.999	0.999	0.999	0.999	0.999	0.998	0.999	0.999	0.999
Precision	0.842	0.88	0.936	0.909	0.846	0.873	0.791	0.868	0.912	0.931	0.944	0.914	0.911	0.983	0.912	0.929
DI	0.825	0.887	0.932	0.874	0.884	0.772	0.784	0.851	0.898	0.925	0.93	0.877	0.888	0.909	0.851	0.897
S2S(mm)	1.152	1.130	1.084	1.401	1.825	1.977	2.287	1.551	0.903	1.386	1.142	2.019	1.895	1.023	1.781	1.450

4.4 Discussions and Concluding Remarks

The main goal of the current study is to develop a framework for accurate and efficient segmentation of all cardiac substructures from both cardiac CT and MR images. The main strength of the proposed method is to train multiple CNNs from scratch and to allow an adaptive fusion strategy for information maximization in pixel labeling despite the limited data and hardware support. Our findings indicate that MO-MP-CNN can be used as an efficient tool to delineate cardiac structures with high precision, accuracy, and efficiency.

Technically, one may question why we did not employ a completely 3D CNN approach instead of utilizing a multi-planar fusion of multiple 2D CNNs. As discussed in [12], the lack of a large number of 3D images restricts the depth of CNN training, which may result in sub-optimal implementation. Hence, training large number of 2D slices is much more feasible than utilizing a 3D approach with the current algorithm. In the instance of plentiful GPU processing power and 3D

imaging data, training would be optimized using a 3D CNN.

Another limitation of our work stems from the use of the *softmax* function in the last layer of the proposed network. To explore whether the information loss due to class normalization in this step is significant, further research should be undertaken using information from the layer before the *softmax* in fusion part and with comparison to the current system. Finally, further work is needed to establish comparative evaluation of different deep neural network approaches such as *ResNet*, *U-net*, and others. While deeper networks are desirable to achieve higher precision in segmentation tasks, lack of 3D data is a significant limitation for training such a system. Data augmentation and transfer learning have been shown to adequately address such challenges to a certain degree, but there is currently no research proving the optimality of such networks relative to the availability of data at hand.

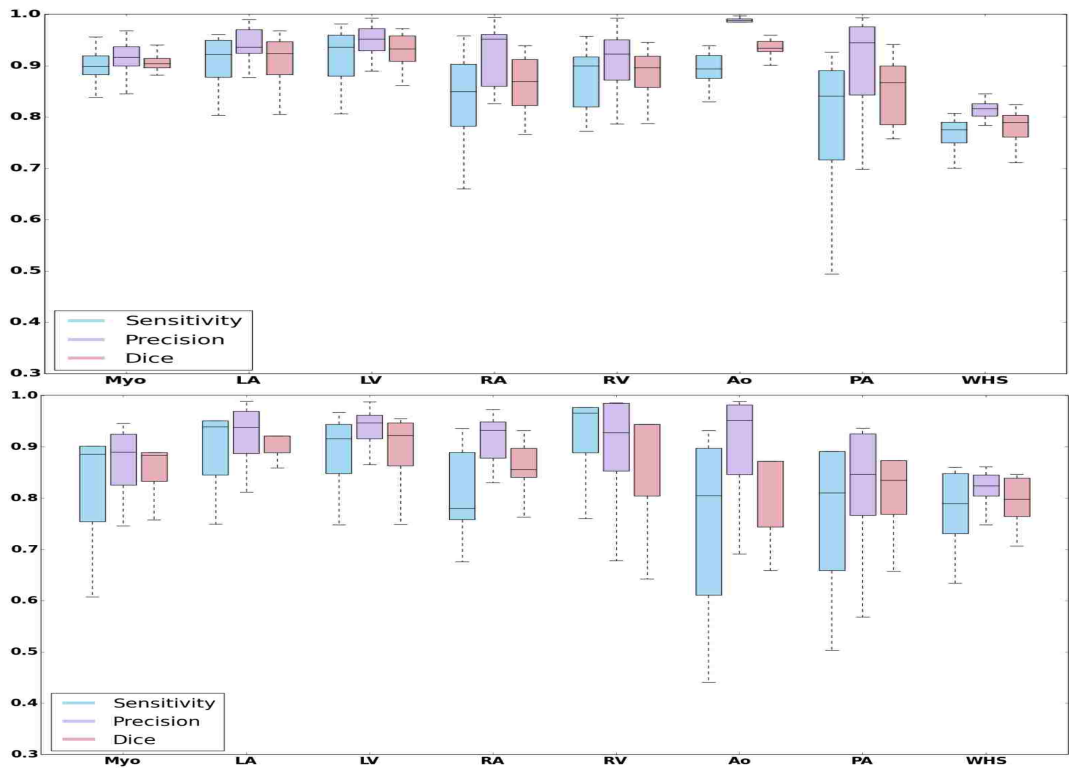


Figure 4.5: Box plots for sensitivity, precision, and Dice index for each structure and WHS. Top figure is for CT dataset and bottom figure is for MR dataset

CHAPTER 5: AUTOMATICALLY DESIGNING CNN ARCHITECTURES UNDER LIMITED COMPUTATIONAL SOURCES

The proposed algorithms and their results are published in the following paper:

- **Aliasghar Mortazi**, Ulas Bagci (2018) Automatically Designing CNN Architectures for Medical Image Segmentation. In: Shi Y., Suk HI., Liu M. (eds) Machine Learning in Medical Imaging. (MLMI), MICCAI, 2018. Lecture Notes in Computer Science, vol 11046. Springer, Cham.

5.1 Overview

The overview of the proposed method is illustrated in Figure 5.1. The hyperparameters of the network are considered as policies to be learned during PG training. To our best of knowledge, this is the first study to find optimum hyperparameters of a given network with policy gradient directly. Moreover, our proposed baseline architecture of densely connected encoder-decoder CNN and the use of *Swish* function as an alternative to *ReLU* are novel and superior to the existing systems. Lastly, our study is the first medical image segmentation work with a fully automated algorithm that discovers the optimal network architecture.

5.2 Policy Gradient

Policy gradient is a class of reinforcement learning (RL) algorithms and relied on optimization of parametrized policies with respect to a expected return (reward) [49]. Unlike other RL methods (such as *Q-Learning*), the PG learns the policy function directly to maximize receiving rewards.

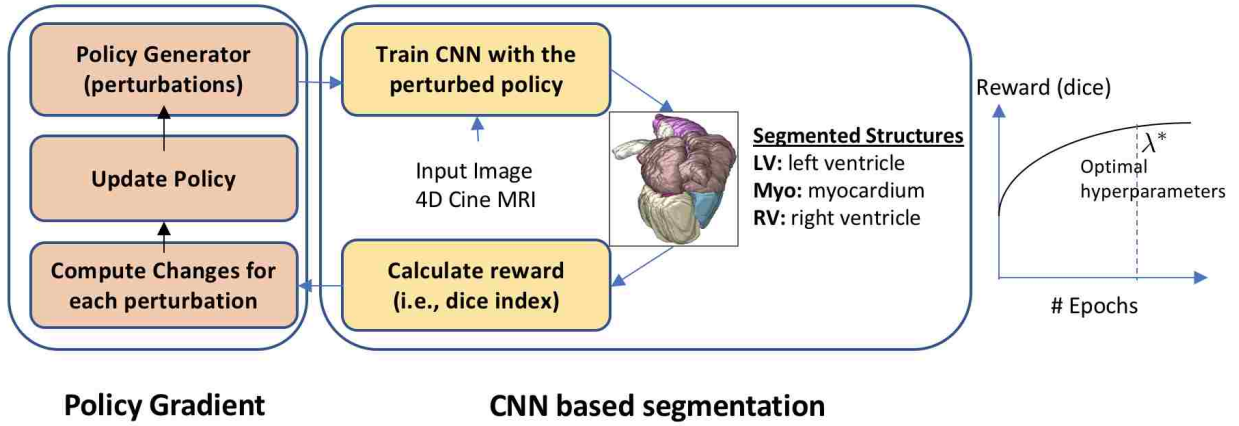


Figure 5.1: Overview of proposed method. First, the policy is initialized randomly and then P perturbations are generated. The network is trained with each perturbation and reward from each perturbation is calculated. The policy will be updated accordingly and the process will be repeated until no significant changes in the reward. Reward is simply set as dice coefficient for evaluating how good the segmentation is.

In our setting, we consider each hyperparameter of the network as a policy, which can be learned during network training. Assume that we have a policy $\pi_0 = \{\theta_1, \theta_2, \dots, \theta_N\}$, indicating the hyperparameters of the network, where N is the number of hyperparameters (dimensions). Our objective is to learn these hyperparameters (i.e., policies) by maximizing a receiving reward. In segmentation task, this reward can be anything measuring the goodness of segmentations such as dice index and Hausdorff distances. Once we randomly initialize hyperparameters, we generate new policies by randomly perturbing the policies in each dimension. Note that each dimension represents an exploration space for hyperparameters such as filter width, height, and etc. Let $P(\pi_0) = \{\pi_1, \pi_2, \dots, \pi_p\}$ be p random perturbations generated near π_0 , represented as $\pi_i = \pi_0 + \Delta_i$ for $i \in \{1, 2, \dots, p\}$. For each random perturbation, $\Delta_i = \{\delta^1, \delta^2, \dots, \delta^N\}$, we assume that δ^d is randomly chosen from $\{-\epsilon^d, 0, +\epsilon^d\}$ for every $d \in \{1, 2, \dots, N\}$ where epsilon is derivative of a function y with respect to x (Later we will define x and y for each dimension in Section 5.4).

The network is trained with these p generated policies, and reward (segmentation outcome) is ob-

tained for each policy. Finally, the maximal reward (i.e., highest dice coefficient) is determined to set the optimal network architecture hyperparameters accordingly. To estimate the partial derivative of the policy function for each dimension, each perturbation is grouped to non-overlapping categories of negative perturbation, zero perturbation, and positive perturbation: C_-^d , C_0^d , and C_+^d such that $\pi_i^d \in \{C_-^d, C_0^d, C_+^d\}$. The perturbations are generated to make sure each category has approximately $p/3$ members. Then, the absolute reward for each category is calculated as a mean of all the rewards $Ave^d = \{Ave_-^d, Ave_0^d, Ave_+^d\}$ for each dimension d . Based on this average reward, the initial policy is updated accordingly:

$$\pi_{0,new}^d = \begin{cases} \pi_0^d - \varepsilon^d & \text{if } Ave_-^d > Ave_0^d \text{ and } Ave_-^d > Ave_+^d \\ \pi_0^d + 0 & \text{if } Ave_0^d \geq Ave_-^d \text{ and } Ave_0^d \geq Ave_+^d \\ \pi_0^d + \varepsilon^d & \text{if } Ave_+^d > Ave_0^d \text{ and } Ave_+^d > Ave_-^d \end{cases} \quad (5.1)$$

The pseudo-code for policy gradient is given in Algorithm 1.

Algorithm 1 Policy Gradient's algorithm

- 1: Initialize π_0 randomly
 - 2: **for** $e=1$:epochs **do**
 - 3: Generate p randomly perturbation of $P(\pi_0) = \{\pi_1, \pi_2, \dots, \pi_p\}$
 - 4: **for** $i=1$: p **do**
 - 5: Train network with policy π_i
 - 6: Calculate reward
 - 7: **for** $d=1$: N **do**
 - 8: $Ave_+^d \leftarrow$ Average rewards for C_+^d
 - 9: $Ave_0^d \leftarrow$ Average rewards for C_0^d
 - 10: $Ave_-^d \leftarrow$ Average rewards for C_-^d
 - 11: **Update** $\pi_{0,new}^d$ based on Equation 1.
-

5.3 Proposed Backbone Architecture for Image Segmentation

As it has been shown in [8, 12], the encoder-decoder architecture is well design deep learning architecture for the segmentation tasks. More recently, the densely connected CNN [30] has been shown that connecting different layers lead into more accurate results for detection problem. Based on this recent evidence, a densely connected encoder-decoder is proposed herein as a new CNN architecture and we use this as our baseline architecture to optimize. The proposed baseline architecture is illustrated in Figure 5.2. Dense blocks consist of four layers, each layer includes convolution operation following by batch normalization operation (BN) and *Swish* activation function [50] (unlike commonly used ReLu). Also, a concatenation operation is conducted for combining the feature maps (through direction (axis) of the channels) for the last three layers. In other words, if the input to l^{th} layer is \mathbf{X}_l , then the output of l^{th} layer can be represented as:

$$F(\mathbf{X}_l) = Conv(BN(Swish(\mathbf{X}_l))), \quad (5.2)$$

where $Swish(x) = xSigmoid(\beta x)$ and as it is discussed in [50], the **Swish** was shown to be more powerful than ReLu since parameter β can be learned during training to control the interpolation between linear function ($\beta = 0$) and ReLu function ($\beta \approx \infty$). Since we are doing concatenation before each layer (except the first one), so the output of each layer can be calculated only by considering the input and output of first layer as:

$$F(\mathbf{X}_l) = F\left(\parallel_{l'=0}^{l'=l-1} F(\mathbf{X}_{l'})\right) \quad for \quad l \geq 1 \quad and \quad l = \{1, 2, \dots, L\}, \quad (5.3)$$

where \parallel is the concatenation operation. For initialization $F(\mathbf{X}_{-1})$ and $F(\mathbf{X}_0)$ are considered as ϕ and \mathbf{X}_l , respectively, which ϕ is an empty set and there are L layers inside each block.

The decoder part of the CNN consists of three dense blocks and two transition layers. The decoder

transition layers can be *average pooling* or *max pooling* and decrease the size of the image by half. In the encoder part, we have same architecture as decoder part except that the transition layers are bilinear interpolation (i.e., unpooling). Each of the decoder transition doubles the size of the feature maps and at the end of this part, we obtain features maps as the same size as input images. Finally, the output of the decoder is passed through a convolution and softmax to produce the probability map. *ADAM optimizer* with a learning rate of 0.0001 is selected for training and *Cross Entropy* is used as a loss function. The other hyperparameters of network such as number of filters, filter heights, and widths for each layer are discussed in next section.

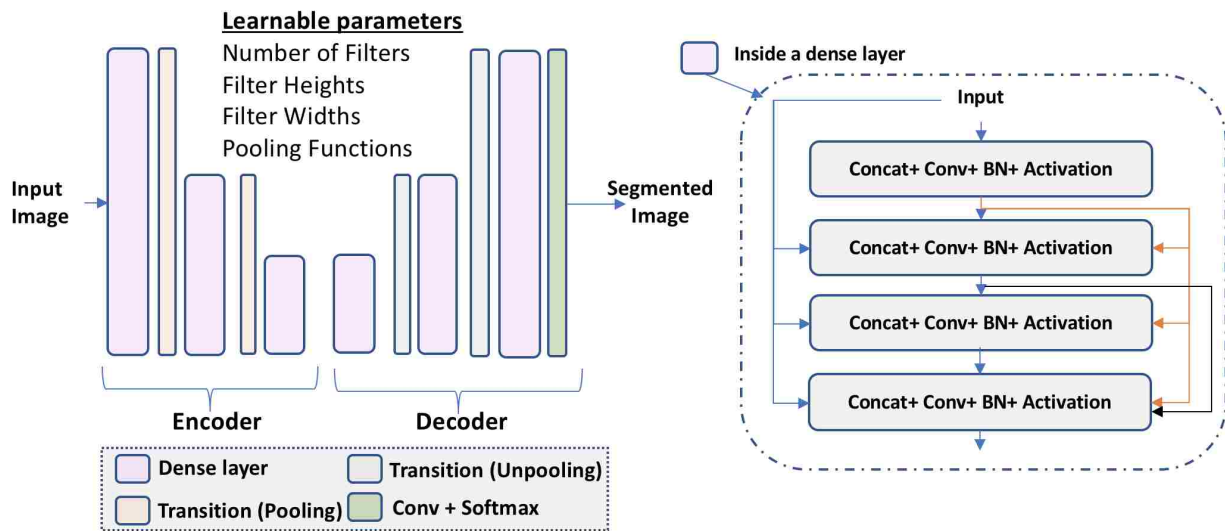


Figure 5.2: Details of the baseline architecture. We combine encoder-decoder based segmentation network with densely connected architecture as a novel segmentation network, which has less parameters to tune and more accurate. Concat: concatenation, BN: batch normalization, and conv: convolution.

5.4 Learnable Hyperparameters

Following hyperparameters are learned automatically with our proposed architecture search algorithm: number of filters, filter height, and filter width for each layer. Additionally, type of pooling

layer was considered as learnable hyperparameters in our setting. Totally, there are 76 parameters (N) to be learned: 3 parameters (filter size, height, and weight) for each of 25 layers (last layer has fixed number of filters), and 2 additional hyperparameters (average or max pooling) for down-sampling layers. More specifically:

- **Number of filters:** The number of filters (NF) for each layer is chosen from function $y_{NF} = 16x_{NF} + 16$ which $x_{NF} = \{1, 2, \dots, 12\}$.
- **Filter height:** The filter height (FH) for each layer is chosen from function $y_{FH} = 2x_{FH} + 1$ which $x_{FH} = \{0, 1, \dots, 5\}$.
- **Filter width:** The filter width (FW) for each layer is chosen from function $y_{FW} = 2x_{FW} + 1$ which $x_{FW} = \{0, 1, \dots, 5\}$.
- **Pooling functions:** The pooling layer is chosen from function $y_{pooling} = x_{pooling}$ which $x_{pooling} = \{0, 1\}$ which '0' represents max pooling and '1' represents average pooling.

The number of generated perturbation p is considered as 42 (experimentally) and in order to decrease the computational cost, each network is trained for 50 epochs, which is adequate to determine a stable reward for the network. The average of dice index for the last 5 epochs on the held-out validation set is considered as reward for the reinforcement learning.

5.5 Experiments and Results

Dataset: For investigating the performance of proposed method, a dataset from Automatic Cardiac Diagnosis Challenge (ACDC-MICCAI Workshop 2017) are used [2]. These data set are composed of 150 cine-MR images including 30 normal cases, 30 patients with myocardium infarction, 30 patients with dilated cardiomyopathy, 30 patients with hypertrophic cardiomyopathy, and 30 patients

with abnormal RV. While 100 cine-MR images were used for training (80) and validation (20), the remaining 50 images were used for testing with online evaluation. Also, for fair validation in training procedure, four subjects from each category have been chosen. The binary mask ground truth of three substructures were provided by challenge organizers for training: right ventricle (RV), myocardium of left ventricle(Myo.), and left ventricle (LV) at two time point of end-systole (ES) and end-diastole (ED).

The MR images were obtained using two MRI scanners of different magnetic strengths (1.5T and 3.0T). Cine MR images were acquired in breath hold (and gating) with a SSFP sequence in short axis. Particularly, a series of short axis slices cover the LV from the base to the apex, with a thickness of 5 mm (or sometimes 8 mm) and sometimes an inter-slice gap of 5 mm. The spatial resolution goes from 1.37 to 1.68 $mm^2/pixel$ and 28 to 40 volumes cover completely or partially the cardiac cycle. An example from this data set is shown in Figure 5.3.

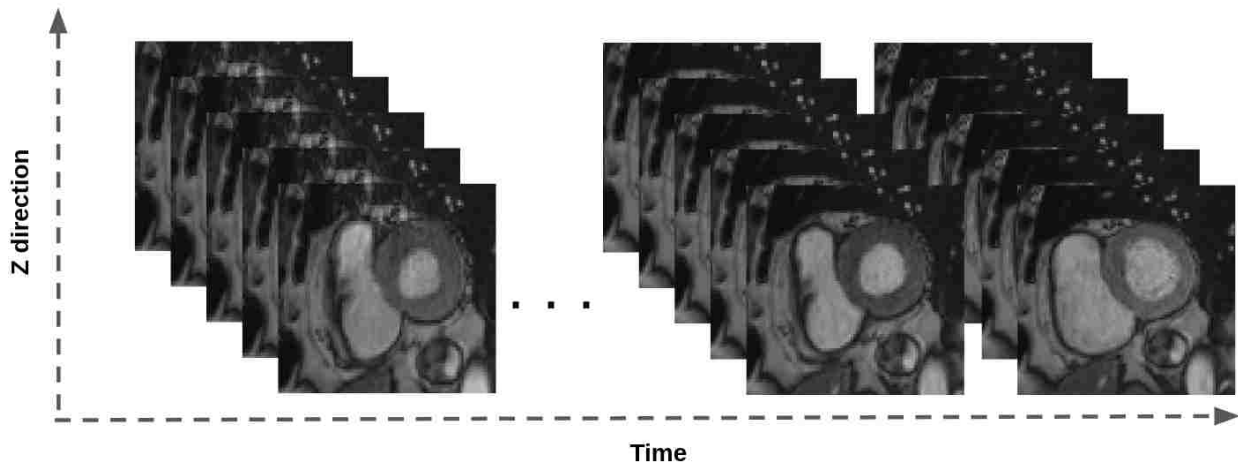


Figure 5.3: An example for cine-MR (4D) cardiac image from ACDC data set ([2]).

Implementation details: We calculated dice index (DI) and Hausdorff distance (HD) to evaluate segmentation accuracy (blind evaluation through challenge web page on the test data). The quantitative results for LV, RV, and Myo as well as mean accuracy (Ave.) are shown in Table 5.2. Twenty

images were randomly selected out of the 100 training images as validation set. After finding optimized hyperparameters, the network with learned hyperparameters was trained fully with the augmented data. The augmentation was done with in-plane rotation and scaling (Table 5.1). The number of images increased by factor of five after augmentation.

Table 5.1: Data augmentation

Data augmentation	
Methods	Parameters
Rotation	$k \times 45, k \in [-1, 1]$
Scale	$\epsilon[1.3, 1.5]$
Training Images	
# of Images	Image size
8470	200×200

Post-Processing: To have a fair comparison with other segmentation methods, which often use post-processing for improving their segmentation results, we also applied post-processing to refine (improve) the overall segmentation results of all compared methods. We presented our results with and without post-processing in Table 5.2. Briefly, a 3D fully connected Conditional Random Field (CRF) method was used to refine the segmentation results, taking only a few additional milliseconds. The output probability map of the CNN is used as unary potential and a Gaussian function was used as pairwise potential. Finally, a connected component analysis was applied for further removal of isolated points.

Comparison to other methods: The performances of the proposed segmentation algorithm in comparison with state-of-the-art methods are summarized in Table 5.2. The DenseCNN (with *ReLU* and with *Swish*) is the densely connected encoder-decoder CNN designed by experts, and its use in segmentation tasks recently appeared in some few applications, but never used for cardiac segmentation before. Filter sizes were all set to 3×3 in DenseCNN and growth rates were considered as 32, 64, 128, 128, 64, and 32 for each block from beginning to the end of the network,

respectively. Also, the average pooling is chosen as the pooling layer. These values were all found after trial-error and empirical experiences, guided by expert opinions as dominant in this field. The 2D U-Net, as one of the state of the arts, is the original implementation of the U-Net architecture proposed by Ronneberger et al. in [14] was used for comparison too. Although we apply our algorithm into 2D setting for efficiency purpose, one can apply it to 3D architectures once memory and other hardware constraints are solved. The details of the learned architecture with the proposed method is shown in Figure 5.4.

We obtained the final architecture design in 10 days of continuous training of a workstation with 15 GPUs (Titan X). Unlike the common CNN architecture designs (expert approach), which requires months or even years of trial-and-error and experience guided search, the proposed search algorithm found optimal (or near-optimal) segmentation results compared to the state of the art segmentation architectures within days.

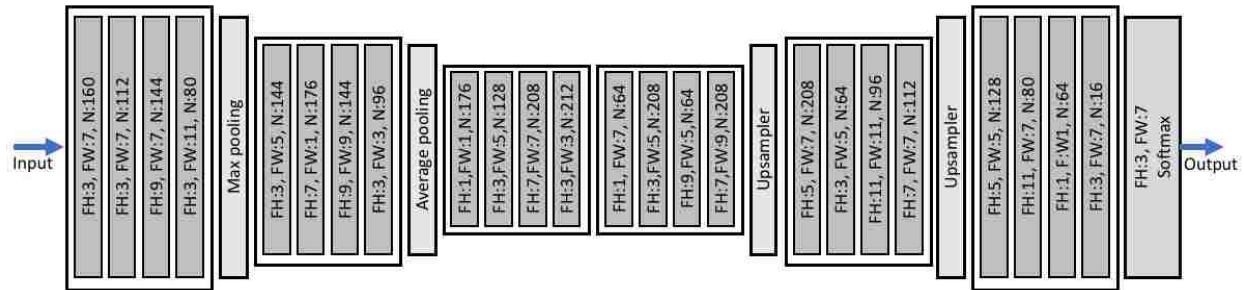


Figure 5.4: Details of the optimally learned architecture by the proposed method. Note that connections among layers inside of each block are same as dense layers.

5.6 Discussions and Conclusion Remarks

We proposed a new deep network architecture to automatically segment cardiac cine MR images. Our architecture design was fully automatic and based on policy gradient reinforcement learning.

Table 5.2: DI and HD for all methods and substructures.

Methods		2D-UNET	DenseCNN (ReLU)	DenseCNN	Proposed	Proposed+CRF
DI	LV	0.904	0.913	0.922	0.921	0.928
	RV	0.868	0.826	0.834	0.857	0.868
	MYO	0.847	0.832	0.845	0.838	0.849
	Ave.	0.873	0.857	0.867	0.872	0.882
HD (mm)	LV	9.670	9.15	8.937	8.99	8.90
	RV	14.37	16.35	16.31	14.27	14.13
	MYO	12.13	11.32	11.28	10.70	10.66
	Ave.	12.06	12.27	13.02	11.32	11.23

After baseline network was structured based on densely connected encoder-decoder network, the policy gradient algorithm automatically searched the hyperparameters of this network, achieving the state of the art results. Note that our hypothesis was to show that it was possible to design CNN automatically for medical image segmentation with similar or better performance in accuracy, and much better in efficiency. It is because expert-design networks require extensive trial-and-error experiments and may take even years to design. Our study has opened a new venue for designing a segmentation engine within a short period of time. Our study has some limitations due to its proof of concept nature. One interesting way to extend the proposed model will be to learn hyperparameters conditionally in each layer (unlike independent assumption of the layers). With the availability of more hardware sources, one may explore many more hyperparameters, such as ability to put more layers than basic model, defining skip-connections, and exploring different activation functions instead of ReLU and other default ones. One may also avoid increasing search space and still perform a good architecture design automatically by choosing the base-architecture more powerful ones such as the SegCaps (i.e., segmentation capsules) [51].

CHAPTER 6: INCORPORATING PRIOR KNOWLEDGE AND LABEL NOISE IN MEDICAL IMAGE SEGMENTATION

- **Aliasghar Mortazi**, Naji Khosravan, Drew A. Torigian, Sila Kurugol, Ulas Bagci (2019) A Weakly Supervised Segmentation by A Deep Geodesic Prior. In: Shi Y., Suk HI., Liu M. (eds) Machine Learning in Medical Imaging. (MLMI), MICCAI 2019. Lecture Notes in Computer Science, Springer, Cham

6.1 Overview

We hypothesize that, if modeled correctly, prior information can lead to a more robust segmentation even when the labels are noisy (i.e., labels annotated by non-experts). To test this hypothesis, we propose a novel method for learning the prior from the geodesic maps of multiple objects. Then, an AE-like network is used to generate the original binary images from their corresponding geodesic maps. Finally, the features from the trained AE are used as a prior to be integrated into the segmentor for better guidance and performance improvement.

6.2 Proposed Weakly-Supervised Approach

Our framework consists of two main components: (1) the segmentation network (or *segmentor* in short), and (2) the geodesic prior learning network. While our segmentation network assigns a class label to each pixel of the input image, our second network (AE) learns a prior from geodesic maps, generated for each object of interest (for multiple objects). We anticipate (and show later in the

results section) that incorporating a well-designed prior into the segmentation network improves its performance, especially in the presence of inaccurate labels.

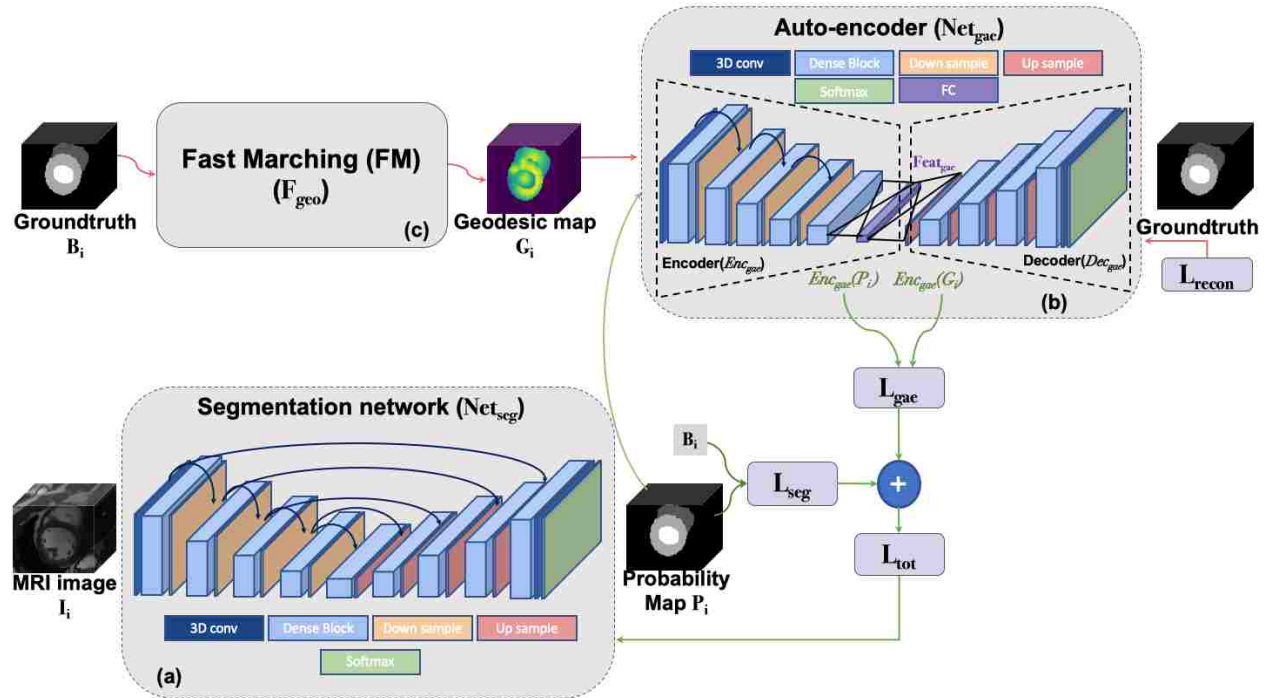


Figure 6.1: The proposed framework has two main components: (1) segmentation network: assigns a class label to each pixel, and, (2) AE which learns a prior from geodesic maps. Green arrows show the flow of training of the segmentor and red arrows show the flow of training of GAE. Note that in test phase only segmentor is used. Also, the in our method the noisy annotations are used for whole training process.

The overview of our approach is illustrated in Figure 6.1. The segmentation network (Net_{seg}) (Figure 6.1(a)) is an encoder-decoder architecture with 3D kernel convolutions. We utilize skip connections in the form of dense connection throughout this network [52]. For the prior learning network (AE), noisy annotations (or binary ground truths) are used to generate geodesic maps (Figure 6.1(c)). Then, the geodesic auto-encoder (GAE) is designed to generate binary ground truths from geodesic maps. Once trained, GAE can be used to calculate two sets of bottleneck features: features resulted from feeding the geodesic map to GAE, and features resulted from feeding the

corresponding Net_{seg} 's output probability map to GAE. Finally, the distance between these two feature vectors are used to form an extra term in the loss function of the Net_{seg} (Figure 6.1).

We define the segmentation network as a function, mapping a gray-scale 3D input image $I_i (I_i \in \mathbb{R}^3)$ into a probability map P_i , ($i \in \{1, 2, \dots, N\}$), N being number of 3D images:

$$P_i = Net_{seg}(I_i, \theta_{seg}), \quad (6.1)$$

where θ_{seg} are the parameters of Net_{seg} , trained to minimize \mathcal{L}_{tot} defined in Equation 6.2. A geodesic map, on the other hand, is generated from ground truth binary images B_i as $G_i = F_{geo}(B_i)$ and then a GAE network (Net_{gae}) is trained to generate binary image from this corresponding geodesic map (explained in Section 6.4). The GAE consists of an encoder, a fully connected (FC) layer, and a decoder. The encoder and FC layers are the feature extraction (Enc_{gae}) parts, mapping the input geodesic map to the feature vector $Feat_{gae} = Enc_{gae}(G_i)$ of length L_{feat} . The decoder (Dec_{gae}) reconstructs the corresponding binary image(s) from the $Feat_{gae}$. Hence, the geodesic network can be formulated as:

$$\hat{B}_i = Net_{gae}(G_i, \theta_{gae}) = Dec_{gae}(Enc_{gae}(G_i, \theta_{enc}), \theta_{dec}), \quad (6.2)$$

where θ_{gae} are the parameters of Net_{gae} , trained to minimize the binary map reconstruction loss \mathcal{L}_{recon} in Net_{gae} . \mathcal{L}_{recon} is a cross-entropy loss between ground-truth and Net_{gae} 's output and $\theta_{gae} = \theta_{enc} \cup \theta_{dec}$. Since Net_{gae} is designed to learn the relation between geodesic maps and their corresponding binary maps, $Feat_{gae}$ contains high-level features inferred from shapes and texture of the objects of interests. This encoded knowledge can be used as an extra term of supervision for better training of Net_{seg} . For each training sample i , we calculate a loss function $\mathcal{L}_{gae}(Enc_{gae}(P_i), Enc_{gae}(G_i))$ to be back-propagated into the segmentation network along with

loss of the segmentor itself. The total loss function of the segmentator is then represented as:

$$\mathcal{L}_{tot} = \sum_i \mathcal{L}_{seg}(P_i, B_i) + \mathcal{L}_{gae}(Enc_{gae}(P_i), Enc_{gae}(G_i)). \quad (6.3)$$

We first train θ_{gae} with \mathcal{L}_{recon} . Once trained, θ_{seg} is updated with the loss function \mathcal{L}_{tot} , while θ_{gae} are fixed.

6.3 Network Architecture for Segmentation

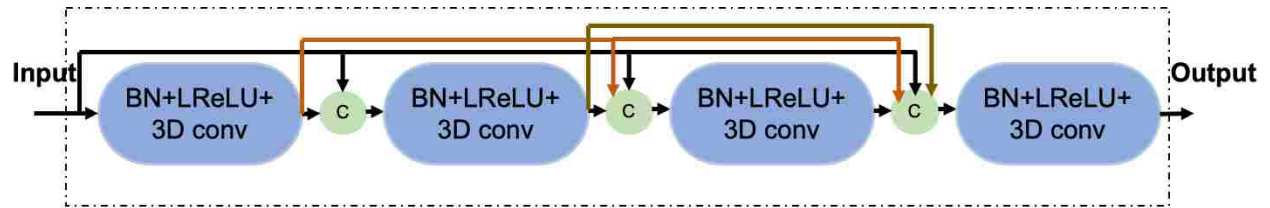


Figure 6.2: Dense Block (DB) content. C : concatenation operation, BN : Batch normalization, $LReLU$: Leaky ReLu activation, and $3D\ conv$: 3D convolution with a $3 \times 3 \times 3$ filter.

We extend fully convolutional dense nets, called *Tiramisu* [21], from 2D to fully 3D. The details of the adapted network are shown in Figure 6.1(a). The encoder and decoder include four dense blocks, each. Within dense blocks, there are four 3D convolution layers followed by a *Batch Normalization* (BN) layer with *Leaky ReLu* nonlinear activation. The size of the convolution kernels are set to $3 \times 3 \times 3$ in all convolution layers (Figure 6.2). The number of filters in the first convolution layer and the *growth rate* are set to 16 for an optimal performance after extensive explorations. The number of output filters in a dense block is $N_f(\mathbf{X}_l) = N_f(\parallel_{l'=0}^{l-1} N_f(\mathbf{X}_{l'}))$, where $l = \{1, 2, \dots, L\}$ and \parallel is the concatenation.

The encoder includes four 3D max pooling operations as transition layers after each dense block. The pooling operation is set to downsample the input size by 2 in x - y plane. Downsampling is not

applied to z direction due to its low resolution (but could be applied for other settings). Similarly, the decoder includes four up-samplers (using bilinear interpolation) as transition layers. Each up-sampler is designed to double the size of its input. Finally, the last layer contains a convolution layer following by a *softmax* function to introduce a notion of probability map in the output. We use *Adam* optimizer to minimize the \mathcal{L}_{tot} in Net_{seg} . \mathcal{L}_{seg} and \mathcal{L}_{gae} are designed with *Cross Entropy* and *Mean Square Error* functions, respectively.

6.4 Learning a Geodesic Prior

Most existing literature related to prior incorporation into segmentation utilize *accurate* binary labels for extracting shape information [39, 40]. Unlike the mainstream studies, we propose to use geodesic maps to increase robustness of the priors when dealing with *noisy* labels, which has never been done before. This approach can particularly be beneficial when the object has complex boundary information to be delineated. In this study, geodesic maps are generated from labels, regardless of being noisy or clean. We expect the proposed geodesic map to capture more information than conventional shape priors.

For each object in our images, we compute an independent geodesic distance map from its binary map B_i by using the Fast Marching (FM) approach [53]. FM is a numerical method to solve boundary value problems of the Eikonal as:

$$\begin{cases} F(x)|\Delta T(x)| = 1, & \forall x \notin S, \\ F(x) = 0, & \forall x \in S, \end{cases} \quad (6.4)$$

describing the evolution of a contour as a function of time, $T(x)$, with the speed of $F(x)$ in the normal direction at a point x on the propagating surface starting from the zero-level S . With a

specified speed, $F(x)$, the time when the contour crosses point x can be computed by solving equation 6.4. In this setting, the special case of $F(x) = 1$ gives the signed distance of every point x from S . In our case, since we have multiple objects (i.e., 3 objects: LV, RV, Myocardium), we defined S as the center of mass of the all closed objects. In our experimental setup, we have also an object with non-Jordan surface (i.e., Myocardium, having donuts shape). In order to include such objects in the geodesic computation, we simply define the the skeleton of the shape and the distances of all the points within each object are computed from their zero line contours S . These maps (obtained from each object) are combined in n -channels (i.e., 3 in our experiments: LV, RV, Mayo) and fed into the auto-encoder as described below.

The proposed AE architecture (Net_{gae}), for learning prior information, is illustrated in Figure 6.1(b). This architecture is very similar to the segmentor with a FC layer in the middle (instead of a convolution layer) to generate deep geodesic features. Also, in order to increase the robustness of these features, there is no skip connections from encoder to decoder. Both encoder and decoders include four DBs and the filters size in each convolution layer was set to $3 \times 3 \times 3$. The growth rate for the encoder part is set to 16 (empirically) as in the segmentor and *ADAM* optimizer is used to minimize \mathcal{L}_{recons} (*Cross Entropy* loss).

6.5 Experiments and Results

To show the robustness of our algorithm and its performance on noisy labels, we ran all the experiments on both expert as well as two levels of noise in the labels (L1 and L2). We reported Dice Index (DI) and Hausdorff distance (HD) in Table 7.2. Also, in cases where we were dealing with noisy labels, there was an upper bound for the performance of the networks. This upper bound was due to lack of information in the presence of noise. To have a sense of this upper bound, for the sake of a more extensive and fair comparison, DI and HD of generated noisy labels with respect

to clean ground truths are reported in this table (*Upper boundary* columns). Higher DI and lower HD indicate a superior segmentation performance. While training of the networks were done using weak/noisy labels, validation was performed on expert/accurate labels.

Data set: Data set which used for evaluating proposed method in this chapter is data set from ACDC challenge/workshop in MICCAI 2017 and it is same data set used in Section 5.5 in Chapter 5. Please refer to that Section for more details about data set.

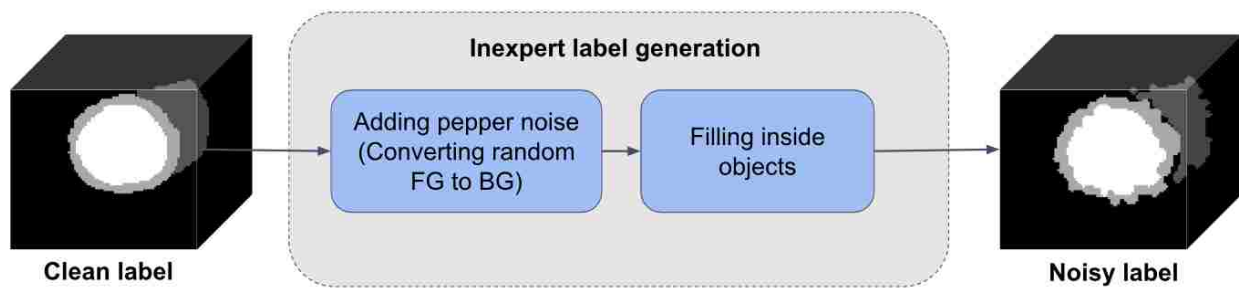


Figure 6.3: Generating noisy labels. Binary images went through a two-step process of adding pepper noise and filling inside object which makes the boundaries inaccurate.

Generating noisy labels: The current annotations at ES and ED in ACDC dataset are considered as expert annotations (as clearly defined by the challenge organizers). Usually, the inexpert annotations include some under-segmentation and/or over-segmentation. This is due to lack of naive annotator’s knowledge in finding the edges. Thus, in order to mimic such inexpert annotations (weak labels), we manipulated the ground truths as follows: first we obtained the outer shell of each object by applying *erosion* to the binary image and then calculate the difference between the original binary image and eroded one. Then, salt-pepper noises were added to each object’s binary shell randomly and *filling* was applied to the shell. This process effected *only* edges of the objects without changing the background, resulting in a shape with distorted boundaries. Finally, eroded binary edges was added to the distorted shell. A sample of weak labels vs. expert labels is shown in Figure 6.3. Participating radiologists confirmed the weak labels through visual evaluations.

Baseline models for comparisons: We have conducted several baseline architectures in order to show the strength of our proposed method. First, the segmentor was trained only with \mathcal{L}_{seg} and without using prior information with both of the inexpert and expert annotations. Also, in order to illustrate the advantage of using the geodesic maps instead of binary maps in modeling shape information, the segmentation network was trained with the prior information obtained from binary AE (*segmentation + binary labels shape*). In this baseline, the AE was trained to reconstruct binary map in its output from its corresponding binary input map (instead of geodesic in our method). The results of this baseline (*Binary Prior*) and our proposed method (*Geodesic Prior*) for inexpert and expert annotations are reported in Table 7.2.

Table 6.1: DI and HD are reported for both expert and inexpert labels.

Labels		Expert Labels			Inexpert Labels(L1)				Inexpert Labels(L2)			
Methods		Seg. Net.	Binary Prior	Geodesic Prior	Seg. Net.	Binary Prior	Geodesic Prior	Upper boundary	Seg. Net.	Binary Prior	Geodesic Prior	Upper boundary
DI	LV	0.854	0.879	0.885	0.831	0.867	0.878	0.880	0.811	0.856	0.873	0.869
	RV	0.803	0.851	0.847	0.791	0.828	0.836	0.835	0.762	0.811	0.831	0.824
	MYO	0.771	0.816	0.826	0.762	0.798	0.810	0.812	0.751	0.780	0.809	0.801
	Ave.	0.809	0.849	0.853	0.795	0.831	0.841	0.842	0.775	0.816	0.838	0.831
HD (mm)	LV	14.79	10.08	10.14	15.87	12.57	11.73	11.75	14.89	13.03	11.65	11.81
	RV	17.77	13.77	13.45	18.89	17.01	15.14	15.15	17.57	17.53	15.44	15.76
	MYO	16.53	12.58	12.04	16.91	13.95	12.73	12.70	16.47	15.12	12.88	13.12
	Ave.	16.36	12.14	11.88	17.22	14.51	13.20	13.20	15.64	15.23	13.32	13.56

Implementation Details: As a pre-processing step, we applied the anisotropic filtering to reduce noise from MRI, histogram matching to standardize MRI intensities, and all images were resized to $200 \times 200 \times 10$ by using B-spline interpolation. First the *GAE* was trained (early-stopping) and then the deep geodesic features ($Feat_{gae}$) were extracted from training data. Then, during training of Net_{seg} the output probability maps of the Net_{seg} were passed through Enc_{gae} and then the loss (\mathcal{L}_{gae}) between two feature vector was calculated and back-propagated through the Net_{seg} . Finally, Conditional Random Field is used for post-processing. we used 80 MR images for training, the 20 images were used as validation, and the 50 images were used for test (with online evaluation). Also, we have used NVIDIA Quadro P6000 GPU for training the networks.

6.6 Discussions and Conclusion Remarks

In this study, we propose a novel framework incorporating a deep geodesic prior information into the segmentation framework. Our AE network is capable of learning high-level features from generated geodesic maps for multiple objects. We show that our proposed approach outperforms the state-of-the-art methods both on clean and noisy labels with several key advantages. First, incorporating prior information improves segmentation results even with imperfect ground truths. Second, more specifically, shape prior is shown to be useful both in Euclidean and Geodesic distance based evaluations, and geodesic priors are shown to be more accurate than the former. Furthermore, the proposed network is capable of performing fully 3D image segmentation unlike most 2D methods in the literature, and can handle multiple objects too. One may argue to obtain inexpert annotations directly from inexpert annotators for more realistic evaluations, but the regulations for medical imaging data sharing in general (and ACDC challenge in particular) didn't allow to get such annotations. Our future work will comprehensively test our hypothesis for different components of our present study such as different clinical imaging problem, large number of inexpert annotators, and inspecting the results based on object type and size.

CHAPTER 7: CHOOSING THE RIGHT OPTIMIZERS FOR MEDICAL IMAGE SEGMENTATION

7.1 Overview

In this chapter, we study how to design (and choose) right optimizers for deep learning based medical image segmentation problems. In the literature, the most of the optimization methods have been designed, tested, and evaluated for classification problems where prediction space is limited to hundred of cases typically. However, in medical image segmentation problems, the prediction of parameter space is much more larger because each pixel is predicted for a label; hence, conventional choice of optimizer may not necessarily be right for the segmentation problem at hand. There is a strong need to tune optimizers for segmentation problems. While adaptive optimizers such as ADAM and ADAGrad are more popular than other conventional (non-adaptive) stochastic gradient descent (SGD) based approaches (due to their fast convergence), recent studies show that adaptive optimizers can suffer from poor generalization. In this chapter, we first show that adaptive optimizers can overfit faster compared to classical SGD optimizers and resulting in poor generalization in segmentation problems. Second, to address this limitation of adaptive optimizers, we propose a new optimization method based on the Nesterov function, which is computationally cheaper and it has better generalization capabilities than adaptive optimizers. We have performed our segmentation experiments by using cardiac imaging dataset from ACDC challenge from MICCAI 2017. Our extensive evaluations showed that the proposed optimizer can get better results than other optimizers in deep learning literature with similar or less computational burden in multi-object segmentation settings. Our proposed optimizer based baseline segmentation methods obtained more than 2% improvement in dice metric in comparison with the state-of-the-art optimizers.

7.2 Introduction

In segmentation problems, deep networks are often taken from classification networks and they are re-designed for handling pixel level classification instead of image level labeling. While doing that researchers develop different loss functions, make new architectural designs and do pruning and other optimization strategies to adjust the problem settings in favor of pixel level classification (segmentation) than detection or image classification (e.g., diagnosis). However, to date there has been no in-depth exploration about how optimization methods are affecting the segmentation results and which optimization methods are the right choice for the particular problem at hand.

Optimization methods are the key elements in training neural network. There has been controversial results in the literature about the characteristics of available optimization methods. Majority of the neural network optimizers have been tested and evaluated for the classification tasks such as image classification which usually dimension of output prediction (i.e number of classes are around 1000), which is significantly different from segmentation problem with output prediction dimension with same size as input (i.e., 200×200 as in our experiments). Hence, these differences between classification and segmentation problems imply that a different investigation and method may be needed for optimization. In this chapter, for the first time in the literature we investigate the effects of different neural network optimizers for medical image segmentation problems and introduce a new simple optimizer to address the current drawbacks of conventional optimizers.

Non-adaptive vs. adaptive optimizers: One of the dominant optimization algorithms is stochastic gradient descent (SGD), which is simple and performing well across many applications. However, it has the disadvantage of scaling the gradient uniformly (i.e., the same way) in all directions (for each parameter). Another challenge is to choose appropriate value for learning rate (LR). Since LR is fixed in SGD based approaches, it is critical to set it appropriately as it can directly affect both convergence speed and accuracy of neural networks. There have been several works

introduced to address this problem by allowing to change LR during training, called "adaptive optimizers". Based on the history of change in gradients, LR is adapted in each iterations. Examples of such methods consists of ADAM [16], ADAGrad [15], and RMSProp [54]. Adaptive optimizers provide, in general, faster training experiences, thus lead to widespread use in deep learning applications.

The same story applies for the optimizers with *momentum*. Momentum optimizers [42] was introduced to speed up the convergence by considering the changes in last iteration with a multiplier which called "momentum". Again, choosing a proper value for momentum rate (MR) was challenging at first, then some adaptive optimizers such as Adaptive Momentum Optimizer [16] (ADAM) was introduced to dynamically change the MR in addition to changing LR adaptively. For the past few years, adaptive optimizers such as ADAGrad [15] and ADAM [16] dominated the deep learning field due to their fast convergence.

Despite their popularity, adaptive optimizers may converge to different minima points in comparison with classical SGD approaches. In other words, they have likely worse generalization ability and out-of-sample behavior than non-adaptive optimizers as evidenced by increasing number of studies published recently [17, 18, 19]. Towards a better generalization ability, researchers turned back to original SGD approaches but with new attempts to solve convergence speed. For instance, *YellowFin* optimizer [19] proved that properly hand tuned LR and MR obtained better results than ADAM. Although it was a proof-of-concept study showing evidence for counter-intuitive idea of non-adaptive methods, in practical applications hand-tuning those rates is challenging and time consuming. In another attempt, a cyclic learning rate (CLR) was introduced in [18] to change the learning rate according to a cycle (i.e triangle or Gaussian), proposing a practical solution to hand-tuning requirement without manual burden. The only disadvantage of the CLR is that a fixed momentum rate may limit the search state, and may fail to find optimal solution.

Summary of our contributions: By motivated from [18], herein we introduce a new variant of CLR, called *Cyclic Learning/Momentum Rate (CLMR)*, in which both learning rate and momentum rate are determined in a cyclic manner during training. This new optimizer has two advantages over adaptive optimizers: (1) it is computationally much cheaper than their adaptive counterparts. (2), it generalizes better than adaptive optimizers. Furthermore, CLMR leads to better results than its conventional baselines such as SGD and CLR. Lastly, we investigate the effect of changing the frequency of cyclic function in training and generalization and suggest the optimum frequency values. We also systematically explore several other optimizers commonly used in medical image segmentations, and compare their performance as well as generalization ability in multi-object segmentation settings by using cardiac MRI images (cine-MRI).

The rest of the chapter is organized as follows. In section 7.3, we introduce the background information for neural network optimizers, their notations, and use in medical image segmentation. In section 7.4, we give the details of the proposed method and network architectures that segmentation experiments have been conducted. Experimental results are summarized in Section 7.5. Section 7.6 concludes the paper with discussions and future works.

7.3 Background

7.3.1 Current Optimization Methods in Deep Learning

Optimizing parameters of neural networks have been challenging from beginning due to huge number of parameters need to be trained. Generally, the current optimizers can be categorized regarding having a fixed LR/MR, or having adaptive LR/MR, or having cyclic/changing LR/MR.

7.3.1.1 Optimizers with fixed LR/MR

Gradient Descent (GD), SGD and Mini-batch GD were first optimizers used for training neural networks. The updating rule for these optimizers include only the value of last iteration as it is shown in Eq. 7.1. Choosing appropriate value for learning rate is challenging in these optimizers, since if LR is very small, then convergence is very slow; and if LR is very high, the optimizer will oscillate around global minima (instead of converging):

$$\theta_i = \theta_{i-1} - \alpha \nabla_{\theta_i} J(\theta_i), \quad (7.1)$$

where θ is the network parameters, α is the learning rate, J is a cost function for minimization (function of θ , X (input), and Y (labels)). The equation 7.1 can be considered as an updating rule for GD, SGD, and mini-batch GD by choosing X and Y as all samples, a single sample, and a batch of samples, respectively.

The Momentum optimizer [42] was introduced to speed up the convergence of optimization by considering the value of all past iterations with a rate which called *momentum*. The update rule for this optimizer is:

$$\theta_i = \theta_{i-1} - \alpha \nabla_{\theta_i} J(\theta_i) - \beta(\theta_{i-1} - \theta_{i-2}), \quad (7.2)$$

in which β is momentum rate (MR). As it is shown in equation 7.2, the past iterations don't play any role in cost function and cost function is only calculated with respect to current iteration. Also, similar to LR, choosing a proper value for momentum rate is again challenging and it has correlation with chosen value for LR too. Then, *Nesterov accelerated gradient* [43] (NAG) was introduced to accelerate the convergence by including information of previous iterations in

calculating gradient of cost function as shown in following equation:

$$\theta_i = \theta_{i-1} - \alpha \nabla_{\theta_i} J(\theta_i - \beta(\theta_{i-1} - \theta_{i-2})) - \beta(\theta_{i-1} - \theta_{i-2}) \quad (7.3)$$

The NAG optimizer usually performs better in convergence speed and accuracy in comparison with other fixed LR/MR optimizers.

7.3.1.2 Optimizers with adaptive LR and MR

One of the major drawbacks of optimizers with fixed LR/MR is the lack the history of gradient of past iterations when adapting the LR and MR. *ADAGrad* [15] is one of fist adaptive LR optimizers used in deep learning area and it basically adapts the learning rate for each parameter in network by dividing gradient of each parameter by its sum of the squares of gradient as it is shown in following equation:

$$\theta_i = \theta_{i-1} - \alpha \frac{1}{\sqrt{G_i + \epsilon}} \circ \nabla_{\theta_i} J(\theta_i), \quad (7.4)$$

where G_i is a diagonal square matrix with the size of number of parameters and each element of its diagonal is equal to the sum of square of gradient of its corresponding parameter as follows:

$$G_i = \sum_{i=1}^I (\nabla_{\theta_i} J(\theta_i))^2, \quad (7.5)$$

in which i stands for current iteration. One of the drawbacks of *ADAGrad* is gradient vanishing due to accumulation of all past square gradients. *ADADelta* and *RMSProp* solve this problem by

considering a limited window for summing past square gradient (instead of all of them).

The most popular optimizer which is used in most of the deep learning application is called ADAPtive Momentum optimizer (ADAM)[16]. ADAM optimizer updating rule used past squared gradient (as scale) and also like momentum, it keeps exponentially decaying average of past gradient:

$$\theta_i = \theta_{i-1} - \alpha_i \frac{\beta_1 \nabla_{\theta} J(\theta_{i-2}) - (1 - \beta_1) \nabla_{\theta} J(\theta_{i-1})}{\sqrt{\beta_2 + \epsilon}} \circ \nabla_{\theta_i} J(\theta_i), \quad (7.6)$$

As mentioned before, one of the disadvantages of adaptive learning methods is its computational cost because they are required to calculate and keep all the past gradients and their squares to update next parameters. Also, the adaptive learning optimizer may converge to different minima point in comparison with fixed learning rate optimizers which is worse in generalization [17, 18, 19].

CLR was proposed to change the learning rate during training and doesn't need additional computational cost in comparison to classical SGD optimizers. It is based on the idea of fixed learning optimizer and changes global LR in a cyclic manner. Figure 7.3a. shows an example use of CLR.

7.4 Methods

Our goal is to find the right optimizer, for medical image segmentation problems under constraint of better generalization and low computational cost. We choose four different state-of-the-art architectures for segmentation which have been state-of-the-art architectures in the literature (with some modification with respect to our application) which are described as follows:

7.4.1 CNN Architectures

Encoder-Decoder Architecture: This architecture is simply consists of the encoder and decoder part as illustrated in Figure 7.1, **without** considering red skip connections. The filter size in all the layer are 3×3 and each encoder and decoder part include 5 *CNN blocks* and each *CNN blocks* consist of different number of layers as mentioned in Table 7.1. Also, the number of filters in each *CNN block* are a fixed number and they are mentioned in Table 7.1 for each layer. Each layer within the *CNN block* includes *Convolution+Batch normalization+ReLU* as activation function (*CBR*).

U-Net Architecture: This architecture is similar to the Encoder-Decoder architecture as illustrated in Figure 7.1 **with** red skip connections from encoder part to decoder part. The number of layers and filters for each block are mentioned in table 7.1.

DenseNet Architecture: Two different DenseNet architectures are used in this section. First, the architecture in Figure 7.1 with dense blocks (DBs) and skip connections is DenseNet_1. Then, in order to use higher growth rate (GR), in DenseNet_2, in the end of each block a convolution layer with kernel size of 1×1 is used to decrease number of its input filters by **C** rate, which **C** is equal 2 in this paper. The GR in DenseNet_2 increased to 24 (from 16 in DenseNet_1) while the number of parameters decreased (Table 7.1).The number of CBR layers and also number of parameters are mentioned in Table 7.1.

7.4.2 Dense Block

Within the DB, a concatenation operation is done for combining the feature maps (through direction (axis) of the channels) for the last three layers. So, if the input to l^{th} layer is \mathbf{X}_l , then the output

Table 7.1: Number of layers in each block of different architectures and number of parameters

	Enc_Dec	U-Net	DenseNet_1 (GR=16)	DenseNet_2 (GR=24)
Block 1	6 layers,#filters=32	6 layers,#filters=32	6 layers	6 layers
Block 2	8 layers,#filters=64	8 layers,#filters=64	8 layers	8 layers
Block 3	11 layers,#filters=128	11 layers,#filters=128	11 layers,	11 layers
Block 4	15 layers,#filters=256	15 layers,#filters=256	15 layers	15 layers
Block 5	20 layers,#filters=512	20 layers,#filters=512	20 layers	20 layers
Block 6	20 layers,#filters=512	20 layers,#filters=512	20 layers	20 layers
Block 7	15 layers,#filters=256	15 layers,#filters=256	15 layers	15 layers
Block 8	11 layers,#filters=128	11 layers,#filters=128	11 layers	11 layers
Block 9	8 layers,#filters=64	8 layers,#filters=64	8 layers	8 layers
Block 10	6 layers,#filters=32	6 layers,#filters=32	6 layers	6 layers
# of params (in millions):	77.5	79.1	7.7	8.8

of l^{th} layer can be shown by equation 7.7:

$$F(\mathbf{X}_l) = CBR(\mathbf{X}_l) \quad (7.7)$$

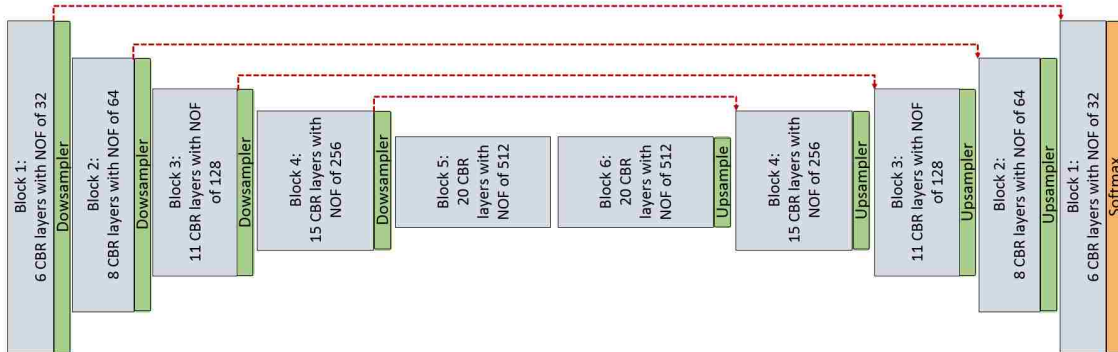


Figure 7.1: CNN Architecture is used for pixel-wise segmentation. The architecture with CNN blocks and without red skip connections is Encoder-Decoder architecture. The architecture with red skip connection and CNN blocks (Figure 7.2a) is U-Net and with Dense block (Figure 7.2b) is Tiramisu

Since we are doing concatenation before each layer (except first one), so the output of each layer can be calculated only by considering the input and output of first layer as following:

$$F(\mathbf{X}_l) = F\left(\overset{l'-l-1}{\underset{l'=0}{\frown}} F(\mathbf{X}_{l'})\right) \quad \text{for } l \geq 1, \quad (7.8)$$

and $l = \{1, 2, \dots, L\}$,

where \frown is defined as concatenation operation. For initialization $F(\mathbf{X}_{-1})$ and $F(\mathbf{X}_0)$ are considered as $\{\}$ and \mathbf{X}_1 , respectively, which $\{\}$ is an empty set and there are L layers inside of the block.

Let \mathbf{K}_{out} be the number of output features for each layer (channel out), and \mathbf{K}_{in_1} the number of input features for first layer (channel in). Then, the feature maps growth (channel out) for second, third, \dots , and L^{th} layer are $\mathbf{K}_{\text{out}} + \mathbf{K}_{\text{in}_1}$, $2\mathbf{K}_{\text{out}} + \mathbf{K}_{\text{in}_1}$, \dots , and $(L - 1)\mathbf{K}_{\text{out}} + \mathbf{K}_{\text{in}_1}$, respectively. The growth rate for the DB is the same as last layer.

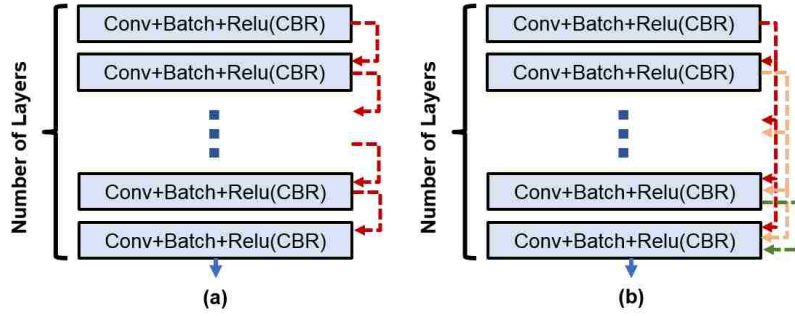


Figure 7.2: (a) CNN block used in Enc-Dec and UNet architectures, (b) Dense block used in Tiramisu architectures

7.4.3 Cyclic Learning/Momentum Rate Optimizer (CLMR)

As it is discussed by Smith in [18], having a cyclic learning rate (CLR) can be a more effective way than having the adaptive optimizer specifically in generalization. Smith [18] introduced a pre-defined cycle (such as triangle or Gaussian function) that a learning rate is changing according to that cycle. Here, we hypothesize (and show later in the results part) that having a cyclic momentum in Nesterov optimizer (Eq. 7.2) can lead to better accuracy in segmentation tasks. As a reminder, momentum in Eq. 7.2 is used to consider the past iterations by a coefficient called momentum. Hence, choosing the proper value for momentum were challenging from beginning and some adaptive methods such as *ADAM*, were trying to solve this issue with adjusting momentum rate adaptively, but they have their own challenges as mentioned before.

We propose to change the MR in similar way to LR. For this, we considered cyclic triangle function for both MR and LR (7.3). $cycle_{lr}$ and $cycle_{mr}$ determine the period of triangle function for LR and MR respectively, and defined as:

$$cycle_{lr} = C_{lr} \times It, \quad (7.9)$$

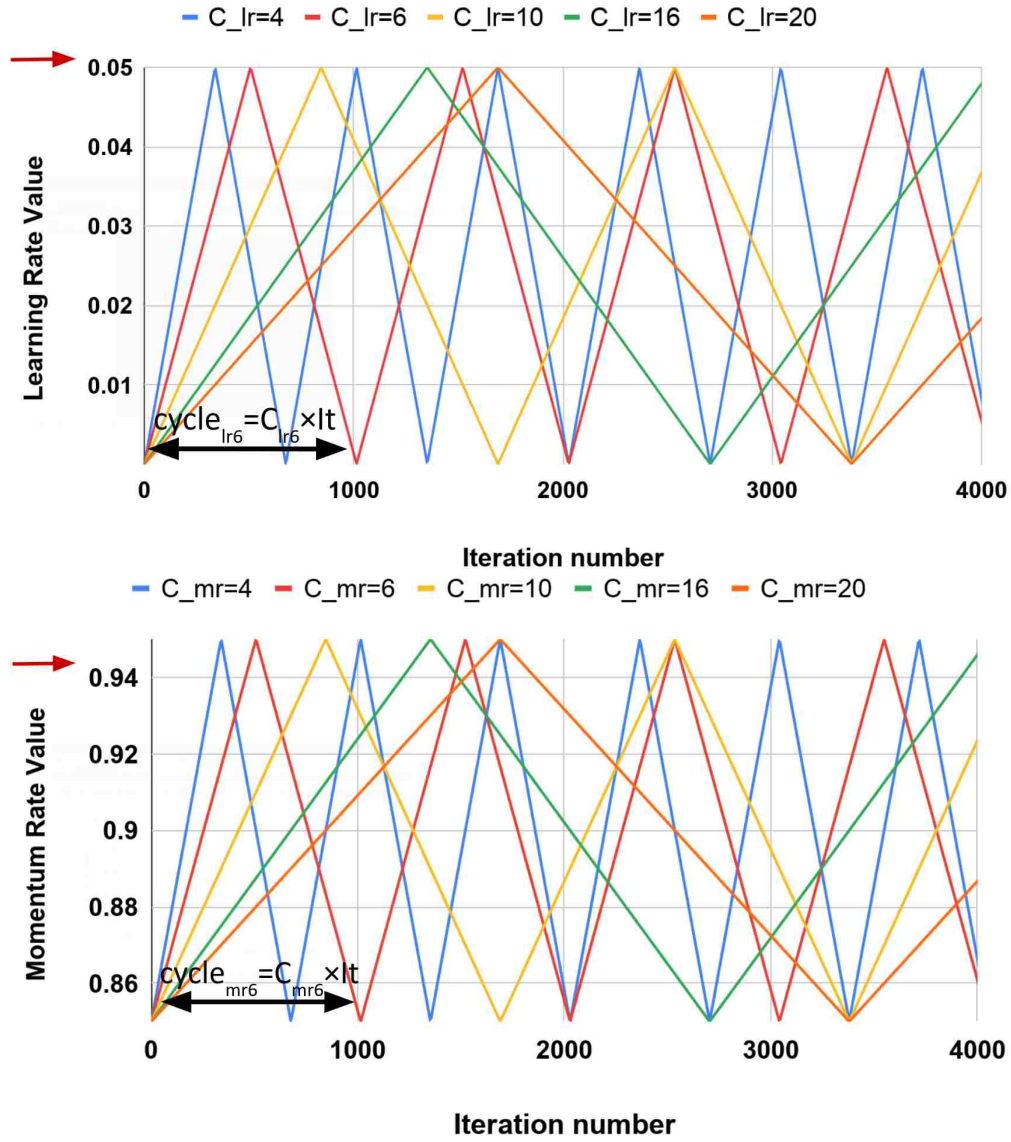


Figure 7.3: CLR and MLR functions. (a) Learning rate triangle function for different C_{lr} values with $min_{lr} = 0.0005$ and $max_{lr} = 0.05$. (b) Momentum rate triangle function for different C_{mr} values with $min_{mr} = 0.85$ and $max_{mr} = 0.95$.

$$cycle_{mr} = C_{mr} \times It, \quad (7.10)$$

where C_{lr} and C_{mr} are positive even integer numbers, It is number of iteration per each epoch.

In Figure 7.3a and 7.3b, the cyclic function for different values of C_{lr} and C_{mr} are illustrated. Moreover, the value of LR during whole training can be determined from equation 7.11:

$$LR = \begin{cases} 2 \times \frac{max_{lr} - min_{lr}}{C_{lr} \times It} \times i + min_{lr}, & \text{for } (N) \times cycle_{lr} \leq i < \frac{2N+1}{2} \times cycle_{lr}, \\ -2 \times \frac{max_{lr} - min_{lr}}{C_{lr} \times It} \times i + 2max_{lr} - min_{lr}, & \text{for } \frac{2N+1}{2} \times cycle_{lr} \leq i < (N+1) \times cycle_{lr}, \end{cases} \quad (7.11)$$

where max_{lr} and min_{lr} are maximum and minimum values of the LR function. i is the iteration indicator during whole training process and $i \in \{1, 2, \dots, It \times Ep\}$ which Ep is total number of epochs in training. And N is a set of natural number. The value of MR can be also determined from following equation:

$$MR = \begin{cases} 2 \times \frac{max_{mr} - min_{mr}}{C_{mr} \times It} \times i + min_{mr}, & \text{for } (N) \times cycle_{mr} \leq i < \frac{2N+1}{2} \times cycle_{mr}, \\ -2 \times \frac{max_{mr} - min_{mr}}{C_{mr} \times It} \times i + 2max_{mr} - min_{mr}, & \text{for } \frac{2N+1}{2} \times cycle_{mr} \leq i < (N+1) \times cycle_{mr}, \end{cases} \quad (7.12)$$

where max_{mr} and min_{mr} are maximum and minimum values of LR function.

Equations 7.11 and 7.12 are used to determine the values of LR and MR in each iteration during training. One of the challenges in using these cyclic LR and MR functions are determining the values of some variables in the equations including max_{lr} , min_{lr} , and C_{lr} for LR; and also max_{mr} , min_{mr} , and C_{mr} for MR. For finding the the max_{lr} and min_{lr} values, as it suggested in [18], one can run the networks with different LR values for few epochs and then these values are chosen according to how network accuracy is increasing and decreasing. However, the proposed solution

seems fine when only LR is changing that can not apply when both LR and MR are changing, since the changing value of one them can affect the accuracy of the other one. It means one needs to run huge number of networks in order to determine the optimum values of max_{lr} , min_{lr} , max_r , and min_{mr} which is not computationally feasible. Also, a heuristic method is suggested in [18] to find the best value of C_{lr} .

In this chapter, we propose a way to find best cyclic functions with minimum computational cost. We consider fixed values for max_{lr} , min_{lr} , max_r , and min_{mr} parameters and we choose these values in a way to cover a practical range of values for both LR and MR (illustrated in Figure 7.3). Then, we did a reasonable heuristic search for finding the appropriate amount of C_{lr} and C_{mr} from the values which are shown in Figure 7.3. Since, changing the values of C_{lr} and C_{mr} lead to change the value of LR and MR in different iterations, thus there is no need to find the optimum values for minimum and maximum. It is worthy to note that even different values of C_{lr} and C_{mr} can affect the results of each other and we did search in 2D space of C_{lr} and C_{mr} to find their optimum values.

7.5 Experiments and results

7.5.1 Data

Data set which used for evaluating proposed method in this chapter is data set from ACDC challenge\workshop in MICCAI 2017 and it is same data set used in Section 5.5 in Chapter 5. Please refer to that Section 5.5 for more details about data set.

7.5.2 Implementation details

The networks are trained for fixed number of epochs (100) and it has been make sure they are trained fully. All the images were resized to 200×200 in short axis by using b-spline interpolation. Then, as pre-processing steps, we applied anisotropic filtering and histogram matching to whole data set. The total number of 2D slices for training were about 1690 and batch size of 10 were chosen for training. Thus, the number of iteration per epoch is $\frac{1690}{10} = 169$ and we have total number of iteration $100 \times 169 = 16900$ during training. The Cross Entropy loss function were chosen for minimization. All the networks are implemented on Tensorflow with using NVIDIA TitanXP GPUs.

Also, in the test data set, as the post processing steps, we applied 3D conditional random field following by selecting connecting component to output of the network.

7.5.3 Results

We calculated Dice Index (DI) and also Cross Entropy (CE) loss on validation set for investigating our proposed optimizer along with other optimizers. In Figures 7.5a and b, the CE and DI curves versus iterations for U-Net architecture for different optimizers are illustrated. As these curves are showing, the DI in U-Net with Adam optimizer is increasing rapidly and sharply at very beginning and then it is almost fixed. But, although our proposed optimizer (CLMR($C_{lr}=20$, $C_{mr}=20$)) is not learning as fast as ADAM optimizer at ver beginning, but it gets higher than ADAM curve finally (this phenomena is more clear in CE curves). The quantitative results on test set in Table 7.2 support the same conclusion. Also, same pattern is happening for DenseNet_2 architecture in Figure 7.6. This is confirming our hypothesis that adaptive optimizers are converging faster but to different local minima point in comparison with classical SGD optimizers.

As Figure 7.5 shows the U-Net architecture with CLMR optimizer performs much better than its CRL optimizer counterpart. This proves that having a cyclic momentum rate can yield to better efficiency than having a simple cyclic learning rate. The results on test set between CLR and CLMR optimizer in Table 7.2 also support this conclusion.

In addition, the curves of DI and CE among different architectures which trained by Adam and CLMR are illustrated in Figure 7.4 a and b. Although, the DenseNet_2 has less parameters in comparison with other architectures, but it gets better results than the other architectures. These curves reveals some important points about using different architectures: first, for all different architecture CLMR optimizer is working better than ADAM optimizer which indicate the power of proposed cyclic optimizer. Second, DenseNet architectures are getting better results than U-Net and Enc_Dec architectures, however the number of parameters in U-Net and Enc_Dec are almost 10 times of DenseNet architectures (Table 7.1). This points indicates that having a over-parameterized architectures can saturate in the training quickly, while having local skip connections can solve the mentioned problem. Third, comparison between curves of DenseNet_1 and DenseNet_2 shows that having higher GR is more important than having dense block with high number of parameters. Since, DenseNet_2, with GR=24, got better results in comparison with DenseNet_1 with twice of number of parameters in end of each dense block in comparison to DenseNet_2 and GR=16. These results are supported with the dice metric obtained from test data and are mentioned in Table 7.2.

Finally, the DI on test data set with online evaluation for different architectures with different optimizers are summarized in Table 7.2. In order to have better comparison, the box plot of all methods are illustrated in Figure 7.7. As this figure shows the dice statistic obtained from CLMR is similar or better than other optimizers.

In addition, qualitative results for different methods are illustrated in Figure 7.8 and Figure 7.9. Figure 7.8 shows the contours for RV, Myo., and LV in ED for different methods and architectures

and also ground-truth across four slices from Apex to Base. Usually, segmentation of RV near the Apex are more harder than others, since RV is almost vanishing at this point and as it is clear some of the methods couldn't even detect the RV at slices near Apex. Figure 7.9 shows the contours for RV, Myo., and LV in ES for different methods and architectures and also ground-truth across four slices from Apex to Base. Since at Es heart is at minimum volume, thus it is more difficult to segment substructures. Also, as it is clear in Figure 7.9, RV near Apex and Base is disappeared, however most of the methods recognize other structures as RV at Base slice. The contours generated with DenseNet_2 method is more similar to ground-truth in both ED and ES, which shows the generalizability of proposed method.

Table 7.2: DI in the test data set with online evaluation.

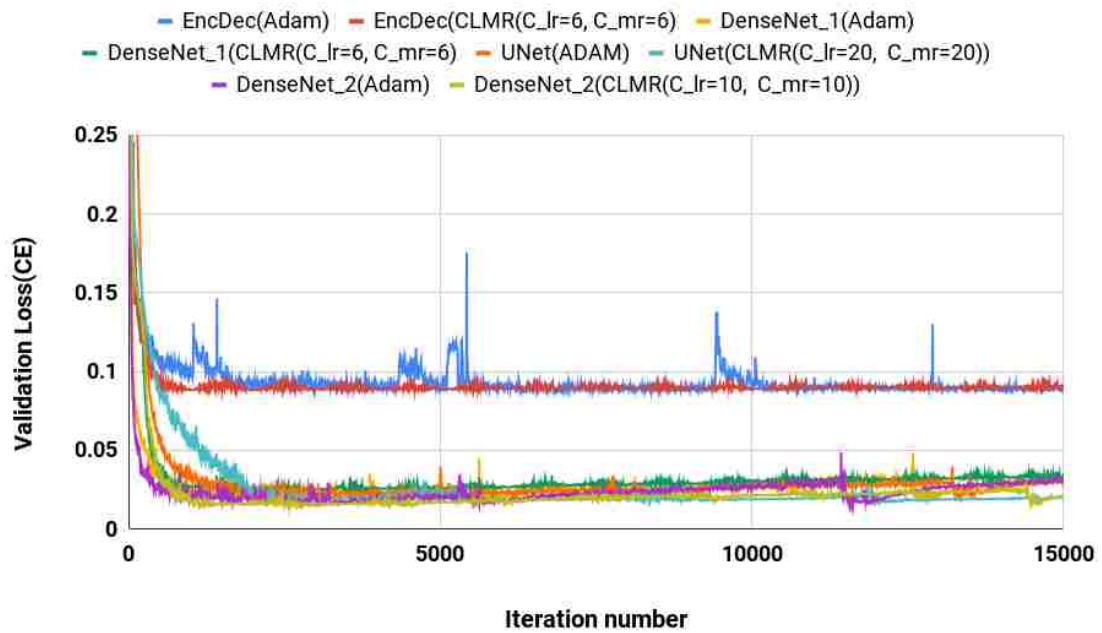
		Adam	Nesterov	CLR	CLMR
Enc_Dec	<i>RV</i>	0.3272	0.1309	0.3833	0.4336
UNet		0.8574	0.5968	0.8618	0.8820
DenseNet_1		0.8802	0.6936	0.8961	0.8957
DenseNet_2		0.8781	0.7232	0.8910	0.9049
Enc_Dec	<i>Myo</i>	0.1473	0.1492	0.1692	0.1686
UNet		0.8628	0.6486	0.8588	0.8631
DenseNet_1		0.8787	0.7170	0.8834	0.8960
DenseNet_2		0.8796	0.7196	0.8904	0.8999
Enc_Dec	<i>LV</i>	0.4950	0.3260	0.4972	0.5418
UNet		0.9238	0.7670	0.8936	0.9360
DenseNet_1		0.9376	0.8465	0.9351	0.9393
DenseNet_2		0.9196	0.8449	0.9378	0.9478
Enc_Dec	<i>Ave.</i>	0.3232	0.1687	0.3499	0.3814
UNet		0.8813	0.6708	0.8714	0.8937
DenseNet_1		0.8988	0.7524	0.9049	0.9103
DenseNet_2		0.8924	0.7626	0.9064	0.9176

7.6 Discussions and Conclusion Remarks

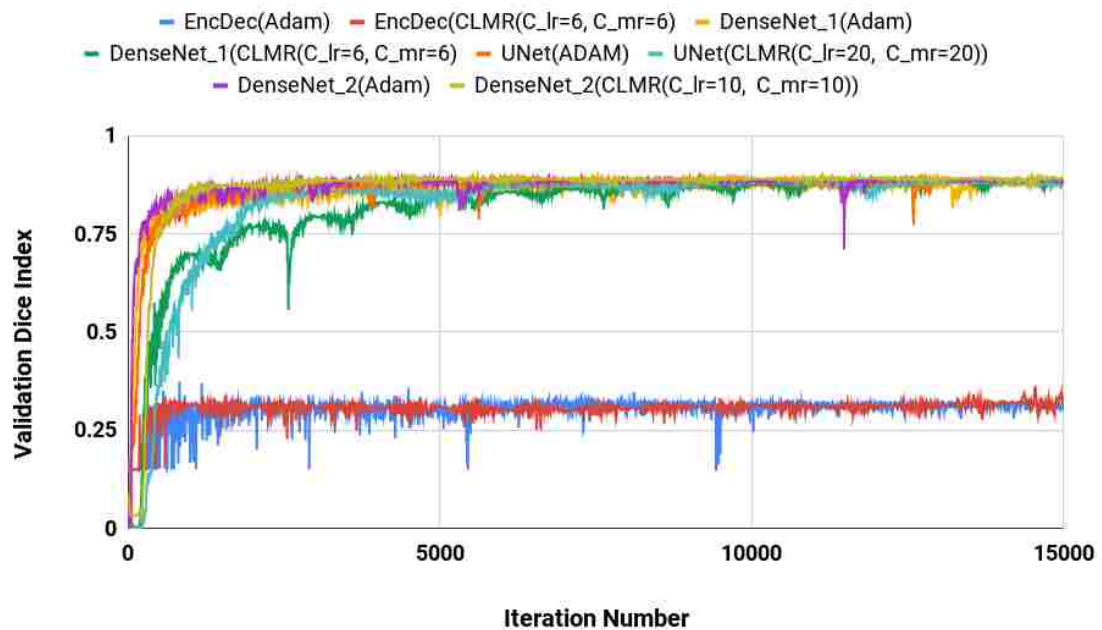
We proposed a new cyclic optimization method (CLMR) to solve the current problems in deep learning optimization. We hypothesized that having a cyclic learning/momentum function can yield to better generalization in comparison to adaptive optimizers, then we showed in the results part that CMLR is doing better than adaptive optimizers.

Also, our method could do better than other cyclic methods (such as CLR) by considering momentum changes in Nesterov optimizer as a cyclic function too. Finding the parameters of these cyclic functions are complicated due to the correlation which exists between LR and MR function. So, we formulated both LR and MR functions and we suggested a method to find the parameters of these cyclic functions with reasonable computational cost.

The proposed method is just a beginning to new generation of optimizers which can generalize better than adaptive ones. One of the challenges in designing these kind of optimizers are setting the parameters of cyclic functions which need further investigation. One can learn these parameters with a neural network or reinforcement learning in an efficient manner. i.e the max_{lr} , min_{lr} , max_r , min_{mr} , C_{lr} , and C_{mr} can be learned by an policy gradient reinforcement learning approach.

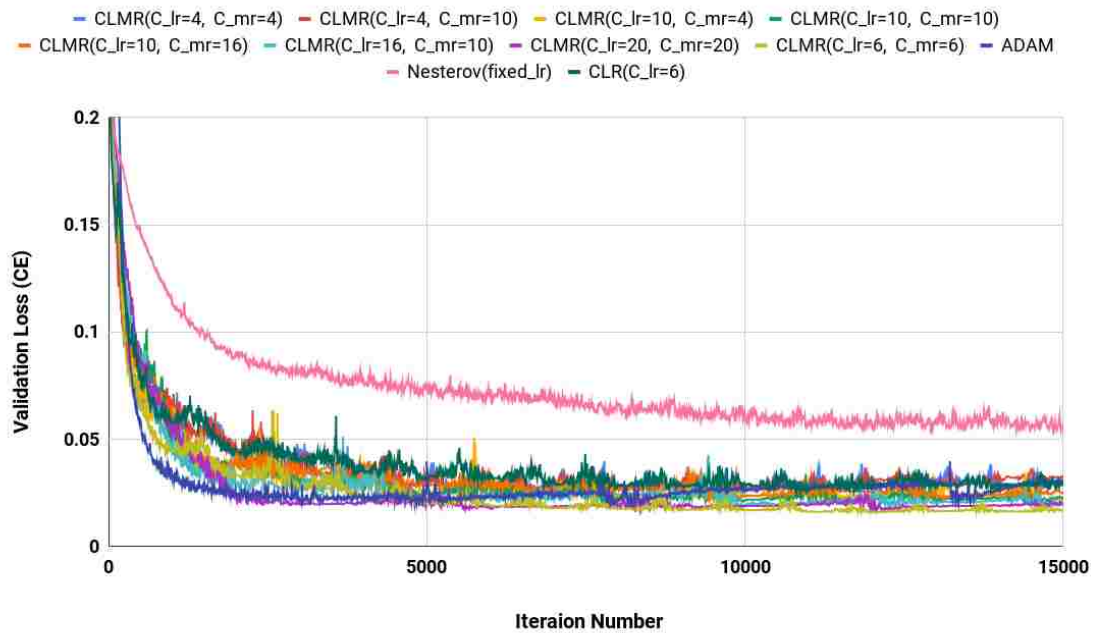


(a) Cross Entropy loss in validation set for four different architectures

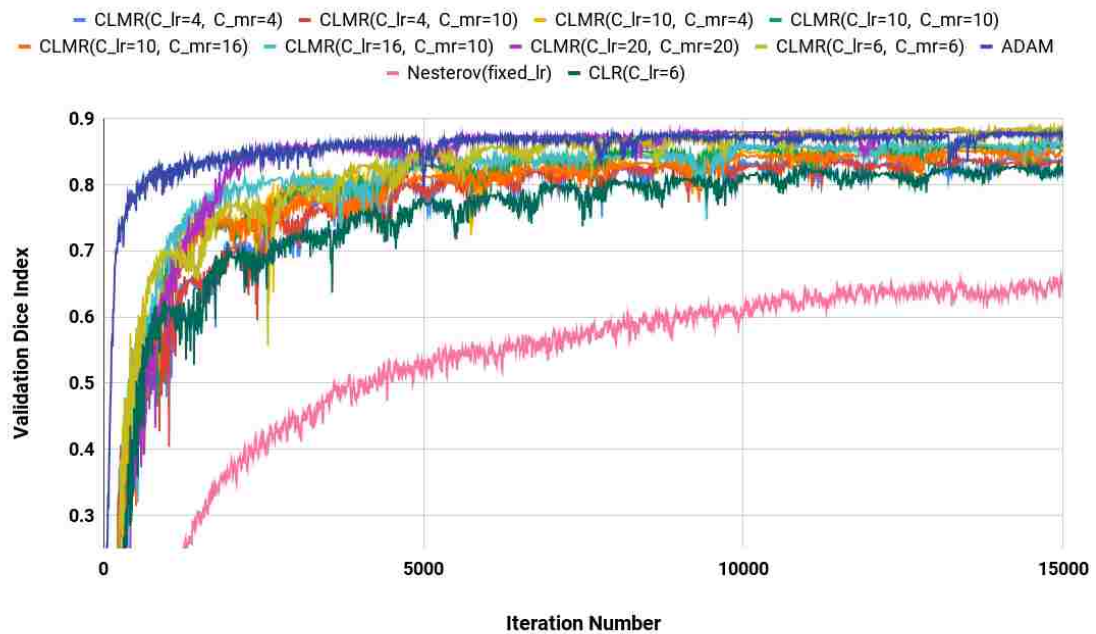


(b) Dice index in validation set for four different architectures

Figure 7.4: Validation loss and dice index for four different architectures with ADAM and CLMR optimizers

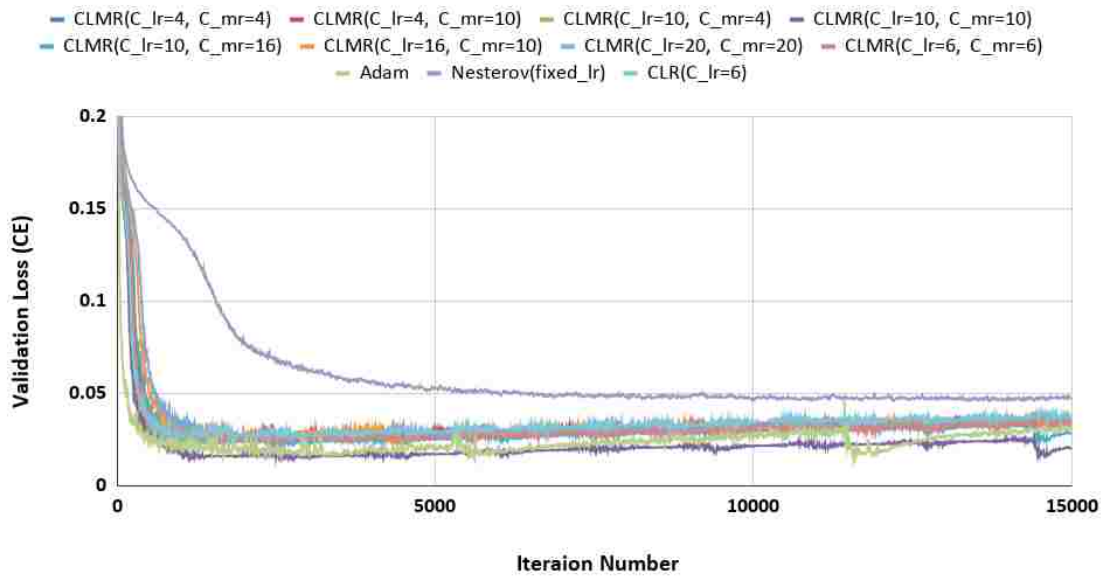


(a) Cross Entropy loss in validation set for DenseNet_2 architecture

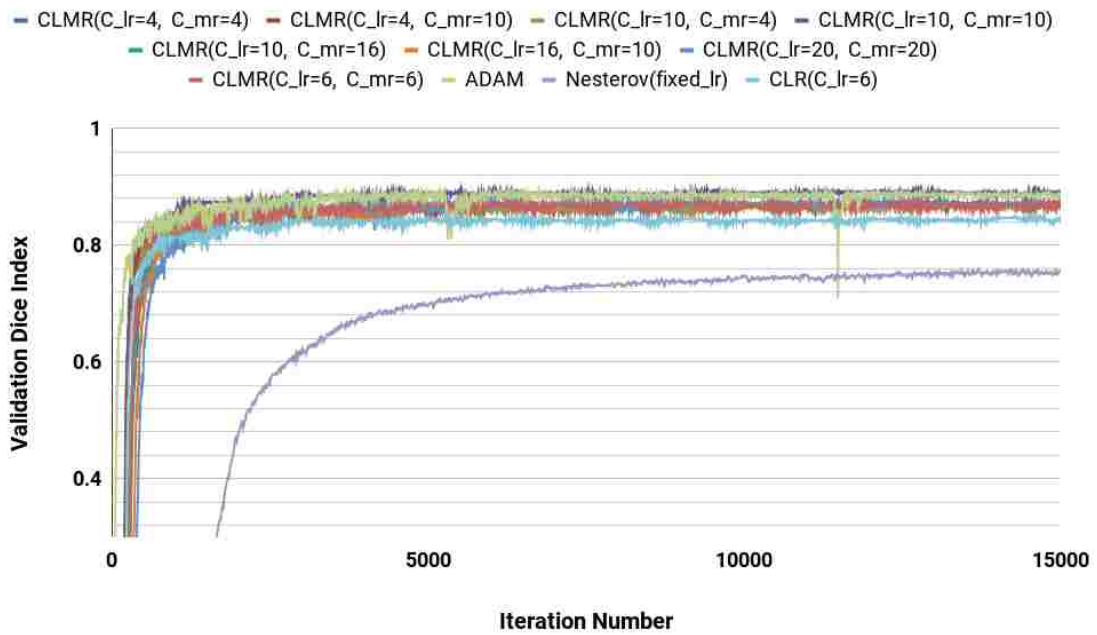


(b) Dice index in validation set for UNet architecture

Figure 7.5: Validation loss and dice index for DenseNet_2 architecture with different values of C_{lr} and C_{mr}



(a) Cross Entropy loss in validation set for DenseNet_2 architecture



(b) Dice index in validation set for UNet architecture

Figure 7.6: Validation loss and dice index for UNet architecture with different values of C_{lr} and C_{mr}

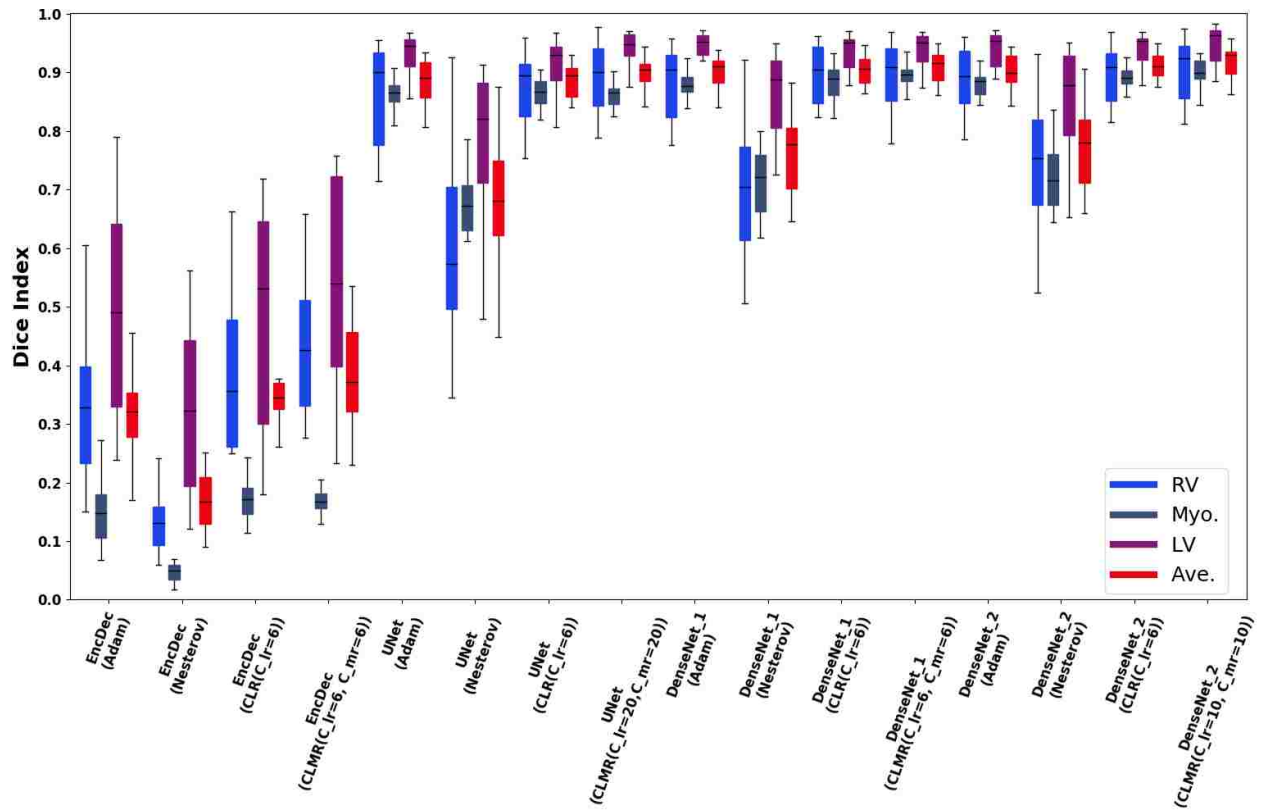


Figure 7.7: Box plots for DI in test data set for RV, Myo., LV and also average of them (Ave.).

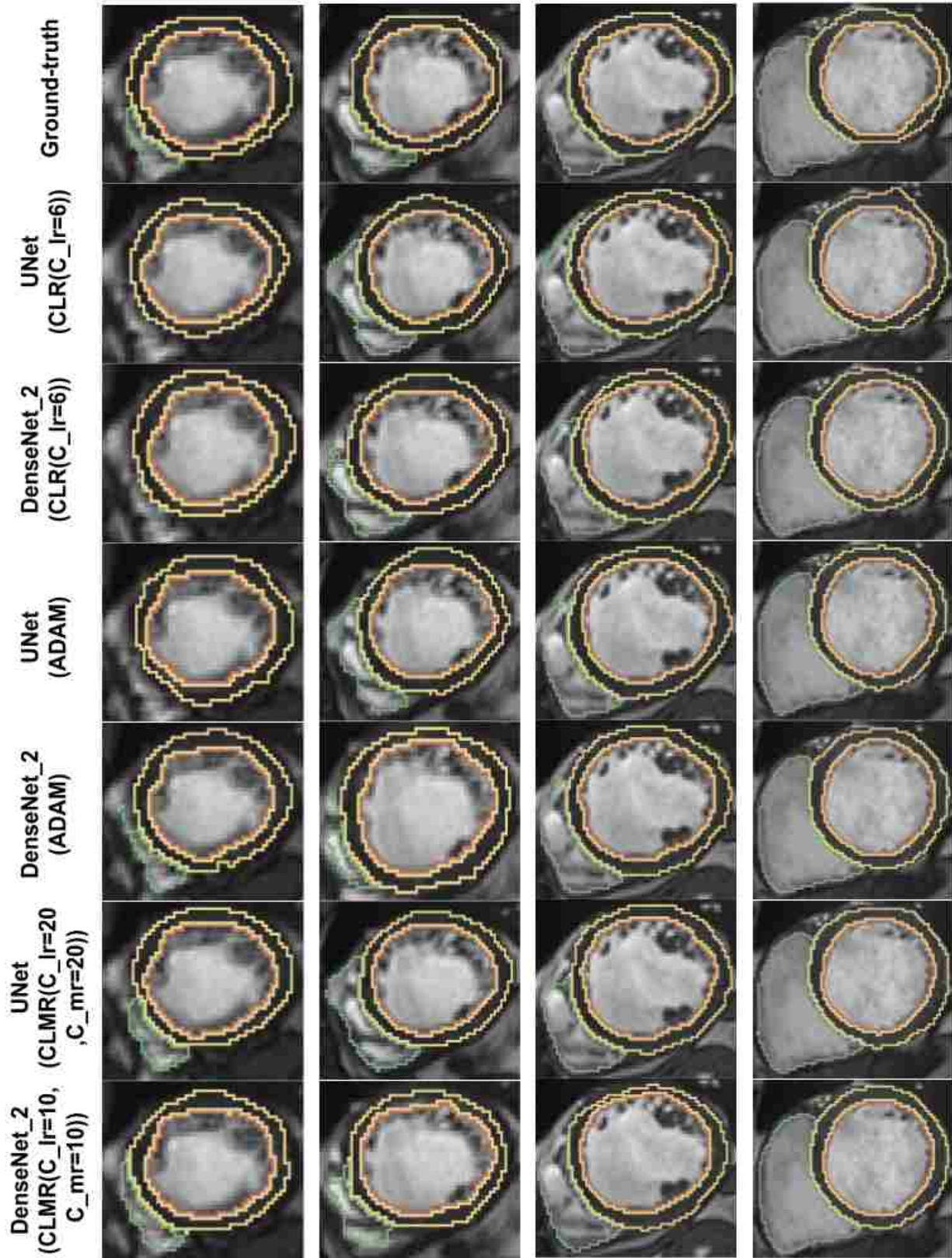


Figure 7.8: Qualitative results for ground-truth and different methods for same subject in end-diastole from Apex to Base for four slices (from right to left). Green, yellow, and brown contours are showing RV, myo, and LV, respectively.

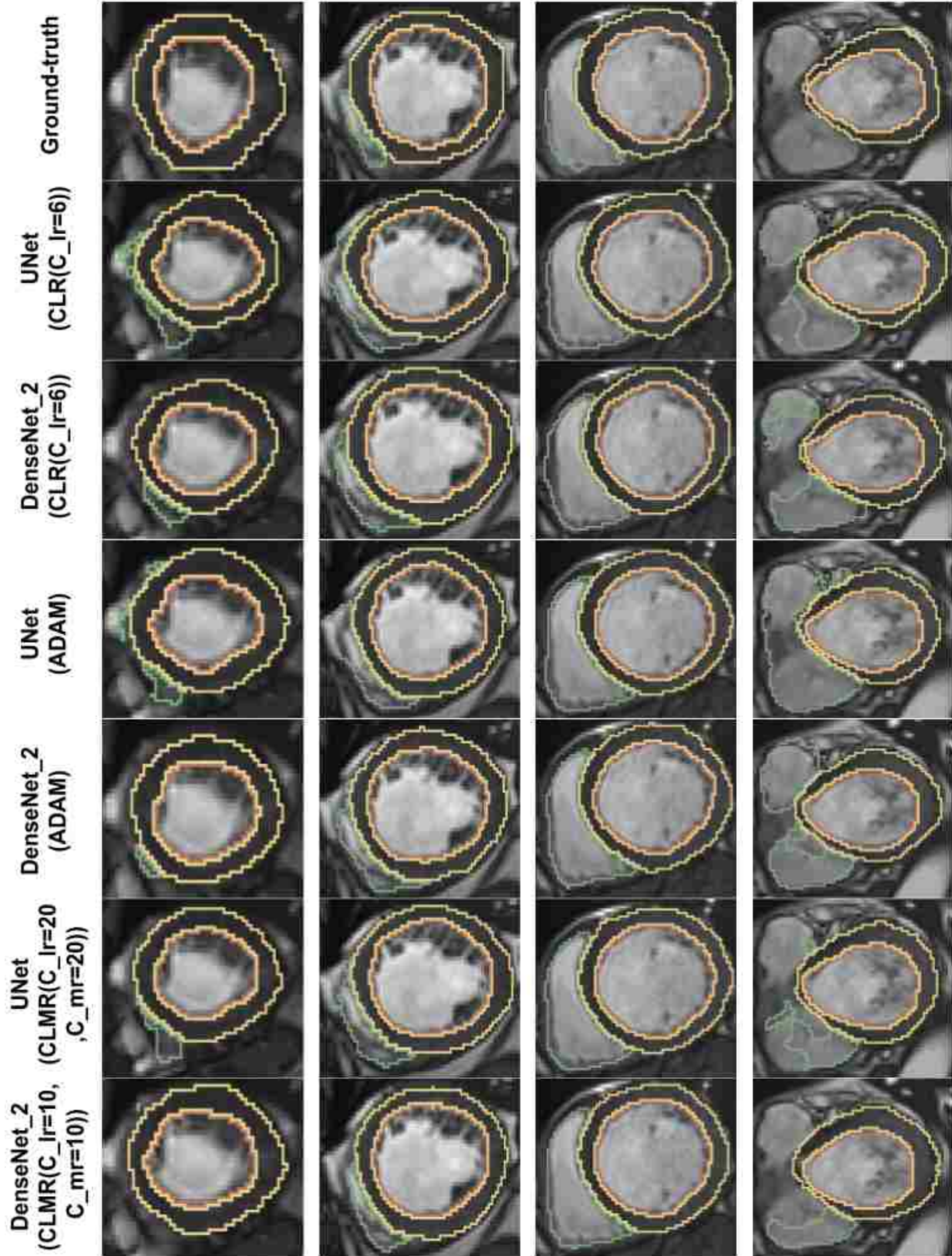


Figure 7.9: Qualitative results for ground-truth and different methods for same subject in end-systole from Apex to Base for four slices (from right to left). Green, yellow, and brown contours are showing RV, myo, and LV, respectively.

CHAPTER 8: CONCLUSION AND FUTURE WORKS

In this chapter, the remarks of dissertation are concluded and summarized. Also, suggestions for future works based on the proposed methods in the dissertation are given.

8.1 Conclusion

In this dissertation, we proposed methods to optimize segmentation deep learning algorithms for medical image segmentation in different aspects such as dimensionality of input images, different modalities and multi-organs, hyperparameters of CNN architectures, using noisy and inexpert labels in training, and designing efficient and accurate optimizers for medical images analysis. In Chapter 3, we proposed a method to handle dimensionality problem in medical images. Since the images in medical images are 3D or 4D, then the CNN methods demands a large number of parameters with huge amount of computational resources to be trained. In Chapter 3, this problem were solved by proposing a 2.5D method following by adaptive fusion. Three 2D CNN networks were trained for each planes of image (axial, sagittal, and coronal) and then in order to utilize the information from each plane in outputs of these CNNs, an adaptive fusion method were introduced and applied. So, the proposed algorithm demanded same amount of parameters and computational resources as 2D networks, while it used information of 3D image to do binary segmentation. Furthermore, in order to train faster (since there are three networks to be trained) a new loss function based on Z-loss were introduced and used. The proposed approach applied to anatomical modeling of LA and PPVs from MRI images due to their importance in diagnosis and predicting several cardiac diseases. In Chapter 4, we extended the proposed method in previous chapter to make it applicable to do multi-class segmentation in multi-modality medical images. Segmenting different organs in different image modalities in clinic with a generic method is challenging due to high

variations among the organs and also numerous modalities in medical images. The proposed approach applied to CT and MRI image to segment seven different substructures from cardiac. In the Chapter 5, we proposed a computational efficient method to address the problem of finding the optimum CNN architecture for medical image segmentation task. Current methods to find the optimum architecture for a given task are computationally expensive and need huge number of resources, however our proposed methods can automatically find optimum hyperparameters of network with reasonable and limited computational resources. We applied our method to segment cardiac substructures from cine-MRI (4D MRI) images which can be useful in measuring ejection fraction in order to diagnosis or predicting heart stroke. In Chapter 6, a new approach proposed based on the learning segmentation from geodesic maps obtained from prior information form a semi auto-encoder network. For first time in the literature, the noisy and inexpert labels were used to train the segmentor as well as to obtain prior information. The information obtained from geodesic prior were integrated into the loss function of weakly supervised segmentation network to guide it in the absence of accurate labels. Finally, in Chapter 7, we investigated the effect of different optimization methods, such as adaptive and cyclic optimizers, in the performance of deep networks in medical image segmentation. We introduced a new cyclic learning /momentum rate optimizer, called CLMR, which computationally is cheaper than adaptive optimizer while it is generalizing better in comparison with them. Both learning rate and momentum rate are changing in cyclic manner in new proposed optimizer. However new optimizer is not converging as fast as adaptive optimizers, but at the end it outperformed them on both validation and test data sets.

8.2 Future Work

In Chapter 3 and 4, we introduced an adaptive fusion method which assigned weights to output of each CNNs according to segmented object. The segmented objects were obtained after applying

the softmax function in the last layer of the proposed network. So, to explore whether the information loss due to class normalization in this step is significant, further research should be undertaken using information from the layer before the softmax in fusion part and compared with the current system. In addition, the weights which are assigned to each network can be learned by Multilayer perceptron network during the training and current calculated weights can be used as initialization.

In Chapter 5, there is no constraint during learning the hyperparameters of network. However, some constraints such as number of parameters in network can be added to learning process. Also, other architecture hyperparameters such as skip connections among the layers can be added to initial policy set to be learned. Another work which can be done in this research is choosing different and better architecture as a baseline architecture.

In Chapter 6, the semi auto-encoder is first fully trained to map geodesic map to its corresponding binary map, and then the segmentor started to train. One of the works which can be done is training whole system in end-to-end manner. i.e both semi auto-encoder and segmentor networks can be trained simultaneously.

LIST OF REFERENCES

- [1] Wufeng Xue, Gary Brahm, Sachin Pandey, Stephanie Leung, and Shuo Li, “Full left ventricle quantification via deep multitask relationships learning,” *Medical image analysis*, vol. 43, pp. 54–65, 2018.
- [2] Olivier Bernard, Alain Lalande, Clement Zotti, Frederick Cervenansky, Xin Yang, Pheng-Ann Heng, Irem Cetin, Karim Lekadir, Oscar Camara, Miguel Angel Gonzalez Ballester, et al., “Deep learning techniques for automatic mri cardiac multi-structures segmentation and diagnosis: Is the problem solved?,” *IEEE transactions on medical imaging*, vol. 37, no. 11, pp. 2514–2525, 2018.
- [3] Neeraj Sharma and Lalit M Aggarwal, “Automated medical image segmentation techniques,” *Journal of medical physics/Association of Medical Physicists of India*, vol. 35, no. 1, pp. 3, 2010.
- [4] Alain Jungo, Raphael Meier, Ekin Ermis, Marcela Blatti-Moreno, Evelyn Herrmann, Roland Wiest, and Mauricio Reyes, “On the effect of inter-observer variability for a reliable estimation of uncertainty of medical image segmentation,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2018, pp. 682–690.
- [5] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, pp. 436, 2015.
- [6] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

- [7] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in *Advances in neural information processing systems*, 2015, pp. 91–99.
- [8] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” *arXiv preprint arXiv:1511.00561*, 2015.
- [9] “Cardiovascular Diseases (cvds),” <http://www.who.int/mediacentre/factsheets/fs317/en/>, 2007, [Online; accessed 30-June-2017].
- [10] Suman S Kuppahally and et al., “Left atrial strain and strain rate in patients with paroxysmal and persistent atrial fibrillation relationship to left atrial structural remodeling detected by delayed-enhancement mri,” *Circulation: Cardiovascular Imaging*, vol. 3, no. 3, 2010.
- [11] Abdelaziz Daoudi, Saïd Mahmoudi, and Mohammed Amine Chikh, “Automatic segmentation of the left atrium on ct images,” in *International Workshop on Statistical Atlases and Computational Models of the Heart*. Springer, 2013, pp. 14–23.
- [12] Aliasghar Mortazi, Rashed Karim, Kawal Rhode, Jeremy Burt, and Ulas Bagci, “Cardiacnet: Segmentation of left atrium and proximal pulmonary veins from mri using multi-view cnn,” in *MICCAI*. Springer, 2017, pp. 377–385.
- [13] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” *arXiv preprint arXiv:1606.00915*, 2016.
- [14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *MICCAI*. Springer, 2015.

- [15] John Duchi, Elad Hazan, and Yoram Singer, “Adaptive subgradient methods for online learning and stochastic optimization,” *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.
- [16] Diederik P Kingma and Jimmy Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [17] Ashia C Wilson, Rebecca Roelofs, Mitchell Stern, Nati Srebro, and Benjamin Recht, “The marginal value of adaptive gradient methods in machine learning,” in *Advances in Neural Information Processing Systems*, 2017, pp. 4151–4161.
- [18] Leslie N Smith, “Cyclical learning rates for training neural networks,” in *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on. IEEE*, 2017, pp. 464–472.
- [19] Jian Zhang and Ioannis Mitliagkas, “Yellowfin and the art of momentum tuning,” *arXiv preprint arXiv:1706.03471*, 2017.
- [20] Jelmer M Wolterink, Tim Leiner, Max A Viergever, and Ivana Isgum, “Automatic segmentation and disease classification using cardiac cine mr images,” *arXiv preprint arXiv:1708.01141*, 2017.
- [21] Simon Jégou, Michal Drozdal, David Vazquez, Adriana Romero, and Yoshua Bengio, “The one hundred layers tiramisù: Fully convolutional densenets for semantic segmentation,” in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on. IEEE*, 2017, pp. 1175–1183.
- [22] Tobon-Gomez and et al., “Benchmark for algorithms segmenting the left atrium from 3d ct and mri datasets,” *IEEE TMI*, vol. 34, no. 7, 2015.

- [23] Birgit Stender, Oliver Blanck, Bo Wang, and Alexander Schlaefer, "Model-based segmentation of the left atrium in ct and mri scans," in *International Workshop on Statistical Atlases and Computational Models of the Heart*. Springer, 2013.
- [24] Xiahai Zhuang and Juan Shen, "Multi-scale patch and multi-modality atlases for whole heart segmentation of mri," *Medical image analysis*, vol. 31, pp. 77–87, 2016.
- [25] X Zhuang, S Ourselin, R Razavi, DLG Hill, and DJ Hawkes, "Automatic whole heart segmentation based on atlas propagation with a priori anatomical information," *Medical Image Understanding and Analysis-MIUA*, pp. 29–33, 2008.
- [26] Peng Peng, Karim Lekadir, Ali Gooya, Ling Shao, Steffen E Petersen, and Alejandro F Frangi, "A review of heart chamber segmentation for structural and functional analysis using cardiac magnetic resonance imaging," *Magma (New York, NY)*, vol. 29, pp. 155, 2016.
- [27] Rudra PK Poudel, Pablo Lamata, and Giovanni Montana, "Recurrent fully convolutional neural networks for multi-slice mri cardiac segmentation," *arXiv preprint arXiv:1608.03974*, 2016.
- [28] MR Avendi, Arash Kheradvar, and Hamid Jafarkhani, "A combined deep-learning and deformable-model approach to fully automatic segmentation of the left ventricle in cardiac mri," *Medical image analysis*, vol. 30, pp. 108–119, 2016.
- [29] Gongning Luo, Ran An, Kuanquan Wang, Suyu Dong, and Henggui Zhang, "A deep learning network for right ventricle segmentation in short-axis mri," in *Computing in Cardiology Conference (CinC), 2016*. IEEE, 2016, pp. 485–488.
- [30] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten, "Densely connected convolutional networks," *arXiv preprint arXiv:1608.06993*.

- [31] Naji Khosravan, Aliasghar Mortazi, Michael Wallace, and Ulas Bagci, “Pan: Projective adversarial network for medical image segmentation,” *arXiv preprint arXiv:1906.04378*, 2019.
- [32] Aliasghar Mortazi, Jeremy Burt, and Ulas Bagci, “Multi-planar deep segmentation networks for cardiac substructures from mri and ct,” *arXiv preprint arXiv:1708.00983*, 2017.
- [33] Aliasghar Mortazi, Jeremy Burt, and Ulas Bagci, “Deep learning for cardiac mri: Automatically segmenting left atrium and proximal veins with human level performance,” .
- [34] Barret Zoph and Quoc V Le, “Neural architecture search with reinforcement learning,” *arXiv preprint arXiv:1611.01578*, 2016.
- [35] Kenneth O Stanley, David B D’Ambrosio, and Jason Gauci, “A hypercube-based encoding for evolving large-scale neural networks,” *Artificial life*, vol. 15, 2009.
- [36] Nate Kohl and Peter Stone, “Policy gradient reinforcement learning for fast quadrupedal locomotion,” in *Robotics and Automation, 2004. Proceedings. ICRA’04. 2004 IEEE International Conference on*. IEEE, 2004, vol. 3.
- [37] Aliasghar Mortazi and Saeed Bagheri Shouraki, “Using embodiment theory to train a set of actuators with different expertise to accomplish a duty: An application to train a quadruped robot for walking,” in *2015 23rd Iranian Conference on Electrical Engineering*. IEEE, 2015, pp. 589–594.
- [38] Aliasghar Mortazi and Ulas Bagci, “Automatically designing cnn architectures for medical image segmentation,” in *International Workshop on Machine Learning in Medical Imaging*. Springer, 2018, pp. 98–106.

- [39] Clement Zotti, Zhiming Luo, Alain Lalande, and Pierre-Marc Jodoin, “Convolutional neural network with shape prior applied to cardiac mri segmentation,” *IEEE journal of biomedical and health informatics*, 2018.
- [40] Ozan Oktay, Enzo Ferrante, Konstantinos Kamnitsas, Mattias Heinrich, Wenjia Bai, Jose Caballero, Stuart A Cook, Antonio de Marvao, Timothy Dawes, Declan P O’Regan, et al., “Anatomically constrained neural networks (acnns): application to cardiac image enhancement and segmentation,” *IEEE transactions on medical imaging*, vol. 37, no. 2, pp. 384–395, 2018.
- [41] Aliasghar Mortazi, Naji Khosravan, Drew A Torigian, Sila Kurugol, and Ulas Bagci, “Weakly supervised segmentation by a deep geodesic prior,” in *International Workshop on Machine Learning in Medical Imaging*. Springer, 2019, pp. 238–246.
- [42] Ning Qian, “On the momentum term in gradient descent learning algorithms,” *Neural networks*, vol. 12, no. 1, pp. 145–151, 1999.
- [43] Yurii Nesterov, “A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$,” in *Doklady AN USSR*, 1983, vol. 269, pp. 543–547.
- [44] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han, “Learning deconvolution network for semantic segmentation,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1520–1528.
- [45] Alexandre de Brébisson and Pascal Vincent, “The z-loss: a shift and scale invariant classification loss belonging to the spherical family,” *arXiv preprint arXiv:1604.08859*, 2016.
- [46] Maria A Zuluaga and et al., “Multi-atlas propagation whole heart segmentation from mri and cta using a local normalised correlation coefficient criterion,” in *International Conference on Functional Imaging and Modeling of the Heart*. Springer, 2013.

- [47] Xiahai Zhuang, Lei Li, Christian Payer, Darko Štern, Martin Urschler, Mattias P. Heinrich, Julien Oster, Chunliang Wang, Örjan Smedby, Cheng Bian, Xin Yang, Pheng-Ann Heng, Aliasghar Mortazi, Ulas Bagci, Guanyu Yang, Chenchen Sun, Gaetan Galisot, Jean-Yves Ramel, Thierry Brouard, Qianqian Tong, Weixin Si, Xiangyun Liao, Guodong Zeng, Zenglin Shi, Guoyan Zheng, Chengjia Wang, Tom MacGillivray, David Newby, Kawal Rhode, Sebastien Ourselin, Raad Mohiaddin, Jennifer Keegan, David Firmin, and Guang Yang, “Evaluation of algorithms for multi-modality whole heart segmentation: An open-access grand challenge,” *Medical Image Analysis*, vol. 58, pp. 101537, 2019.
- [48] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al., “TensorFlow: Large-scale machine learning on heterogeneous distributed systems,” *arXiv preprint arXiv:1603.04467*, 2016.
- [49] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour, “Policy gradient methods for reinforcement learning with function approximation,” in *Advances in neural information processing systems*, 2000, pp. 1057–1063.
- [50] Prajit Ramachandran, Barret Zoph, and Quoc V Le, “Searching for activation functions,” 2017.
- [51] Rodney LaLonde and Ulas Bagci, “Capsules for object segmentation,” *arXiv preprint arXiv:1804.04241*, 2018.
- [52] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger, “Densely connected convolutional networks,” in *CVPR*, 2017, vol. 1, p. 3.
- [53] Alexandru Telea, “An image inpainting technique based on the fast marching method,” *Journal of graphics tools*, vol. 9, no. 1, pp. 23–34, 2004.

[54] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky, “Neural networks for machine learning lecture 6a overview of mini-batch gradient descent,” .