Automatic Analysis of Chest CT in Systemic Sclerosis Using Deep Learning

Jingnan Jia

Colophon

About the cover:

The cover was created by DALL-E 3 and modified by Jingnan Jia. On the front cover, you are greeted by a painting of seriously diseased lungs. The painting is drawn by a robotic hand that stretched out from the left. Turning to the back cover, you will find that the hand comes from an infant robot, and the screen on the robot's belly shows that it is learning from chest CT scans. The whole scene is to express that deep learning models are currently in development to automatically diagnose lung diseases. With enough anatomical knowledge, you may notice that the painting is inaccurate, e.g. lung lobes are missing. This means that deep learning may still make serious mistakes. In summary, although the current application of deep learning is still in its infancy, its future is bright.

Automatic Analysis of Chest CT in Systemic Sclerosis Using Deep Learning
Jingnan Jia

ISBN: 978-94-6506-140-5
Thesis layout & cover designed by Jingnan Jia
Printed by Ridderprint, the Netherlands

© 2024 Jingnan Jia, Leiden, the Netherlands
All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without permission in writing from the copyright owner.

Automatic Analysis of Chest CT in Systemic Sclerosis Using Deep Learning

Proefschrift

ter verkrijging van
de graad van doctor aan de Universiteit Leiden,
op gezag van rector magnificus prof.dr.ir. H. Bijl,
volgens besluit van het college voor promoties
te verdedigen op dinsdag 10 september 2024
klokke 11:30 uur

door

Jingnan Jia
geboren te Linfen, Shanxi, China
in 1993

Promotor:	Prof. dr. ir. M. Staring
Co-promotor:	Dr. B. C. Stoel
Leden promotiecommissie:	Prof. dr. A.H.M. van der Helm-van Mil Prof. dr. H.A. Marquering *Amsterdam University Medical Center, Amsterdam* dr. C. Jakobs *Radboud University Medical Center, Nijmegen* dr. T. Alderliesten

The research in this thesis was performed at the Division of Image Processing (LKEB), Department of Radiology of Leiden University Medical Center, The Netherlands.
This work was carried out in the ASCI graduate school. ASCI dissertation series number: 457.

Financial support for the publication of this thesis was kindly provided by:
LKEB,
ASCI research school,
Library of Leiden University,
Stichting NVLE Fonds,
Guerbet Nederland,
Canon Medical Systems Nederland,
Mr. Willem Bakhuys Roozeboomstichting.

Contents

List of abbreviations **v**

1 Introduction **1**

1.1 Pulmonary anatomy 1
1.2 Systemic sclerosis (SSc) 2
1.3 Pulmonary function tests 3
1.4 Chest HRCT 5
1.5 Deep learning on chest HRCT 6
1.6 Thesis outline 7

2 Multi-task semi-supervised learning for pulmonary lobe segmentation **11**

2.1 Introduction 13
2.2 Methods 13
  2.2.1 Model design 13
  2.2.2 Loss function and evaluation metrics 14
  2.2.3 Training strategies 15
  2.2.4 Learning rate optimization 15
2.3 Dataset and data augmentation 16
  2.3.1 Dataset 16
  2.3.2 Data augmentation 17
2.4 Experiments and results 17
  2.4.1 Implementation details 17
  2.4.2 Impact of multi-scale input & adaptive learning rate 17
  2.4.3 Impact of auxiliary tasks & training strategies 18
  2.4.4 Comparison with existing networks 18
2.5 Conclusions 18
2.6 Acknowledgements 19

3 **Explainable fully automated CT scoring of systemic sclerosis related interstitial lung disease by cascaded regression neural networks and its comparison with experts** **21**
3.1 Introduction 23
3.2 Related work 25
3.2.1 Automated level selection 25
3.2.2 Automated scoring 25
3.2.3 Automated scoring for SSc-ILD 26
3.3 Materials and methods 26
3.3.1 Dataset 26
3.3.2 Network design 29
3.3.3 Techniques to overcome label imbalance 30
3.3.4 Heat map visualization: network explainability 33
3.3.5 Statistical analysis and evaluation 34
3.3.6 Network implementation details 34
3.4 Experiments and results 35
3.4.1 SSc-ILD scoring performance 35
3.4.2 Comparison with human experts 35
3.4.3 Heat map explanation and its evaluation 38
3.5 Discussion 38
3.5.1 Comparison to related work 42
3.5.2 Explanation and discussion on results 42
3.5.3 Limitations 45
3.6 Conclusions 45
3.7 Acknowledgments 45
3.8 Appendix 46

4 **Automatic pulmonary function estimation from chest CT scans using deep regression neural networks: the relation between structure and function in systemic sclerosis** **51**
4.1 Introduction 53
4.2 Materials and methods 54
4.2.1 Dataset 54
4.2.2 Network design 57
4.2.3 Relation between lung structure and function 58
4.2.4 Evaluation metrics and statistical analysis 59
4.3 Experiments and results 60
4.3.1 Experiment setting 60
4.3.2 Manually designed biomarkers (Q1) 60

4.3.3 Transfer learning (TL) versus trained from scratch (Q2) 61
4.3.4 Estimation of PFTs%pred: direct versus indirect (Q3) 61
4.3.5 1-class versus 4-class (Q4) 63
4.3.6 Contribution of the different chest regions (Q5) 64
4.3.7 Comparison standard repeatability criteria (Q6) 69
4.3.8 Multiple variable regression analysis (Q7) 70
4.4 Discussion and conclusions 71
4.5 Acknowledgements 74

5 Using 3D point cloud and graph-based neural networks to improve the estimation of pulmonary function tests from chest CT 75
5.1 Introduction 77
5.2 Related work 79
5.2.1 Point cloud and point cloud neural netowk (PNN) 79
5.2.2 Graph and graph neural network (GNN) 80
5.3 Methodology 80
5.3.1 Acquisition of CT images and PFTs measurements 82
5.3.2 Dataset preparation 82
5.3.3 PNN-Vessel design 83
5.3.4 GNN-Vessel design 84
5.3.5 Combination of different networks 85
5.3.6 Evaluation metrics and statistical analysis 86
5.4 Experiments and results 87
5.4.1 Experimental settings 87
5.4.2 E1: Training strategies and hyper-parameters for PNN-Vessel . . 87
5.4.3 E2: Training strategies and hyper-parameters for GNN-Vessel . . 90
5.4.4 E3: Combination of different networks 91
5.5 Discussion 93
5.6 Conclusions 96
5.7 Declaration of interests 96
5.8 Data availability 96
5.9 Acknowledgments 97

6 Summary, discussion and future work 99
6.1 Summary 99
6.2 Discussion and future work 100
6.3 General conclusions 102

7 Samenvatting, discussie en toekomstig werk 103
7.1 Samenvatting 103

7.2 Discussie en toekomstig werk 105
7.3 Algemene conclusies 107

8 Supplementary material 109
seg-metrics: a Python package to compute segmentation metrics 109
8.1 Background 111
8.2 Related packages 111
8.3 Our `seg-metrics` package 111
    8.3.1 Overlap-based metrics 112
    8.3.2 Distance-based metrics 114
8.4 Installation 115
8.5 Use cases 115
8.6 Comparison to other packages 118
8.7 Limitation and future work 119
8.8 Availability and requirements 120
8.9 Acknowledgments 120

References 121

List of publications 133

Acknowledgements 135

Curriculum Vitae 137

List of abbreviations

CT	computed tomography	1
SSc	systemic sclerosis	1
ILD	interstitial lung disease	2
HRCT	high resolution computed tomography	3
PFTs	pulmonary function tests	3
DLCO	diffusion capacity for carbon monoxide	3
FEV$_1$	forced expiratory volume in 1 second	3
FVC	forced vital capacity	3
TLC	total lung capacity	3
CO	Carbon monoxide	3
RV	residual volume	4
mm Hg	millimeter mercury	4
BMD	bone mineral density	7
RAM	regression attention maps	8
DNNs	deep neural networks	13
MSE	mean squared error	14
MSD	mean surface distance	14
HD95	95th percentile Hausdorff distance	14
DSC	Dice similarity coefficient	14
STD	standard deviation	14
EAT	equally alternating training	15
FAT	focused alternating training	15
COPD	chronic obstructive pulmonary disease	16
CNNs	convolution neural networks	25
CTD	connective tissue disease	26
UCTD	undifferentiated connective tissue disease	29
VEDOSS	very early diagnosis of systemic sclerosis	29

CI	confidence intervals	34
ICC	intra-class correlation coefficient	34
κ	Cohen's linearly weighted kappa	34
GANs	generative adversarial networks	45
Perc85	85th percentile density	53
BiLSTM	bidirectional long short-term memory	54
GAP	global average pooling	57
FC	fully connected	57
TL	transfer learning	58
MAPE	mean absolute percentage error	59
$\mathbf{V_{lung}}$	lung volume	61
$\mathbf{V_{vessel}}$	vessel volume	61
MLPs	multi-layer perceptions	79
SA	set abstraction	79
PNN	point cloud neural network	78
GNN	graph neural network	78
GCN	graph convolution network	84
GATConv	graph attention convolution	84
IN	instance normalization	84
CCNorm	coordinate center normalization	88
LR	learning rate	88
MIS	medical image segmentation	111
TP	true positive	112
FN	false negative	112
FP	false positive	112
TN	true negative	112
HD	Hausdorff distance	114
MDSD	median surface distance	115
PyPI	Python Package Index	115

1

Introduction

This thesis focuses on the automatic analysis of chest Computed tomography (CT) in systemic sclerosis (SSc) using deep learning. In this introduction, we provide an overview of lung anatomy and pulmonary circulation. We then introduce systemic sclerosis and two tools that aid the diagnosis: pulmonary function tests and high-resolution CT imaging. We proceed with presenting deep learning methods on chest CT, and their potential to aid in the analysis. At the end of the introduction, we provide an outline of this thesis.

1.1 Pulmonary anatomy

The human lungs consist of a left and right lung, as shown in Figure 1.1, with similar but asymmetrical anatomy. Each lung is composed of lung lobes, divided by fissures. The right lung has two fissures; the horizontal and oblique fissures, and three lobes, namely the right upper lobe (RUL), the right middle lobe (RML) and the right lower lobe (RLL). The left lung has two lobes, the left upper lobe (LUL) and the left lower lobe (LLL), divided by the oblique fissure in the left lung [1].

The major function of the lungs is to perform gas exchange, which requires both ventilation of air and blood circulation (see Figure 1.2). This blood supply contains deoxygenated blood and travels to the lungs where erythrocytes, also known as red blood cells, pick up oxygen to be transported to tissues throughout the body [1].

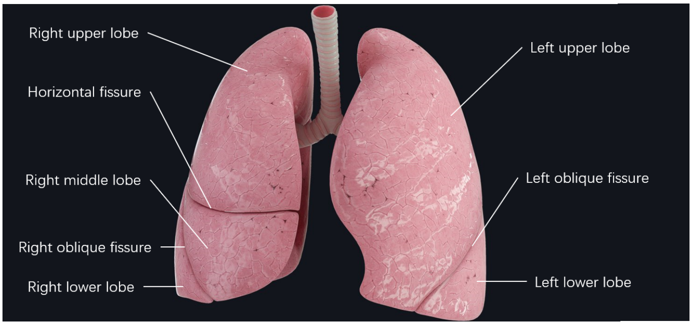


Figure 1.1: Lung lobes (modified and adopted from `https://3d4medical.com/blog/auscultation-of-the-lungs`).

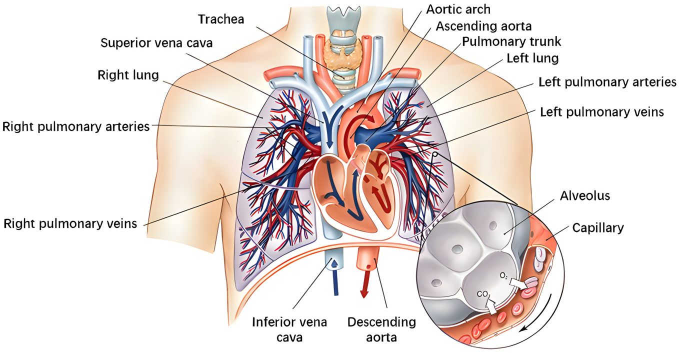


Figure 1.2: Pulmonary circulation (modified and adopted from `https://3d4medical.com/blog/auscultation-of-the-lungs`).

The pulmonary artery is a vessel that arises from the pulmonary trunk and carries deoxygenated, arterial blood to the alveoli. The pulmonary artery branches multiple times as it follows the bronchi, and each branch becomes progressively smaller in diameter. One arteriole and an accompanying venule supply and drain one pulmonary lobule. As they approach the alveoli, the pulmonary arteries become the pulmonary capillary network. The pulmonary capillary network consists of tiny vessels with very thin walls that lack smooth muscle. The capillaries branch and follow the bronchioles and structure of the alveoli. It is at this point that the capillary wall meets the alveolar wall, creating the respiratory membrane. Once the blood is oxygenated, it drains from the alveoli by multiple pulmonary veins, which exit the lungs through the hilum [1, 2]. The hilum is also the place of entrance of bronchi and pulmonary arteries.

1.2 Systemic sclerosis (SSc)

SSc is an immune-mediated rheumatic disease that is characterized by aberrant immune activation, vascular injury followed by defective neovascularization and impaired remodeling and extensive tissue fibrosis of the skin and internal organs [3, 4]. Although systemic sclerosis is uncommon, it has a high mortality — greater than any other rheumatic disease [3]. Cardiopulmonary involvement including Interstitial lung disease (ILD) importantly increases mortality risk [5, 6, 7]. ILD is present in up to 80% of patients with SSc [8]. The heterogeneous expression of this rare disease poses challenges to both the patient and clinician, particularly with regard to predicting the development of serious internal organ involvement. Although early and accurate diagnosis and classification might improve patient outcomes, clinicians often struggle to diagnose SSc early in the disease course [9]. Screening strategies facilitate timely recognition of life-threatening complications and initiation of targeted therapies to

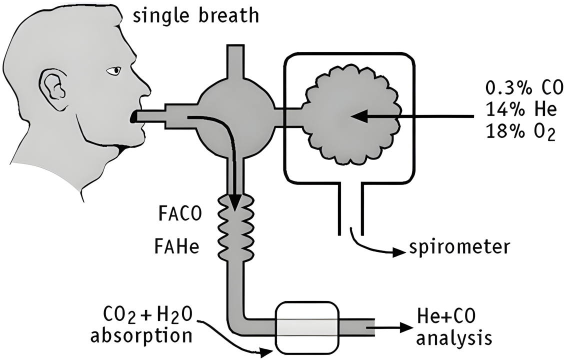


Figure 1.3: Illustration of DLCO measurement. He: Helium. CO: Carbon monoxide. H_2: Hydrogen. H_2O: Water. CO_2: carbon dioxide. FACO: Fraction of Carbon Monoxide in Alveolar Gas. FAHE: Fraction of Helium in Alveolar Gas. (adopted from [17]).

slow their progression [10].

SSc-related ILD (SSc-ILD) is diagnosed by **pulmonary function tests (PFTs)** and chest **high resolution computed tomography (HRCT)** [8]. Although PFTs are central to monitoring ILD progression, they have limited sensitivity in diagnosing ILD [11]. PFTs are associated with intra-individual variation during repeated measurements and extra-pulmonary factors (e.g., oral fibrosis, myopathy and fatigue) can influence the results. Proposed definitions of progressive ILD based on PFT and HRCT, can help identify patients with progressive fibrosing phenotype who could benefit from more aggressive or alternative therapies [12]. A clinical practice guideline defined progressive pulmonary fibrosis as fulfilling at least two of three criteria (worsening symptoms, radiological progression, and physiological progression) within the past year with no alternative explanation in a patient with an ILD other than idiopathic pulmonary fibrosis [13]. Involvement of experienced thoracic radiologists is central for identifying more subtle progression of ILD on HRCT [13]. Invasive procedures, including both bronchoalveolar lavage and lung biopsy, are typically only done in case of diagnostic uncertainty.

1.3 Pulmonary function tests

To evaluate progression of SSc-ILD, various PFTs are used as key measures [14, 15, 16], such as the diffusion capacity for carbon monoxide (DLCO), forced expiratory volume in 1 second (FEV_1), forced vital capacity (FVC) and total lung capacity (TLC).

- DLCO. DLCO is a measure of the lungs' ability to transfer gas from inspired air to the bloodstream [18] (see Figure 1.3). Carbon monoxide (CO) is used for this test because it has a higher affinity for hemoglobin (200 to 250 times that

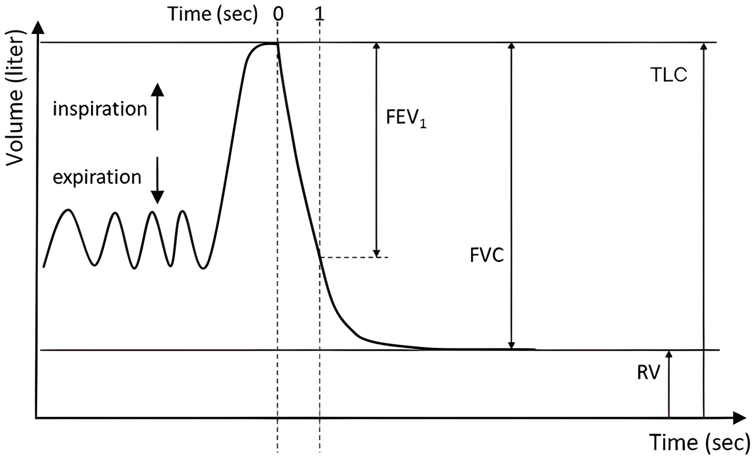


Figure 1.4: Breathing curve measured by spirometry. RV: residual volume, FEV$_1$: forced expiratory volume in 1 second, FVC: forced vital capacity, TLC: total lung capacity (adopted and modified from `https://bronchiectasis.com.au/bronchiectasis/diagnosis-2/lung-function`).

of oxygen), and it follows the same pathway as that of oxygen to finally bind with hemoglobin. Oxygen is not preferred since its uptake is limited by cardiac uptake and total body consumption [19]. DLCO is expressed as mL/min/mm Hg, and represents volume in mL of CO transferred per minute for each millimeter mercury (mm Hg) of pressure difference across the total available functioning lung gas exchange surface [20].

- FEV$_1$. FEV$_1$ is the volume of air exhaled during the first second of the FVC maneuver (Figure 1.4). It tends to be lower in lung diseases that obstruct the airway, such as asthma or emphysema.

- FVC. A breathing curve begins with the patient inhaling as deeply as he or she can. Then the patient exhales forcefully as long as possible; the amount of exhaled volume in this manner is defined as the FVC (Figure 1.4).

- TLC. Even after exhaling as long and hard as possible, some air remains in the lungs; this is called the residual volume (RV). The RV plus the FVC equals the TLC (see Figure 1.4). The RV (and hence TLC) cannot be measured by spirometry. Rather, they must be measured by special tests that require the patient either to breathe an inert gas such as helium (the concentration of which is measured in the expired air, from which the residual volume is calculated) or to sit in an airtight booth in which the pressure is measured as he or she breathes.

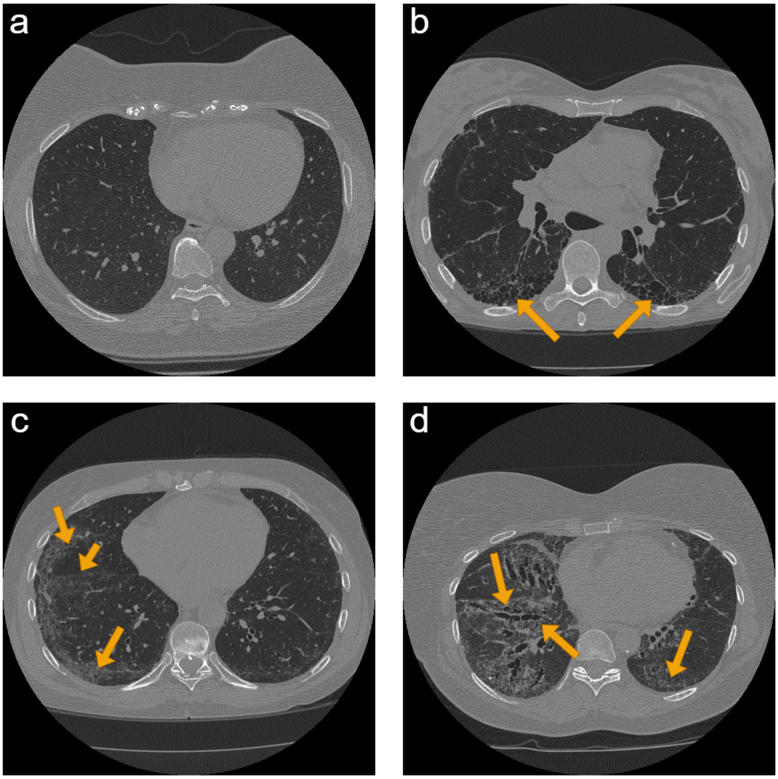


Figure 1.5: Slices of CT scans from a SSc patient with different patterns (arrows). a) normal case, b) with reticulation pattern, c) with ground glass opacity pattern, d) with reticulation and ground glass opacity patterns.

PFTs can, however, not always be performed if there is a risk of disease transmission, e.g. in patients with COVID-19, active tuberculosis or other airborne infectious diseases [21, 22]. In addition, some patients, who have hemoptysis or had surgery in the past month, or other contraindications [23, 24], like aneurysmatic abnormalities and ischemic stroke, are not able to perform PFTs because the forced exhalation during spirometry may increase the risk of complications [9].

1.4 Chest HRCT

According to expert consensus, PFTs should be ordered in all patients with SSc and repeated regularly to monitor the progress of SSc [25]. However, some patients with SSc have normal FVC and DLCO values despite the presence of fibrosis visible on CT [26]. Therefore, CT also plays an important role in the accurate diagnosis of SSc.

CT, as a non-invasive imaging technique, is the gold standard for the detection of ILD in SSc disease [8]. A chest CT scan offers a more intricate visualization compared to a standard chest X-ray. It captures multiple views of the lungs, which are then combined into three-dimensional, cross-sectional representations, visualizing the organs' dimensions, contours and internal architecture. CT could provide valuable findings including the pattern of ILD (e.g. ground glass opacity and reticulation shown in Figure 1.5) and the severity or extent of fibrosis, findings that correlate with disease prognosis [27]. Expert consensus guidelines recommend that CT should be performed in all SSc patients to screen for ILD [25].

1.5 Deep learning on chest HRCT

Although HRCT scans could provide detailed information, there are still some challenges for the diagnosis of lung disease from CT scans. For example, an HRCT normally includes over 1000 slices [28], which significantly increases the diagnostic workload for radiologists. In addition, image interpretation by human experts is limited due to human subjectivity, the large variations across interpreters, and fatigue [28]. In order to overcome these challenges, deep learning techniques were introduced to automate the analysis of chest CT images.

Machine learning, and specifically deep learning, can be divided to supervised, unsupervised, semi-supervised and weakly-supervised learning. Typically, deep learning models are trained using labeled data, called supervised learning. However, for tasks where manually generating labels is laborious and expensive, the use of unsupervised learning methods is of significant value [29]. Unsupervised learning tries to reveal the structure within the data on its own [30]. Semi-supervised learning utilizes both labeled and unlabeled data. Finally, weakly supervised learning tries to learn from coarse-grained or inaccurate labels [31].

Deep learning can be applied to a variety of areas in CT imaging, but not limited to, the following research areas.

- **Image segmentation**, which aims to assign labels to pixels so that the pixels with the same label form a segmented object. For analysis and quantification from chest CT scans, automated segmentation of major anatomical structures or lesions is an important prerequisite [28]. For instance, to quantify the severity of SSc-ILD on CT scans, it is normally necessary to segment the region of SSc-ILD and lungs using deep learning. Another example is COVID-19 severity scoring, which could be automatically derived from the lobe [32] and lesions segmentation results [33] by computing the percentage of affected parenchymal tissue per lobe. For lung and lobe segmentation, a great number of deep-learning-based methods outperformed the best interactive methods [28].

- **Landmark detection**, which aims to localize the anatomical landmarks [34, 35] from 3D CT images. It can help experts to locate areas or scans of interest before detailed analysis of disease [36]. In recent years, there are a great number of works on landmark localization on chest CT, e.g. localization of the upper and lower edge of lungs [37], the lumbar vertebra [38, 39, 40], localization of the heart, ascending aorta, aortic arch and descending aorta [36, 41].

- **Biomarker estimation**, which aims to estimate biomarkers directly from chest CT images. For example, the estimation of Agatston scores [42], coronary calcium scores [43, 44], bone mineral density (BMD) and the percentage of lung emphysema [45].

Despite the great success of deep learning in the aforementioned tasks, several significant challenges exist in the realm of automatic analysis of lung CT imaging for systemic sclerosis (SSc). First, the main challenge is the limited availability of annotated medical imaging datasets, which restricts the training and validation of deep learning models. High-quality labeled data requires experienced experts and remains labor-intensive and time-consuming. Second, labels are normally unbalanced, because normally most patients just have slight disease while rarely patients have very severe disease. This limits the generalizability of deep learning models. Third, training deep learning models on 3D HRCT images requires substantial computational resources of GPU for both processing and storage. As a compromise for limited GPU resources, low-resolution CT is often a necessary solution.

In this thesis, we explore different methods to overcome these challenges. For instance, to overcome the lack of large annotated dataset, we use non-annotated datasets by semi-supervised learning in Chapter 2 and use synthetic CT images to augment training dataset in Chapter 3. To overcome the requirement of high GPU memory for 3D HRCT images, we converted the grid 3D HRCT images to point clouds and graphs, and developed networks based on them in Chapter 5. This decreased the required size of datasets and models exponentially.

1.6 Thesis outline

The aim of this thesis is to develop automatic methods focusing on quantifying disease severity of SSc disease based on CT images. We design two routes, direct and indirect, to achieve this aim. The research topics of each chapter and the connections between them are summarized in Figure 1.6. On the indirect route, we first obtain the segmentation of lungs, lobes and vessels (Chapter 2). Then we estimate PFT based on the segmented vessels (Chapter 5). On the direct route, we aim to directly estimate the PFT and score ILD from CT in Chapters 4 and 3, respectively.

Chapter 2 describes the developed deep-learning-based network for lobe segmentation. We propose a multi-task semi-supervised model that can leverage information of multiple structures from unannotated datasets and datasets annotated with different structures. A focused alternating training strategy is presented to balance the different tasks. We evaluate the trained model on an external independent CT dataset. Due to the lack of a Python package to calculate complete metrics for medical image segmentation, we develop and publish a user-friendly Python package to calculate the overlap-based and distance-based segmentation metrics (see **Chapter 8 Supplementary material**).

Chapter 3 presents the developed deep learning framework to automate SSc-ILD scoring. The automated framework is a cascade of two neural networks. The first network selects five craniocaudal positions for scoring. Subsequently, for each level, a second network estimates the ratio of three patterns to the total lung area. To overcome the score imbalance in the second network, we propose a method to augment the training dataset with synthetic data. To explain the network's output, a heat map method is introduced to highlight the candidate ILD regions.

Chapter 4 shows a proposed deep-learning-based framework to automatically estimate PFT results from chest CT scans of SSc patients. We use segmented lungs and vessels to mask the CT images separately to explore how different regions influence the estimation of PFTs. We also propose regression attention maps (RAM), which can show the contribution of different regions.

In **Chapter 5** we extend the work of **Chapter 4** to improve PFT estimation performance. We develop a point cloud neural network (PNN-Vessel) and graph neural network (GNN-Vessel), based on the point cloud data and graph data of centerlines, respectively. We also explore how to achieve the best estimation performance by combining different networks.

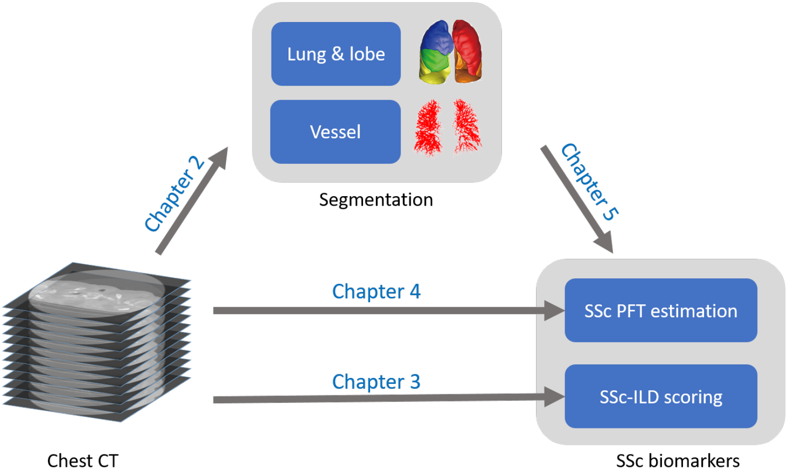
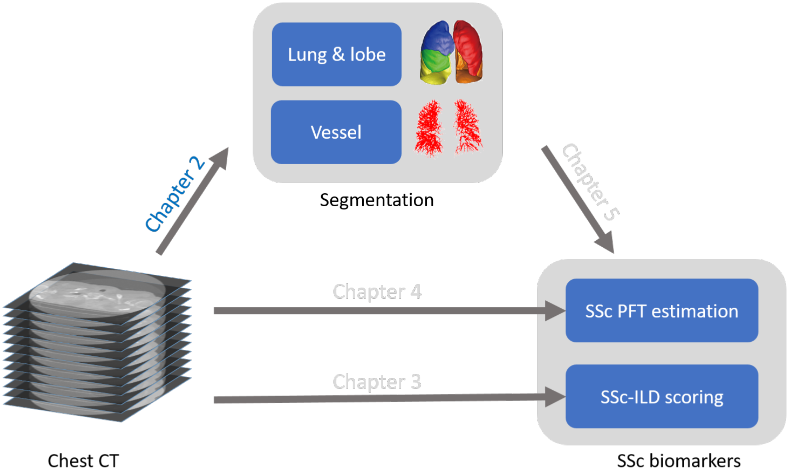


Figure 1.6: Overview of the research topics in this thesis.

Chapter 6 and 7 summarize the overall achievements, implications and future work of this thesis in English and Dutch, respectively.

2

Multi-task semi-supervised learning for pulmonary lobe segmentation



This chapter was adapted from:

Jia, Jingnan, Zhiwei Zhai, M. Els Bakker, Irene Hernández-Girón, Marius Staring, and Berend C. Stoel. "Multi-task semi-supervised learning for pulmonary lobe segmentation." In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pp. 1329-1332. IEEE, 2021.

Abstract

Pulmonary lobe segmentation is an important preprocessing task for the analysis of lung diseases. Traditional methods relying on fissure detection or other anatomical features, such as the distribution of pulmonary vessels and airways, could provide reasonably accurate lobe segmentations. Deep learning based methods can outperform these traditional approaches, but require large datasets. Deep multi-task learning is expected to utilize labels of multiple different structures. However, commonly such labels are distributed over multiple datasets. In this chapter, we proposed a multi-task semi-supervised model that can leverage information of multiple structures from unannotated datasets and datasets annotated with different structures. A focused alternating training strategy is presented to balance the different tasks. We evaluated the trained model on an external independent CT dataset. The results show that our model significantly outperforms single-task alternatives, decreasing the mean surface distance from 7.174 mm to 4.196 mm. We also demonstrated that our approach is successful for different network architectures as backbones.

2.1 Introduction

Computed Tomography (CT) plays an important role in the diagnosis of lung diseases. The appearance of these diseases is, however, diverse and complex. Some lung diseases predominantly affect certain lobes [46], as each lobe has an independent airway and vascular system. Therefore, lobe segmentation is an important preprocessing step in the automated interpretation of lung CT, in order to quantify lung disease in specific regions of interest.

Traditional lobe segmentation algorithms [47] combine information from fissures, bronchi and pulmonary vessels. Recently, deep neural networks (DNNs) improved the performance in lobe segmentation considerably [48, 49]. However, most of them require hundreds [49] or even thousands [48] of annotated CT scans, which is both laborious and time-consuming. Since the lack of large annotated datasets is a common challenge in medical imaging research, we propose to pool several smaller datasets, each with annotations of different structures, and present a strategy to leverage these multiple annotations.

Different methods utilizing multiple annotations have been proposed, based on multi-task or semi-supervised learning. Multi-task models were proposed to extract multi-label information from one dataset [50], or to pool several different datasets of different organs for pre-training or transfer learning [51]. Semi-supervised learning was proposed for brain MRI segmentation, where a segmentation network and a reconstruction network, sharing the same encoder, were trained with annotated data and unannotated data, respectively [52].

In this chapter, we combined multi-task and semi-supervised learning for lobe segmentation. Since the distribution of vessels can help improve lobe segmentation, auxiliary chest CT datasets with vessel annotations were added to train our model. This requires however a proper balance between the training of the different subnets.

The main contributions of this chapter are therefore: 1) A multi-task semi-supervised network for lobe segmentation that can utilize information from distinct datasets with annotations of different anatomical structures; and 2) A focused alternating training strategy to let the model train different tasks alternatively on different datasets, and still focus on the main task.

2.2 Methods

2.2.1 Model design

Our network is composed of three subnets, sharing the same encoder (Figure 2.1): two segmentation nets for segmenting lobes and vessels (called LobeNet and VesselNet, respectively) and one reconstruction net (ReconNet). The two segmentation nets are adapted from V-Nets [53], consisting of an encoder and decoder (see Figure 2.2). The

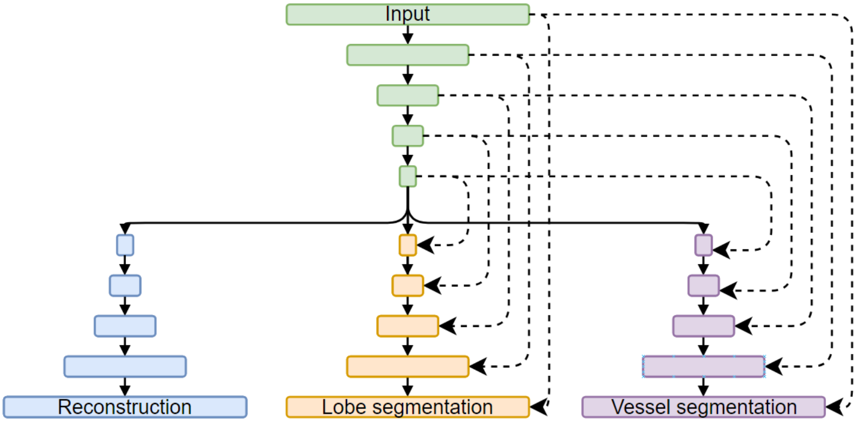


Figure 2.1: Framework of proposed model. Solid lines represent convolution paths; dashed lines denote skip connections.

only difference between the two segmentation nets is the number of output channels of the last layer. LobeNet outputs six channels, representing one background and five lobes. VesselNet has two output channels, representing background and vessels. The ReconNet architecture is similar to the segmentation nets. It omits, however, any skip connections from encoder to decoder, in order to introduce an information bottleneck. Because skip connections would allow the network to copy information from early layers and skip the dimensionality reduction in the encoder [52].

The model is fed with a pair of 3D patches at different fields of view, to provide local and global context (see Figure 2.2). One patch is cropped from the CT images with the original resolution, and the other is cropped from a down-sampled image with the same center as the first one.

2.2.2 Loss function and evaluation metrics

Because of the imbalanced labels in the lobe and vessel datasets, the two segmentation tasks were trained using weighted Dices: $loss_{seg} = 1 - \sum_{i=1}^{m} \frac{Dice_i}{V_i}$, where $Dice_i$ represents the Dice of i^{th} class, and V_i represents the volume of i^{th} class. The mean squared error (MSE) was used as the loss function to reconstruct the input images. To evaluate the model performance, the mean surface distance (MSD), 95th percentile Hausdorff distance (HD95) and average Dice similarity coefficient (DSC) with standard deviation (STD) were calculated using a publicly available tool developed by us [54] (see **Chapter 8 Supplementary material** of the thesis). To test the significance of the differences between two pairs of results, a Wilcoxon signed rank test was used. A P-value less than 0.05 is considered to be statistically significant.

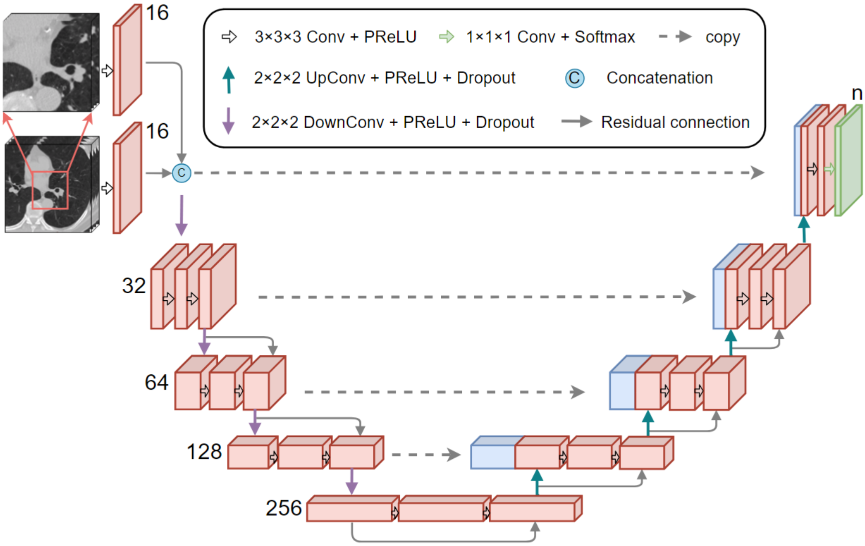


Figure 2.2: Architecture of the segmentation subnets.

2.2.3 Training strategies

Since the dataset of each subnet has its own annotation, the three subnets cannot be trained jointly. Therefore, we examined two training strategies, where different subnets are trained alternatingly with different datasets (see Figure 2.3).

The first training strategy is called equally alternating training (EAT), where the gradients of each subnet are updated once during each iteration. In the second training strategy, the gradients of the main subnet (LobeNet) is updated more frequently (twice in each iteration), i.e. after finishing one of the auxiliary subnets (VesselNet or ReconNet). Since this strategy focuses more on the LobeNet, we called this focused alternating training (FAT).

2.2.4 Learning rate optimization

Training a multi-task network is challenging, because different tasks have different loss scales and require different learning rates. Balancing different tasks is required so that they can help the main task instead of competing against it. Therefore, we introduce an adaptive learning rate to ensure the step sizes for the auxiliary tasks are always smaller than for the main task.

The learning rate of LobeNet was fixed as 1e-4, whereas for VesselNet and ReconNet they were initialized at 1e-5, and subsequently updated adaptively during training: $lr = \lambda \cdot lr_{lobe} \cdot \frac{loss_{lobe}}{loss_{lobe}+loss_{aux}}$, where $loss_{aux}$ is the auxiliary loss, and λ controls the adaptive rate. In this chapter $\lambda = 0.1$ was used.

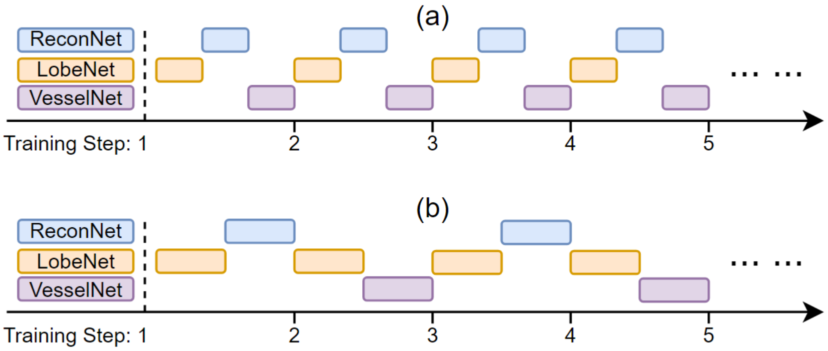


Figure 2.3: Training strategies. (a) equally alternating training; (b) focused alternating training.

2.3 Dataset and data augmentation

2.3.1 Dataset

Four datasets were used: two separate datasets from our institution, GLUCOLD [55] and SSc [56], were used for segmenting lobes and vessels, respectively; one dataset, LUNA16 [57], for the reconstruction task; and an independent dataset, LOLA11 [55], was used for external testing. This study was approved by the ethics committees of Leiden University Medical Center.

GLUCOLD consists of 22 CT scans of chronic obstructive pulmonary disease (COPD) patients. Masks were generated by region growing and further corrected by two experienced researchers. The data was divided into three subsets for training (17), validation (1) and testing (4).

SSc contains 77 cases with systemic sclerosis. A Graph-cuts method [58] was applied to obtain initial vessel masks. We selected 55 high quality segmentations to form the dataset for VesselNet, and divided them into training (50) and test data (5). As our main goal was to segment lobes, small errors in vessel masks were considered acceptable.

LUNA16 is composed of 888 CT scans, selected from the LIDC dataset, initially intended for lung nodule analysis. By excluding all annotations, only the CT data was used for unsupervised learning in ReconNet. Please note that in principle more images could be added into the training dataset for ReconNet since it does not require annotations.

LOLA11 includes 55 CT scans from various sites. It was not used for training or validation and serves as an independent performance evaluation. LOLA11 organizers annotated visible-only fissures on 9 coronal slices in each case [55]. Our expert (M.E.B) verified the annotations, and corrected them where needed. MSD was calculated

Table 2.1: Performance (DSC/MSD ± STD) of the different models in GLUCOLD. MI: Multi-scale input. AL: adaptive learning rate. ST: single-task training. MT: multi-task training with FAT. **Bold** indicates the best performance (highest DSC or lowest MSD).

MI	**AL**	**DSC**		**MSD (mm)**	
		ST	**MT**	**ST**	**MT**
	✓	0.952±0.022	0.964 ± 0.019	1.834±0.840	1.025 ± 0.480
			0.967 ± 0.017		0.868 ± 0.369
✓		0.960±0.018	0.969 ± 0.015	1.184±0.325	1.003 ± 0.366
	✓		**0.970 ± 0.016**		**0.765 ± 0.294**

based on those slices where fissures were annotated.

2.3.2 Data augmentation

Because of GPU memory limitations, it was not feasible to input the whole high-resolution 3D CT images into the model. Therefore, two 3D patches of size 144 × 144 × 96 voxels with different scales were extracted from the CT images. The images of the training dataset were augmented on-the-fly by linear transformations including random shifts (±5%), rotations (±5%), shearing (±5%), and scaling (±5%).

2.4 Experiments and results

2.4.1 Implementation details

The model was implemented using TensorFlow 1.15. The Adam optimizer was used, batch size was set to 1, and the total number of training steps for the main task (LobeNet) was fixed to 100,000. Multithreading was applied to accelerate data preprocessing. The code to replicate the experiments has been released at `https://github.com/Jingnan-Jia/lobeseg`. Training and validation was performed on an Intel(R) Xeon(R) CPU Gold 6126 @ 2.6GHz machine with 90 GB memory. A single GPU NVIDIA GeForce RTX 2080TI with 11 GB memory was used to accelerate training.

2.4.2 Impact of multi-scale input & adaptive learning rate

We first trained a single LobeNet with single-patch input as a baseline. Then the different techniques were applied to improve it and their effects were examined. The performance of the different techniques is shown in Table 2.1. It can be seen that compared with single-task learning, all multi-task models achieved significant improvements to varying extents. This confirms that information from other anatomical structures can improve lobe segmentation. Among these multi-task models, the model with adaptive learning rate and multi-scale input performed best.

Table 2.2: Performance of combinations of different tasks with multi-scale input and adaptive learning rate on GLUCOLD. VS: VesselNet. RC: ReconNet. **Bold** indicates the best performance (highest DSC or lowest MSD)

Architecture	DSC	MSD (mm)
LobeNet	0.960 ± 0.018	1.834 ± 0.325
LobeNet +RC	0.960 ± 0.027	1.541 ± 0.513
LobeNet +VS	0.968 ± 0.015	0.827 ± 0.339
LobeNet +RC+VS (EAT)	0.964 ± 0.017	1.032 ± 0.495
LobeNet +RC+VS (FAT)	**0.970 ± 0.016**	**0.765 ± 0.294**

2.4.3 Impact of auxiliary tasks & training strategies

Based on multi-scale input and adaptive learning rate, we progressively introduced different auxiliary tasks. The results are presented in Table 2.2. It shows that the introduction of VesselNet or ReconNet alone can improve the performance of LobeNet to varying extents. The combination of the three subnets performed better than single LobeNet but worse than LobeNet + VesselNet if the equally alternating training strategy was used. The focused alternating training achieved the best performance. Therefore, focusing to the main task was successful, while still making use of the auxiliary tasks.

2.4.4 Comparison with existing networks

To compare our model with existing methods, we replaced LobeNet by FRV-Net [50] and applied multi-task training strategies to it. The models were tested independently on the corrected LOLA11 (Table 2.3). Compared with the single-task LobeNet, our model achieved a significant ($p<0.005$) improvement. Furthermore, compared with FRV-Net and its multi-task version, our proposed model still achieved a competitive performance. Moreover, our proposed multi-task training strategies also worked for FRV-Net to some extent. This indicates the generalizability of our methods.

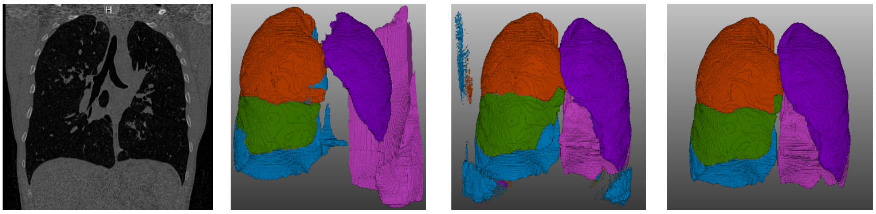
Figure 2.4 and Figure 2.5 show the qualitative results of different networks. From the 3D and 2D views, it can be seen that the initial segmentation from LobeNet has significant false positive voxels outside the lungs. The segmentation from FRV-Net could not predict the oblique fissure exactly. Our model was able to precisely predict both fissures and lung surfaces.

2.5 Conclusions

In this chapter, a multi-task semi-supervised learning model was proposed for pulmonary lobe segmentation, which can utilize unannotated datasets and annotated datasets of different anatomical structures. Multi-scale input, adaptive learning rate and focused alternating training strategy were introduced to balance different tasks.

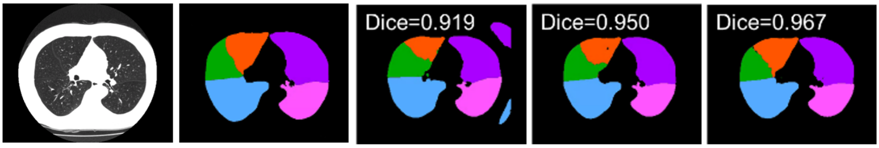
Table 2.3: Comparison of single-task and multi-task models with LobeNet or FRV-Net as the main subnet. †denotes significantly better than the single-task model ($p<0.05$ in Wilcoxon signed rank test).

Architecture	MSD (mm)	HD95 (mm)	Architecture	MSD (mm)	HD95 (mm)
LobeNet	7.124 ±3.432	27.980 ±11.908	FRV-Net	5.847 ±2.990	27.944 ±15.541
LobeNet +RC+VS	$4.696^{\dagger}$ ±2.813	$19.383^{\dagger}$ ±9.904	FRV-Net +RC+VS	5.135 ±3.705	22.528 ±17.269



Original CT LobeNet FRV-Net Our model

Figure 2.4: 3D view of results from one example in LOLA11.



Original CT Ground truth LobeNet FRV-Net Our model

Figure 2.5: 2D view of results from one example in GLUCOLD.

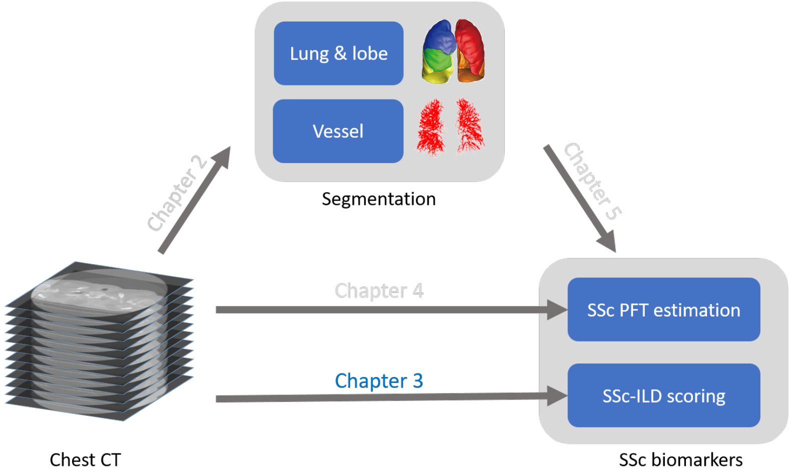
Experiments show that our multi-task semi-supervised model outperformed single-task LobeNet and FRV-Net. Considering it has been successfully implemented on LobeNet and FRV-Net, our model along with the training methods shows the potential to improve other DNN architectures as well.

2.6 Acknowledgements

We would like to thank the LOLA11 Challenge organizers for the data collection and the creation of ground-truth labels. This work was performed using the computational resources from the Academic Leiden Interdisciplinary Cluster Environment (ALICE) provided by the Leiden University. This work is supported by the China Scholarship Council No.202007720110. The authors report no conflicts of interest.

3

Explainable fully automated CT scoring of systemic sclerosis related interstitial lung disease by cascaded regression neural networks and its comparison with experts



This chapter was adapted from:

Jia, Jingnan, Marius Staring, Irene Hernández-Girón, Lucia JM Kroft, Anne A. Schouffoer, and Berend C. Stoel. "Prediction of lung CT scores of systemic sclerosis by cascaded regression neural networks." In *Medical Imaging 2022: Computer-Aided Diagnosis*, vol. 12033, pp. 837-843. SPIE, 2022.

Jia, Jingnan, Irene Hernández-Girón, Anne A. Schouffoer, Jeska K. De Vries-Bouwstra, Maarten K. Ninaber, Julie C. Korving, Marius Staring, Lucia JM Kroft, and Berend C. Stoel. "Explainable fully automated CT scoring of systemic sclerosis related interstitial lung disease by cascaded regression neural networks and its comparison with experts." (*submitted*).

Abstract

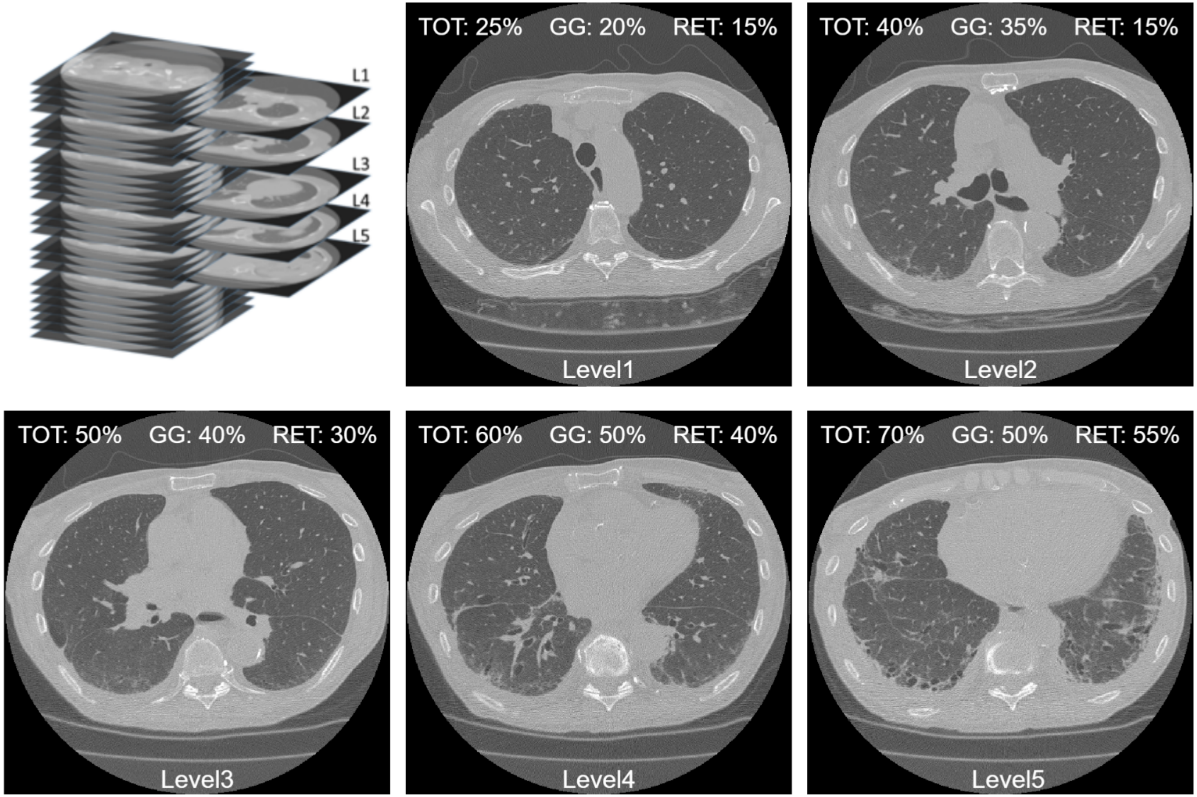
Visual scoring of interstitial lung disease in systemic sclerosis (SSc-ILD) from CT scans is laborious, subjective and time-consuming. This study aims to develop a deep learning framework to automate SSc-ILD scoring. The automated framework is a cascade of two neural networks. The first network selects the craniocaudal positions of the five scoring levels. Subsequently, for each level, the second network estimates the ratio of three patterns to the total lung area: the total extent of disease (TOT), ground glass (GG) and reticulation (RET). To overcome the score imbalance in the second network, we propose a method to augment the training dataset with synthetic data. To explain the network's output, a heat map method is introduced to highlight the candidate interstitial lung disease regions. The explainability of heat maps was evaluated by two human experts and a quantitative method that uses the heat map to produce the score. The results show that our framework achieved a κ of 0.66, 0.58, and 0.65, for the TOT, GG and RET scoring, respectively. Both experts agreed with the heat maps in 91%, 90% and 80% of cases, respectively. Therefore, it is feasible to develop a framework for automated SSc-ILD scoring, which performs competitively with human experts and provides high-quality explanations using heat maps.

3.1 Introduction

Systemic sclerosis (SSc) is a rare autoimmune connective tissue disease affecting different organs with high mortality [59], of which interstitial lung disease (ILD) is the primary cause [60]. The extent of interstitial lung disease in systemic sclerosis (SSc-ILD) on lung CT images has been identified as an independent predictor of disease progression and mortality in patients with SSc [61]. Quantification of SSc-ILD extent is also needed for treatment initiation and evaluation of its efficacy [60]. Several scoring systems have been proposed to quantify SSc-ILD from chest CT scans [34] and the most useful and commonly used quantitative method in the clinical setting is proposed by Goh and colleagues [34, 35]. In this scoring system, CT images are scored in five axial slices, corresponding to levels of: a) origin of the great vessels; b) main carina; c) pulmonary venous confluence; d) halfway between the third and fifth level; e) 1 cm above the right hemi-diaphragm [35]. At each level, three visual scores are estimated as the percentage of total lung area that is covered by: total disease extent (TOT), ground-glass opacities (GG) and reticular patterns (RET), as shown in Figure 3.1. TOT area is the union of the areas from GG and RET, making TOT scores less than or equal to the sum of GG and RET scores. Each of these scores is used in risk stratification, where TOT and RET are associated with mortality [35], while GG can differentiate SSc-ILD from idiopathic pulmonary ILD [62].

Nonetheless, visual scoring remains a challenging task, because of difficulties in recognizing different patterns and estimating its ratio to the whole lung. From Figure 3.1 it is conceivable how difficult it is to identify different patterns and estimate their ratios for each level, especially when GG and RET overlap. Therefore, an atlas was published to provide a guiding consensus document to reduce inter-observer variability [63]. Despite this, ILD scoring is still laborious and dependent on rater experience. Therefore, an automatic scoring tool is needed to overcome these limitations [64, 65] and to make the scoring available for clinical practice. An automated scoring tool would consist of two steps: 1) selecting the five levels (axial CT slices) according to anatomical landmarks; and 2) estimating the score for each selected slice by recognizing the different patterns and estimating their proportion to the total lung area. Recently, deep neural networks have been proposed for anatomical level localization [39] and quantification of imaging features [61, 66, 67], which are closely related to the two steps needed for automated ILD scoring. While several methods combined the two steps together to estimate other imaging biomarkers [68, 69, 70, 71], there are few published works applied on SSc-ILD scoring.

The purpose of this study was, therefore, to build a fully automated framework to select the five anatomical levels and score the extent of SSc-ILD for each level directly, without needing manual segmentations. The main contributions of this chapter are as



L1
L2
L3
L4
L5
TOT: 25% GG: 20% RET: 15%
Level1
TOT: 40% GG: 35% RET: 15%
Level2
TOT: 50% GG: 40% RET: 30%
Level3
TOT: 60% GG: 50% RET: 40%
Level4
TOT: 70% GG: 50% RET: 55%
Level5

Figure 3.1: Illustration of five levels in the same CT volume from one patient of systemic sclerosis. Interstitial lung disease scores from human experts are annotated on the top of each image. The level numbers are annotated at the bottom. TOT = total disease extent, GG = ground glass opacity, RET = reticular pattern.

follows:

- To the best of our knowledge, we are the first to propose an automated framework for SSc-ILD scoring without pixel-wise fibrosis annotation.
- We introduced a data synthesis method to generate training images with exact SSc-ILD scores, leading to significant improvement in the SSc-ILD scoring.
- We proposed replacement-based heat maps, which can visually explain the network's output, making the framework more easily acceptable for clinicians. The reliability of heat maps was evaluated by an automatic evaluation method and by two human experts.
- Our framework performed competitively to experienced human experts, while costing only several seconds.

3.2 Related work

An automated scoring framework may consist of two networks: 1) level selection (i.e. localization) from 3D medical images; and 2) scoring of the resulting 2D image slices. This section therefore reviews related studies of these two topics and their application on SSc-ILD.

3.2.1 Automated level selection

Existing methods show that, despite trained on global image-level labels, convolution neural networks (CNNs) have a remarkable ability to localize objects-of-interest [72]. Level selection aims to localize the anatomical level or anatomical landmarks [34, 35] from 3D CT images. In recent years, there are a great number of works on landmark localization in 3D medical images (see Table 3.1), e.g. localization of the upper and lower edge of lungs [37] in chest CT images, the lumbar vertebra [38, 39, 40] in spine CT images, the anatomical landmarks in cardiac ultrasound images [73], anatomical structure localization in CT images [36, 41], probe localization in liver CT images [72]. The network design of these works all consist of a feature extractor followed by several fully connected layers. The feature extractors consist of several CNN blocks (a CNN layer, a normalization layer and an activation layer) with residual connections [40]. Although these works are all developed for non-SSc patients, the success of regression networks on the aforementioned works shows its potential on level selection of SSc patients.

3.2.2 Automated scoring

A large number of deep neural networks has been proposed for scoring tasks in general medical imaging, which were not focused on scoring SSc-ILD. An indirect method is to develop a segmentation network and score images based on lesion segmentation results, such as idiopathic pulmonary fibrosis [74]. The limitation of such an indirect scoring method is that it requires pixel-wise segmentation labels. A direct method is to develop a network to output the score directly without any segmentation. If the scoring output contains less than 5 categories, researchers usually treat it as a classification task, such as Gleason scoring of prostate cancer in histopathology images [75, 76, 77], grading of ulcerative colitis in endoscopic images [78], grading of abnormalities in knee MRI [79], diabetic retinopathy grading in eye fundus images [80], osteoarthritis severity grading in knee MRI [81], fibrosis estimation [82]. When the scoring is a real (floating point) number or contains more than 5 categories, regression neural networks are preferred, e.g. Agatston scoring in chest CT images [42], ventricle function indices estimation in ultrasound images [83], coronary calcium scoring in chest CT scans [43, 44], bone mineral density (BMD) and the estimation of percentage of lung emphysema from CT scans [45]. Because we aimed to estimate the ratio of fibrosis to the total lung area without segmentation, a regression neural network was adopted in our work.

3.2.3 Automated scoring for SSc-ILD

To the best of our knowledge, there are no automated level selection methods published for SSc scoring. In addition, there are only two automated scoring frameworks developed for SSc patients (see Table 3.1). Chassagnon et al. [61] developed networks, which could output the fibrosis areas and severity quantification for SSc patients. However, their work used segmentation networks to output the pixel labels as a basis for computing the final biomarkers, which is time-consuming and requires laborious manual pixel-wise annotations for training. Since pixel-wise annotations for GG and RET are even more difficult to obtain due to unclear boundaries between the two patterns, only TOT patterns have been segmented to assess SSc-ILD. In contrast to only segmenting TOT pattern, Su et al. [84] developed a severity scoring model for connective tissue disease (CTD) associated ILD (CTD-ILD, including SSc-ILD) that could segment GG, RET and honeycombing patterns, separately. This also requires laborious pixel-wise annotations. As far as we know, there are no published methods on automated scoring of SSc-ILD without the need for pixel-wise fibrosis annotations.

3.3 Materials and methods

The proposed two-step framework is shown in Figure 3.2. A level selection network (L-Net) selects the five anatomical levels from the input 3D CT scans. Subsequently, five 2D slices were selected according to the five level positions and an SSc-ILD scoring network (S-Net) estimates three scores (TOT, GG and RET) for each input 2D slice.

3.3.1 Dataset

The dataset was collected retrospectively and consisted of de-identified high-resolution CT scans of 230 SSc patients, who were referred to our multidisciplinary healthcare

Table 3.1: Summary of related works for automated scoring frameworks on medical imaging.

Patients	Task	Network	Target	Dataset
Non-SSc	localization	regression	lung upper and lower edge [37]	chest CT
			lumbar vertebra [40]	spine CT
			anatomical structures [41]	body CT
			anatomical plane landmarks [73]	cardiac ultrasound
			probe localization [72]	liver CT
	scoring	segmentation	idiopathic pulmonary fibrosis [74]	chest CT
		classification	grading of ulcerative colitis [78]	endoscopic
			grading of abnormalitie [79]	knee MRI
			diabetic retinopathy grading [80]	eye fundus
			osteoarthritis severity grading [81]	knee MRI
			fibrosis estiation [82]	chest CT
		regression	ventricle function indices [83]	cardiac ultrasound
			coronary calcium scoring [43, 44]	chest CT
			percentage emphysema [45]	chest CT
			Agatston scoring [42]	chest CT
SSc included	scoring	segmentation	SSc-ILD assessing [61]	chest CT
			CTD-ILD assessing [84]	chest CT

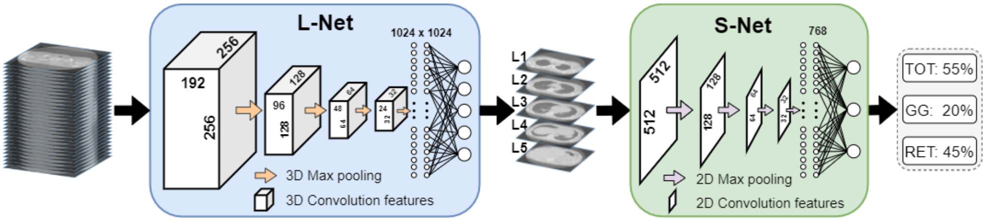


Figure 3.2: Proposed framework for SSc-ILD scoring based on two cascaded neural networks. L-Net outputs five values of anatomical levels. S-Net outputs three values for automatic SSc-ILD scoring. TOT = total disease extent, GG = ground glass pattern, RET = reticular pattern.

program [85] for suspected SSc (Table 3.2). The CT scans were performed with an Aquilion 64 scanner (Canon Medical Systems), with 120 kVp, median tube current 140 mA, median CTDIvol 8.2 mGy, rotation time 0.4 seconds, collimation 64×0.5 mm and 0.8 helical beam pitch [86]. Images were reconstructed with filtered back projection and using an FC86 kernel, 0.64×0.64 mm median pixel spacing, and a slice thickness and increment of 0.5 and 0.3 mm, respectively. Approval of all ethical and experimental procedures and protocols was granted by the Institutional Review Board of the LUMC under Protocol Nos. P09.003/SH/sh, REU 036/SH/sh, REU 043/SH/sh, and B19.008/KB/kb. CT data were accessed for this retrospective study on 18-03-2021. The data was anonymized previously, therefore there was no access to data that could identify individual participants during or after data collection. The dataset was randomly split into three groups with 135, 45 and 50 patients for training, validation and testing, respectively.

Before training the L-Net, all CT scans were resized to a fixed size of $256 \times 256 \times 256$-pixel matrix. After resizing, the pixel spacing along the x and y axis was 1.26 mm. The average spacing along the z-axis was 1.2 mm. CT values were truncated between -1500 HU and 1500 HU. The world positions of five levels for each CT scan were manually selected by a rheumatologist (Observer-1, 5-year experience) and a radiologist (Observer-2, 20-year experience) in consensus. Subsequently, we converted the world positions of the five levels to relative slice numbers in the resampled 3D CT scans (the bottom slice was regarded as number 0, corresponding to the base of the lung) [87]. These slice numbers were used as the ground truth. To increase the image diversity for training the L-Net, we randomly cropped patches with a fixed size of $256 \times 256 \times 192$ (ordered by xyz) on-the-fly during training. These patches always covered all five levels and could also be fitted into the GPU memory of 11 GB.

While L-Net was trained and tested using the down-sampled CT volumes, S-Net used the 2D axial slices of five levels with the original in-plane resolution (512×512) from the aforementioned 230 scans. High-resolution images include fine spatial

Table 3.2: Dataset properties of systemic sclerosis patients.

Characteristic	**Patients, (n=230)**	
Age [years] (standard deviation)	54	(14.0)
Female (%)	191	(83.0)
Interstitial lung disease detected on CT (%)	96	(41.7)
Anti-centromere antibodies positive (%)	88	(38.3)
Anti-topoisomerase antibodies positive (%)	56	(24.3)
Pulmonary arterial hypertension (%)	8	(3.5)
Disease Subset:		
Non-cutaneous (%)	29	(12.6)
Diffuse cutaneous (%)	62	(27.0)
Limited cutaneous (%)	133	(57.8)
Alternative diagnosis* (%)	6	(2.6)

* Morphea scleroderma, undifferentiated connective tissue disease (UCTD), UCTD with Sjögren's syndrome, puffy fingers without systemic disease, and two cases of very early diagnosis of systemic sclerosis (VEDOSS).

details, which can help to distinguish and grade the three fibrosis patterns. All 2D slices were scored in consensus by two experts to obtain the ground truth. Additionally, to evaluate inter/intra-observer agreement 16 patients (80 axial slices) were randomly selected from the testing dataset and the two experts scored them independently. Then they independently scored the same 80 axial slices again after six weeks. The TOT, GG and RET scores can range from 0% to 100%, and were estimated with a precision of five percent (Appendix Figure A3.1), following the protocol by Goh et al [35]. To augment our dataset, two neighboring slices (above and below the chosen slice of each level) were added for training. Because the spacing of neighboring slices is only 0.3 mm, we assumed that these represent the same score. In addition, the 2D training images were augmented on the fly by random rotation (±30°), scale (95% – 105%) and shift (±10 pixels) along the horizontal and vertical direction.

3.3.2 Network design

Inspired by [39], we experimented with different 3D VGG [88] networks as the structure L-Net (Figure 3.3-A), including VGG11, VGG16 and VGG19. Deeper networks like 3D ResNet50 [89] would lead to GPU memory problems with the same input patch size, so no deeper networks were tested further. Therefore, VGG11 was finally selected for L-Net. As for the S-Net, we evaluated different 2D networks with different capacities including SqueezeNet [90], VGG11,16, and 19 [88], ResNet18 [89], ResNet50, ResNeXt50 [91], DenseNet [92], ShuffleNet [93], ConvNeXt [94], and finally decided to adopt ConvNeXt for S-Net due to its state of the art performance (Figure 3.3-B). Compared with the original VGG11 proposed in [88], L-Net extends all convolutional

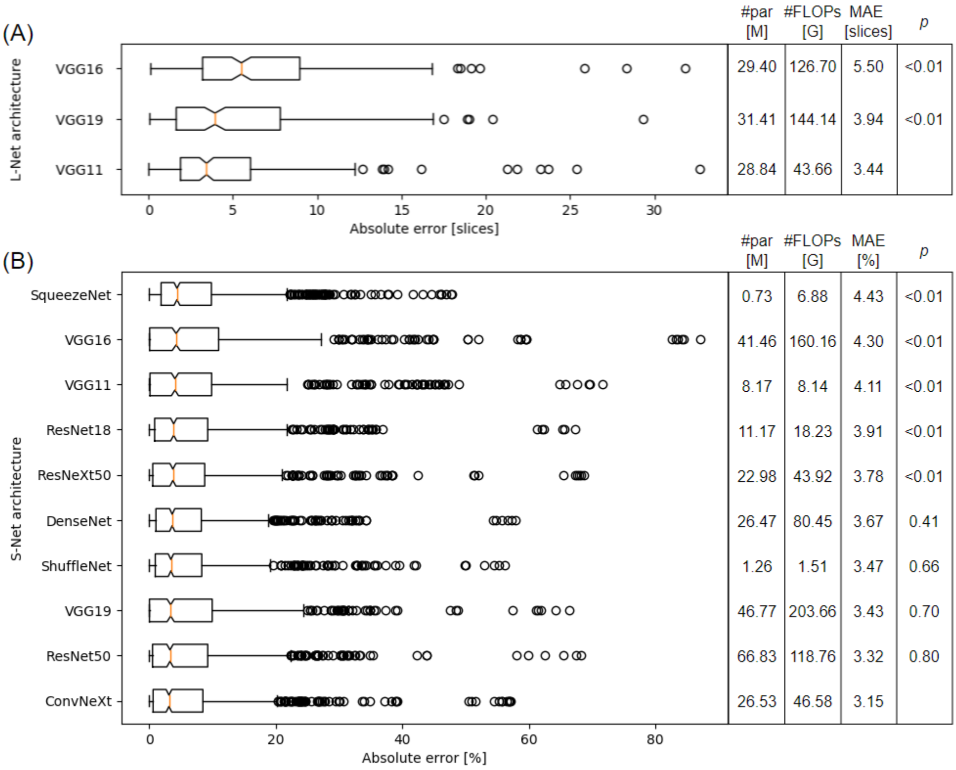


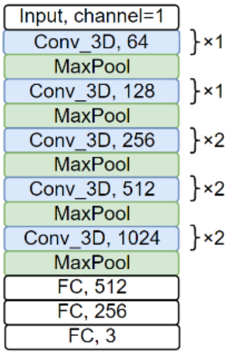
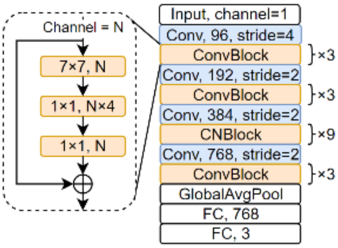
Figure 3.3: Network architecture selection for L-Net (A) and S-Net (B). #par: number of trainable parameters; #FLOPs: number of floating point operations; MAE: mean absolute error. VGG11 and ConvNeXt achieve the lowest MAE for L-Net and S-Net, respectively. P-values in (A) were obtained by the Wilcoxon signed rank test between each network and VGG11; P-values in (B) were obtained by the Wilcoxon signed rank test between each network and ConvNeXt. A P-value less than 0.05 is considered statistically significant.

and max-pooling layers from 2D to 3D. The feature extractor (convolutional layers) of the S-Net was initialized by pre-trained weights from ImageNet [94], while the fully connected layers were initialized randomly using a normal distribution. The architecture and training details of L-Net and S-Net are shown in Table 3.3.

3.3.3 Techniques to overcome label imbalance

From Appendix Figure A3.1, we could find that the score distribution is highly askew —some high scores even do not exist in the training dataset. The noticeable score imbalance with 50% of TOT scores being 0 would limit the networks' performance. Therefore, to ensure a balanced score distribution during training, we developed a sampling method that randomly selects training images with a probability inversely

Table 3.3: Details of network design and training scheme for L-Net and S-Net.

Net	L-Net (VGG-11)	S-Net (ConvNeXt)
Structure	 Input, channel=1 Conv_3D, 64 ×1 MaxPool Conv_3D, 128 ×1 MaxPool Conv_3D, 256 ×2 MaxPool Conv_3D, 512 ×2 MaxPool Conv_3D, 1024 ×2 MaxPool FC, 512 FC, 256 FC, 3	 Channel = N 7×7, N 1×1, N×4 1×1, N Input, channel=1 Conv, 96, stride=4 ConvBlock ×3 Conv, 192, stride=2 ConvBlock ×3 Conv, 384, stride=2 CNBlock ×9 Conv, 768, stride=2 ConvBlock ×3 GlobalAvgPool FC, 768 FC, 3
Dimension	3D	2D
Batch size	4	10
Training time (h)	2.8	4.9
Parameters (M)	28.84	22.53
Training epochs	500	500
Learning rate	0.0001	0.0001

proportional to the ratio of each TOT score [95]. In this way, the scores that appear less frequently (i.e. higher scores) would be used for training more frequently. To further address the label imbalance and to increase data diversity, we synthesized training images with SSc-ILD scores that are lacking in the original dataset, by digitally inserting GG and RET patterns into healthy training images.

The flowchart of data synthesis is shown in Figure 3.4-A. First, we created two patches full of different patterns, one for GG and one for RET, by manually extracting local CT patches from training images with high scores in these two patterns separately. Subsequently, the healthy training images (TOT=0) were augmented by the digital insertion of these patterns. The candidate lesion regions for the pattern insertion were randomly generated by defining up to three ellipses with random centers, orientations and axes lengths (lengths range from 1 to 150 pixels), which were then cropped by the lung mask to ensure the patterns will be inserted in the lung area only. The lung mask was automatically generated by a multi-atlas based automatic lung segmentation algorithm [96]. To avoid introducing unrealistic borders during pattern insertion, the inserted patterns gradually fade out at the edge, by a Gaussian decay in intensity. To increase the pattern diversity of synthetic data, we always applied random rotation (±180°) and scale (95% – 105%) to the patterns before each pattern insertion. The disease severity scores were then adapted according to the area of inserted patterns. Some synthetic image examples and their scores were shown in Figure 3.4-B. The

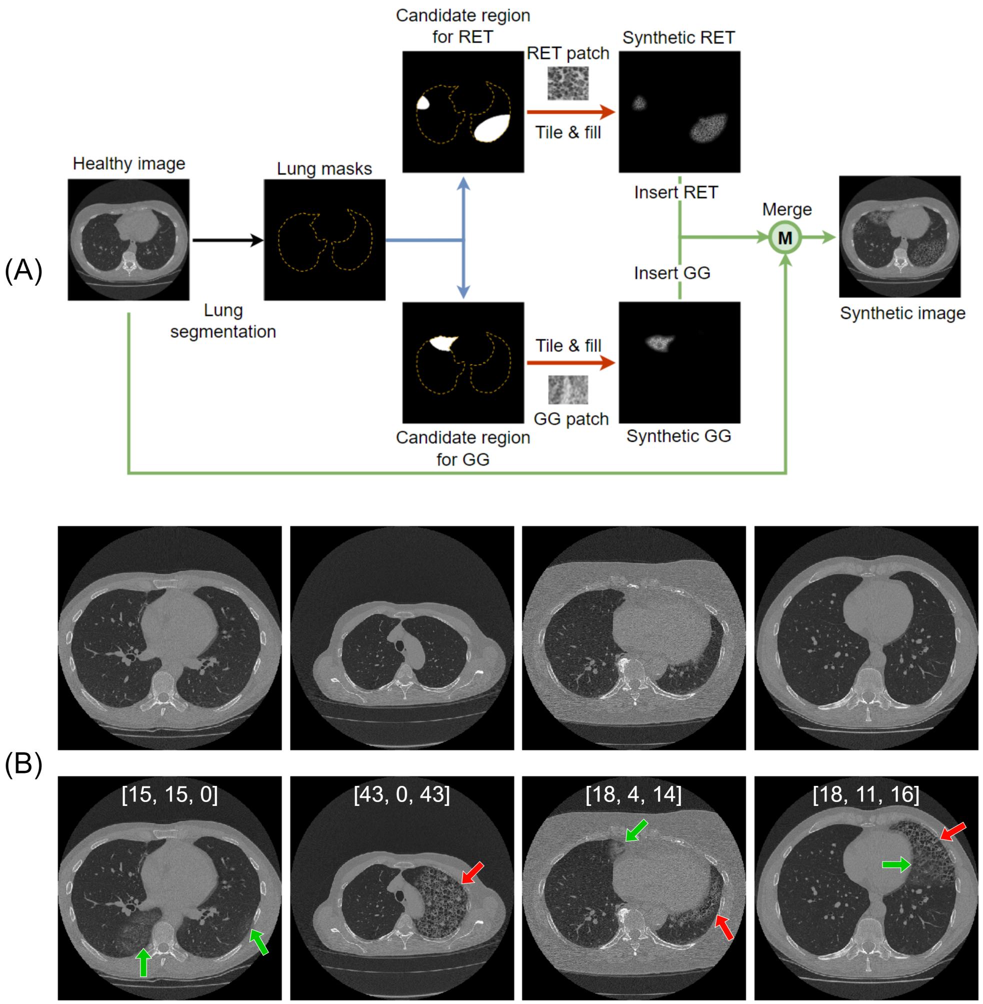


Figure 3.4: Data synthesis flowchart and examples. (A) Flowchart to synthesize images with different disease patterns. Blue arrows indicate the generation of random candidate lesion regions, red arrows indicate the filling of patterns; green arrows indicate the insertion of patterns. (B) Four pairs of synthetic examples. The upper row shows the original images; the lower row shows the corresponding synthetic images. Green arrows point to GG; red arrows point to RET. Different pattern combinations are shown from left to right: only GG, only RET, GG and RET without overlap, GG and RET with overlap. The scores of these synthetic images are shown on the image in the order of [TOT, GG, RET]. TOT = total disease extent, GG = ground glass pattern, RET = reticular pattern.

synthetic data constitute half of the training dataset, while validation and testing were performed on real patient data only.

3.3.4 Heat map visualization: network explainability

Application in clinical practice is limited, if the output of a deep learning network is difficult to explain. Therefore, inspired by the occlusion-based visualization method [97], we developed a replacement-based method to generate heat maps, indicating which areas were important for the S-Net in recognizing different disease patterns. The method details are as follows.

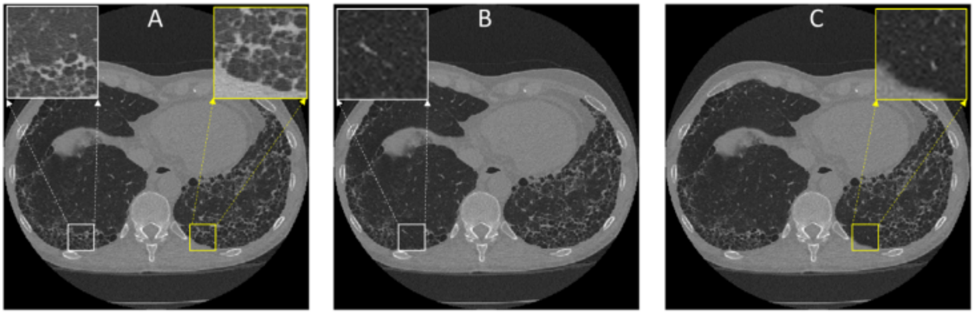


Figure 3.5: Illustration of the replacement-based heat map. (A) Original CT slice. (B) A patch in the lung is replaced by a healthy patch. (C) A patch at the edge of the lung is replaced by a healthy patch while the pixels outside the lung were kept unchanged.

A rectangular patch full of healthy lung tissue, in advance cropped from a lung region of healthy slices, was used to cover the test image from top-left to bottom-right step-by-step, separately. The patch size was 64×64 pixels with a step size of 16. As shown in Figure 3.5-A, the white box and yellow box are two positions with lung fibrosis to different extents. After we replaced the two patches separately with the healthy patch, we can obtain Figure 3.5-B and C. The output score from the S-Net of the original test image was regarded baseline. Each time we slide the healthy patch, the original image was occulted by the healthy patch at a different position. We could obtain the new score from S-Net and record its change at each position. A heat map of the image is then generated using the magnitude of the score change (ΔP). A negative score change ($\Delta P < 0$) implies that the network regarded the original patch as diseased since the score decreased after concealing the area with healthy tissue. If the output score remains unchanged ($\Delta P = 0$), the original patch was already considered healthy. A score increase ($\Delta P > 0$) means that the network produced a false positive, since the inserted healthy patch was apparently classified as diseased. To make sure that the replaced pixels are in the lung area, the rectangular healthy patch was cropped by the lung mask before each replacement (Figure 3.5-C). The patch edge fades gradually by linearly increasing transparency to make it more natural.

This replacement-based heat map was inspired by the occlusion-based visualization method [97]. The difference is that the occlusion-based method would cover the original image using a patch with a constant value, which would introduce artifacts, while our replacement-based method covers the original image using a patch cropped from a healthy CT scan, which still includes lung texture and makes the generated image more natural.

In order to evaluate the performance of the heat maps, blinded to the network's output, two human experts independently rated their agreement with the heat maps using a Likert scale, with five labels (1-5): "Strongly disagree", "Disagree", "Neutral", "Agree" and "Strongly agree" , using dedicated software (Appendix Figure A3.2).

Additionally, we developed an automatic method to evaluate the heat map explainability. By thresholding the heat map, the different patterns were segmented and their areas were divided by the total lung area, to obtain a derived SSc-ILD score. Subsequently, we tested the network's consistency by the correlation between the derived SSc-ILD score and the network's output. The optimal threshold was obtained from the validation dataset by varying the threshold from -4% to 0% and selecting the one with the smallest mean absolute error (MAE) between the derived SSc-ILD score and S-Net output.

3.3.5 Statistical analysis and evaluation

To evaluate our networks, the following statistical analyses were performed by an in-house Python 3.8 script with corresponding libraries.

The MAE, standard deviation (STD) and 95% confidence intervals (CI) are reported. To evaluate the inter-observer agreement, Cohen's linearly weighted kappa (κ) [98] and intra-class correlation coefficient (ICC) [36] were used. κ was calculated by scikit-learn 0.24.2 [99]. ICC was calculated by pingouin 0.4.0 [100] based on a single-rating, absolute-agreement, 2-way mixed-effects model [98]. To statistically test differences between groups, a paired T-test and Wilcoxon signed rank test were performed, as implemented by scikit-learn 0.24.2. A P value of less than 0.05 was considered to indicate a statistically significant difference. All metrics were calculated based on the testing dataset unless stated otherwise.

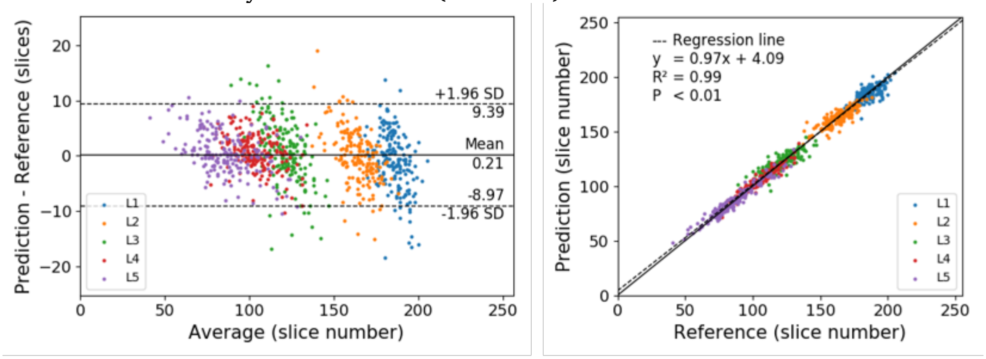
3.3.6 Network implementation details

Our neural networks, L-Net and S-Net, were implemented using PyTorch 1.7.1 (`https://pytorch.org`). For both networks, the loss function was the mean squared error (MSE). The Adam optimizer was used with a learning rate of 1e-4, a weight decay of 1e-4 and 500 epochs. Multithreading was used to accelerate the on-the-fly data augmentation. The workstation for training and validation was equipped with an Intel(R) Xeon(R) CPU Gold 6126 @ 2.6GHz with 90 GB memory and a GPU RTX 2080TI with 11 GB memory. The source code and trained models are published at `https://github.com/Jingnan-Jia/ssc_scoring` to facilitate reproduction of results.

3.4 Experiments and results

3.4.1 SSc-ILD scoring performance

First, we trained and evaluated the L-Net (Figure 3.6) and S-Net (Table 3.4), separately. Subsequently, an end-to-end framework was built as a cascade of the trained L-Net and S-Net (called L&S-Net), in which the input slices for S-Net were automatically selected by the L-Net (Table 3.5). For none of the levels, the automatic scoring results of L&S-Net showed any significant differences as compared to solely S-Net which received the manually selected slices (Table 3.5).



(A) (B)

Figure 3.6: Testing results of the L-Net in selecting slices on the five levels, L1 – L5. (A) Bland-Altman plot and (B) Correlation plot. The average spacing between slices was 1.2 mm. The ICC of the five consecutive levels was 0.72, 0.84, 0.81, 0.96 and 0.97. No significant bias was observed among the five levels (P=0.20, 0.93, 0.42, 0.49, and 0.76, respectively).

3.4.2 Comparison with human experts

The inter- and intra-observer agreement in the sub-group of 16 patients (80 axial slices) from the testing dataset were compared with our proposed method (Table 3.6). The inter-observer agreement was higher during the second scoring session. The intra-observer agreement of Observer-2 was higher than Observer-1, and the inter- and intra-observer agreement in GG scoring was always lower than in TOT and RET scoring.

For scoring TOT, our automatic method was close to the first rating by Observer-1 ($Obs1_{T1}$), but Observer-2 was closer to the consensus than our method. For GG the model had a fair agreement with human consensus, while the observers agreed moderately, and for RET the model's agreement was moderate, but moderate/substantial for observers. Except for the second GG scoring by Observer-2 ($Obs2_{T2}$, $P < 0.05$), other human observations did not perform significantly better than our method. The Bland-Altman plots (Appendix Figure A3.3) illustrate the performance of an individual

Table 3.4: SSc-ILD scoring performance from S-Net with different technique combinations.

Experiments	**DS***	**BS**	**PT**	**TOT**			**GG**			**RET**		
(i)				MAE [%]↓	κ↑	ICC↑	MAE [%]↓	κ↑	ICC↑	MAE [%]↓	κ↑	ICC↑
	-	-	-	8.87 (12.55)	0.42	0.59	6.62 (11.64)	0.31	0.38	6.59 (9.30)	0.42	0.58
(ii)	-	-	✓	7.85 (10.71)	0.53	0.72	5.96 (10.03)	0.45	0.60	5.81 (8.42)	0.53	0.72
	-	✓	-	8.30 (11.35)	0.46	0.62	5.89 (10.88)	0.40	0.48	5.96 (8.70)	0.48	0.65
	✓	-	-	6.74 (9.59)	0.59	0.77	5.07 (9.38)	0.54	0.67	5.08 (7.76)	0.57	0.75
(iii)	-	✓	✓	6.87 (9.61)	0.59	0.77	5.09 (9.50)	0.54	0.66	5.11 (7.45)	0.59	0.78
	✓	✓	-	7.98 (11.31)	0.48	0.64	5.43 (9.82)	0.47	0.60	5.52 (8.56)	0.52	0.69
	✓	-	✓	6.26 (8.67)	0.63	0.82	4.76 (8.81)	0.58	0.70	4.76 (6.70)	0.62	0.82
(iv)	✓	✓	✓	**5.90 (8.77)**	**0.66**	**0.83**	**4.66 (8.83)**	**0.58**	**0.71**	**4.49 (6.70)**	**0.65**	**0.84**

[*] DS = Data synthesis, BS = Balanced sampling, PT = Pre-training, TOT = total disease extent, GG = ground glass pattern, RET = reticular pattern, Mean absolute error (MAE) is followed by the standard deviation (STD) between parentheses, κ = Cohen's weighted kappa, ICC = Intra-class correlation coefficient. ↓ = lower is better, ↑ = higher is better. **Bold** numbers indicate the best performance. (i). Baseline established by training the S-Net from scratch without balanced sampling or synthesized data. (ii). Introducing either pre-trained weights from ImageNet, balanced sampling or data synthesis. (iii). Combination of two of the three techniques. (iv). Combination of all three techniques which obtained the best performance. Therefore, the proposed method contains all three techniques.

Table 3.5: MAE comparison of SSc-ILD scoring between the whole framework (L&S-Net) and sole S-Net.

Level	MAE of TOT*			MAE of GG			MAE of RET		
	LS-Net [%]	S-Net [%]	P	LS-Net [%]	S-Net [%]	P	LS-Net [%]	S-Net [%]	P
1	4.20 ± 7.94	6.28 ± 11.57	0.33	2.34 ± 6.88	2.81 ± 6.71	0.57	3.49 ± 6.25	5.14 ± 9.67	0.42
2	4.61 ± 6.89	4.74 ± 7.46	0.06	3.11 ± 7.67	3.33 ± 7.62	0.88	3.35 ± 5.67	3.72 ± 6.15	0.49
3	5.61 ± 8.00	6.17 ± 8.63	0.64	4.65 ± 8.42	4.82 ± 8.19	0.56	4.33 ± 5.24	4.69 ± 6.13	0.74
4	7.12 ± 8.87	6.88 ± 9.23	0.17	5.88 ± 9.58	5.98 ± 9.59	0.10	4.99 ± 6.75	4.93 ± 6.65	0.90
5	8.12 ± 10.00	7.58 ± 9.99	0.11	7.13 ± 10.49	6.75 ± 10.41	0.40	6.15 ± 8.26	5.74 ± 7.78	0.11
ALL	5.86 ± 8.46	5.90 ± 8.77	0.21	4.56 ± 8.80	4.66 ± 8.83	0.13	4.40 ± 6.55	4.49 ± 6.70	0.28

[*] Mean absolute error (MAE) ± standard deviation (STD) is presented expressed as %, TOT = total disease extent, GG = ground glass pattern, RET = reticular pattern, ALL = Calculated based on the results from all the five levels.

human score ($\mathrm{Obs2}_{\mathrm{T2}}$) and our automatic network.

The average time of fully automated scoring for the five levels is less than ten seconds per patient, while human experts need around ten minutes (around 2.5 minutes to identify five levels and another 7.5 minutes to score three patterns of five levels).

3.4.3 Heat map explanation and its evaluation

The replacement-based heat maps of the automatic scoring for the three different patterns are shown in Figure 3.7 of different patients from the testing subset. The proposed visualization method can show areas of different patterns and display the severity with different colours. The yellow and red areas in the heatmaps (Figure 3.7) denote the negative score change after the area has been covered by a healthy patch, which means that the original patch is "diseased". A red area means more severe and more obvious patterns than a yellow area. The green and blue areas mean that the network produced a false positive, since the inserted healthy patch was apparently classified as diseased. The heat maps can also help to find the cause of errors, as shown in the last row in Figure 3.7, where the GG scoring result (30%) is far lower than the ground truth (90%). From the heat maps, we can see only about 30% of the whole lung was activated (yellow and red area) and the GG pattern was missed in around 1/3 of the right lung (blue area). Alternative heat maps are presented in Appendix Figure A3.4 to indicate false negatives and positives.

From the semi-quantitative evaluation of the heat maps, Observer1 rated the heat maps with “Strongly Agree” or “Agree” in 97.0%, 94.2% and 89.8% of cases for TOT, GG and RET, respectively (Figure 3.8, upper row). Ratings of “Strongly agree” or “Agree” by Observer2 occurred in 84.0%, 85.8% and 70.2% of cases for TOT, GG and RET, respectively. Thus, on average they agreed in 90.5%, 90.0% and 80.0% with the heat maps, respectively.

After applying an optimized threshold value (Appendix Figure A3.5) to the heat maps on the testing dataset, a significant linear correlation was found between the heat map-derived SSc-ILD score and the L&S-Net’s output (Figure 3.8, lower row). For TOT, GG and RET, 84%, 87% and 83% of the S-Net’s output variation can be explained by the heat maps, respectively.

3.5 Discussion

In this study, we developed a deep learning framework to perform fully automated SSc-ILD scoring in chest CT scans. By cascading two separate networks, the framework was able to select the five anatomical levels from 3D CT scans and then quantify the extent of three different disease patterns for each level. The training of the framework only needs visual scores as the ground truth without the requirement of prior manual segmentations. Heat maps can intuitively explain the network’s output, and can be

Table 3.6: SSc-ILD scoring performance of human experts (Observer-1 and Observer-2) in two scoring sessions (T1 and T2, with 6 weeks interval) and our proposed network in a subset of 16 patients from the testing dataset.

Agreement	**Comparison**	**TOT***			**GG**			**RET**		
		MAE [%] ↓	**κ ↑**	**ICC↑**	**MAE [%] ↓**	**κ ↑**	**ICC↑**	**MAE [%] ↓**	**κ ↑**	**ICC↑**
Inter-observer	Obs1_{T1} vs Obs2_{T1}	5.25	0.59	0.76	4.00	0.54	0.66	3.81	0.61	0.82
	Obs1_{T2} vs Obs2_{T2}	4.25	0.67	0.88	3.94	0.58	0.80	3.44	0.63	0.86
Intra-observer	Obs1_{T1} vs Obs1_{T2}	4.38	0.63	0.83	3.50	0.56	0.73	3.06	0.67	0.84
	Obs2_{T1} vs Obs2_{T2}	3.50	0.74	0.89	3.69	0.62	0.74	2.69	0.72	0.90
Comparison with ground truth (GT)	Obs1_{T1} vs GT	7.06 (0.41)	0.51	0.73	5.63 (0.51)	0.44	0.68	4.94 (0.37)	0.56	0.76
	Obs1_{T2} vs GT	6.19 (0.14)	0.58	0.82	5.38 (0.78)	0.46	0.59	4.75 (0.13)	0.58	0.78
	Obs2_{T1} vs GT	6.56 (0.42)	0.58	0.80	5.38 (0.32)	0.48	0.63	4.63 (0.18)	0.61	0.84
	Obs2_{T2} vs GT	4.94 (0.40)	0.67	0.86	4.94 (0.001)†	0.55	0.75	4.19 (0.18)	0.63	0.80
	L&S-Net vs GT	6.40	0.54	0.79	6.13	0.39	0.55	4.44	0.61	0.84

[*] TOT = total disease extent, GG = ground glass pattern, RET = reticular pattern, GT = Ground truth (human consensus score), MAE = mean absolute error, κ = Cohen's weighted kappa, ICC = Intra-class correlation coefficient. Data between parentheses are P values from the Wilcoxon signed rank test comparing observers with our proposed method. $\text{Obs1}_{\text{T1/T2}}$ and $\text{Obs2}_{\text{T1/T2}}$ denote the observations from Observer-1 and Observer-2 at the first or second scoring session, respectively.

[†] Significantly better than our method ($P<0.05$).

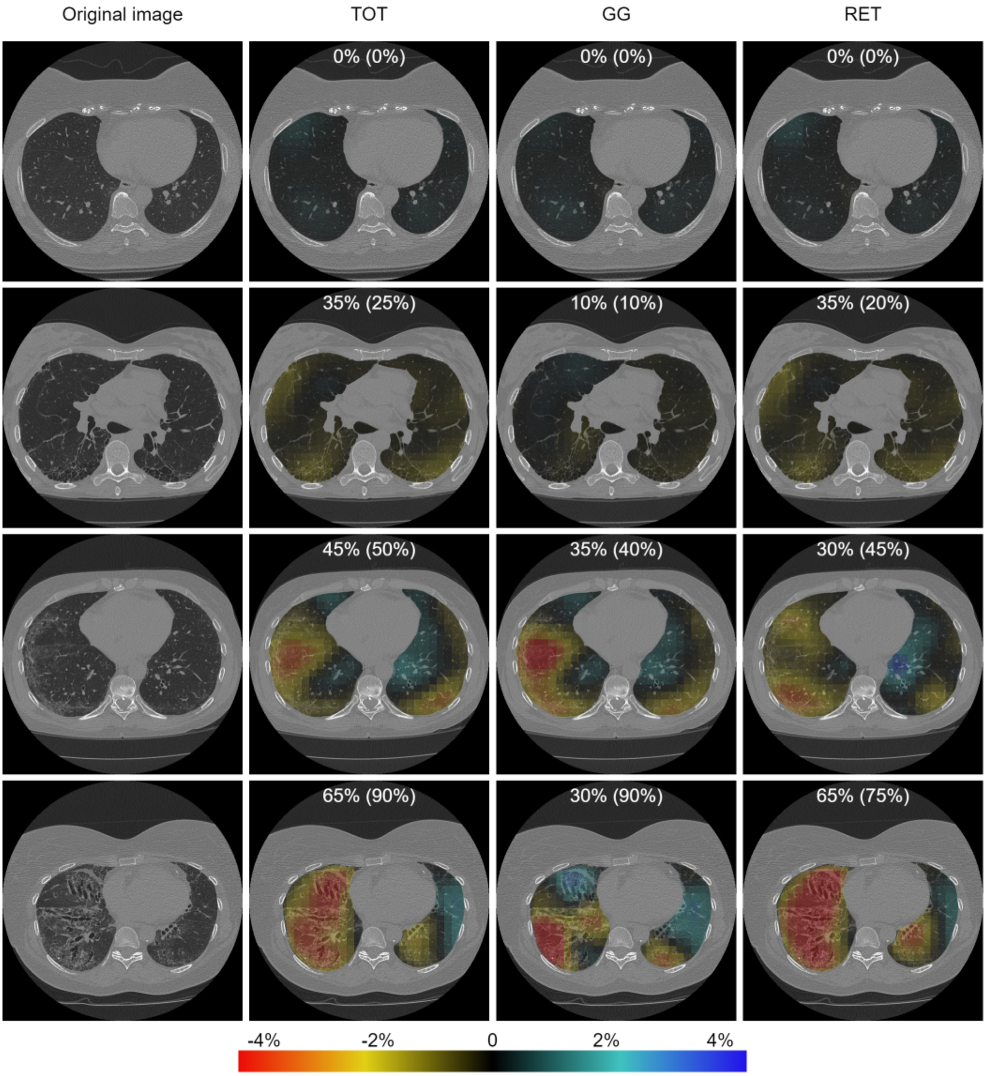


Figure 3.7: Heat map visualization for various test images. Each row represents one axial slice from a different patient. The first column is the original image and the subsequent three columns show the heat maps of the three disease patterns. Different colours represent the magnitude of score change. From top to bottom, the images show increasing disease severity. The automatic ILD score is shown on the top of each image followed by the ground truth (human consensus) between parentheses. TOT = total disease extent, GG = ground glass pattern, RET = reticular pattern.

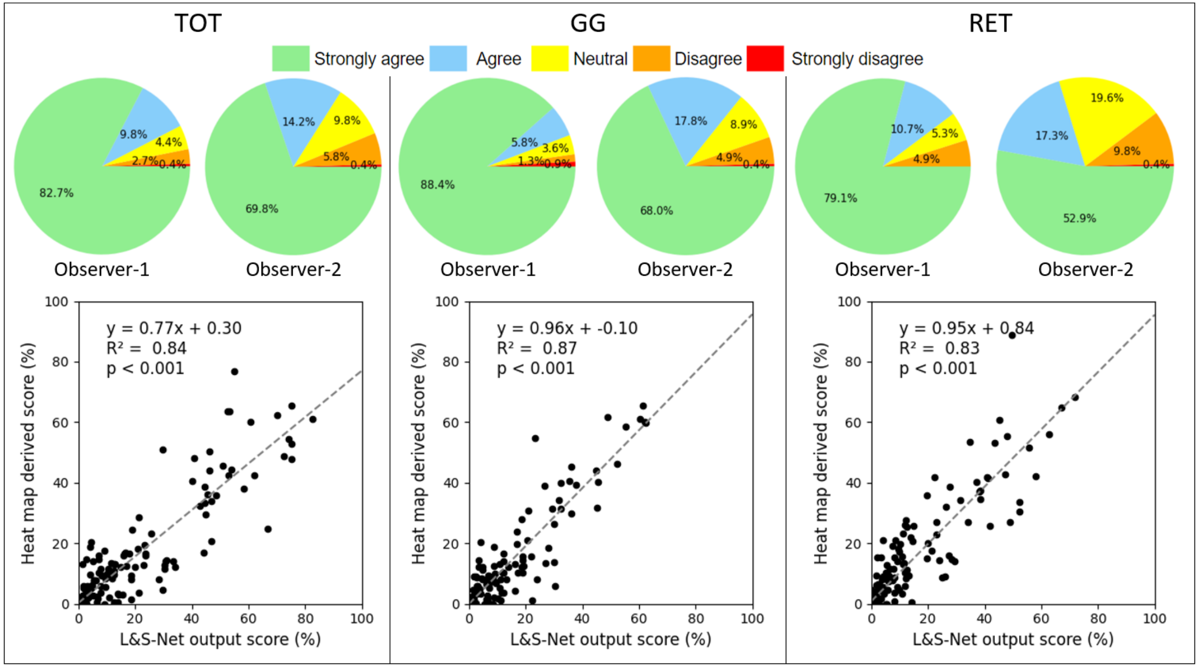


Figure 3.8: Heat map performance evaluated by the two observers (pie charts in the upper row) and the association between the heat map derived ILD scoring and L&S-Net's output (scatter plots in the lower row), to indicate its explainability. TOT = total disease extent, GG = ground glass pattern, RET = reticular pattern.

used to derive coarse segmentations of the different patterns that are consistent with the network's output. Our framework has the potential to serve as an alternative to visual SSc-ILD scoring of lung involvement in systemic sclerosis.

3.5.1 Comparison to related work

Our work shows the feasibility to select the positions of five scoring levels and estimate the percentage of radiological patterns using regression neural networks. This provides the possibility to automate other pattern-percentage based scoring systems [34, 101, 102]. To the best of our knowledge, this is the first fully automated quantification of SSc-ILD without pixel-wise annotation. In addition to manual scoring method, the two most recent and relevant deep learning based scoring methods are shown in Table 3.7, which segmented SSc-ILD on CT scans to quantify the severity of disease [61, 84]. Compared with their approach, with a similar number of patients, there are three main differences. First, our two-step framework consists of a 3D level selection network and a 2D scoring network, while they assumed that slices were already manually selected beforehand and only proposed a 2D network for lesion segmentation. Second, our score regression network just need image-wise scores for training, while their segmentation-based score assessment requires time-consuming pixel-wise annotations, which is more difficult to obtain. Third, our framework can score SSc-ILD in 10 seconds while other manual or semi-automated methods require more time.

3.5.2 Explanation and discussion on results

Our framework consists of two networks, trained independently: L-Net for automatic level selection and S-Net for automatic ILD scoring. For the L-Net, the selection of the first level is more difficult (ICC = 0.72) than other levels because indicating the origin of the great vessels is variable as it was not defined precisely. Nevertheless, the automatically selected levels were accurate enough, because the ultimate scoring did not show significant differences compared with the single S-Net's performance with manually annotated slices. This may be due to the fact that disease patterns appear and disappear only gradually from one slice to the other.

In this chapter we demonstrated that different network structures with different capacities did not necessarily show a significant difference (see Figure 3.3). That implies that network design is not the bottleneck for our task. Our further investigation showed that the key issue, hindering the network performance, is the low quality of dataset. Therefore, we improved the training method by introducing synthetic training images that significantly improved the network's performance.

For the ILD scoring network, the pre-trained weights, balanced sampling and the proposed data synthesis all helped to steadily improve the network's performance for all three patterns. Our random ILD insertion method is very effective and easy to implement, which only requires two small patches fully covered by GG and RET.

Table 3.7: Comparison between different methods for SSc-ILD scoring.

Method	**Method description**	**Label**	**Labelling difficulty**	**Number of patients**	**MAE [%]**	**Time**
Manual	Manual slice selection + manual scoring	NA*	NA	NA	7.06 to 4.19	10 minutes
Segmentation-based [61]	Manual slice selection + automatic scoring	Pixel-wise	Difficult	208	NA	3 minutes
Segmentation-based [84]	Manual slice selection + automatic scoring	Pixel-wise	Difficult	298	NA	3 minutes
Ours (regression-based)	Automatic slice selection + automatic scoring	Image-wise	Easy	230	6.40 to 4.44	10 seconds

[*] NA: not applicable, MAE: mean absolute error, Time: estimated scoring time per patient, TOT = total disease extent, GG = ground glass pattern, RET = reticular pattern, HC = honeycombing.

Generally, ILD in SSc has a specific distribution, e.g. classical subpleural sparing earlier in the disease [103], which was not simulated by our synthesis method. The scoring results were however not affected by this limitation, since the neural network only needs to estimate the ratio of ILD, irrespective of the location of ILD. Nevertheless, there is still some space for improvement in GG scoring. Also for human experts, GG is more difficult to define and identify than RET patterns, because of the limited spatial resolution of CT and consequential "partial volume" effect. Moreover, some GG patterns resemble noise from image acquisition or reconstruction. Conversely, reticular lesions are larger than voxel size and can be visually or automatically identified as structures, such as thickened interlobular septa or thickened airways causing pathological reticular patterns. Our proposed network may help in distinguishing noise from actual pathological ground glass lesions when noise patterns can be identified.

With the help of the replacement-based heat map, we visualized which areas contribute to which scores respectively. Two experts evaluated the heat maps independently and both gave very satisfactory ratings. After we applied a threshold to the heat map, the ratio of different patterns to the total lung area was consistent with the automatic ILD scores by L&S-Net. The quantitative measurement shows that our proposed heat maps can accurately explain the L&S-Net's output. This can increase the clinicians' confidence in the network's output. A heat map “highlights” the detected pathology that may help the physician with a quick image interpretation. Exploring the heat map can also be used to check the quality of the automatic score. The heat map could be regarded as a coarse segmentation of TOT, GG and RET. Normally it is not practical to have large datasets of SSc ILD pattern segmentations because it is very time-consuming and laborious. The heat maps can act as an initial step to obtain manual segmentation reducing annotation time. From this perspective, we successfully obtained a coarse ILD pattern segmentation network for SSc patients without the requirement of a segmentation ground truth. Compared with the normal binary segmentation [61], the advantage of our heat map is that it also gives an indication of the severity of a disease pattern, as shown by the colour, instead of a binary classification into either healthy or disease.

We observed that human experts gave higher ratings to the GG heat maps than RET, although the MAE of automatic GG scoring was actually consistently higher than RET. This can be explained by the fact that human experts have more confidence in recognizing RET patterns, so they use more strict criteria for RET heat maps. Since they were less confident in their GG recognition, reflected by the lower inter-/intra-observer agreement, this resulted in more tolerance for GG heat maps.

In the visual ILD scoring system, the use of only five anatomical levels has been a compromise, for clinical practice. It is already very time-consuming and laborious to manually select the five levels and score three patterns for each level (taking around ten minutes). Our method, however, could automatically complete the level selection and ILD scoring in several seconds. In addition, our method can be easily extended to score all slices of the entire CT volume, which is practically not feasible for humans.

3.5.3 Limitations

Our method has some limitations. The L-Net was initialized with a random distribution instead of pre-trained weights. This may be improved if pre-trained weights from a large 3D medical image dataset are available. The quality of data synthesis could clearly be improved further. The current pattern insertion method may distort the structure of airways and vessels and introduce some periodic artefacts. In future research, generative adversarial networks (GANs) [104] may generate more realistic synthetic patterns. The data used in this study is from a single model CT scanner within a single healthcare programme with tightly-controlled acquisition and reconstruction parameters. Because of the lack of publicly available independent testing dataset, whether this method could be used across a range of CT scanners, sites and protocols still needs to be verified. The ILD scores of our synthetic training images were obtained by the ratio of different patterns, while the ILD scores of the real images were estimated by the human observer without any pattern segmentation or contours. Therefore, there may be a systematic bias between them, which could contribute to the disagreement between our framework and experts.

3.6 Conclusions

In conclusion, we proposed the first fully automated framework to estimate scores for ground glass opacities, reticular patterns and total disease extent from 3D CT scans, specific for systemic sclerosis. The output scores can be clearly explained by the replacement-based heat maps. The results show its potential as an objective alternative for visual scoring of systemic sclerosis and could be extended to other applications where a diagnosis is based on scores at different anatomical levels.

3.7 Acknowledgments

This work was performed using the computational resources from the Academic Leiden Interdisciplinary Cluster Environment (ALICE) provided by the Leiden University. This work is supported by the China Scholarship Council No.202007720110.

3.8 Appendix

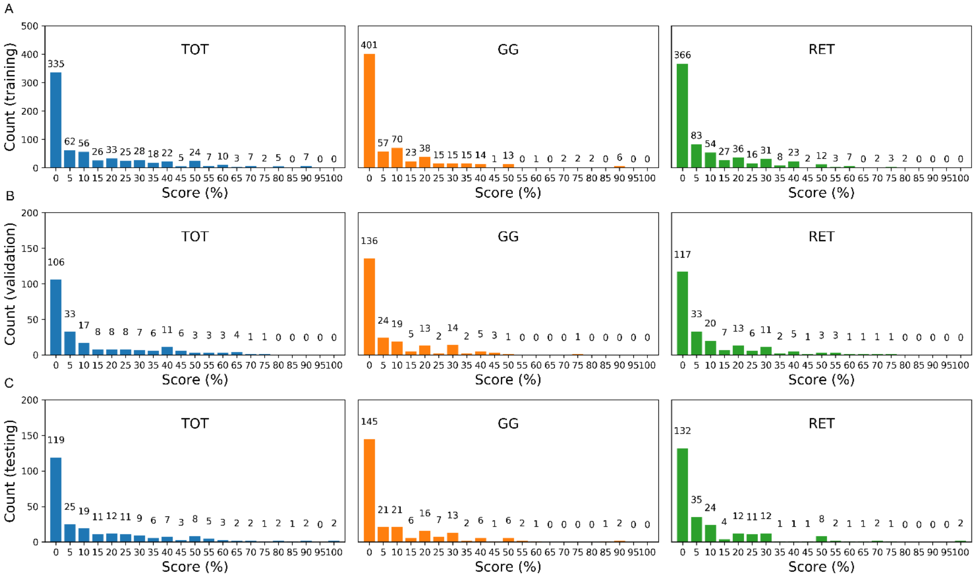
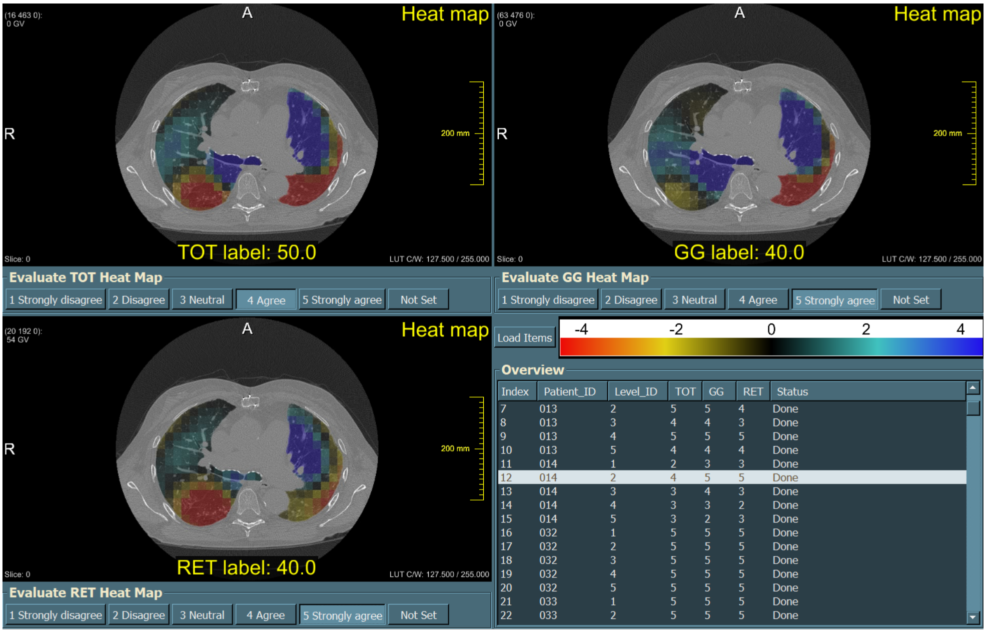


Figure A3.1: Score distribution of axial slices for three patterns in different datasets. The scores are estimated with a precision of five percent, leading to 21 bins for each pattern in (A) training, (B) validation and (C) testing dataset. TOT = total disease extent, GG = ground glass pattern, RET = reticular pattern.



Index	Patient_ID	Level_ID	TOT	GG	RET	Status
7	013	2	5	5	4	Done
8	013	3	4	4	3	Done
9	013	4	5	5	5	Done
10	013	5	4	4	4	Done
11	014	1	2	3	3	Done
12	014	2	4	5	5	Done
13	014	3	3	4	3	Done
14	014	4	3	3	2	Done
15	014	5	3	2	3	Done
16	032	1	5	5	5	Done
17	032	2	5	5	5	Done
18	032	3	5	5	5	Done
19	032	4	5	5	5	Done
20	032	5	5	5	5	Done
21	033	1	5	5	5	Done
22	033	2	5	5	5	Done

Figure A3.2: Interface of the dedicated software for the heat map evaluation using the Likert scale. A dedicated software was developed to help the experts evaluate the heat maps easily. The interface was developed using MeVisLab 3.1.1 (`https://www.mevislab.de`).

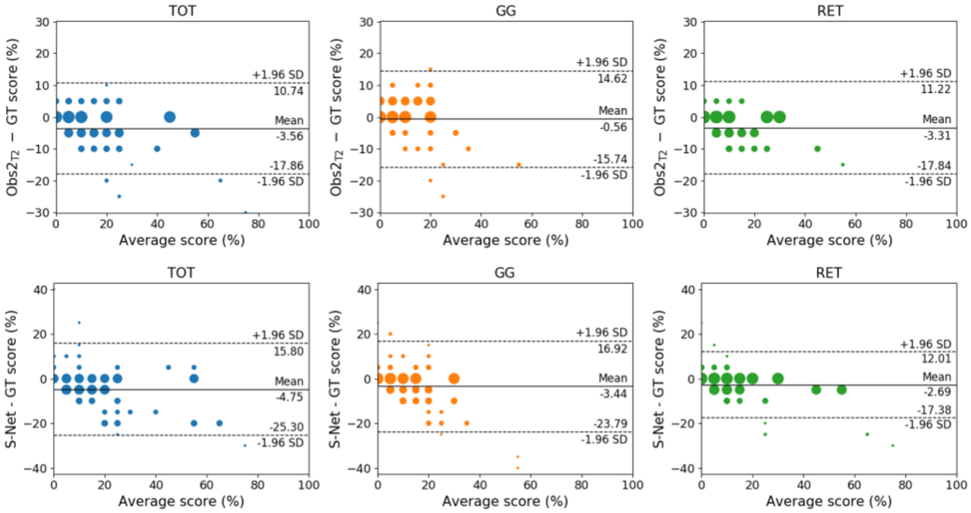


Figure A3.3: Bland-Altman plots of the scores by Observer-2 at the second scoring session (upper) and our L&S-Net (lower). The size of the circles indicates the numbers of cases with the same outcome. GT = Ground truth (human consensus), TOT = total disease extent, GG = ground glass pattern, RET = reticular pattern, SD = standard deviation.

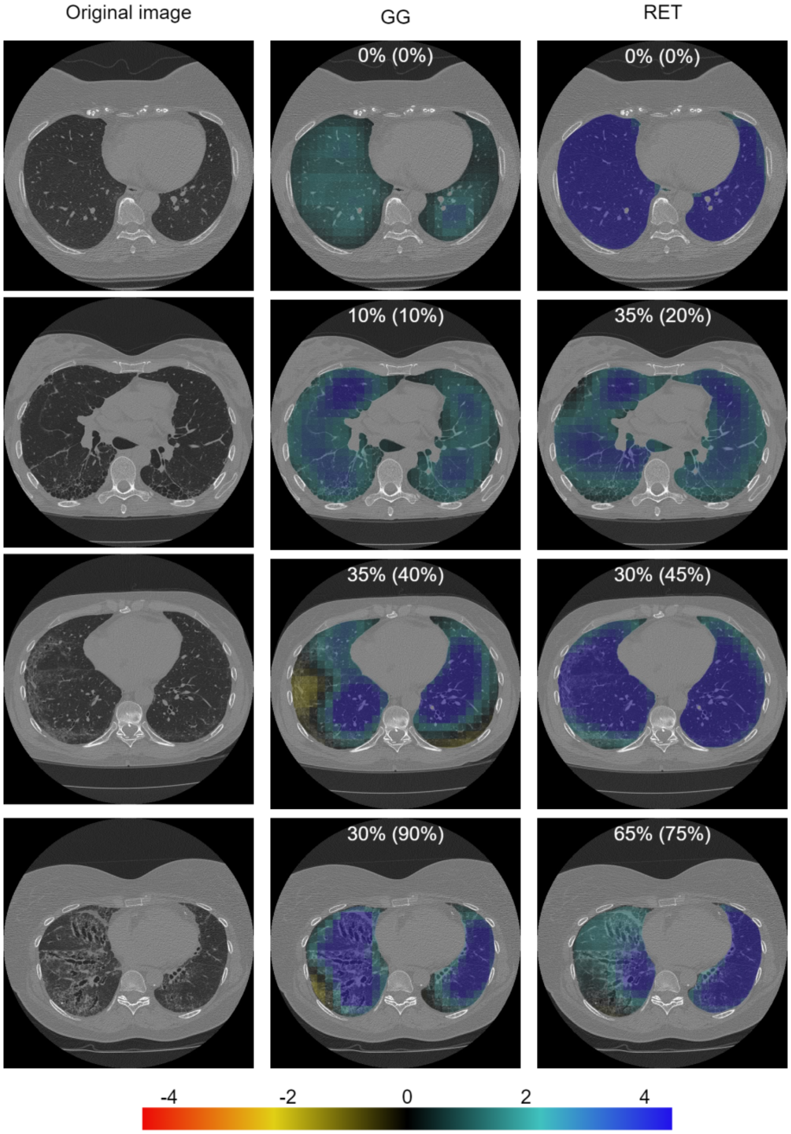


Figure A3.4: Alternative heat maps for various testing images. While the original heat maps are obtained by replacement by healthy patches, these heat maps were generated from the replacement by two different disease patterns. The first column shows the original image. The second and third column were generated by covering the original image with patches with ground glass patterns (GG) and reticular patterns (RET), respectively. Different colours represent the magnitude of score change. The blue colour represents the cases where the region does not include the specific disease pattern. The cyan and black region indicates disease patterns. Yellow and red region are the false negative parts. From top to bottom, the images show increasing disease severity. Scoring results are shown on the top of each image followed by the ground truth between parentheses.

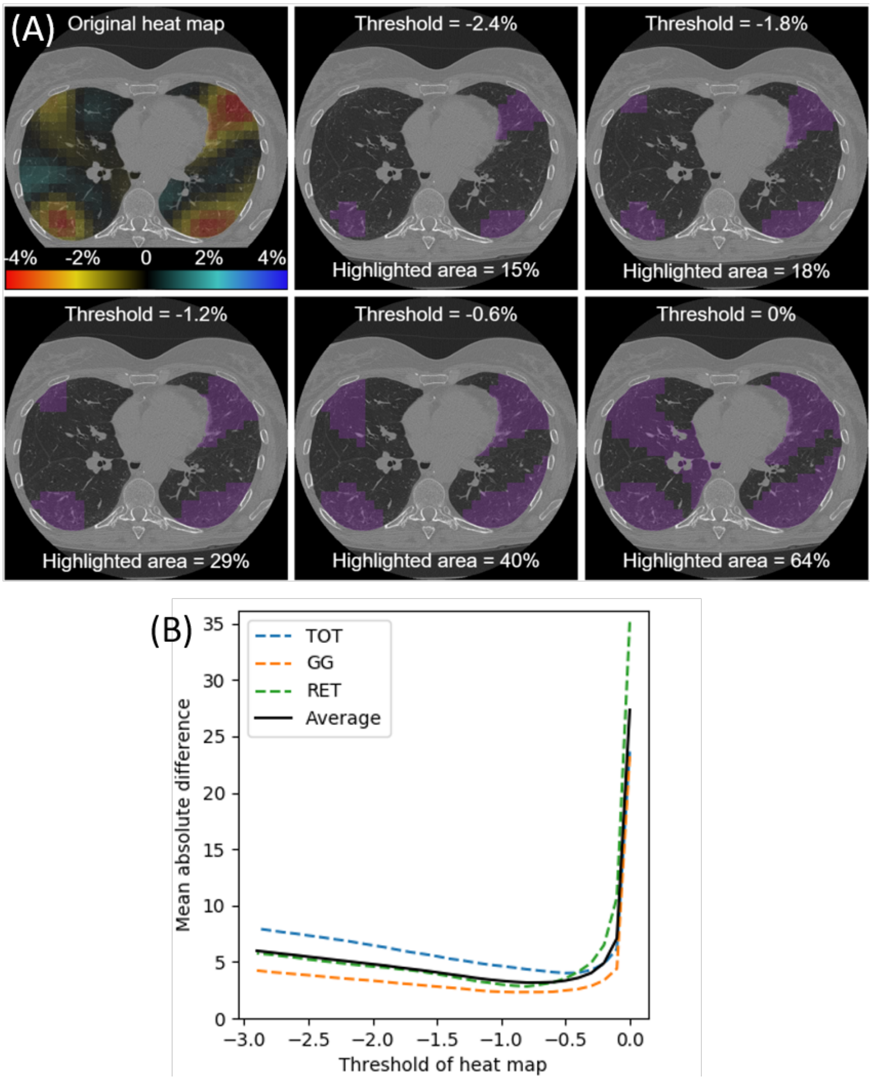
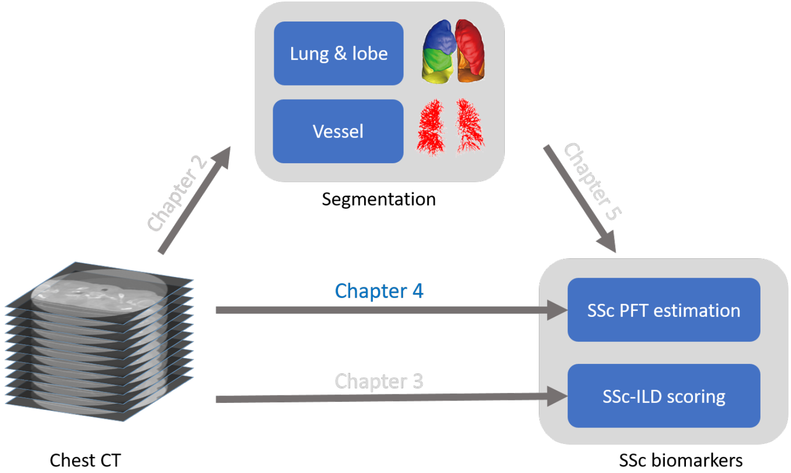


Figure A3.5: Exploration of the optimal heat map threshold. (A) Examples of the highlighted area of different thresholds in heat maps. Upper left is the original heat map (network's output score is 40%) showing score change at each position after the replacement by a healthy patch. Score decrement (yellow and red colours) denotes the positions of disease pattern. The remaining images show the highlighted area by purple colour when applying different thresholds to the original heat map. The ratio of highlighted area to the total lung is shown at the bottom of each image. (B) Correlation between heat map threshold and interstitial lung disease score difference between heat map derived method and our network output. This figure is obtained based on validation dataset. TOT = total disease extent, GG = ground glass pattern, RET = reticular pattern.

4

Automatic pulmonary function estimation from chest CT scans using deep regression neural networks: the relation between structure and function in systemic sclerosis



This chapter was adapted from:

Jia, Jingnan, Emiel R. Marges, Jeska K. De Vries-Bouwstra, Maarten K. Ninaber, Lucia JM Kroft, Anne A. Schouffoer, Marius Staring, and Berend C. Stoel. "Automatic Pulmonary Function Estimation From Chest CT Scans Using Deep Regression Neural Networks: The Relation Between Structure and Function in Systemic Sclerosis." *IEEE Access* 11 (2023): 135272-135282.

Abstract

Pulmonary function tests (PFTs) play an important role in screening and following-up pulmonary involvement in systemic sclerosis (SSc). However, some patients are not able to perform PFTs due to contraindications. In addition, it is unclear how lung function is affected by changes in lung structure in SSc. Therefore, this study aims to explore the potential of automatically estimating PFT results from chest CT scans of SSc patients and how different regions influence the estimation of PFTs. Deep regression networks were developed with transfer learning to estimate PFTs from 316 SSc patients. Segmented lungs and vessels were used to mask the CT images to train the network with different inputs: from entire CT scan, lungs-only to vessels-only. The network trained on entire CT scans with transfer learning achieved an ICC of 0.71, 0.76, 0.80, and 0.81 for the estimation of DLCO, FEV_1, FVC and TLC, respectively. The performance of the networks gradually decreased when trained on data from lungs-only and vessels-only. Regression attention maps showed that regions close to large vessels were highlighted more than other regions, and occasionally regions outside the lungs were highlighted. These experiments show that apart from the lungs and large vessels, other regions contribute to PFT estimation. In addition, adding manually designed biomarkers increased the correlation (R) from 0.75, 0.74, 0.82, and 0.83 to 0.81, 0.83, 0.88, and 0.90, respectively. This suggests that that manually designed imaging biomarkers can still contribute to explaining the relation between lung function and structure.

4.1 Introduction

Systemic sclerosis (SSc) is a rare immune-mediated connective tissue disease that affects different organs. Interstitial lung disease (ILD) is, however, the leading cause of morbidity and mortality, and up to 90% of SSc patients have pulmonary function abnormalities [14]. To evaluate progression of SSc-ILD, various pulmonary function tests (PFTs) are used as key measures, such as the diffusion capacity for carbon monoxide (DLCO), forced expiratory volume in 1 second (FEV_1), forced vital capacity (FVC) and total lung capacity (TLC) [14, 15, 16]. In clinical practice, PFTs are expressed either in absolute values or in percent predicted values (abbreviated as PFTs%pred, including DLCO%pred, FEV_1%pred, FVC%pred and TLC%pred). PFTs%pred are obtained by the standardization of the absolute values according to the patients' characteristics to avoid biases from sex, ethnicity and height [105]. A PFT%pred below 100% then represents a lung function that is lower than the average lung function in a population with the same age, gender, etc., with upper and lower limits of normal (usually 1.64 STD). Both absolute and percent predicted are commonly used clinically as outcome measures for progression of SSc-ILD [15].

PFTs can, however, not always be performed if there is a risk of disease transmission, e.g. in patients with COVID-19, active tuberculosis or other airborne infectious diseases [21, 22]. In addition, some patients, who have hemoptysis or had surgery in the past month, or other contraindications [23, 24], like aneurysmatic abnormalities and ischaemic stroke, are not able to perform PFTs because the forced exhalation during spirometry may increase the risk of complications [9]. Therefore, alternative methods to estimate PFT are of great interest. Because CT could provide high-resolution details of the lungs, it is regarded the gold standard for diagnosing SSc-ILD [17]. In previous research, quantitative biomarkers have been extracted from chest CT images of SSc patients, which correlate with PFTs [96]. Therefore, when PFTs are not possible and CT scans have been made for SSc patients, it is of great interest to see if CT could be used to estimate PFT.

Apart from being an alternative to PFTs, PFT estimation from CT scans can also be used to study the relation between structure and function as the lungs become affected by SSc-ILD. Initially, imaging biomarkers were designed for SSc to explicitly describe lung structure and subsequently determine their correlation with lung function. For SSc patients with fibrosis, Goh *et al.* [106] designed a visual fibrosis scoring system, which correlated with FVC (R = -0.40). For SSc patients without fibrosis visible on CT, Zhai *et al.* [96] found that two vascular tree-based biomarkers (α and β) , which represent the lung vessel radius histogram, correlated with DLCO%pred (R=-0.29 and 0.32, respectively). For SSc patients with or without fibrosis, Ninaber *et al.* [16] found that lung density, measured by the 85th percentile density (Perc85) from CT scans, correlated significantly with DLCO%pred (R=-0.49) and FVC %predicted (R=-0.64).

Apart from these manually designed biomarkers, an altogether different approach would be to develop a deep learning model that is trained to output PFT prediction values directly, with or without fibrosis visible on CT scans. Subsequently, the trained model could be studied in detail to explore the relation between lung structure from CT and lung function from PFTs.

To the best of our knowledge, we are the first to estimate PFTs for SSc patients. There are no works to estimate PFTs for SSc patients previously. The most relevant and recent works on automatic estimation of PFTs from chest CT using deep learning [21, 107] are not for SSc patients. Choi, *et al.* [21] developed a network to estimate FEV_1 and FVC for patients before their first lung cancer surgery. Their network consisted of a ResNet-50 for feature extraction and a bidirectional long short-term memory (BiLSTM) network for PFT prediction. Park, *et al.* [107] trained two separate I3D networks to estimate FEV_1 and FVC, respectively, for subjects at risk of lung cancer. It is unclear if their models could be applied directly to SSc patients. In addition, both methods estimate FEV_1 and FVC only, lacking DLCO and TLC. For determining SSc-ILD progression, however, TLC and especially DLCO are important measurements, the latter of which is most predictive of adverse outcomes, including death [15]. Therefore, the aim of this study was to 1) develop a deep learning model to estimate DLCO, FEV_1, FVC and TLC for SSc patients from their CT scans; and 2) explore the contribution of different anatomical regions, and provide explanations from a clinical perspective.

The remaining chapter is organized as follows. Section 2 describes the datasets and methods we used for the prediction of PFT. Detailed experiments and results are shown in Section 3. Finally, section 4 discusses the experiments, explains the results and concludes the chapter.

4.2 Materials and methods

4.2.1 Dataset

In this study, we retrospectively selected 333 patients who were referred to our targeted outpatient health care program (combined care in systemic sclerosis) between April 2009 and October 2015 in Leiden University Medical Center. Because of the diagnosis of SSc according to the referring rheumatologist, or a strong suspicion for SSc, they underwent high-resolution CT scans, followed by pulmonary function tests. As shown in Figure 4.1, we excluded seven patients with a CT-PFT interval greater than ten days, nine patients with incomplete PFTs, and one patient with a low-quality CT scan, resulting in 316 CT-PFT pairs. The dataset was split into two disjoint groups: 252 for four-fold training and cross-validation, and 64 for testing. The research protocol was

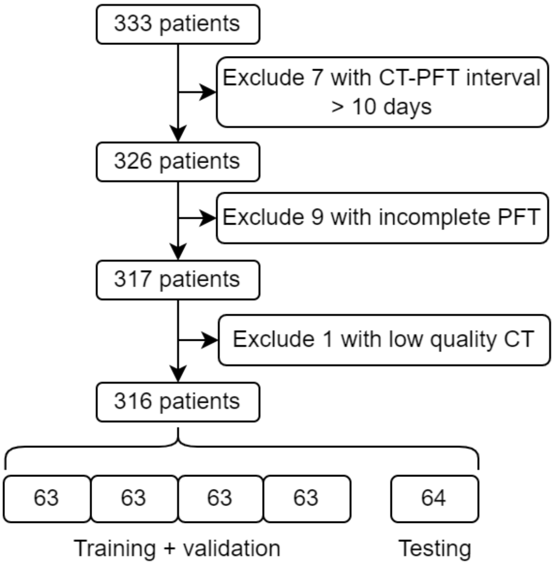

333 patients
Exclude 7 with CT-PFT interval > 10 days
326 patients
Exclude 9 with incomplete PFT
317 patients
Exclude 1 with low quality CT
316 patients
63
63
63
63
64
Training + validation
Testing

Figure 4.1: Flowchart of the dataset inclusion and partition.

granted approval by the local Medical Ethics Committee and written informed consent was provided by all patients.

CT scanning. All subjects underwent scanning at full inspiration without contrast enhancement using an Aquilion 64 CT scanner (Canon Medical Systems), configured at 120 kVp, a median tube current of 140 mA, a rotation time of 0.4 seconds, a collimation of 64 × 0.5 mm and a helical beam pitch of 0.8; leading to a median CTDI_{vol} of 8.2 mGy. The images were reconstructed with filtered back projection and an FC86 kernel, with a median pixel spacing of 0.64 mm × 0.64 mm, with a slice thickness and increment of 0.5 and 0.3 mm, respectively.

PFT measurements. PFTs were performed by an experienced technologist using a spirometer under ERS/ATS guidelines [108, 18] including single-breath diffusion capacity for carbon monoxide corrected for haemoglobin concentration (DLCO), forced expiratory volume in 1 second (FEV_1), forced vital capacity (FVC) and total lung capacity (TLC). While DLCO was measured in units of mm/Hg/min, FEV_1, FVC, TLC were measured in units of liter. The PFT percent predicted values (PFTs%pred) were calculated with the latest official conversion equations and reference values [109, 110, 111]. Clinical characteristics of the 316 patients are shown in Table 4.1.

Data preprocessing and augmentation. Because of GPU memory limitations, we first down-sampled all CT scans to an isotropic spacing of 1.5 mm, as illustrated in Figure 4.2, resulting in a median image size of 256 × 256 × 240 voxels. Next,

Table 4.1: Dataset characteristics

Characteristic	Patients (n=316)
Age, mean ± standard deviation [years]	53.4 ± 14.6
Female (%)	258 (81.4)
Interstitial lung disease detected on CT (%)	
Anti-centromere antibodies positive (%)	
Anti-topoisomerase antibodies positive (%)	
Pulmonary arterial hypertension (%)	
Disease Subset:	
Non-cutaneous (%)	
Diffuse cutaneous (%)	
Limited cutaneous (%)	
Alternative diagnosis* (%)	
Pulmonary Function, Mean ± Std	
DLCO, [mL/min/mm Hg]	5.55 ± 1.92
FEV$_1$, [L]	2.62 ± 0.77
FVC, [L]	3.32 ± 0.97
TLC, [L]	4.91 ± 1.23
DLCO%, [%]	71.95 ± 20.10
FEV$_1$%, [%]	89.32 ± 17.69
FVC%, [%]	90.58 ± 18.97
TLC%, [%]	85.62 ± 17.07

we performed intensity truncation to clip voxel values between -1500 and 1500 HU to remove some artifacts. Then we applied padding, if necessary, to guarantee a minimum image size of 240 × 240 × 240 voxels. To subsequently augment the training data, a random 3D patch of a fixed size (240 × 240 × 240 voxels, which was ensured to cover the whole lung area) were cropped from each volume as they are fed into the model. In different epochs, different 3D patches were cropped from each CT for training. The epoch number is the number of 3D patches cropped from each CT. In the validation and testing phase, we used 3D patches of 240 × 240 × 240 voxels at the center position, from ($x0$-120, $y0$-120, $z0$-120) to ($x0+120$, $y0+120$, $z0+120$) where ($x0$, $y0$, $z0$) is the coordinates of the center point of each validation and testing CT image. To investigate the contribution of different chest regions, we masked the CT images using various masks. Lung masks were obtained by a multi-atlas based method [96], while vessel masks were acquired using a graph-cut based vessel segmentation network [58]. The segmentation of lung and vessel masks was obtained by an in-house script in MeVisLab 2.7.1 (VC12-64). The implementation details could be found at the online document which were released along with the original paper [58] (http://links.lww.com/JTI/A114). The source code of our in-house script for the segmentation could be found at `https://github.com/Zhiwei-Zhai/`

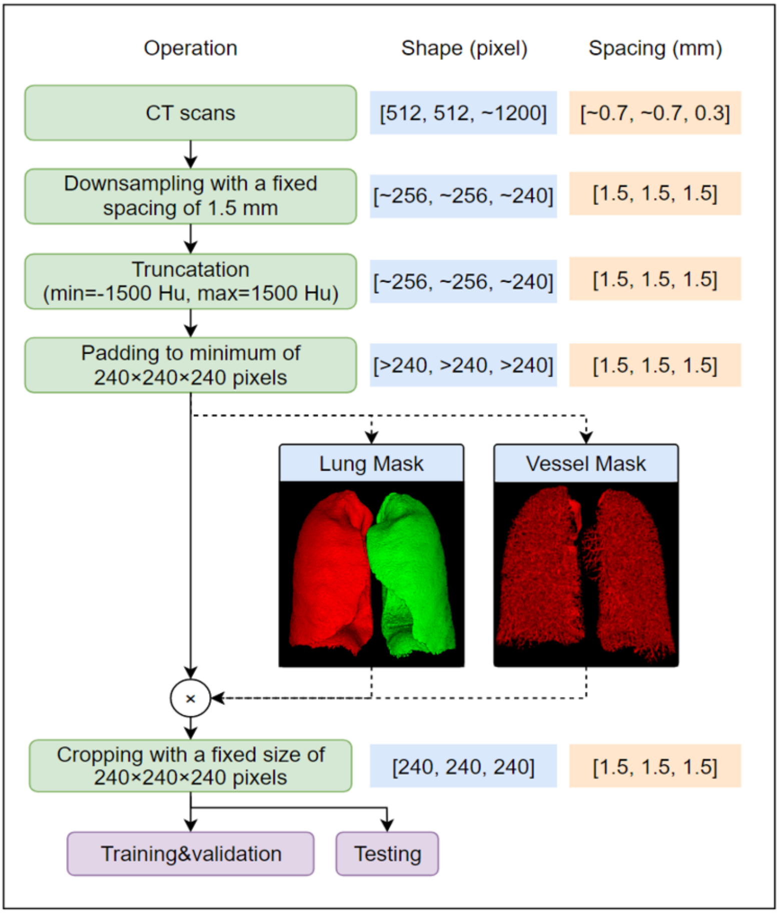


Figure 4.2: CT scan preprocessing procedure.

`Vessel-Segmentation-Using-Graph-cuts`. No additional data augmentation was performed.

4.2.2 Network design

The network was adapted from X3D [112], which was originally designed for video recognition. The original paper proposed a series of networks with different capacities. An X3D of medium size (X3D-M) was selected as the architecture of our network, to account for limited GPU memory. As illustrated in Figure 4.3, the network consists of several convolution and max-pooling layers, followed by four ResNet blocks with max-pooling layers between each of them, and finally one global average pooling (GAP) layer and two fully connected (FC) layers. The output of the last FC layer has four values, representing the four (absolute) PFT parameters, simultaneously estimated in

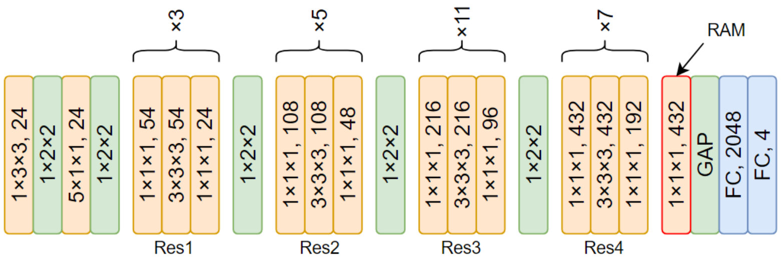


Figure 4.3: X3D-M structure. The whole network consists of 3D convolution layers (orange boxes), max-pooling layer (green boxes) and two fully connected layer (blue box). Kernel size (xyz) and channel number are denoted by the first three numbers and the last number, respectively.

one network. We also developed four separate networks with 1-class outputs for each of them, estimating the different PFT parameters, separately. The comparison between these 1-class and 4-class networks will be shown later in Q4 of Section Experiments and Results.

To increase network performance, we introduced transfer learning (TL), in which the network was initialized by the weights trained from another domain. Although it may achieve better performance if the source domain is similar with the target domain, the lack of large annotated lung CT dataset makes it impractical to apply pre-trained weights from lung CT dataset. However, TL has been widely used in deep learning because it was reported to improve network performance significantly even if the source domain is different [113]. Therefore, our X3D-M network was pre-trained on Kinetics, a human action video dataset [112, 114] (pytorch.org/hub/facebookresearch_pytorchvideo_x3d). Although there are other 3D networks which may also works on our task, X3D-M is the network which was released recently, achieved the SOTA performance, publish their pre-trained weights from Kinetics dataset, and could be fit into our GPU with memory of 11 GB.

We studied two ways to estimate PFTs%pred: 1) directly, by one network that is trained to estimate PFTs%pred directly; and 2) indirectly, where the absolute PFTs are obtained by a network, from which the PFTs%pred are subsequently calculated by the official conversion equations [109, 110, 111].

4.2.3 Relation between lung structure and function

After we obtained the optimized network and training method, we performed two strategies to understand how the network derived the estimation and how each chest region (such as muscle, lung, vessel, etc.) contributed to the PFT estimation. The first strategy was to train multiple networks with various inputs: whole CT image, lungs-only (by excluding the volume outside the lungs), left or right lung-only, vessels-only and the binarized version of vessels-only. The difference in performance between the different networks implies the contribution of these different regions. The second strategy was 3D regression activation mapping (RAM-3D), which is a variant of the Grad-CAM [115] on 3D regression tasks. The original Grad-CAM was designed for 2D image classification [115], which could generate heat maps to highlight the important regions for classification by convolutional neural networks (CNNs). Inspired by that, Wang *et al.* proposed a RAM for 2D image regression [116]. In this work, we extended this RAM from 2D to 3D to highlight areas of interest in the 3D CT volumes for the PFT estimation. To capture detailed regional information, we computed the gradient for the linear output layer with respect to the feature maps of the convolution layer right before the GAP layer (marked in Figure 4.3).

4.2.4 Evaluation metrics and statistical analysis

The performance of the proposed deep learning networks was evaluated on two separate datasets: a four-fold cross-validation dataset and a separate testing set. The optimization of network structure and training strategy was based on the four-fold cross-validation results. The testing dataset was used only for the final performance assessment, and for comparison our network's performance with standard repeatability criteria of PFT measurements.

We used various metrics to evaluate the agreement between our network output and measured values (from spirometry). The mean absolute error (MAE) was used to reflect the absolute agreement. Because the unit and scale of the four PFTs are different, we used the mean absolute percentage error (MAPE), which is the ratio of MAE to the real measurements, to reflect the relative uncertainty of prediction. MAE and MAPE were calculated as follows:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_e - y_m|, \tag{4.1}$$

$$\mathrm{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n} \frac{|y_e - y_m|}{y_m}, \tag{4.2}$$

where i is the index of samples and n represents the total number of samples, y_e is the network's estimated value, and y_m is the measured PFT value.

We used the Pearson correlation coefficient (R) to indicate the linear correlation. An absolute value of R below 0.1 indicates a negligible correlation, a value between 0.1 and 0.39 indicates a weak correlation, between 0.4 and 0.69 a moderate correlation, between 0.7 and 0.89 a strong correlation, and over 0.9 indicates a very strong correlation [117]. The intra-class correlation coefficient (ICC) is a measure of reliability, which represents not only the absolute agreement but also the linear correlation. ICC was calculated by Pingouin 0.4.0 [100] based on a single-rating, absolute-agreement, 2-way mixed-effects model [98]. ICC values below 0.5 indicate poor reliability, between 0.5 and 0.75 moderate reliability, between 0.75 and 0.9 good reliability, and any value above 0.9 indicates excellent reliability [98].

To statistically test differences between groups, a Wilcoxon signed-rank test was performed, as implemented by scikit-learn 0.24.2. A p-value of less than 0.05 was considered to indicate a statistically significant difference. Bland-Altman plots were used to analyze the mean differences (bias) and limits of agreement. These statistical analyses were performed by an in-house python 3.8 script with corresponding libraries.

In addition, we applied multiple variable regression analysis using IBM SPSS Statistics (version 27) software (IBM, Armonk, USA), to determine if manual biomarkers could contribute to the prediction from the developed networks.

4.3 Experiments and results

We sequentially conducted a series of experiments to answer the following questions and optimize our method, based on the answers to these questions: **Q1:** How well can traditional manually designed features predict PFTs in our dataset? **Q2**: Does our network benefit from transfer learning? **Q3**: For PFTs%pred estimation, is the direct estimation better than the indirect estimation? **Q4**: How does a 1-class network perform compared to a 4-class network? **Q5**: How much do the different chest regions contribute to the PFT estimation? **Q6**: How does our method perform compared to standard repeatability criteria for PFTs? **Q7**: Are manual biomarkers still valuable for SSc patients given our automatic method?

4.3.1 Experiment setting

Our neural networks were implemented using PyTorch 1.11.0 (https://pytorch. org). The loss function was the mean squared error (MSE), and a batch size of 1 was used. The Adam optimizer was used with a learning rate of 1e-4 and a weight decay of 1e-3. Multithreading was used to accelerate the on-the-fly data augmentation. The training will stop when the validation loss does not decline in 25 consecutive epochs or once 100 epochs have been completed. The workstation for training and validation was equipped with an Intel(R) Xeon(R) CPU Gold 6126 2.6GHz with 90 GB memory and a NVIDIA GPU GeForce RTX 2080TI with 11 GB memory. Our code and trained models are publicly available via GitHub (https://github.com/Jingnan-Jia/PFT) for the convenience of reproducing our method or applying our model to other datasets.

4.3.2 Manually designed biomarkers (Q1)

First, the correlation between different PFTs and PFTs%pred are shown in Figure 4.4 (A) and (B), respectively. Consistent with the literature, high correlations with $p<0.001$ were found among the four PFTs with R ranging from 0.71 to 0.95, and lower correlations with still $p<0.001$ among the four PFTs%pred with R from 0.41 to 0.88. We applied previously developed manual quantification methods on our CT dataset to obtain various imaging biomarkers including lung volume (V_{lung}), vessel volume (V_{vessel}), Perc85 [16], α and β [96]. The correlation between manually designed biomarkers and measured PFTs and PFTs%pred are presented in Figure 4 (C). V_{lung} was significantly correlated with PFTs and PFTs%pred with $p<0.05$. V_{vessel} showed no significant correlation with any PFTs except TLC values ($p=0.03$). Perc85 correlated significantly with all PFTs%pred ($p<0.001$), which is consistent with a previous report [3]. α and β showed no significant correlations with any of the PFTs%pred. With the absolute PFT measures, β still showed a significant correlation, with R ranging from 0.28 to 0.34, but α did not show any significant correlations.

4.3.3 Transfer learning (TL) versus trained from scratch (Q2)

The performance of the network based on TL was compared with the network trained from scratch, see Table 4.2. It is shown that the R and ICC values increased and MAE values decreased after the introduction of TL. The standard deviation also decreased, which means that the networks with TL were more stable than those trained from scratch. This finding was verified by the scatter plots of the two networks (Figure 4.6), where the regression lines of the network with TL were closer to the identity line than the networks without TL. Figure 4.7 shows the Bland-Altman plots of networks without or with TL. The plots display the differences between the automatically estimated PFTs and measured PFTs against their mean. From Figure 4.6 and 4.7, we can observe that the network trained from scratch tended to give conservative estimations: close to the mean value of measurements.

Therefore, the images with lower PFTs were overestimated and higher PFTs were underestimated. After the introduction of TL, these pattern disappeared in FEV_1, FVC and TLC, whereas in DLCO a similar but less prominent pattern remained. This indicates that the network trained with TL achieves a better agreement to measured PFTs. In addition, Figure 4.5 shows that TL could speed up the training: decreasing the training epochs from 68 to 14. Therefore, we used TL in all the following experiments.

4.3.4 Estimation of PFTs%pred: direct versus indirect (Q3)

Table 4.3 shows the performance of estimating PFTs%pred for the two methods. The indirect estimation achieved ICC values of 0.60, 0.60, 0.69 and 0.75 for DLCO%pred, FEV_1%pred, FVC%pred and TLC%pred, respectively. These ICC values were higher than those of the direct method (ICC=0.30, 0.47, 0.50 and 0.53). The indirect method also achieved higher R and lower MAE values. Therefore, all the following networks were trained to estimate the absolute PFTs first.

Table 4.2: Comparison between networks trained with or without transfer learning. $P^{\dagger}$: Significance of the mean error difference between the networks with or without TL. $P^{\ddagger}$: Significance of the mean error difference between the 4-class and 1-class network.

TL	**Out**	**Metrics**	**DLCO**					**FEV1**					**FVC**					**TLC**				
			1	**2**	**3**	**4**	**All**	**1**	**2**	**3**	**4**	**All**	**1**	**2**	**3**	**4**	**All**	**1**	**2**	**3**	**4**	**All**
No	4-class	R	0.72	0.55	0.57	0.47	0.54	0.77	0.54	0.78	0.63	0.62	0.80	0.59	0.78	0.71	0.68	0.86	0.68	0.77	0.76	0.71
		ICC	0.64	0.51	0.40	0.44	0.48	0.73	0.52	0.61	0.52	0.58	0.77	0.59	0.68	0.60	0.65	0.85	0.64	0.68	0.69	0.70
		MAE	0.96	1.27	1.36	1.21	1.25	0.39	0.49	0.39	0.42	0.46	0.45	0.63	0.45	0.48	0.56	0.50	0.86	0.63	0.56	0.70
		MAPE	20%	25%	26%	29%	26%	16%	22%	18%	15%	19%	17%	24%	13%	15%	17%	13%	20%	13%	14%	15%
Yes	4-class	R	0.77	0.74	0.73	0.80	0.71	0.85	0.74	0.81	0.75	0.77	0.86	0.82	0.91	0.81	0.83	0.85	0.85	0.88	0.68	0.82
		ICC	0.76	0.73	0.68	0.66	0.71	0.76	0.74	0.82	0.72	0.76	0.82	0.81	0.90	0.71	0.8	0.82	0.85	0.82	0.67	0.81
		MAE	0.90	1.00	1.14	0.99	1.04	0.41	0.38	0.38	0.42	0.41	0.44	0.41	0.38	0.56	0.49	0.60	0.48	0.71	0.79	0.63
		MAPE	19%	21%	23%	19%	22%	19%	16%	15%	20%	18%	15%	15%	13%	20%	16%	12%	10%	16%	17%	13%
		$P^{\dagger}$	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
	1-class	R	0.68	0.79	0.75	0.66	0.71	0.85	0.70	0.82	0.75	0.76	0.86	0.80	0.87	0.77	0.78	0.91	0.89	0.88	0.87	0.85
		ICC	0.67	0.65	0.73	0.62	0.70	0.80	0.70	0.81	0.74	0.76	0.80	0.77	0.86	0.67	0.76	0.90	0.84	0.85	0.79	0.84
		MAE	0.99	0.93	1.07	1.10	1.07	0.32	0.38	0.32	0.46	0.38	0.39	0.42	0.42	0.52	0.51	0.43	0.56	0.50	0.62	0.55
		MAPE	22%	21%	22%	25%	23%	15%	17%	14%	22%	16%	16%	13%	17%	19%	17%	10%	13%	12%	16%	12%
		$P^{\ddagger}$	0.51	0.68	0.57	0.74	0.60	0.22	0.01	0.10	0.61	0.13	0.15	0.01	0.26	0.39	0.18	0.80	0.75	0.66	0.34	0.61

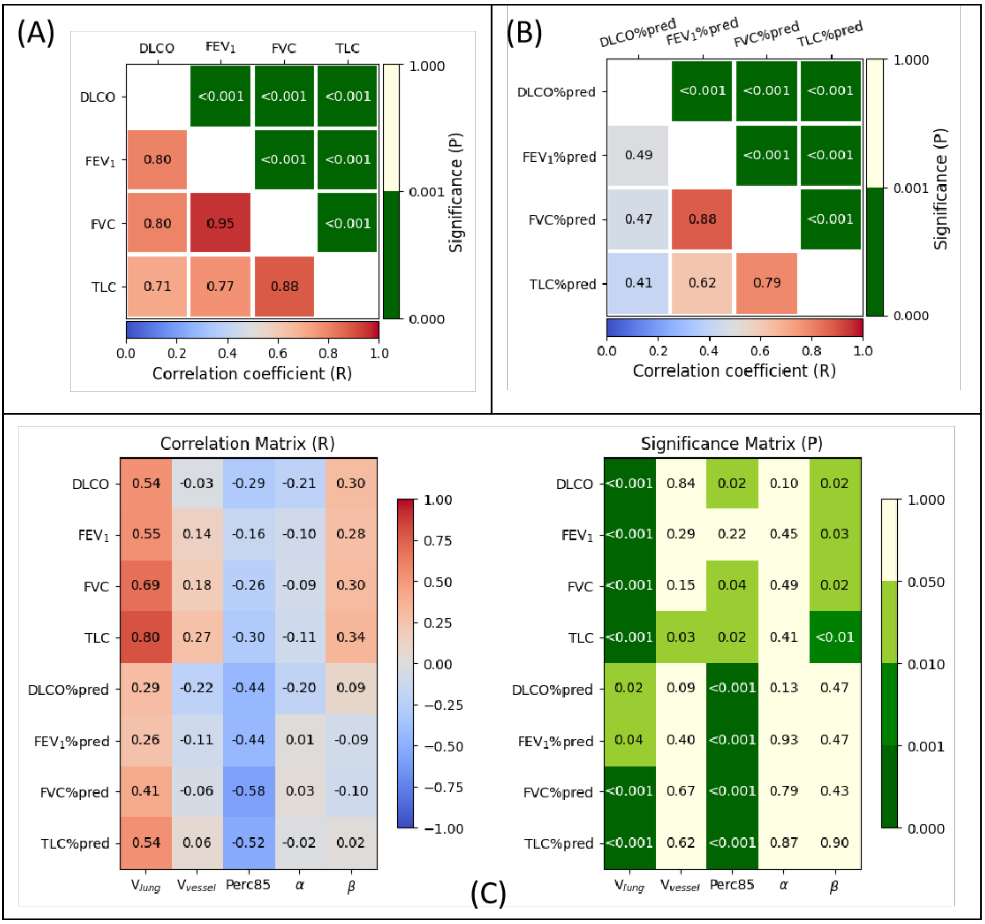


Figure 4.4: Pearson correlation coefficients and corresponding significance levels: (A) between different absolute PFTs; (B) between different PFTs%pred; and (C) between manually designed biomarkers and PFTs/PFTs%pred, from the testing dataset.

4.3.5 1-class versus 4-class (Q4)

Table 2 shows that the ICC values of the 4-class network (ICC=0.71, 0.76, 0.80, and 0.81) were similar to the ICC values of the four 1-class networks (ICC=0.70, 0.76, 0.76, and 0.84). The R and MAE values for the two network designs were also similar. *p*-values of 0.60, 0.13, 0.18 and 0.61 indicate that the results of the four networks with 1-class output did not show a significantly difference compared to the 4-class network. Because the 4-class network can output four PFTs at a same time, which saves training & inference time and GPU memory, all the following networks were trained with a 4-class output.

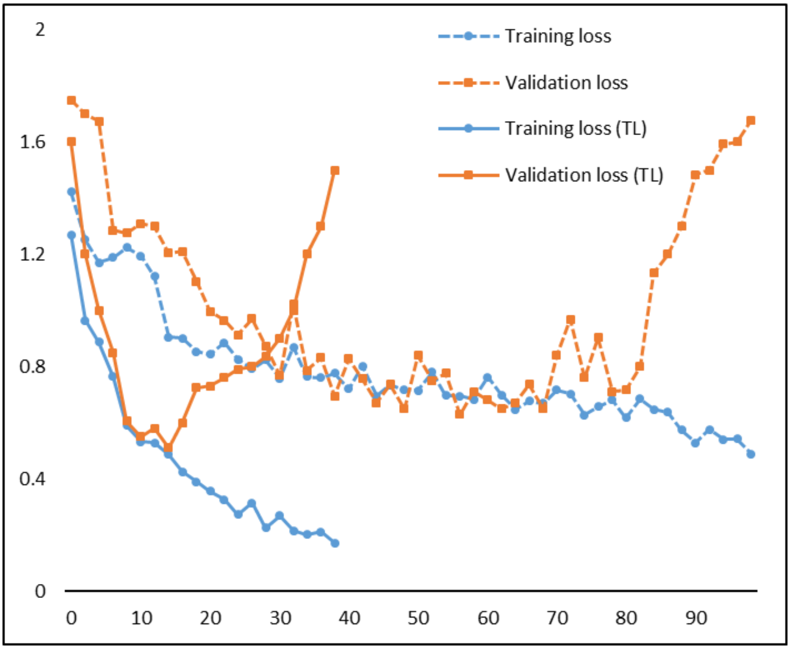


Figure 4.5: Comparison of training and validation curves with or without transfer learning (TL) on the same fold.

Table 4.3: PFT% estimation comparison between two methods.The units of MAE are mL/min/mm Hg for DLCO and liter for FEV_1/FVC/TLC.

Method	Metrics	DLCO%	FEV_1%	FVC%	TLC%
Direct	R	0.32	0.47	0.51	0.54
	ICC	0.30	0.47	0.50	0.53
	MAE	17.30	14.88	14.98	13.59
	MAPE	29%	18%	18%	17%
Indirect	R	0.60	0.65	0.74	0.76
	ICC	0.60	0.60	0.69	0.75
	MAE	13.85	14.79	13.76	11.04
	MAPE	22%	18%	16%	13%

4.3.6 Contribution of the different chest regions (Q5)

The PFT estimation performance of our proposed networks are summarized in Figure 4.8 and Table 4.4. DLCO was always the most difficult parameter to estimate, followed by FEV_1 and then FVC and TLC. The network trained on the whole CT volume achieved the best performance (highest ICC and R, as well as lowest MAE values). The network trained on both lungs performed slightly worse. The performances for the left and right lung were similar, which implies similar contribution of left and right lung to the

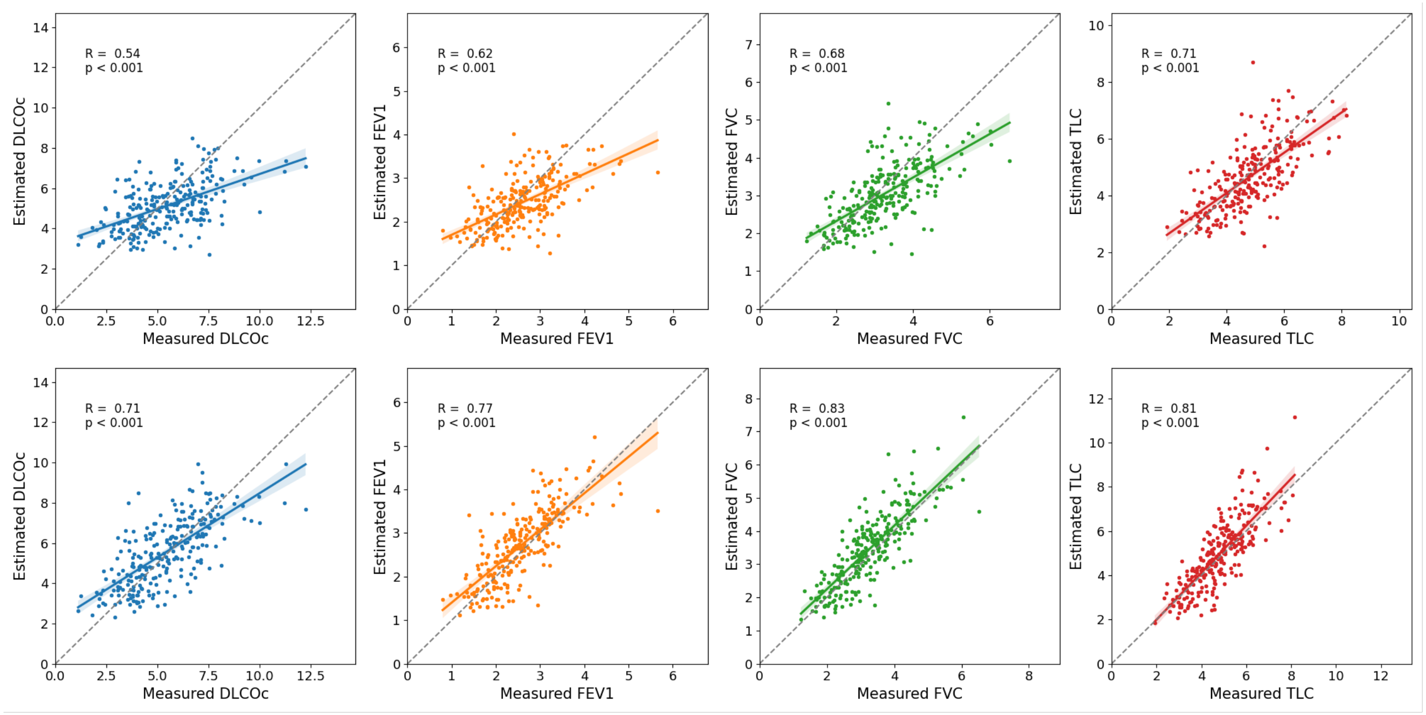


Figure 4.6: Scatter plots comparison between networks without (upper) and with (lower) transfer learning. Each image shows the identify line (dot line), regression line (solid line) and the 95% confidence intervals (shaded areas).

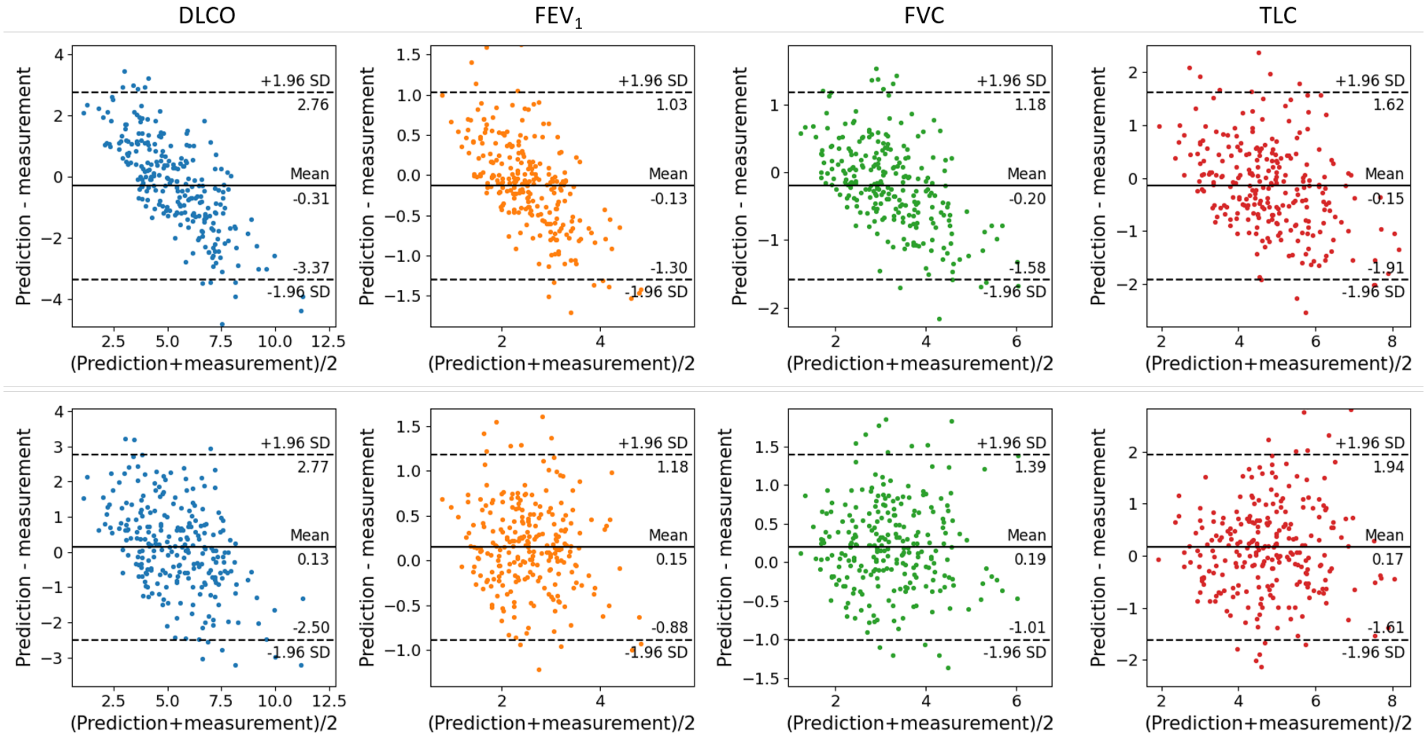


Figure 4.7: Bland-Altman plots comparison between networks without (upper) and with (lower) transfer learning. The mean difference and the limits of agreement (mean ± 1.96 × SD, where SD is the standard deviation of the differences) are also shown on the plots.

PFT estimation. The network trained on pulmonary vessels performed worse than the previous networks for FEV_1, FVC and TLC estimation, but better for DLCO estimation. The network trained on binarized vessels (1 as foreground and 0 as background) achieved similar ICC and MAE values and slightly higher R values, as compared to the network trained on gray scale vessels. The networks trained on gray scale vessels performed the worst compared to the other networks, but still better than the manually designed vessel based biomarkers α (R=-0.29) and β (R=0.32).

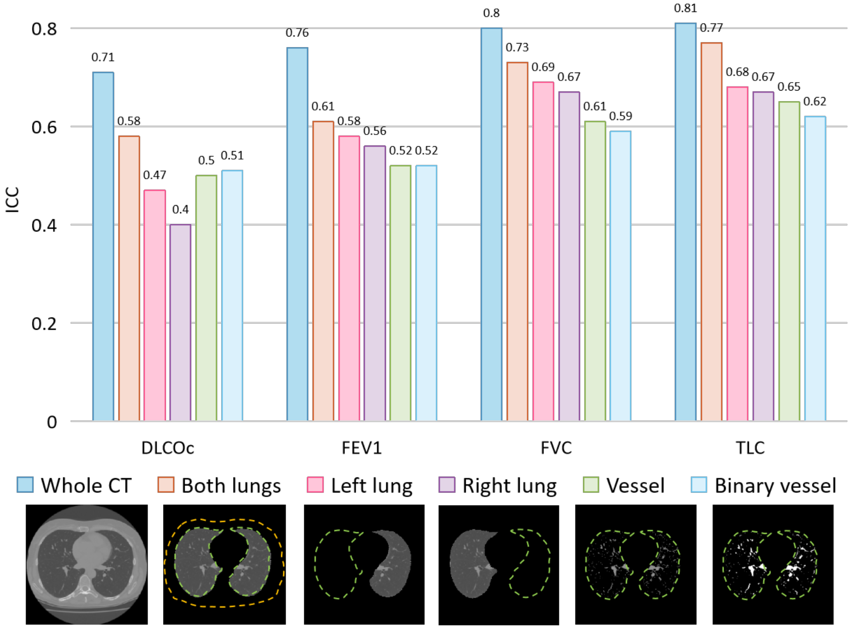


Figure 4.8: Performance of the networks trained on different regions of the chest. The bars with different colors represent networks trained by different regions, which are illustrated at the bottom.

If we look at the RAMs generated from the network trained on whole CT, for some patients, the highlights are limited to the lungs, see Figure 4.9 (left, row of Whole CT). For some other patients, the highlights also appeared outside the lungs (i.e. in the chest wall), see Figure 4.9 (right, row of Whole CT). For all networks, the two center regions of the two lungs, where the larger pulmonary vessels are located, were highlighted the most. This pattern applied to the RAMs of all networks. The coronal views of RAMs were vertically discontinuous; this is because the X3D-M network only applied pooling layers along the x and y axes, while leaving the z axis free of pooling layers, before the layer where our RAM_3D was applied. That led to a narrow reception field along the z axis.

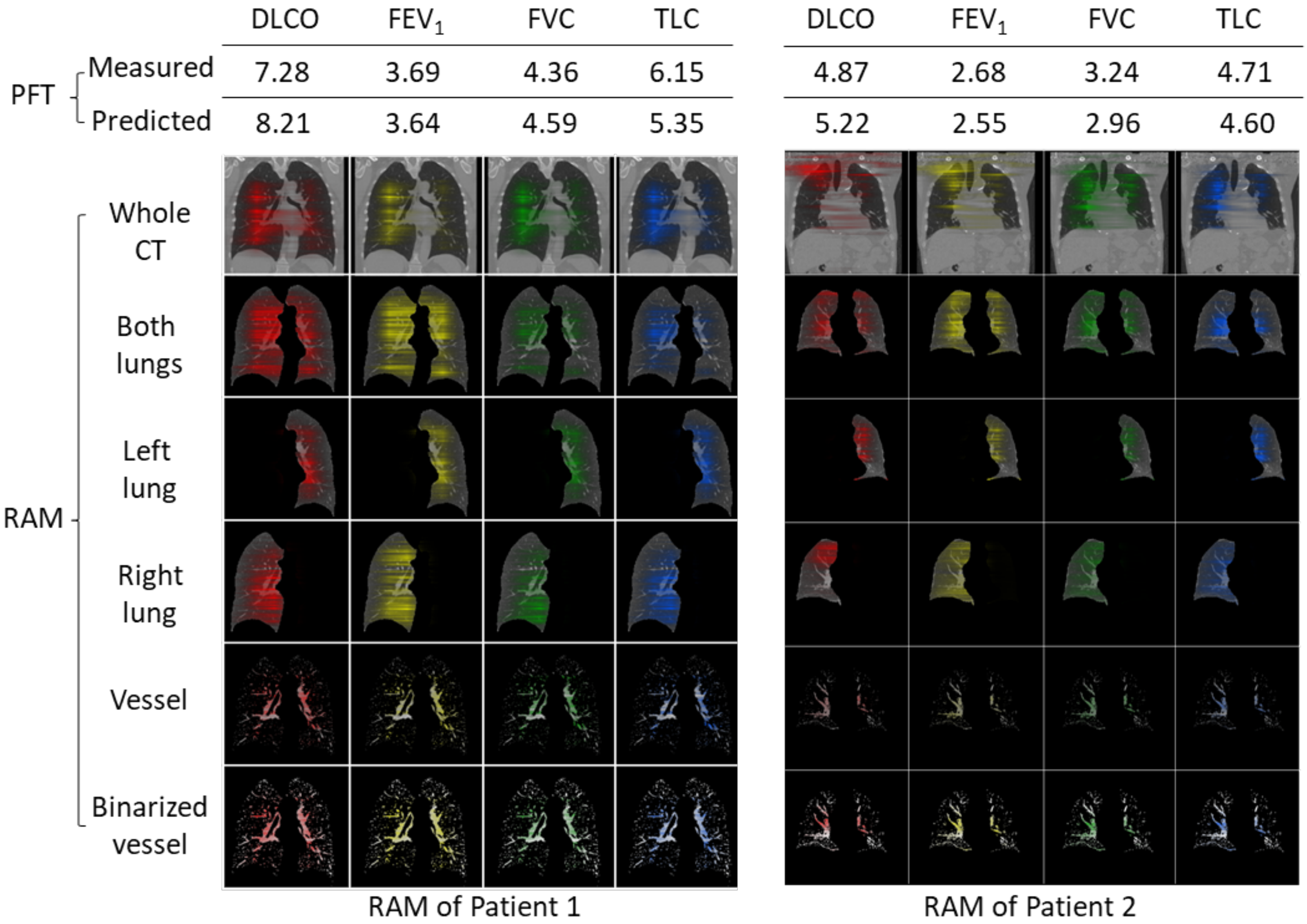


Figure 4.9: RAMs of two patients for different networks (coronal view). RAMs of different rows are generated from the networks trained by different regions. Red, yellow, green and blue highlight the attention of networks on DLCO, FEV$_1$, FVC and TLC, respectively.

Table 4.4: Performance of the networks trained from different inputs. The units of MAE are mL/min/mm Hg for DLCO and liter for FEV_1/FVC/TLC. DLCO%: DLCO percent prediction; FEV_1%: FEV_1 percent prediction; FVC%: FVC percent prediction; TLC%: TLC percent prediction.

Input	Metrics	DLCO	FEV_1	FVC	TLC	Mean
Whole CT	R	0.71	0.77	0.83	0.82	0.78
	MAE	1.04±0.86	0.41±0.36	0.49±0.42	0.63±0.48	0.64±0.53
	MAPE	22%±24%	18%±18%	16%±13%	13%±9%	17%±16%
Both lungs	R	0.59	0.62	0.73	0.78	0.68
	MAE	1.19±0.98	0.45±0.42	0.49±0.44	0.57±0.49	0.68±0.58
	MAPE	27%±32%	20%±22%	16%±15%	13%±12%	19%±20%
Left lung	R	0.47	0.58	0.69	0.68	0.61
	MAE	1.29±1.02	0.48±0.43	0.56±0.45	0.74±0.58	0.77±0.62
	MAPE	28%±30%	20%±19%	18%±14%	16%±13%	21%±20%
Right lung	R	0.45	0.59	0.69	0.69	0.61
	MAE	1.31±1.02	0.45±0.41	0.52±0.44	0.69±0.55	0.74±0.61
	MAPE	30%±39%	19%±20%	17%±17%	15%±14%	20%±23%
Vessels	R	0.51	0.53	0.62	0.66	0.58
	MAE	1.28±1.01	0.53±0.46	0.58±0.49	0.73±0.55	0.78±0.63
	MAPE	27%±30%	23%±22%	19%±16%	16%±12%	21%±20%
Binarized vessels	R	0.57	0.55	0.64	0.68	0.61
	MAE	1.28±1.04	0.52±0.45	0.62±0.50	0.76±0.56	0.80±0.64
	MAPE	27%±29%	22%±22%	19%±18%	16%±13%	21%±21%

4.3.7 Comparison standard repeatability criteria (Q6)

After comparing our method to previous works, putting the results of our network into clinical perspective is still needed. Estimating PFTs from CT scans by human experts is impractical for obvious reasons, therefore we could not compare our method with human observations. However, we could compare our results with the theoretically best obtainable result, as determined by the officially recommended repeatability criteria for spirometric measurements. The PFT measures are normally obtained by means of three repetitions of the measurements [17]. According to the most recent official standard on pulmonary function testing [17], the repeatability for DLCO and TLC obtained by the helium dilution technique between technically acceptable measurements should be within 10% of the average value. The repeatability criterion for FEV_1 and FVC is that differences should be lower than 0.15 L [17]. To have a fair comparison between different PFTs, the acceptable errors of 0.15 L for FEV_1 and FVC were

Table 4.5: Comparison between official repeatability criteria and the relative error of our method.

	DLCO (%)	FEV_1 (%)	FVC (%)	TLC (%)
Repeatability criteria	10	6	5	10
Our method (MAPE ± STD)	20 ± 18	19 ± 14	15 ± 12	13 ± 11

Table 4.6: Multivariable stepwise linear regressioin analysis for DLCO, FEV_1, FVC and TLC.

DLCO	
Parameter	R
NetDLCO	0.75
NetDLCO+V_{lung}	0.78
NetDLCO+V_{lung}+β	0.81

FEV_1	
Parameter	R
$NetFEV_1$	0.74
$NetFEV_1$+Perc85	0.83

FVC	
Parameter	R
NetFVC	0.82
NetFVC+Perc85	0.87
NetFVC+Perc85+V_{lung}	0.88

TLC	
Parameter	R
NetTLC	0.83
NetTLC+Perc85	0.88
NetTLC+Perc85+V_{lung}	0.90

divided by the mean measured values in Table 4.1, obtaining a percentage error of 6% ($RE_{FEV_1} = 0.15/\bar{y}_{FEV_1} = 0.15/2.62 \approx 6\%$) and 5% ($RE_{FVC} = 0.15/\bar{y}_{FVC} = 0.15/3.32 \approx 5\%$). As shown in Table 4.5, the repeatability criteria is 10%, 6%, 5% and 10% for DLCO, FEV_1, FVC and TLC, respectively.

4.3.8 Multiple variable regression analysis (Q7)

A multivariable regression analysis was performed to evaluate if manual biomarkers could still contribute to the estimation of PFTs, in addition to the estimation of our method. Multivariable stepwise linear regression was performed with DLCO as the dependent variable and the network-estimated DLCO (NetDLCO), Perc85, α, β, V_{lung}, and V_{vessel} as independent variables. We performed similar analyses for FEV_1, FVC and TLC (Table 4.6). The multivariable stepwise regression analysis showed that the inclusion of V_{lung} and β could significantly improve the regression coefficient of DLCO (from R =0.75 to R =0.81). Similarly, by including V_{lung} and Perc85, the estimation of FEV_1, FVC and TLC could also be significantly improved. Therefore, although we have developed automatic networks that outperformed manually designed biomarkers as single predictors, these manual biomarkers could still contribute further to the estimation of PFTs.

Table 4.7: Comparison Between Our Method and Previous Automatic Methods for the Estimation on PFTs. Because Previous Methods Did Not Estimate DLCO and TLC, the Corresponding Results are Not Included. NR: Not Reported

Method	**Study population**	**Backbone**	**#Subjects**	**MAE**		**R**	
				FEV	**FVC**	**FEV**	**FVC**
[21]	Lung cancer	ResNet [118]	546	0.33	0.37	0.73	0.82
[107]	Risk of lung cancer	I3D [119]	16148	0.22	0.22	NR	NR
Our method	SSc	X3D [113]	316	0.41	0.49	0.77	0.83

4.4 Discussion and conclusions

This chapter demonstrated that it is feasible to automatically estimate comprehensive PFTs and PFTs%pred from chest CTs, using deep learning. Our results indicate that CT scans can assist in estimating PFTs with considerable predictive accuracy.

To the best of our knowledge, there is no currently published work on estimating PFT values for SSc patients. The only two works [107, 21] that aimed to automatically estimate PFTs from CT were developed based on 546 subjects diagnosed with lung cancer [5] or 16148 subjects with a risk of developing lung cancer [107], as shown in Table 4.7. The R values of our method is slightly higher than Choi, et al. [21], while the MAE of our method is higher than the other two works. Because we have already applied X3D with transfer learning, which achieved state-of-the-art performance on video recognition, possible reasons of the performance gap may include: 1) Dataset sizes are different. Considering the best MAE was achieved by the network trained on the largest number of subjects (16148 patients), there is still potential benefit from increasing the training set size for our network. 2) Different disease has different pathogenesis, leading to different difficulties in learning the relation between function and structure. In SSc patients, for example FVC may remain stable while DLCO significantly decline over time [118]. Therefore, compared with previous work, which only estimate FVC and FEV$_1$, our work estimates a more comprehensive set of PFT measurements (DLCO, FEV$_1$, FVC and TLC) for SSc patients, rendering it more clinically relevant for SSc patients, that is likely of additional clinical value. The comparison is for reference only, since it is based on different datasets sizes, different networks and different diseases. Implementing the two methods on our dataset to have an absolute fair comparison is impractical because the other two methods did not have public available pre-trained weights as what we have for X3D from Kinetics dataset.

The observed correlation between α, β and PFT in our study differs from the original report [58]. This is because patients with lung fibrosis were excluded in the original report, whereas our dataset comprised 80% CT images with various degrees of fibrosis. In patients with lung fibrosis, fibrotic areas led to over-segmentation of vessels, decreasing the correlation between the α and β calculated and PFT.

Estimating TLC was consistently more successful than for the other three PFT measurements. The MAE and prediction uncertainty in percentage (represented by MAPE) of TLC are always lower than the others. This could be explained by the fact that lung volume calculated by simply counting the number of voxels in both lungs is already strongly correlated with TLC, as measured by spirometry [29]. The estimation of DLCO consistently underperformed compared to the other three measures, since gas exchange is less correlated with TLC. While the agreement between estimated PFTs and the measured PFTs ranged from moderate to good, the agreement between directly estimated PFTs%pred and measured PFT%pred ranged from poor to moderate. This finding is consistent with a previous report [107]. This can be attributed to the challenge of estimating reference equations for diverse population groups. Therefore we proceeded our research on estimating absolute PFTs, because 1) estimating PFTs%pred indirectly was more accurate than a direct estimation; and 2) this approach is more flexible as other PFT biomarkers, such as FEV_1/FVC [14] and FVC/DLCO [120], can then also be derived from the estimated absolute PFT values.

From the comparison between networks trained by different regions of CT scans, we found that networks trained on the whole CT image could achieve the best performance. CT masked by both lungs produced slightly inferior results, suggesting that tissue outside the lung area still contribute to the estimation of PFTs to some extent. This observation could be verified by Figure 8, where some regions outside lungs are highlighted for Patient 2 in the first row (network trained on the whole CT) while regions outside lungs are not highlighted for Patient 1. This suggests that the interaction between the chest wall and intercostal muscles contribute to PFTs in some patients. This is consistent with the clinical knowledge that stronger intercostal muscles combined with a compliant chest wall will have a positive effect on PFTs [121]. In contrast, chest wall stiffness, as sometimes observed in patients with SSc, may negatively influence PFTs. RAMs of different networks trained on different regions of CT have similar patterns: the entire lung is highlighted to different extents while the center regions of lungs are highlighted mostly. This implies that the networks for estimating PFTs need global information of the whole CT, while focusing more on the center regions where the largest vessels are located. This is consistent with findings in the previous study [107]. Apparently, it would be of greater clinical value if we could further extract what the contributors are. However, limited by the low resolution of current RAM techniques, we could not give more detailed contributors. Because a visualization centered on model interpretability would bridge the divide between AI-driven analyses and clinical practitioners, we will research more detailed visualization methods in our future work.

It is surprising that networks trained solely on grayscale vessels or binarized vessel masks still achieved R and ICC values over 0.5 for all four PFT measures. This implies that, in addition to vessel radius histogram information (used by α and β), the spatial structure of the vascular tree plays a more significant role in estimating PFTs.

Currently, there are no established guidelines for the level of precision required to implement new techniques in clinical practice for predicting PFTs. The repeatability criteria to measure PFTs is the standard for spirometry, which is the upper limit of any methods which aim to replace spirometry. At the current stage, our method could not perform competitively with spirometry if we compare our MAE with the repeatability criteria of spirometry. In addition, our method has not been prospectively validated, so it can only be used in research at present. Nevertheless, our method still 1) verified the possibility to estimate PFT, especially DLCO, from CT scans for patients of systemic sclerosis. 2) paved the way for more accurate methods and foster medical community to establish standards and regulations for such methods in the future. It would be beneficial to witness its integration into the clinical (randomized) trials in the future.

The multiple variable regression analysis showed that previous manually designed biomarkers could further explain variation in PFTs. This observation implies that if we add manually designed biomarkers as extra input to the networks, we might improve networks further in future research.

There are some limitations to our research. Because of the lack of public available 3D network weights pre-trained by lung CT images, we applied TL from Kinetics dataset, which may not optimal for PFTs estimation. In the future, we will explore the potential of network weights pre-trained from lung CT scans once we have the access to large annotated lung CT datasets. In addition, due to the need to protect healthy individuals from radiation exposure, it is not feasible to design a prospective experiment to collect CT and PFT pairs for a healthy control group. As a result, the retrospectively collected CT-PFT pairs in our study do not include healthy participants. Consequently, it remains uncertain whether our trained network can be applied to distinguish lung-structure relations in SSc patients from those in healthy individuals. Moreover, all SSc patients in this study were scanned with the same scanner at the same center. Therefore, additional experiments involving other patients and scanners are necessary to verify the generalizability of our deep learning method in the future. To achieve optimal performance for new scanners, we may need to fine-tune our model based on new datasets. If more image modalities are available in the future, we can explore the potential scalability of our methods on other modalities. Therefore, external validation is needed to be imbedded in clinical (randomized) trials. For now, the method can only be used for clinical research.

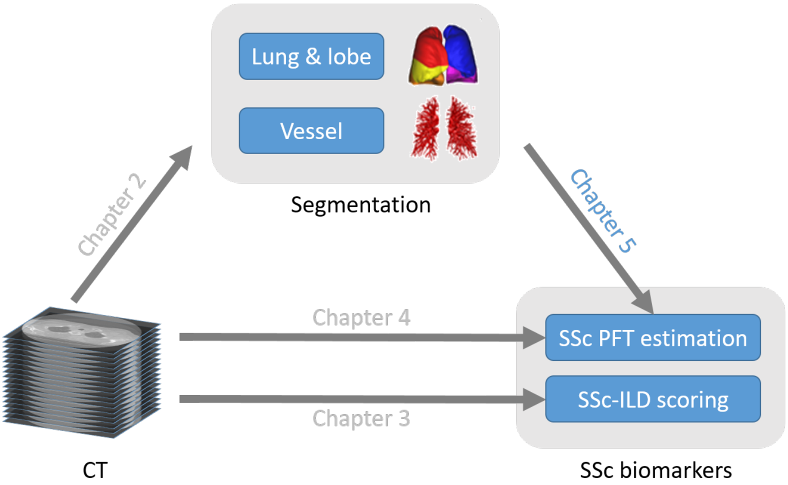
In conclusion, our method can automatically and comprehensively estimate PFTs for SSc patients. This can help to estimate lung function for patients who are unable to perform these tests, while there are CT scans available. The method can form a basis for studying the relation between function and structure, since we found for example that regions outside the lungs also contribute to the estimation of PFTs. For future work, we will investigate how to extract the contributors outside the lungs in more detail, which would be of great clinical value.

4.5 Acknowledgements

This work is supported by the China Scholarship Council No.202007720110. This work was performed using the computational resources from the Academic Leiden Interdisciplinary Cluster Environment (ALICE) provided by the Leiden University.

5

Using 3D point cloud and graph-based neural networks to improve the estimation of pulmonary function tests from chest CT



This chapter was adapted from:

Jia, Jingnan, Bo Yu, Prerak Mody, Maarten K. Ninaber, Lucia JM Kroft, Anne A. Schouffoer, Marius Staring, and Berend C. Stoel. "Using 3D point cloud and graph-based neural networks to improve the estimation of pulmonary function tests from chest CT." (*submitted*)

Abstract

Pulmonary function tests (PFTs) are important clinical metrics to measure the severity of interstitial lung disease for systemic sclerosis patients. However, PFTs cannot always be performed by spirometry if there is a risk of disease transmission or other contraindications. In addition, it is unclear how lung function is affected by changes in lung vessels. Therefore, convolution neural networks (CNNs) were previously proposed to estimate PFTs from chest CT scans (CNN-CT) and extracted vessels (CNN-Vessel). Due to GPU memory constraints, however, these networks used down-sampled images, which causes a loss of information on small vessels. Previous literature has indicated that detailed vessel information from CT scans can be helpful for PFT estimation. Therefore, this chapter proposes to use a point cloud neural network (PNN-Vessel) and graph neural network (GNN-Vessel) to estimate PFTs from point cloud and graph-based representations of pulmonary vessel centerlines, respectively. After that, we combine different networks and perform multiple variable step-wise regression analysis to explore if vessel-based networks can contribute to the PFT estimation, in addition to CNN-CT. Results showed that both PNN-Vessel and GNN-Vessel outperformed CNN-Vessel, by 14% and 4%, respectively, when averaged across the intra-class correlation coefficient (ICC) scores of four PFTs metrics. In addition, compared to CNN-Vessel, PNN-Vessel used 30% of training time (1.1 hours) and 7% parameters (2.1 M) and GNN-Vessel used only 7% training time (0.25 hours) and 0.7% parameters (0.2 M). We combined CNN-CT, PNN-Vessel and GNN-Vessel with the weights obtained from multiple variable regression methods, which achieved the best PFT estimation accuracy (ICC of 0.748, 0.742, 0.836 and 0.835 for the four PFT measures respectively). The results verified that more detailed vessel information could provide further explanation for PFT estimation from anatomical imaging.

5.1 Introduction

Systemic sclerosis (SSc) is a rare immune-mediated connective tissue disease that affects different organs. Up to 90% of SSc patients have lung involvement, with pulmonary function abnormalities such as interstitial lung disease (ILD) [14]. To evaluate progression of SSc-ILD, various pulmonary function tests (PFTs) are used as key measures, such as diffusion capacity for carbon monoxide (DLCO), forced expiratory volume in 1 second (FEV_1), forced vital capacity (FVC) and total lung capacity (TLC) [14, 15, 16]. PFTs, however, can not always be performed if there is a risk of disease transmission, e.g. in patients with COVID-19 [21, 22] or other contraindications such as myocardial infarct, pulmonary embolism or ascending aortic aneurysm [23, 24].

CT scans provide high-resolution details of the lungs, and is hence regarded the gold standard for diagnosing SSc-ILD [122, 35]. Therefore, when PFTs are not possible and CT scans have been made previously of SSc patients for other purposes, it could be of interest to estimate PFTs from CT images. In addition, the clinical research community is interested to investigate the relation between structural (especially vascular) changes due to SSc and pulmonary function. Previous work has extracted some quantitative biomarkers from chest CT of SSc patients, which correlated with PFTs [16, 35, 96, 123]. For instance, two lung vascular tree-based biomarkers, extracted from the lung vessel radius histogram, were found to be correlated with DLCO [96]. However, these biomarkers are investigated only in terms of their correlations with PFTs but do not estimate PFT values directly. Thus, there is a need to investigate the use of high-resolution pulmonary vessel masks from CT scans to directly estimate PFTs.

Deep neural networks have shown some promise for PFT estimation from CT scans [21, 107, 124] due to their powerful capability to extract features from images. To the best of our knowledge, there are just three works on the PFT estimation using deep learning from CT scans. The first work is for patients with lung cancer [21]. A network was developed to estimate FEV_1 and FVC, which achieved R values of 0.73 and 0.82, respectively. Their network consisted of a ResNet-50 [89] for feature extraction and a bidirectional long short-term memory (BiLSTM [125]) network for PFTs prediction. The second work is for subjects at risk of lung cancer [107]. Two separate networks were trained to estimate FEV_1 and FVC, respectively. Both of the previous methods estimated FEV_1 and FVC only, lacking DLCO and TLC. In addition, none of them were developed for SSc patients. The third work for PFT estimation is for SSc patients [124]. The potential of CNNs was verified to estimate the complete set of PFTs. Intra-class correlation coefficient (ICC) values were achieved for DLCO, FEV_1, FVC and TLC as 0.71, 0.76, 0.80 and 0.81, respectively, from CT scans of SSc patients (see Figure 5.1-I, CNN-CT). In addition, motivated by the manually extracted vascular tree-based biomarkers [96], it was verified that CNN trained by 3D binary vessels could also be used to estimate complete PFTs (with lower accuracy) [124] (see Figure 5.1-J, CNN-Vessel).

The limitation of this previous work for SSc patients [124] is that the 3D images were down-sampled before being fed into networks, due to GPU memory limitation (see Figure 5.1-B and D). This led to approximately 20× loss of detailed vessel information. Down-sampling is a compromise that has to be made, because even with a batch size of 1, the GPU usage was still at least 11 GB, and to use the original resolution CT, a single GPU of at least 220 GB is needed, which does not yet exist. Because the absence of small vessels was reported to influence PFTs significantly [96], we assumed that the better performance could be achieved by overcoming the information loss of small vessels during the development of PFT-estimation networks. Therefore, the goal of this study was to explore the possibility of improving the deep-learning-based PFT estimation performance by efficiently utilizing detailed vessel information.

Given that most voxels of 3D binary vessel images are background, we propose to convert binary vessel images to two different data formats – point clouds and graphs, to efficiently utilize the relevant vessel voxels. A point cloud is a sparse representation of the binary vessel image and contains structural information on the pulmonary vessels. Such a representation could be used to develop a point cloud neural network (PNN) To extract structural features relevant to PFT estimation. However, point clouds do not explicitly represent the geometric relationships between points. Given that points in vessels are not completely independent but belong to different sub-branches, such information could be useful to PFT-estimation networks. Therefore, we also built graphs of vessels by using edges to explicitly connect the points in the same sub-branches. A graph neural network (GNN) could be developed based on the graphs of vessels.

Thus, our contributions are as follows

- We, for the first time, propose to use a point cloud neural network (PNN-Vessel) and a graph neural network (GNN-Vessel) to estimate PFTs from pulmonary vessels.
- We explored the architectures and training strategies for PNN-Vessel and GNN-Vessel.
- We explored how pulmonary vasculature influences pulmonary function and verified that higher resolution of vessels, which include more small vessels, could lead to higher PFT estimation accuracy.
- We, for the first time, successfully combined CNN, PNN and GNN together and achieved the best PFT estimation performance.

The remaining chapter is organized as follows. In Section 2, the related work was reviewed. Then, our solution for the PFT estimation based on point clouds and graphs is described in Section 3. Detailed experiment setting and results are shown in Section 4. Finally, in Section 5, the experiment results are discussed.

5.2 Related work

5.2.1 Point cloud and point cloud neural netowk (PNN)

A 3D point cloud is a set of discrete data points in space, which is commonly used in 3D scanners, LIDAR and RGB-D cameras [126]. In a 3D point cloud, each point position has its set of Cartesian coordinates (x, y, z) and other extra features like color and depth [127]. A point cloud is stored as an unordered set of vectors with shape of (N, D+C), where N is the number of points, D is the dimension of the coordinates (normally 3 for points in 3D space) and C is the number of extra features.

In the past years, a great number of different neural networks were proposed for point cloud data. PointNet is a pioneering deep neural network that directly analyzes point clouds for both classification and segmentation [128]. The basic idea of PointNet is to learn a spatial encoding of each point through a sequence of shared multi-layer perceptions (MLPs) and then aggregate all individual point features to a global point cloud signature by a global pooling [128]. Before the global pooling, each point is processed identically and independently, which ensures invariance to permutations but also ignores the local context. To overcome this limitation, PointNet++ [129] was released subsequently and became one of the most influential networks by leveraging local neighborhoods at multiple scales. PointNet++ hierarchically abstracts features of point clouds using a number of set abstraction (SA) blocks. An SA block consists of a subsampling layer to down-sample the incoming points, a grouping layer to query neighbors for each point, and a simplified PointNet to extract and aggregate features. Because PointNet++ affirmed the significance of leveraging local neighboring information, more networks were proposed to aggregate spatially-local correlation information. In PointCNN [130], an X-Conv operator was introduced, which can weight and permute input features before they are processed by a typical convolution. This is the generalization of a CNN on the point cloud domain. Inspired by the success of transformers in natural language processing [131] and image analysis [132], Point Transformer [133] networks were designed with self-attention layers for point clouds.

The latest network, which achieved the best performance on most of the benchmarks, is PointNeXt [134]. It used the same design of PointNet++ [129] but applied different hyper-parameters and training strategies for different tasks. It concluded that by just optimizing training strategies (e.g. data augmentation and optimization techniques) for different tasks, PointNet++ could exceed the current state of the art. Inspired by this works, the PNN-Vessel described in this chapter also uses the design of

PointNet++ and explores the optimal training strategies and hyper-parameters for PFT estimation.

5.2.2 Graph and graph neural network (GNN)

A graph is a data structure that models a set of objects (nodes) and their relationships (edges) [135]. Because of the unstructured nature of graph data, CNNs cannot be applied directly to graphs. Therefore, a great number of networks were proposed to mimic the principle of CNNs by aggregating information from neighboring nodes using different aggregation strategies [136, 137, 138, 139, 140]. ChebConv [140] is one of the earliest attempts, and approximates spectral graph convolutions using Chebyshev polynomials. GCN [137] simplifies ChebConv by utilizing only the first two Chebyshev polynomials while still outperforming it on real-world datasets. The core operations of a GCN is aggregating (average or maximum) neighboring features, followed by an MLP layer to increase/decrease feature dimensions and a non-linear activation function. Simplifying Graph Convolution (SGConv) [136] is a simplified version of GCN, which aims to reduce computational costs without sacrificing too much performance. GraphConv [138] was proposed as a hierarchical version of k-GNNs, based on the k-dimensional Weisfeiler-Lehman (WL) algorithm, which is able to work with the fine- and coarse-grained structures of a given graph. Graph Isomorphism Network Convolution (GINConv) [139] implements graph isomorphism tests [141] in a neural network. Graph Attention Network Convolution (GATConv) [142] was introduced with the concept of self-attention mechanisms to graph convolutions. Each node computes attention coefficients with all of its neighbors, allowing different neighbors to contribute differently to the updated node representation. The key difference between PNN-Vessel and GNN-Vessel is that PNN-Vessel regards all neighboring points that have nearest distances, whereas GNN-Vessel only considers points as neighbors if they have an direct edge between them. This is critical in our task where two neighboring points in a vascular tree should belong to the same vessel branch. Therefore, we proposed the GNN-Vessel to estimate PFTs from our built vessel graph dataset.

5.3 Methodology

In this section, we describe how to convert 3D grid vessel images to point cloud and graph datasets. We then separately illustrate our two proposed networks: PNN-Vessel and GNN-Vessel.

Two previously developed for PFT estimation are shown in Figure 5.1-I and J, which were trained by down-sampled CT images and down-sampled binary vessel images seperately. The two new methods developed in this chapter are shown in Figure 5.1-K and L. The high-resolution 3D grid vessel images were skeletonized resulting in two types of data. The first type of data is point cloud with coordinates and vessel radius as the features of each point. The second type of data is graph with coordinates

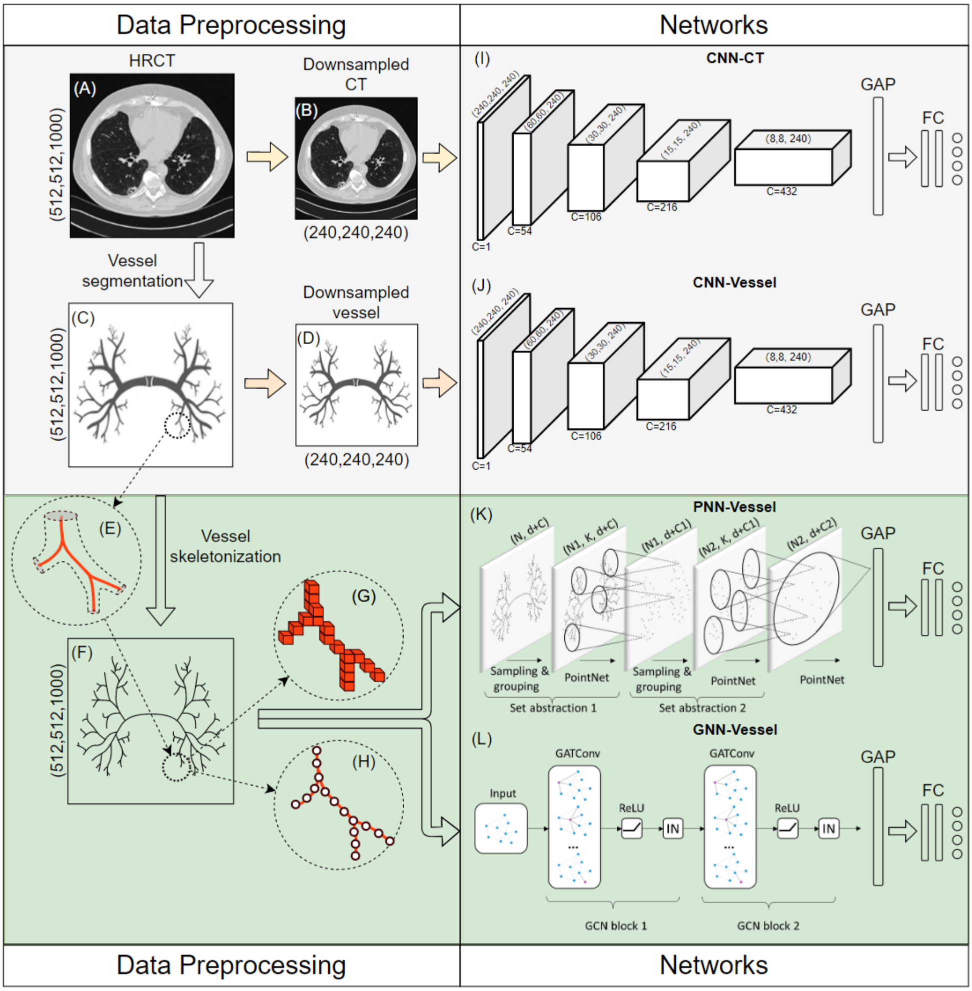


Figure 5.1: Overview of the four PFT estimation methods. The data-preprocessing steps are on the left and neural network architectures on the right. The existing methods [124] are shown on the upper part (gray background), the work proposed in this chapter is on the bottom part (green background). The shape of each grid image is marked next to it. Each of the four networks consists of an encoder and a global average pooling (GAP) layer, followed by a fully-connected (FC) block. The FC block includes two fully connected layers with 1024 nodes for each of them. The four output nodes represent DLCO, FEV_1, FVC and TLC, respectively.

and vessel radius as the features of each node, and edge between two adjacent nodes. PNN-Vessel and GNN-Vessel were then developed for the two datasets, respectively. The details of dataset preparation and network design are described below.

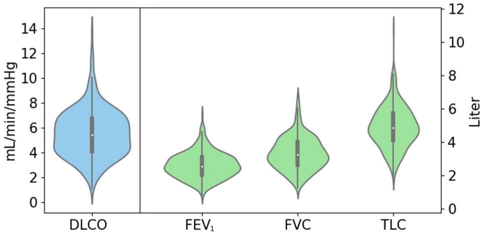


Figure 5.2: Measured PFTs distribution on the whole dataset.

5.3.1 Acquisition of CT images and PFTs measurements

In this study, we retrospective selected a cohort of 316 patients referred to our specialized outpatient health care program (focused on combined care in systemic sclerosis) at Leiden University Medical Center. These patients, identified by the referring rheumatologists as having a definitive diagnosis of Systemic Sclerosis (SSc) or presenting with a strong clinical suspicion, were examined by high-resolution computed tomography (HRCT) scans and subsequent pulmonary function tests (PFTs) within 10 days. All subjects underwent scanning at full inspiration without contrast enhancement using an Aquilion 64 CT scanner (Canon Medical Systems), configured at 120 kVp, a median tube current of 140 mA, a rotation time of 0.4 seconds, a collimation of 64 × 0.5 mm and a helical beam pitch of 0.8; leading to a median CTDIvol of 8.2 mGy. The images were reconstructed with filtered back projection and an FC86 kernel, with a median pixel spacing of 0.64 mm × 0.64 mm, with a slice thickness and increment of 0.5 and 0.3 mm, respectively. PFTs were performed using a spirometer under ERS/ATS guidelines [108, 18] including DLCO, FEV_1, FVC and TLC. While DLCO was measured in units of mm/Hg/min, FEV_1, FVC, TLC were measured in units of liter (Figure 5.2). We divided this dataset into two distinct subsets: 253 CT-PFT pairs were allocated for four-fold training and cross-validation, while the remaining 63 pairs were reserved for the separate testing phase. Written informed consent was provided by all patients. Approval of all ethical and experimental procedures and protocols was granted by the Institutional Review Board of the LUMC under protocol numbers P09.003/SH/sh, REU 036/SH/sh, REU 043/SH/sh and B19.008/KB/kb. For the current specific analysis no separate research protocol was submitted.

5.3.2 Dataset preparation

5.3.2.1 Dataset preparation for CNN-CT and CNN-Vessel

We reproduced previously developed CNN-CT and CNN-Vessel networks [124]. The dataset preparation for CNN-CT and CNN-Vessel are shown in Figure 5.1-A, B, C, and D. After we obtained CT scans (Figure 5.1-A), the corresponding binary vessels masks were automatically segmented by a graph-cut based method [58] (Figure 5.1-C). To develop CNN-CT and CNN-Vessel, we down-sampled all CT scans and binary vessel images to an isotropic spacing of 1.5 mm, then cropped 3D patches of fixed size (240 × 240 × 240 voxels, see Figure 5.1-B and D).

5.3.2.2 Dataset preparation for PNN-Vessel

After we obtained the binary vessel masks, the centerlines of the vessel trees were extracted using skeletonization [56] with the calculated radius embedded in the centerline voxels, as illustrated in Figure 5.1-E and F. After we obtained the centerlines, we converted the points on the centerlines to a list of vectors, each of which includes three elements of spatial coordinates and one element of the corresponding vessel radius (Figure 5.1-G). Before the conversion, the average size of each 3D vessel grid image is about 512 MB with a mean size of 512 × 512 × 1000 pixels along x, y and z axes. After the conversion, the size of each image decreased dramatically to 0.3 MB. Each image is now represented as an array with a shape of N × 4, where N is the number of points on the vessel centerlines, with a mean value of 75,000. 4 means the four features for each point: the coordinates along the x, y and z axes, and the radius at that point.

5.3.2.3 Dataset preparation for GNN-Vessel

As shown in Figure 5.1-H, each vessel centerline tree is defined as a graph and the voxels of the centerlines are defined as nodes in a graph. Each node has four features: coordinates (x, y, z) and radius value (R). The edges exist between two adjacent voxels. The mean number of nodes is 75 k, with the mean number of edges of 132 k.

5.3.3 PNN-Vessel design

We designed a point-cloud-based neural network named PNN-Vessel, which extracts features from 3D point clouds to regress to four PFTs values. As shown in Figure 5.1 (K), it consists of an encoder, a global average pooling (GAP) layer and a decoder. The decoder consists of two FC layers with 1024 channels for each of them. The encoder consists of two set abstraction blocks. The details of the set abstraction blocks are shown in Figure 5.3. The i^{th} set abstraction take an input point cloud of size $N_i \times (D + C_i)$, representing N_i points with D-dim coordinates and C_i-dim feature for each point. It first subsamples N_{i+1} points by a farthest point sampling (FPS) strategy [129]. With the N_{i+1} points as the center, N_{i+1} groups of point sets were extracted and each group included K neighboring points. The N_{i+1} groups of point sets were then passed to a shared PointNet, resulting in N_{i+1} points with D-dim coordinates and C_{i+1}-dim features. Therefore, after a set abstraction block, the number of points was decreased from N_i to N_{i+1} while the number of features was increased from C_i to C_{i+1}. Finally, features from the last set abstraction layer are passed into a GAP layer

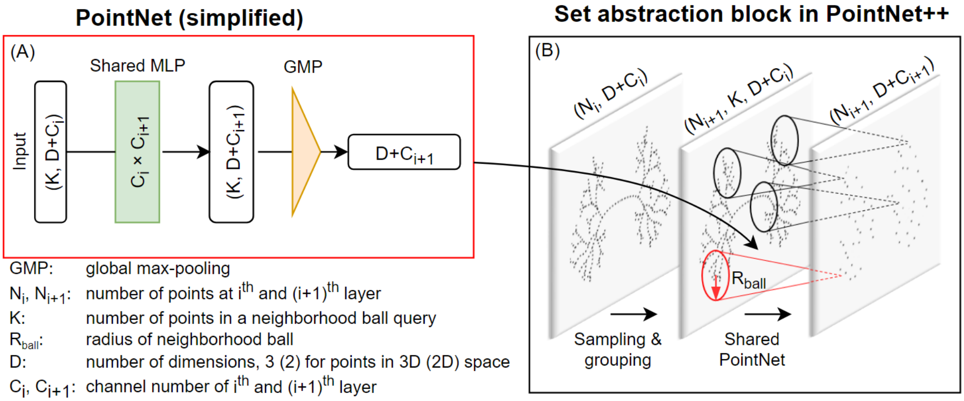

PointNet (simplified)
(A)
Input
(K, D+C_i)
Shared MLP
C_i × C_i+1
(K, D+C_i+1)
GMP
D+C_i+1
GMP: global max-pooling
N_i, N_i+1: number of points at i^th and (i+1)^th layer
K: number of points in a neighborhood ball query
R_ball: radius of neighborhood ball
D: number of dimensions, 3 (2) for points in 3D (2D) space
C_i, C_i+1: channel number of i^th and (i+1)^th layer
Set abstraction block in PointNet++
(B)
(N_i, D+C_i)
(N_i+1, K, D+C_i)
(N_i+1, D+C_i+1)
R_ball
Sampling & grouping
Shared PointNet

Figure 5.3: Architecture of PointNet and PointNet++ [134].

that combines point features followed by two FC layers that output four PFTs values. Inspired by PointNeXt, we successively tuned the hyper-parameters to achieve the optimal combination of hyper-parameters.

5.3.4 GNN-Vessel design

Although the point cloud format offers detailed information about the structure of the pulmonary vessels, they do not leverage connectivity information, an aspect relevant to the tree hierarchy of pulmonary vessels. In the vessel centerlines, points that are physically close to each do not necessarily belong to the same vessel. To utilize the connectivity information, we built a dataset using graphs of vessel centerlines (see Section 5.3.2) and trained the graph neural networks to estimate PFTs. The structure of our designed GNN-Vessel is shown in Figure 5.1 (L). It consists of an encoder, a global average pooling (GAP) layer and a decoder. The decoder consists of two FC layers with 1024 channels for each of them. The encoder consists of two graph convolution network (GCN) blocks. Each GCN block includes one graph attention convolution (GATConv) layer [142], followed by ReLU activation and instance normalization (IN) layers. The GATConv is a combination of a graph neural network and an attention layer. The attention layer helps focusing to the important information from the data instead of global data. As shown in Figure 5.3, the GATConv layer consists of a shared MLP, an attention calculation block and a aggregation block, which correspond to the following equations:

$$\mathbf{x}_i^{'} = F(\mathbf{x}_i^{in}) \tag{5.1}$$

$$\alpha_{ij} = softmax_j(\mathrm{ReLU}(\mathbf{x}_i || \mathbf{x}_j) \cdot \mathbf{w}) \tag{5.2}$$

$$\mathbf{x}_i^{out} = \sum_{j \in S_i} (\alpha_{ij} \cdot \mathbf{x}_i^{'}) \tag{5.3}$$

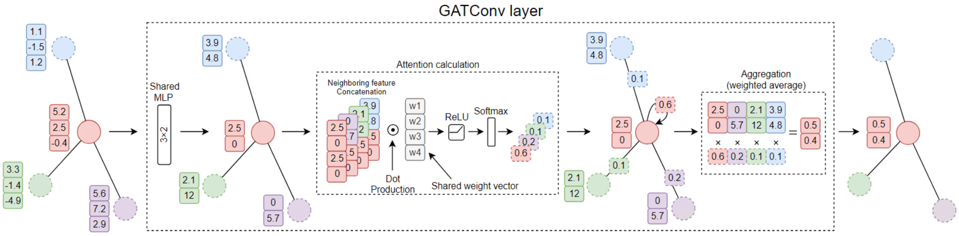


Figure 5.4: Illustration of the graph attention convolution (GATConv) layer [142]. Different colors correspond to different nodes. The feature vector of each node is shown next to the node with the same color.

where $\mathbf{x}_i^{in}$, $\mathbf{x}_i^{'}$, and $\mathbf{x}_i^{out}$ are the input, intermediate and output features of node i. $F(\cdot)$ represents the MLP layer which is shared by all nodes. α_{ij} is the attention weight between node i and j. $||$ denotes the concatenation operation. N_i is the neighborhood of node i in the graph.

5.3.5 Combination of different networks

After we developed PNN-Vessel and GNN-Vessel, it is of great interest to see if the combination of different networks could result in better performance. We proposed two methods to verify this.

In the first method, we developed a combined network (see Figure 5.5) which consists of several different encoders, corresponding to the pre-trained feature extractors for different data formats, and one decoder, which consists of two FC layers. A GAP layer, standardization layer and concatenation layer were used to fuse the features from different encoders. Because of GPU memory limitation, CNN-CT and CNN-Vessel could not be in the encoder at the same time. Given that all image information of vessel has already been included in CT, we chose to exclude CNN-Vessel from the combined network. The three encoders were initialized with the weights from the trained single CNN-CT, PNN-Vessel and GNN-Vessel, separately. The FC layers were initialized from scratch. The whole combined network was trained by 100 epochs with learning rate of 1e-4 and batch size of 1.

In the second method, we performed multiple variable regression analysis to evaluate if the vessel based networks could contribute further to the estimation of PFTs, in addition to the estimation of CNN-CT. Multiple variable stepwise linear regression (a statistical technique used to measure the impacts of several explanatory variables on a dependent variable) was performed with DLCO as the dependent variable and the estimated DLCO from CNN-CT, CNN-Vessel, PNN-Vessel and GNN-Vessel as independent variables. We performed the same analyses for FEV_1, FVC and TLC as dependent variables.

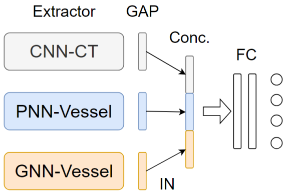


Figure 5.5: Illustration of the combined network. Global average pooling (GAP) was performed on the features from different extractors. After that, these features become 1-dimension vectors. Each of them were then performed instance normalization (IN) before concatenation (Conc.). The concatenated 1-dimension features were then fed into two fully-connected (FC) layers.

5.3.6 Evaluation metrics and statistical analysis

We used various metrics to evaluate the agreement between our network output and measured values (from spirometry). The mean absolute error (MAE) was used to reflect the absolute agreement. Since the unit and scale of the four PFTs are different, we also used the mean absolute percentage error (MAPE), which is the ratio of MAE to the real measurements, to reflect the relative uncertainty of prediction. MAE and MAPE were calculated as follows:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |Y_i - \hat{Y}_i|, \tag{5.4}$$

$$\mathrm{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n} \frac{|Y_i - \hat{Y}_i|}{Y_i}, \tag{5.5}$$

where $i \in N$ is the index of samples, N represents the total number of samples, $\hat{Y}_i$ is the network's estimated value, and Y_i is the measured PFTs value.

We used the Pearson correlation coefficient (R) to indicate the linear correlation. The coefficient R can be interpreted as negligible ($R < 0.1$), weak (0.1 to 0.39), moderate (0.4 to 0.69), strong (0.7 to 0.89) or very strong (over 0.9) [117]. We also used the intra-class correlation coefficient (ICC) as a measure of reliability, which represents not only the linear correlation, but also the absolute agreement. ICC was calculated by Pingouin 0.4.0 [100] based on a single-rating, absolute-agreement, 2-way mixed-effects model [24]. ICC values can be explained as poor (below 0.5), moderate (between 0.5 and 0.75), good (between 0.75 and 0.9) and excellent (above 0.9) reliability [98]. Bland-Altman plots were used to analyze the mean differences (bias) and limits of agreement. These statistical analyses were performed by an in-house python 3.8 script with corresponding libraries. All the following experiment results are based on four-fold cross-validation unless otherwise stated. The multi-variable step-wise linear regression provide the extra regression performance on the hold-out testing dataset.

5.4 Experiments and results

We conducted three sets of experiments to explore: **E1**. training strategies and hyper-parameters for PNN-Vessel, **E2**. training strategies and hyper-parameters for GNN-Vessel and **E3**. strategies to combine different networks.

5.4.1 Experimental settings

Our neural networks were implemented using PyTorch 1.11.0 (`https://pytorch.org`). Mixed-precision [143] was applied during training to minimize the occupation of GPU memory and accelerate training. The loss function was the mean squared error (MSE):

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2, \tag{5.6}$$

where $i \in N$ is the index of samples, Y_i is the vector of measured PFTs, $\hat{Y}_i$ is the vector of predicted PFTs. The Adam optimizer was used with 100 epochs. Multiprocessing was used to accelerate on-the-fly data augmentation. The workstation for training and validation was equipped with an Intel(R) Xeon(R) CPU Gold 2.6 GHz with 90 GB memory and a NVIDIA GPU GeForce RTX 2080TI with 11 GB memory. The trained networks and source code for network development are publicly available at `https://github.com/Jingnan-Jia/PFT_regression` for the convenience of reproducing our method, applying our model to other datasets or using our model as a pre-trained model for other tasks.

5.4.2 E1: Training strategies and hyper-parameters for PNN-Vessel

Because this is the first work on point cloud based network for PFT estimation, it is unclear what are the optimal training strategy and data augmentation techniques. As we mentioned in Section 5.3.3, instead of inventing a new network architecture, we adopted PointNet++ as the backbone of PNN-Vessel and explored the best combination of different hyper-parameters and data augmentation techniques. Inspired by PointNeXt, the explored techniques include (see Table 5.1):

- **Random scaling (±10%).** A method for data augmentation, which randomly scales the coordinates of the **whole** point cloud by a factor in ±10%.

- **Random shifting (±3 mm).** A method for data augmentation, which randomly shifts the coordinates of the **whole** point cloud within 3 mm.

- **L2 Loss regularization (0.01) [144].** It stabilizes training by reducing the sum of the squares of the trainable parameters. The equation of the loss function after the introduction of L2 Loss regularization with weight of 0.01 is

$$L = MSE + 0.01\textstyle\sum_{i=1}^{N} w_i^2, \tag{5.7}$$

where MSE is the original loss function of PNN-Vessel, N is the number of trainable parameters, and w_i is the weight of i^{th} parameter, 0.01 is the weight of L2 loss regularization.

- **Coordinate center normalization (CCNorm).** A method for data augmentation, which makes the mean of the coordinates of the whole point be zero. The equation of CCNorm is
$$Y_{norm} = Y - \overline{Y} \tag{5.8}$$
where Y represents the original coordinates, and $\overline{Y}$ is the mean of coordinates.
- **Tuning the number of input points.** As we mentioned at the end of Section 5.3.2, the number of points for different samples are different. To align input data, we randomly selected a fixed number (N_i in Figure 5.3) of points for each patient to feed PNN-Vessel for each training iteration. For points less than than N_i, some points of the patient were repeatedly sampled. Higher number of input points means more detailed vessel information.
- **Tuning the radius of neighboring query balls [129].** As shown in Figure 5.3, increasing R_{ball} could increase the receptive field for each set abstraction layer. The principle of tuning R_{ball} for PNNs is just like adjusting the kernel size for CNNs.
- **Cosine learning rate (LR) decay [145].** LR is decayed with a cosine annealing for each batch.
- **AdamW optimizer [146].** It was reported to perform better optimization than Adam Optimizer [146].

We did not explore the influence of jittering or label smoothing, which were used in PointNeXt [134], because jittering would destroy the continuity of blood vessels and label smoothing is for classification tasks instead of our PFTs regression tasks.

Table 5.1 shows the influence of the different parameters. The techniques improving the PNN-Vessel performance included using CCNorm, increasing the number of input points from 7 K to 56 K, increasing R_{ball} from 10 mm to 40 mm, and replacing optimizer Adam with a new optimizer AdamW. The techniques that do not improve PNN-Vessel performance included random scaling (±10%), random shifting (±3 mm), L2 Loss regularization (0.01), cosine LR decay. Figure 5.6 shows the scatter plot and Bland-Altman plot of the final optimal network. The output of PNN-Vessel shows moderate correlation with DLCO and FEV_1 (R=0.57 and 0.65) and strong correlation with FVC and TLC (R=0.71 and 0.74). The mean differences for all PFTs (0.03, 0.07, 0.09 and 0.01) are close to 0, indicating that there is no systematic bias between the estimated and measured PFTs.

Table 5.1: Hyper-parameter optimization results for PNN-Vessel based on four-fold cross-validation.

Techniques	ICC				ΔICC[a]	Decision[b]
	DLCO	FEV_1	FVC	TLC		
Baseline[c]	0.38	0.43	0.47	0.45		✓
+ Random scaling (±10%)	0.27	0.30	0.37	0.34	-0.11	✗
+ Random shifting ($\pm 3mm$)	0.38	0.45	0.44	0.45	0	✗
+ L2 Loss regularization (0.01)	0.33	0.46	0.45	0.42	-0.02	✗
+ CCNorm	0.40	0.48	0.54	0.56	+0.06	✓
Input points 56 K	0.46	0.59	0.64	0.65	+0.09	✓
R_{ball} 40 mm	0.52	0.60	0.65	0.69	+0.03	✓
LR decay Cosine	0.53	0.58	0.63	0.66	-0.02	✗
Optimizer AdamW	**0.54**	**0.61**	**0.68**	**0.73**	+0.03	✓

[a] ΔICC: Average change of ICC, comparing to the previous best performance.

[b] Decision: Include the technique (✓) in the following experiments if Δ ICC > 0; exclude the technique (✗) if Δ ICC ≤ 0.

[c] Baseline: batch size is 20; Number of input points is 7 K; R_{ball} is 10; Optimizer is Adam.

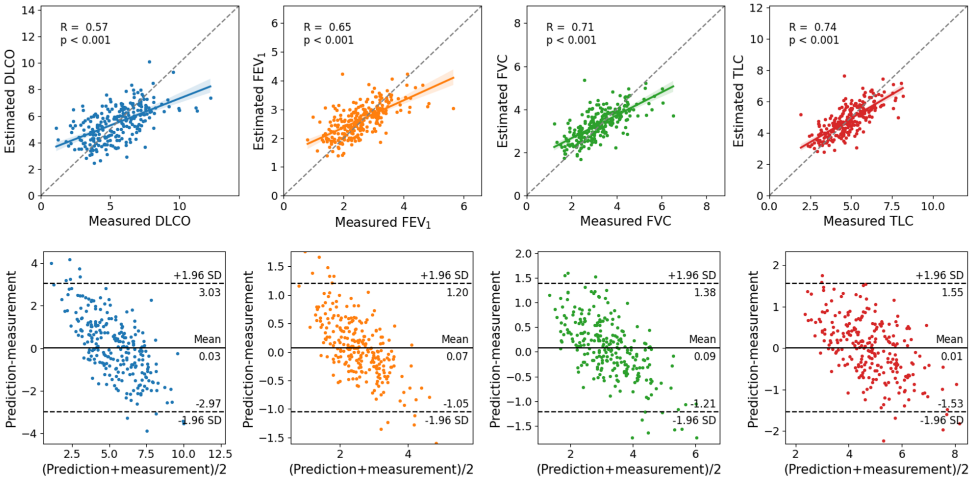


Figure 5.6: PFTs estimation results of PNN-Vessel based on the optimized setting. Upper: scatter plots with the identify line (dotted line), regression line (solid line) and the 95% confidence intervals (shaded areas). Lower: corresponding Bland-Altman plots with the mean difference and the limits of agreement (mean ś 1.96 × SD, where SD is the standard deviation of the differences).

Table 5.2: Influence of number of points in PNN-Vessel based on four-fold cross-validation.

Experiment	Network	Input points (K)	Training time (h)	Params. (M)	GPU memory (GB)	ICC			
						DLCO	FEV_1	FVC	TLC
i	CNN-Vessel	≈ 7*	4.4	30.2	10.2	0.51	0.52	0.59	0.62
ii	PNN-Vessel	7	1.8	2.1	2.6	0.25	0.37	0.49	0.58
iii	PNN-Vessel	14	2.0	2.1	3.4	0.31	0.48	0.58	0.61
iv	PNN-Vessel	28	2.7	2.1	5.1	0.48	0.51	0.60	0.66
v	PNN-Vessel	56	4	2.1	8.8	**0.54**	**0.61**	**0.68**	**0.73**

* Average number of voxels for each down-sampled vessel image

When we compare the PFT estimation performance between CNN-Vessel (Table 5.2-i) and PNN-Vessel (Table 5.2-v), the ICC values of four PFTs were improved from 0.51, 0.52, 0.59 and 0.62 to 0.54, 0.61, 0.68 and 0.73, respectively. Nevertheless, because they have different network structures and different numbers of input points, it is not clear whether such an improvement is due to different network design or different input resolutions. Therefore, we conducted a series of experiments with the number of input vessel points gradually increasing from 7K to 56 K (higher number of input points means higher resolution and more details of vessels). The results are shown in Table 5.2-ii, iii, and iv (higher numbers than 56 K were not applied because of GPU memory limitation). When we gradually increase the number of input points, the PFT estimation performance is gradually increasing as well. When we compare Table 5.2-i and ii, we found that given the similar number of input vessel points, CNN-Vessel performed better than PNN-Vessel. However, when the number of points was increased gradually, the accuracy of PNN-Vessel grew steadily and finally outperformed CNN-Vessel.

5.4.3 E2: Training strategies and hyper-parameters for GNN-Vessel

Since training GNN is much faster than PNN-Vessel, we used Optuna (`https://optuna.org`) to automatically search the optimal hyper-parameters for GNN-Vessel. The search algorithm was set to Tree-structured Parzen Estimator (TPE) [147], which was reported to outperform both manual and random search algorithms. The hyper-parameters that may influence the performance of GNN-Vessel include learning rate, batch size, number of convolution layer, convolution kernel and normalization method. Possible data augmentation techniques include random scaling, random shifting and random sampling. Random scaling or shifting has already reported not to improve the PFT estimation performance (see Table 5.1). Random sampling (e.g. sampling 56 K nodes from the whole graph) would take substantially longer time, because each sampling of points require the re-indexing of remaining edges. Therefore, we decided not to introduce data augmentation techniques for GNN-Vessel. Therefore, the search space for possible hyper-parameters are shown in Table 5.3. The hyper-parameter

Table 5.3: Hyper-parameter search space setting for GNN-Vessel. **Bold** values were finally selected after hyper-parameter search.

Hyper-parameters	Search space (candidate values)
Learning rate	**1e-3**, 1e-4
Batch size	8, 16, **32**, 64
Hidden channels	32, 64, **128**, 256
Convolution layers	1, **2**, 3, 4
Convolution kernel	ChebConv [140], GINConv [139], GCNConv [137], **GATConv** [142], SGConv [136], GraphConv [138]
Normalization	BatchNorm [149], **InstanceNorm** [148], LayerNorm [150], GraphNorm [151], DiffGroupNorm [135]

Table 5.4: Hyper-parameter optimization process for GNN-Vessel based on four-fold cross-validation.

Techniques	ICC				ΔICC[b]
	DLCO	FEV_1	FVC	TLC	
Baseline[a]	0.43	0.48	0.50	0.53	
Batch size 32	0.44	0.47	0.50	0.56	+0.01
Hidden channels 128	0.48	0.51	0.51	0.59	+0.03
Convolution layers 2	0.49	0.49	0.52	0.61	+0.01
Normalization InstanceNorm	0.48	0.51	0.54	0.62	+0.01
Convolution kernel GATConv	0.52	0.52	0.57	0.64	+0.03
Learning rate 1e-3	**0.52**	**0.55**	**0.60**	**0.66**	+0.02

[a] Baseline hyper-parameters: learning rate of 1e-4, batch size of 8, hidden channels of 32, 1 GCNConv layer, and Batch normalization.

[b] ΔICC: Average change of ICC, comparing to the previous best performance.

search stopped after 100 search iterations. We selected some intermediate results to show the influence of different hyper-parameters in Table 5.4. The best PFT estimation performance was achieved with a learning rate of 1e-3, batch size of 32, 128 hidden channels, 2 convolution layers with GATConv kernel [142] and instance normalization [148]. Figure 5.7 shows the scatter plot and Bland-Altman plot of the final optimal GNN-Vessel. The output of GNN-Vessel shows moderate correlation with DLCO, FEV_1, FVC and TLC (R=0.54, 0.58, 0.62 and 0.68).

5.4.4 E3: Combination of different networks

The comparison of different networks proposed in this work (PNN-Vessel and GNN-Vessel) and existing methods (CNN-CT and CNN-Vessel) are shown in Table 5.5. CNN-CT and CNN-Vessel used the same training time (3.7 hours) and have the same trainable parameters (30.1 M). This is because they have the same network design

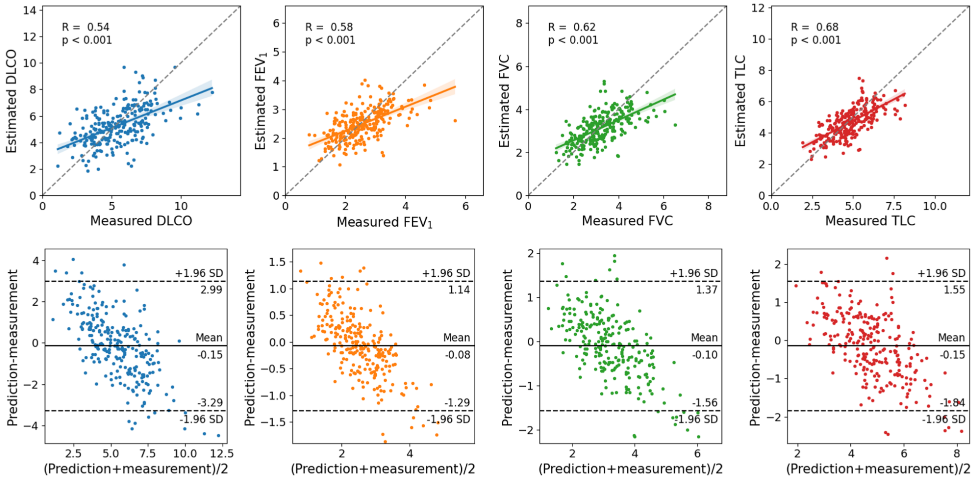


Figure 5.7: PFTs estimation results of GNN-Vessel based on the optimized setting. Upper: scatter plots with the identify line (dotted line), regression line (solid line) and the 95% confidence intervals (shaded areas). Lower: corresponding Bland-Altman plots with the mean difference and the limits of agreement (mean ś 1.96 × SD, where SD is the standard deviation of the differences).

Table 5.5: PFT estimation performance comparison based on four-fold cross-validation between different networks

Network	Training time (h)	Params. (M)	ICC				R				MAE			
			DLCO	FEV$_1$	FVC	TLC	DLCO	FEV$_1$	FVC	TLC	DLCO	FEV$_1$	FVC	TLC
CNN-CT	3.7	30.1	0.71	0.76	0.80	0.81	0.71	0.77	0.83	0.82	1.04	0.41	0.49	0.63
CNN-Vessel	3.7	30.1	0.51	0.52	0.59	0.62	0.57	0.55	0.64	0.68	1.20	0.52	0.62	0.76
PNN-Vessel	1.1	2.1	0.54	0.61	0.68	0.73	0.57	0.65	0.71	0.74	1.15	0.45	0.54	0.67
GNN-Vessel	0.25	0.2	0.52	0.55	0.60	0.66	0.54	0.58	0.62	0.68	1.24	0.49	0.59	0.72

and input sizes. Compared to CNN-Vessel, PNN-Vessel used about 30% training time (1.1 hours) and 7% parameters (2.1 M). GNN-Vessel used 7% training time (only 0.25 hours) and 0.7% parameters (0.2 M). Notably, although GNN-Vessel receive additional edge information, its performance is worse than PNN-Vessel. Nevertheless, both PNN-Vessel and GNN-Vessel outperformed CNN-Vessel, by 14% and 4%, respectively, when averaged across the ICC scores of four PFTs metrics.

Table 5.6 shows the results of the combined network with different feature extractors. It shows that leaving out GNN-Vessel could achieve better performance than leaving out PNN-Vessel, which indicates the PNN-Vessel contributes more than GNN-Vessel. Although none of the combined networks outperformed the individual CNN-CT in Table 5.5 on the average ICC or R values. Most of the MAE values of the combined networks were lower than the counterparts in the individual CNN-CT in Table 5.5.

Table 5.6: PFT estimation performance of combined network with different feature extractors based on four-fold cross-validation

Feature extractor			ICC				R				MAE			
CNN-CT	PNN-Vessel	GNN-Vessel	DLCO	FEV_1	FVC	TLC	DLCO	FEV_1	FVC	TLC	DLCO	FEV_1	FVC	TLC
✓	✓		0.71	0.76	0.79	0.77	0.73	0.78	0.80	0.79	0.99	0.35	0.42	0.58
✓		✓	0.69	0.72	0.76	0.76	0.70	0.77	0.81	0.81	1.02	0.35	0.43	0.53
✓	✓	✓	0.67	0.72	0.75	0.77	0.71	0.78	0.81	0.82	1.01	0.36	0.43	0.52

Table 5.7: Multivariable stepwise linear regressioin analysis based on four-fold cross-validation.

DLCO			
Parameter	R	ICC	MAE
CNN-CT	0.71	0.71	1.04
CNN-CT, PNN-Vessel	0.73	0.72	0.98
CNN-CT, PNN-Vessel, GNN-Vessel	0.75	0.73	0.97

FEV_1			
Parameter	R	ICC	MAE
CNN-CT	0.77	0.76	0.41
CNN-CT, PNN-Vessel	0.79	0.77	0.33

FVC			
Parameter	R	ICC	MAE
CNN-CT	0.83	0.80	0.49
CNN-CT, PNN-Vessel	0.84	0.83	0.38

TLC			
Parameter	R	ICC	MAE
CNN-CT	0.82	0.81	0.63
CNN-CT, PNN-Vessel	0.84	0.83	0.49
CNN-CT, PNN-Vessel, GNN-Vessel	0.85	0.84	0.48

Multivariable step-wise linear regression was performed via four-fold cross-validation dataset with DLCO as the dependent variable and the estimated DLCO from CNN-CT, CNN-Vessel, PNN-Vessel and GNN-Vessel as independent variables. We performed similar analyses for FEV_1, FVC and TLC (Table 5.7). The results in Table 5.7 showed that CNN-CT always performed the best among the four networks. The inclusion of CNN-Vessel could not improve the PFT estimation accuracy (this is why it was not in Table 5.7). The inclusion of PNN-Vessel significantly improved the performance for all measures, while the additional inclusion of GNN-Vessel further improved the performance for DLCO and TLC. After we obtained regression equations from validation dataset, we applied them to the separate testing dataset (see Table 5.8). The testing performance shows the similar tendency with validation performance.

5.5 Discussion

In this chapter, we converted high-resolution binary pulmonary vessels to point cloud and graph data, then proposed PNN-Vessel and GNN-Vessel to estimate PFTs based on these two datasets, respectively. After applying proper training strategies and hyper-parameters, both of them showed considerable improvement compared to the existing CNN-Vessel model, which was trained on low-resolution grids containing vessel masks.

Random scaling (±10%) decreased the accuracy of the PFT estimation significantly

Table 5.8: PFT estimation performance on testing dataset by applying the regression equations obtained from four-fold cross-validation dataset.

DLCO

Parameter	R	ICC	MAE
CNN-CT	0.75	0.73	1.01
CNN-CT, PNN-Vessel	0.78	0.76	0.97
CNN-CT, PNN-Vessel, GNN-Vessel	0.76	0.75	0.99

FEV_1

Parameter	R	ICC	MAE
CNN-CT	0.74	0.73	0.51
CNN-CT, PNN-Vessel	0.76	0.74	0.47

FVC

Parameter	R	ICC	MAE
CNN-CT	0.82	0.80	0.48
CNN-CT, PNN-Vessel	0.83	0.84	0.37

TLC

Parameter	R	ICC	MAE
CNN-CT	0.83	0.80	0.60
CNN-CT, PNN-Vessel	0.83	0.79	0.63
CNN-CT, PNN-Vessel, GNN-Vessel	0.85	0.84	0.49

(see Section 5.4.2). This finding is contrary to the finding in PointNeXt [134]. This is because the tasks in [134] are independent on the scale of the point cloud data, while PFT estimation is dependent on the scale of lung vessels. Larger vessel trees normally represent larger lung sizes, which is directly related to higher PFT values [152].

When we compare Table 5.2-i and ii, CNN-Vessel achieved better performance than PNN-Vessel given a similar number of input points (7K). However, when we gradually increased the number of input points for each point cloud data, the PFT estimation performance of PNN-Vessel gradually increased and finally outperformed CNN-Vessel (Table 5.2-i) using the same GPU (11 GB in RTX2080Ti). It verified our assumption that more detailed vessel information leads to better PFT estimation performance. Contrary to CNN-Vessel, PNN-Vessel, which uses only 2.1 MB trainable parameters, is able to receive high-resolution vessels. If there are no implementation limitations, increasing the number of input points further may achieve better performance.

Although graph data include additional edge information compared to point cloud data, the GNN-Vessel performed worse than PNN-Vessel. One possible reason is their network structures differ, which leads to different learning capacities. Another reason may be because the training set of point cloud data for PNN-Vessel was augmented by random shuffling and sampling, while GNN-Vessel applied no augmentation. It means that PNN-Vessel was trained by numerous different samples, while GNN-Vessel was trained by limited number of different samples. The third reason may be because the pulmonary vessels used in this work were segmented by an automated graph-cut based method [58]. Thus, any imperfections in the segmentation would be amplified in graphs, due to imperfect connectivity. Since point clouds do not contain this explicit connectivity, PNN-Vessel may be able to deal with these imperfections.

Under the current network combination setting, the combined networks did not show significantly better performance than the single CNN-CT on the average ICC or

Table 5.9: Comparison between our method (combination of three networks based on multivariable linear regressioin results) and previous automatic methods for the estimation on PFTs. Because previous methods did not estimate DLCO and TLC, the corresponding results are not included.

Method	**Study population**	**Backbone**	**#Subjects**	**MAE**				**R**			
				DLCO	**FEV$_1$**	**FVC**	**TLC**	**DLCO**	**FEV$_1$**	**FVC**	**TLC**
[21]	Lung cancer	CNN (ResNet) [118]	546	-	0.33	0.37	-	-	0.73	0.82	-
[107]	Lung cancer risk	CNN (I3D) [119]	16148	-	0.22	0.22	-	-	-	-	-
CNN-CT [124]	SSc	CNN (X3D) [113]	316	1.01	0.51	0.48	0.60	0.75	0.74	0.82	0.83
Our method	SSc	CNN, PNN, GNN	316	0.99	0.47	0.37	0.49	0.76	0.76	0.83	0.85

R values. This may be because the training of different networks requires different learning rates, batch sizes, etc. For instance, we set batch size to 1 and learning rate to 1e-4 for the combined networks, while the optimal batch size for individual CNN-CT, PNN-Vessel and GNN-Vessel is 1, 20 and 32, respectively (the training of individual PNN-Vessel and GNN-Vessel could not be converged with batch size of 1) and the optimal learning rate is 1e-4, 1e-3, 1e-3, respectively (CNN-CT with learning rate of 1e-3 would encounter exploding gradients during training). Such conflicts of hyper-parameters may negatively affect the training of combined networks.

From the multi-variable step-wise regression analysis, we found that by including PNN-Vessel and GNN-Vessel, the estimation of all PFTs could be significantly improved to different extents. It verifies our assumption that more detailed vessel information could provide more explanation of PFTs. It is worth noting that the inclusion of CNN-Vessel did not improve the estimation performance of PFTs. This may be because that the information of 3D grid vessels are already included in the 3D CT images. The reason why simple multi-variable linear regression perform better than complex combined network may be that the combined network has too many features to learn. Although the features before FC layers are extracted by different networks, these features are all extracted from vessels or CT images containing vessels. Therefore, there may be a great number of repeated or similar features which increase the learning difficulty of combined networks.

PFT estimation is a topic that few researchers have explored. To the best of our knowledge, there are only three published works [21, 107, 124] for PFT estimation from chest CT (see Table 5.9). Two of them [21, 107] are for patients with (the risk of) lung cancer and only FEV$_1$ and FVC were estimated in the two works. The other one [124] is for patients with SSc, and it estimated four PFTs using a CNN (X3D) network. These three works used different kinds of CNNs (ResNet [7], I3D [8] and X3D [9] respectively) as the backbones. None of them proposed new networks as well. Compared to them, we are the first to estimate PFTs using PNN and GNN instead of CNN. Our method is also the first work which combined CNN, PNN and GNN together and achieved the better performance than CNN [124] alone. The comparison between our method and CNN-CT [124] is fair because we used the same dataset. However, the results of [21, 107] are for reference only, since they are based on different datasets sizes, networks and diseases. Nevertheless, compared with the previous works which, our work estimates a more comprehensive set of PFT measurements for SSc patients with higher accuracy, rendering it more clinically relevant for SSC patients, that is likely of additional clinical value.

Our work may also provide a general solution to overcome the high GPU memory requirement in deep learning. By segmenting the objects of interest and then converting the results to point cloud or graph data type, higher prediction accuracy may be achieved using PNN, GNN or their combination.

There are some limitations to our research. The first limitation is that, in order to use automatic hyper-parameter search tool (Optuna), we did not apply data augmentation for GNN-Vessel. This may limit its performance. In the future, we will explore the efficient data augmentation methods for GNN-Vessel. Another limitation is that this chapter explored the influence of three modalities (CT, point cloud and graph). If more image modalities are available in the future, we can explore the potential scalability of our methods on other modalities.

5.6 Conclusions

In this chapter, we skeletonized pulmonary vessels and proposed PNN-Vessel and GNN-Vessel to estimate PFTs from point cloud and graph-based dataset, respectively. Compared with CNN-Vessel, which was developed based on 3D grid images, PNN-Vessel achieved significantly better performance with shorter training time and GNN-Vessel achieved slightly better performance with substantially shorter training time. Our multiple variable step-wise regression analysis verified that more detailed vessel information could provide more explanation of PFT estimation. We conclude that the detailed geometry of the vessels aids in the estimation of PFTs.

5.7 Declaration of interests

The authors declare no conflicts of interest.

5.8 Data availability

All code used to develop and verify the deep neural networks in this study has been published at `https://github.com/Jingnan-Jia/PFT_regression`. All data and materials used in the analysis can be available upon request for the purposes of reproducing or extending the analysis via the corresponding author, in accordance with local and institutional guidance and legal requirements.

5.9 Acknowledgments

This work was performed using the computational resources from the Academic Leiden Interdisciplinary Cluster Environment (ALICE) provided by the Leiden University. This work is supported by the China Scholarship Council No.202007720110.

6

Summary, discussion and future work

HRCT is an important modality to non-invasively diagnose pulmonary diseases and assess treatment effects. In this thesis, we developed automatic methods to quantify SSc disease, based on HRCT. In this chapter, we summarize the previous chapters and discuss interesting directions of future research.

6.1 Summary

In this thesis, we first provided a general introduction in Chapter 1 about pulmonary anatomy, SSc, PFTs, chest CT and deep learning on chest CT. A lung lobe segmentation method was proposed in Chapter 2, as accurately extracting lungs and lobes is an essential step for later SSc disease analysis. An explainable fully automated SSc-ILD scoring framework was proposed in Chapter 3. This framework could automatically select five levels and estimate the ratio of SSc-ILD to lung area for each level in the order of several seconds. In Chapter 4, an automatic PFT estimation network was developed which could help to understand the relation between lung function and structure and to estimate the PFTs from CT scans for patients with PFTs contraindications. Because of GPU memory limitation, the CT scans used in Chapter 4 were down-sampled. Therefore, Chapter 5 achieved higher PFT regression performance with less training time by converting vessel centerlines from HRCT to point cloud and graph data.

Chapter 2 outlined the development of a deep-learning network for lobe segmentation, emphasizing deep multi-task learning to utilize labels from various structures. These labels are often spread across different datasets. To address this, we introduced a multi-task semi-supervised model capable of harnessing information from unannotated datasets and those annotated with different structures. Additionally, we incorporated multi-scale input and an adaptive learning rate. A focused alternating training strategy was proposed to balance the tasks. We tested the trained model on an independent testing dataset, finding that the combination of multi-task learning, semi-supervised learning, multi-scale input, adaptive learning rate, and the focused alternating training strategy significantly enhanced model performance compared to single-task models. Furthermore, our approach proved effective with various network architectures.

Chapter 3 introduced a deep learning framework for automating SSc-ILD scoring.

This framework consists of two neural networks: the first identifies the craniocaudal positions of the five scoring levels, and the second estimates the ratios of three patterns (total extent of disease, ground glass, and reticulation) relative to the total lung area at each level. To address score imbalance in the second network, we augmented the training dataset with synthetic data. A heat map method was employed to highlight potential interstitial lung disease regions, with its explainability evaluated by two experts and a quantitative method using the heat map to generate scores. The results demonstrated that the automated SSc-ILD scoring framework is competitive with human experts and provides clear explanations through heat maps.

In **Chapter 4**, we introduced a deep-learning framework to automatically estimate pulmonary function tests (PFTs) from chest CT scans of SSc patients. The model used segmented lungs and vessels to mask the CT images, training the network with different inputs: entire CT scans, lungs-only, and vessels-only. This allowed us to assess the influence of different regions on PFT estimation. The network trained on entire CT scans with transfer learning achieved the highest intra-class correlation (ICC) for DLCO, FEV_1, FVC, and TLC. Performance decreased when training on lungs-only and vessels-only data. Regression attention maps indicated that regions near large vessels were often highlighted more, including some areas outside the lungs. These findings suggest that regions beyond the lungs and large vessels contribute to PFT estimation. Moreover, incorporating manually designed biomarkers improved correlation (R) for all four PFTs, indicating their continued relevance in linking lung function and structure.

In **Chapter 5**, we extended the work of **Chapter 4** to improve PFT estimation performance. We developed a point cloud neural network (PNN-Vessel) and graph neural network (GNN-Vessel), based on the point cloud data and graph data of centerlines, respectively. The results show that both PNN-Vessel and GNN-Vessel could outperform CNN-Vessel (CNN network developed on 3D grid vessel masks). It verified that more detailed vessel information could provide more explanation of PFT estimation. By combining CNN-CT (CNN network developed on 3D CT), PNN-Vessel and GNN-Vessel, we could achieve the best PFT estimation performance.

6.2 Discussion and future work

In this thesis, we developed an indirect path and a direct path to automatically analyze CT of SSc patients.

In the indirect path, the whole diagnosis is split into two steps: segmentation and quantification. At each step, a network was developed separately. The advantage of this path is that once an error occurs, it is easier to locate in which step the error occurred. This improves the acceptance and integration into clinical practice. A potential disadvantage is that the final prediction performance is dependent on the segmentation performance. The development of high accuracy 3D segmentation (Chapter 2) networks is challenging because it normally requires time-consuming and laborious pixel-wise annotation from experienced human experts. Therefore, it is an interesting direction in the future to research how to develop high-performance 3D segmentation networks based on small datasets or weak labels.

In the direct path, we automatically scored ILD or predicted PFTs from the CT image directly. The advantage of direct is that unlike the segmentation networks in the indirect path, which require time-consuming voxel-wise manual annotation, the networks in the direct path are all regression tasks, which only need image-wise labels for training. A potential disadvantage is that the whole prediction process is a black box, which makes is difficult to explain the prediction. The current explanation techniques, such as heat maps, are still limited by their low resolution. This lack of transparency raises concerns in critical healthcare applications where understanding the reasoning behind a model’s predictions is crucial. In the future, by enhancing the interpretability, healthcare professionals may trust and validate the outputs of deep learning models, improving their acceptance and integration into clinical practice. In addition, an interesting direction for SSc-ILD scoring (Chapter 3) is to research the influence of the proportion of synthetic training images to score estimation accuracy. We only used half of the training images from synthetic images in Chapter 3. If the ratio of the synthesized images is as high as possible while the score estimation accuracy remains unchanged, the annotation cost would be significantly reduced.

Our thesis is valuable to a wide range of readers. For **medical image algorithm developers**, our thesis provides detailed methods to design and develop deep learning networks for lobe segmentation, ILD scoring and PFT estimation. The source code and models of all networks in this thesis are publicly available for the convenience to reproduce them. We also developed a user-friendly Python package (Chapter 8) to calculate complete metrics for medical image segmentation, which benefit the whole medical image algorithm community. For **radiologists**, they can either use our developed segmentation networks as a preprocessing for further manual analysis of CT scans, or use the whole fully automatic networks directly, which can help relieve them from the arduous and time-consuming task of analyzing CT images. In addition, we verified that PFT is strongly related to the structure and shape of vessels, because the networks trained by binarized vessels and gray-scale vessels achieved similar PFT estimation accuracy. The regions close to the large vessels seem to influence PFTs more than other regions, and occasionally regions outside the lungs, e.g. muscles, also influence PFTs. However, limited by the low-resolution heat maps, we could not understand the more detailed relationship between lung function and structure. For future work, it would be of great clinical value if more detailed and human understandable contributors could be extracted. For **SSc patients**, our automatic PFT estimation network could benefit those who are not applicable to spirometry due to a risk of disease transmission or other contraindications.

Our developed automatic methods for lobe segmentation, SSc scoring and PFT estimation could achieve considerable accuracy within seconds. These deep learning based automatic tools are also helpful for clinical practice. For lobe segmentation, an accurate segmentation in several seconds might accelerate treatment planning. For automatic disease severity scoring, the heat maps generated by our network may help clinicians discover more fibrosic areas. For PFT estimation from CT scans, we provide an alternative for spirometry for patients with contraindications. Currently, there are no standards for the level of accuracy at which they can be applied in clinical practice. Therefore, these developed networks are for research use only. Nerveless, the performance of these networks suggest meaningful contributions to clinical practice. We also hope that this thesis will promote the early establishment of industry standards in related fields, thereby accelerating the transformation of these studies into clinical practice.

The methods and techniques developed in this thesis could also be extended to other topics. For instance, inspired by the observation that lobe segmentation could benefit from related organs (Chapter 2), we could also develop networks for aorta or bronchi segmentation which may also benefit from pulmonary vessels. After that, we can further research whether and how the structure of aorta and bronchi affects lung function. Similarly, the point neural network and graph neural network could also be applied to the centerlines of aorta and bronchi to improve PFT estimation.

6.3 General conclusions

In conclusion, this thesis proposes automatic methods for quantifying SSc disease on CT images from different perspectives: a network for automated lung lobe segmentation, a cascaded network for automated SSc-ILD scoring, and a series of networks for PFT estimation. Our multi-task semi-supervised learning could be generalized to different network backbones. The cascaded network for SSc-ILD scoring performed competitively with human experts and provides high-quality explanations using heat maps. PFT estimation network could be used to study the relation between lung structure and function and may provide an alternatives for SSc patients with contraindications. More detailed vessel information from HRCT provide more explanation on the PFT estimation of SSc patients. All the networks developed in this thesis have a runtime in the order of seconds, substantially improving over conventional methods. To summarize, deep learning has the powerful potential and a variety of applications on the automated analysis of chest CT in SSc.

7

Samenvatting, discussie en toekomstig werk

Hoge resolutie CT (HRCT) is een belangrijke modaliteit voor de niet-invasieve diagnose van longziekten en het beoordelen van behandelingseffecten. In dit proefschrift worden automatische methoden gepresenteerd, die zijn ontwikkeld om Systemische Sclerodermie (SSc) te kwantificeren op basis van HRCT. In dit hoofdstuk vatten we de voorgaande hoofdstukken samen en bespreken we interessante richtingen voor toekomstig onderzoek [1].

7.1 Samenvatting

In dit proefschrift geeft Hoofdstuk 1 een algemene inleiding over de longanatomie, SSc, longfunctiemetingen (PFT's), thorax-CT en deep learning toegepast op thorax-CT. Een methode voor segmentatie van de longkwabben werd voorgesteld in Hoofdstuk 2, omdat het nauwkeurig extraheren van longen en kwabben een essentiële stap is voor latere analyse van SSc. In Hoofdstuk 3 werd een verklaarbaar raamwerk voorgesteld voor het volledig-geautomatiseerd scoren van SSc-ILD. Dit raamwerk kan in enkele seconden vijf anatomische niveaus automatisch selecteren en voor elk niveau de verhouding van SSc-ILD tot het totale longoppervlak schatten. In Hoofdstuk 4 werd een automatisch netwerk voor het schatten van PFT's ontwikkeld, dat kan helpen bij het begrijpen van de relatie tussen longfunctie en longstructuur en bij het schatten van PFT's uit CT-scans voor patiënten met contra-indicaties voor PFT's. Hoofdstuk 5 breidde het werk van Hoofdstuk 4 uit door middellijnen van bloedvaten om te zetten naar puntenwolken en grafen.

Hoofdstuk 2 beschreef de ontwikkeling van een deep-learning netwerk voor lobsegmentatie, met de nadruk op het diep multi-task learning door gebruik te maken van labels van verschillende structuren. Deze labels zijn vaak verspreid over verschillende datasets. Om dit aan te pakken, introduceerden we een multi-task semi-supervised model dat in staat is om informatie te gebruiken van niet-geannoteerde datasets en datasets geannoteerd met verschillende structuren. Daarnaast integreerden we multi-scale invoer en een adaptieve leersnelheid. Een gefocuste alternerende trainingsstrategie werd voorgesteld om de taken in balans te houden. We testten het

---

[1] This Chapter was automatically translated from **Chapter 6** with the help of ChatGPT, followed by further manual adjustments by native Dutch speakers.

getrainde model op een onafhankelijke testdataset en ontdekten dat de combinatie van multi-task learning, semi-supervised learning , multi-scale invoer, adaptieve leersnelheid en de gefocuste alternerende trainingsstrategie de modelprestaties significant verbeterde in vergelijking met single-task modellen. Bovendien bleek onze benadering effectief met verschillende netwerkarchitecturen.

Hoofdstuk 3 introduceerde een deep learning framework voor het automatiseren van SSc-ILD scoring. Dit framework bestaat uit twee neurale netwerken: het eerste identificeert de craniocaudale posities van de vijf score niveaus, en het tweede schat de verhoudingen van drie patronen (totale omvang van de ziekte, matglasafwijkingen en reticulaties) ten opzichte van het totale longoppervlak op elk niveau. Om de score disbalans in het tweede netwerk aan te pakken, vergrootten we de trainingsdataset met synthetische data. Een heatmap-methode werd gebruikt om gebieden met waarschijnlijke interstitiële longziekte te markeren, waarbij de verklaarbaarheid ervan werd geëvalueerd door twee experts en een kwantitatieve methode, die de heatmap gebruikte om scores te genereren. De resultaten toonden aan dat het geautomatiseerde SSc-ILD scoring framework concurrerend is met menselijke experts en duidelijke verklaringen biedt via heatmaps.

In **Hoofdstuk 4** introduceerden we een deep-learning framework om automatisch longfunctieonderzoeken (PFT's) te schatten uit CT-scans van SSc-patiënten. Het model gebruikte gesegmenteerde longen en vaten om de CT-beelden te maskeren, waarbij het netwerk werd getraind met verschillende invoerbeelden: volledige CT-scans, alleen longen en alleen vaten. Dit stelde ons in staat om de invloed van verschillende regio's op de PFT-schatting te beoordelen. Het netwerk, getraind op volledige CT-scans en met transfer-learning, bereikte de hoogste intra-klasse correlatie (ICC) voor DLCO, FEV_1, FVC en TLC. De prestaties namen af bij training op data met alleen longen en alleen vaten. De regressie attention-maps gaven aan dat regio's nabij grote vaten vaak meer werden benadrukt, inclusief enkele gebieden buiten de longen. Deze bevindingen suggereren dat regio's bijdragen aan de PFT-schatting, die buiten de longen en grote vaten liggen. Bovendien verbeterde het incorporeren van handmatig ontworpen biomarkers de correlatie (R) voor alle vier de PFT's, wat wijst op hun blijvende relevantie bij het bestuderen van de relatie tussen longfunctie en structuur.

In **Hoofdstuk 5** hebben we het werk van **Hoofdstuk 4** uitgebreid om de prestaties van PFT-schattingen te verbeteren. We hebben een puntenwolk-neuraal-netwerk (PNN-Vessel) en een graaf neuraal netwerk (GNN-Vessel) ontwikkeld, gebaseerd op, respectievelijk puntenwolken en grafen van de middellijnen. De resultaten tonen aan dat zowel PNN-Vessel als GNN-Vessel beter presteerden dan CNN-Vessel (CNN-netwerk ontwikkeld op 3D-raster bloedvatmaskers). Het bevestigde dat meer gedetailleerde informatie over bloedvaten meer uitleg kon bieden over PFT-schattingen. Door CNN-CT (CNN-netwerk ontwikkeld op 3D-CT), PNN-Vessel en GNN-Vessel te combineren, konden we de beste prestaties voor PFT-schattingen behalen.

7.2 Discussie en toekomstig werk

In dit proefschrift hebben we een indirecte en directe aanpak ontwikkeld om CT-beelden van SSc-patiënten automatisch te analyseren.

In de indirecte aanpak wordt de hele diagnose opgesplitst in twee stappen: segmentatie en kwantificatie. In elke stap werd afzonderlijk een netwerk ontwikkeld. Het voordeel van deze aanpak is dat als er een fout optreedt, het gemakkelijker is om te bepalen in welke stap de fout is opgetreden. Dit verbetert de acceptatie en integratie in de klinische praktijk. Het nadeel is dat de segmentatieprestaties van invloed zullen zijn op de uiteindelijke voorspellingsnauwkeurigheid. De ontwikkeling van zeer nauwkeurige 3D-segmentatienetwerken (Hoofdstuk 2) is een uitdaging omdat het normaal gesproken tijdrovende en arbeidsintensieve pixelgewijze annotaties door ervaren menselijke experts vereist. Daarom is het in de toekomst een interessante richting om te onderzoeken hoe hoogwaardige 3D-segmentatienetwerken kunnen worden ontwikkeld op basis van kleine datasets of zwakke labels.

In de directe aanpak hebben we ILD automatisch gescoord of PFT's voorspeld op basis van het hele beeld. Het voordeel van een directe aanpak is dat in tegenstelling tot de segmentatienetwerken in de indirecte aanpak, die tijdrovende voxel-voor-voxel handmatige annotatie vereisen, de netwerken in de directe aanpak allemaal regressietaken zijn, die alleen labels op beeld-niveau nodig hebben voor training. Het mogelijke nadeel is dat het hele voorspellingsproces een black box is, wat het moeilijk maakt om de voorspelling te verklaren. De huidige verklaringstechnieken, zoals heatmaps, worden nog steeds beperkt door hun lage resolutie. Dit gebrek aan transparantie baart zorgen bij kritische toepassingen in de gezondheidszorg, waarbij het begrijpen van de redenering achter de voorspellingen van een model van cruciaal belang is. Door de interpreteerbaarheid te verbeteren kunnen zorgprofessionals in de toekomst de uitkomsten van deep learning-modellen vertrouwen en valideren, waardoor de acceptatie en integratie ervan in de klinische praktijk wordt verbeterd. Daarnaast is een interessante richting voor SSc-ILD-scores (hoofdstuk 3) het onderzoeken van de invloed van de proportie synthetische trainingsbeelden op de nauwkeurigheid van de scoreschatting. In hoofdstuk 3 hebben we slechts de helft van de trainingsafbeeldingen uit synthetische afbeeldingen gebruikt. Als de verhouding van de gesynthetiseerde afbeeldingen zo hoog mogelijk is terwijl de nauwkeurigheid van de scoreschatting onveranderd blijft, zouden de annotatiekosten aanzienlijk kunnen worden verlaagd.

Dit proefschrift is waardevol voor een breed scala aan lezers. Voor **ontwikkelaars van medische beeldalgoritmen** biedt dit proefschrift gedetailleerde methoden om deep learning-netwerken te ontwerpen en te ontwikkelen voor kwabsegmentatie, ILD-score en PFT-schatting. De broncode en modellen van alle netwerken in dit proefschrift zijn openbaar beschikbaar gemaakt om ze makkelijk te reproduceren. We hebben ook een gebruiksvriendelijk Python-pakket (hoofdstuk 6) ontwikkeld om uitgebreide metingen voor medische beeldsegmentatie te berekenen, wat ten goede komt aan de hele gemeenschap van medische beeldalgoritmen. **Radiologen** kunnen onze ontwikkelde segmentatienetwerken gebruiken als preprocessing voor verdere handmatige analyse van CT-scans, of de volledig automatische netwerken direct gebruiken, wat hen kan helpen bij het verlichten van de zware en tijdrovende taak van het analyseren van CT-beelden. Bovendien hebben we bevestigd dat PFT sterk gerelateerd is aan de structuur en vorm van bloedvaten, omdat de netwerken die zijn getraind door gebinariseerde vaten en grijsschaal-vaten een vergelijkbare nauwkeurigheid van de PFT-schatting bereikten. De regio's dichtbij de grote vaten lijken PFT's meer te beïnvloeden dan andere regio's, en soms ook regio's buiten de longen, b.v. spieren, beïnvloeden ook PFT's. Echter, beperkt door de heatmap's lage resolutie, konden we de meer gedetailleerde relatie tussen longfunctie en structuur niet doorgronden. Voor toekomstig werk zou het van grote klinische waarde zijn als er meer gedetailleerde en voor mensen begrijpelijke bijdragen zouden kunnen worden verzameld. Voor **SSc-patiënten** kan ons automatische PFT-schattingsnetwerk voordelig zijn voor degenen die niet in aanmerking komen voor spirometrie vanwege een risico op ziekteoverdracht of andere contra-indicaties.

Onze ontwikkelde automatische methoden voor lobsegmentatie, SSc-scorebepaling en PFT-schatting kunnen binnen enkele seconden een aanzienlijke nauwkeurigheid bereiken. Deze op deep-learning gebaseerde automatische tools zijn ook nuttig voor de klinische praktijk. Voor lobsegmentatie kan een nauwkeurige segmentatie in enkele seconden de behandelingsplanning versnellen. Voor het automatische scoren van de ziekteactiviteit kunnen de heatmaps die door ons netwerk worden gegenereerd, de clinici helpen om meer fibrosegebieden te ontdekken. Voor PFT-schatting op basis van CT-scans bieden we een alternatief voor spirometrie voor patiënten met contra-indicaties. Aangezien deze deep-learning-netwerken relatief nieuwe tools zijn in hun respectieve toepassingsgebieden, zijn er echter nog geen standaarden over het nauwkeurigheidsniveau waarop ze in de klinische praktijk kunnen worden toegepast. Daarom zijn deze ontwikkelde netwerken alleen voor gebruik voor onderzoek. Desalniettemin suggereert de resultaten van deze netwerken dat ze betekenisvolle bijdragen kunnen leveren aan de klinische praktijk. We hopen ook dat dit proefschrift de vroege vaststelling van industrie-standaarden in vergelijkbare toepassingsvelden zal bevorderen, waardoor de transformatie van deze studies naar de klinische praktijk wordt versneld.

De methoden en technieken die in dit proefschrift zijn ontwikkeld, kunnen ook worden uitgebreid naar andere onderwerpen. Geïnspireerd door de observatie dat kwabsegmentatie baat zou kunnen hebben bij vergelijkbare organen (Hoofdstuk 2), zouden we bijvoorbeeld ook netwerken voor aorta- of bronchiënsegmentatie kunnen ontwikkelen die ook geholpen kunnen worden door informatie van longvaten. Daarna kunnen we verder onderzoeken of en hoe de structuur van aorta en bronchiën de longfunctie beïnvloedt. Op dezelfde manier zouden het puntneurale netwerk en het grafische neurale netwerk ook kunnen worden toegepast op de middellijnen van de aorta en de bronchiën om de PFT-schatting te verbeteren.

7.3 Algemene conclusies

Samenvattend stellen we in dit proefschrift automatische methoden voor, om SSc te kwantificeren uit CT-beelden vanuit verschillende perspectieven: een netwerk voor geautomatiseerde segmentatie van longkwabben, een cascade-netwerk voor het geautomatiseerd scoren van SSc-ILD, en een reeks netwerken voor PFT-schattingen. Ons multitask semi-supervised learning kan worden gegeneraliseerd naar verschillende basisnetwerken. Het cascade-netwerk voor SSc-ILD-score presteerde vergelijkbaar met menselijke experts en biedt duidelijke verklaringen met behulp van heatmaps. Het PFT-schattingsnetwerk kan worden gebruikt om de relatie tussen longstructuur en -functie te bestuderen en kan een alternatief bieden voor SSc-patiënten met contra-indicaties. Meer gedetailleerde informatie over bloedvaten van HRCT biedt meer uitleg over de PFT-schatting bij SSc patiënten. Alle ontwikkelde netwerken in dit proefschrift hebben een rekentijd in de orde van seconden, wat een substantiële verbetering is ten opzichte van conventionele methoden. Samenvattend, geeft deep learning potentieel veel mogelijkheden en een verscheidenheid aan toepassingen bij de automatische analyse van thorax CT in SSc.

8

Supplementary material

seg-metrics: a Python package to compute segmentation metrics

This chapter was adapted from:

Jia, Jingnan, Marius Staring, and Berend C. Stoel "seg-metrics: a Python package to compute segmentation metrics." *medRxiv*, 2024-02.

Abstract

Medical image segmentation (MIS) is an important task in medical image processing. Unfortunately, there is not a out-of-the-box python package for the evaluation metrics of MIS. Therefore, we developed `seg-metrics`, an open-source Python package for MIS model evaluation. Unlike existing packages, `seg-metrics` offers user-friendly interfaces for various overlap-based and distance-based metrics, providing a comprehensive solution. `seg-metrics` supports multiple file formats and is easily installable through the Python Package Index (PyPI). With a focus on speed and convenience, `seg-metrics` stands as a valuable tool for efficient MIS model assessment.

8.1 Background

In the last decade, the research of artificial intelligence on medical images has attracted researchers' interest. One of the most popular directions is automated medical image segmentation (MIS) using deep learning, which aims to automatically assign labels to pixels so that the pixels with the same label from a segmented object. However, in the past years a strong trend of highlighting or cherry-picking improper metrics to show particularly high scores close to 100% was revealed in scientific publishing of MIS studies [153]. In addition, even though there are some papers that evaluate image segmentation results from different perspectives, the implementation of their evaluation algorithms is inconsistent. This is due to the lack of a universal metric library in Python for standardized and reproducible evaluation. Therefore, we proposed to develop an open-source publicly available Python package `seg-metrics`, which aims to evaluate the performance of MIS models. Our package is public available at `https://pypi.org/project/seg-metrics`.

8.2 Related packages

As far as we know, untill the publication date of this package (2020), there are only two open source packages which could perform MIS metrics calculation: `SimpleITK`[154] and `Medpy` [155].

SimpleITK is an interface (including Python, c#, Java, and R) to the Insight Segmentation and Registration Toolkit (ITK) designed for biomedical image analysis. Unfortunately, `SimpleITK` does not support the evaluation of MIS directly. Each evaluation consists of several basic steps, which makes it not user-friendly. **Medpy** is a medical image processing library written in Python. It includes some functions to evaluate MIS. However, it mainly support the operations of binary segmentation results, which limits its wider application scenarios. Therefore, this work aims to develop a Python package specifically for MIS.

8.3 Our `seg-metrics` package

Our `seg-metrics` package supports calculating different evaluation metrics directly in one line of code. The metrics could be divided to overlap-based metrics and distance-based metrics. Overlap-based metrics, define the overlap between the reference annotation and the prediction of the algorithm. It is typically complemented by a distance-based metrics, which could explicitly assess how close the boundaries are between the prediction and the reference [156]. The details of the two categories are described below.

Table 8.1: Confusion matrix (adopted from `https://en.wikipedia.org/wiki/Confusion_matrix`)

Total (P+N)		Prediction	
		Positive (P)	Negative (N)
Reference	Positive (P)	TP	FN
	Negative (N)	FP	TN

8.3.1 Overlap-based metrics

A confusion matrix (see Table 8.1) could be derived when comparing a segmentation (pixel-wise classification) result and its reference. In this table, there are 4 different outcomes:

1. **True positive (TP)**: If the actual classification is positive and the predicted classification is positive, this is called a TP result because the positive sample was predicted correctly.
2. **False negative (FN)**: If the actual classification is positive and the predicted classification is negative, this is called a FN result because the positive sample is incorrectly predicted as being negative.
3. **False positive (FP)**: If the actual classification is negative and the predicted classification is positive, this is called a FP result because the negative sample is incorrectly predicted as being positive.
4. **True negative (TN)**: If the actual classification is negative and the predicted classification is negative, this is called a TN result because the negative sample is predicted correctly.

Based on these four outcomes, we can derive a great number of overlap-based metrics. Their equations are as follows.

- **Dice Coefficient (F1-Score)**

$$Dice = \frac{2 \times |A \cap B|}{|A| + |B|} = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{8.1}$$

- **Jaccard index**

$$Jaccard = \frac{|A \cap B|}{|A \cup B|} = \frac{TP}{TP + FP + FN} \tag{8.2}$$

- **Precision/Positive predictive value (PPV)**

Precision score is the number of true positive results divided by the number of all positive results

$$Precision = \frac{TP}{TP+FP} \tag{8.3}$$

- **Selectivity/Specificity/True negative rate (TNR)**

$$TNR = Specificity = \frac{TN}{TN+FP} \tag{8.4}$$

- **False negative rate (FNR)**

$$FNR = \frac{FN}{TN+FP} \tag{8.5}$$

- **Recall/Sensitivity/Hit rate/True positive rate (TPR)**

  Recall score, also known as Sensitivity, hit rate, or TPR, is the number of true positive results divided by the number of all samples that should have been identified as positive

$$TPR = Sensitivity = \frac{TP}{TP+FN} \tag{8.6}$$

- **False positive rate (FPR)**

$$FPR = \frac{FP}{TP+FN} \tag{8.7}$$

- **Accuracy/Rand Index**

  Accuracy score, also known as Rand index is the number of correct predictions, consisting of correct positive and negative predictions divided by the total number of predictions.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \tag{8.8}$$

- **Volume similarity** Volume similarity measures the absolute size difference of the regions, as a fraction of the size of the sum of reference and segmentation result. There is more than one definations for the volume similarity [157].

  1. The first definition is [157]:

$$VS = 1 - \frac{|V_{pred}| - |V_{gdth}|}{|V_{pred}| + |V_{gdth}|} \tag{8.9}$$

where V_{pred} is the volume of prediction and V_{gdth} is the volume of the ground truth. It ranges from 0 to 1. Higher value means the size (volume) of the prediction is more similar (close) with the size (volume) of the ground truth.

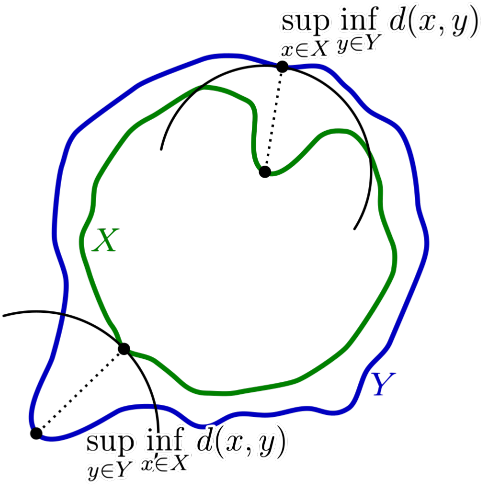

$\sup_{x\in X}\inf_{y\in Y} d(x,y)$
X
Y
$\sup_{y\in Y}\inf_{x\in X} d(x,y)$

Figure 8.1: Hausdorff distance between the green curve X and the blue curve Y.

2. The second definition is:

$$VS = \frac{2\cdot(|V_{pred}| - |V_{gdth}|)}{|V_{pred}| + |V_{gdth}|} \tag{8.10}$$

This definition is from the official tutorial of `SimpleITK` [158]. Negative VS means the volume of prediction is less than the volume of ground truth, which is called **underestimation**. Positive VS means the volume of prediction is greater than the volume of the ground truth, which is called **overestimation**.

In our package `seg_metrics`, we implemented the **second** definition. Please note that none of the two equations represent overlap information. VS only represents the volume size difference between prediction and ground truth.

8.3.2 Distance-based metrics

- **Hausdorff distance (HD)** (see Figure 8.1)

$$HD = \max\left\{\sup_{a\in A}\inf_{b\in B} d(a,b), \sup_{b\in B}\inf_{a\in A} d(b,a)\right\} \tag{8.11}$$

where sup represents the supremum operator, inf is the infimum operator, and $inf_{b\in B} d(a,b)$ quantifies the distance from a point $a \in X$ to the subset $B \subseteq X$.

- **Hausdorff distance 95% percentile (HD95)** is the 95% percentile of surface distances between segmentation and reference.

- **Mean (Average) surface distance (MSD)** is the mean value of surface distances between segmentation and reference [159, 160].

- **Median surface distance (MDSD)** is the median value of surface distances between segmentation and reference.

  Note: These metrics are **symmetric**, which means the distance from segmentation result to reference is the same as the distance from reference to segmentation result.

8.4 Installation

Our package was published in the Python Package Index (PyPI), which is the official third-party software repository for Python. Thus, `seg-metrics` can be directly installed and immediately used in any Python environment using a single line as follows.

```
$ pip install seg-metrics
```

8.5 Use cases

`seg-metrics` is a Python package which outputs the segmentation metrics by receiving one ground truth image and another predicted image. After we import the package by "`from seg_metrics import seg_metrics`", the syntax to use it is as follow (**Note:** all the following cases are based on texttttseg-metrics 1.1.6).

```
from seg_metrics.seg_metrics import write_metrics
write_metrics(labels,
              gdth_path = None,
              pred_path = None,
              csv_file = None,
              gdth_img = None,
              pred_img = None,
              metrics = None,
              verbose = False,
              spacing = None,
              fully_connected = True,
              TPTNFPFN = False)
    """ Parameter description.
    labels: a list of labels to performe the calculation of metrics.
    gdth_path: a (sequence of) path of ground truth.
    pred_path: a (sequence of) path of prediction.
    csv_file: filename to save the metrics.
    gdth_img: a (sequence of) ground truth.
    pred_img: a (sequence of) prediction.
    metrics: metric names.
    verbose: whether to show the animated progress bar
    spacing: spacing of input images.
    fully_connected: whether to apply fully connected border.
    TPTNFPFN: whether to return the confusion matrix.

    return: A dict or a list of dicts which store metrics.
    """
```

More examples are shown below.

- Evaluate two batches of images with same filenames from two different folders.

```
labels = [4, 5 ,6 ,7 , 8]
gdth_path = 'data/gdth'  # folder for ground truth images
pred_path = 'data/pred'  # folder for predicted images
csv_file = 'metrics.csv'  # file to save results

metrics = sg.write_metrics(labels=labels,
                  gdth_path=gdth_path,
                  pred_path=pred_path,
                  csv_file=csv_file)
print(metrics)
```

- Evaluate two images

```
labels = [4, 5 ,6 ,7 , 8]
gdth_file = 'data/gdth.mhd'  # full path for ground truth image
pred_file = 'data/pred.mhd'  # full path for prediction image
csv_file = 'metrics.csv'

metrics = sg.write_metrics(labels=labels,
                           gdth_path=gdth_file,
                           pred_path=pred_file,
                           csv_file=csv_file)
```

- Evaluate two images with specific metrics

```
labels = [0, 4, 5 ,6 ,7 , 8]
gdth_file = 'data/gdth.mhd'
```

```
pred_file = 'data/pred.mhd'
csv_file = 'metrics.csv'

metrics = sg.write_metrics(labels=labels[1:],
                  gdth_path=gdth_file,
                  pred_path=pred_file,
                  csv_file=csv_file,
                  metrics=['dice', 'hd'])
# for only one metric
metrics = sg.write_metrics(labels=labels[1:],
                  gdth_path=gdth_file,
                  pred_path=pred_file,
                  csv_file=csv_file,
                  metrics='msd')
```

- Select specific metrics. By passing the following parameters to select specific metrics.

```
# ----------Overlap based metrics---------------
- dice:         Dice (F-1)
- jaccard:      Jaccard
- precision:    Precision
- recall:       Recall
- fpr:          False positive rate
- fnr:          False negtive rate
- vs:           Volume similarity
# ----------Distance based metrics---------------
- hd:           Hausdorff distance
- hd95:         Hausdorff distance 95% percentile
- msd:          Mean (Average) surface distance
- mdsd:         Median surface distance
- stdsd:        Std surface distance
```

For example:

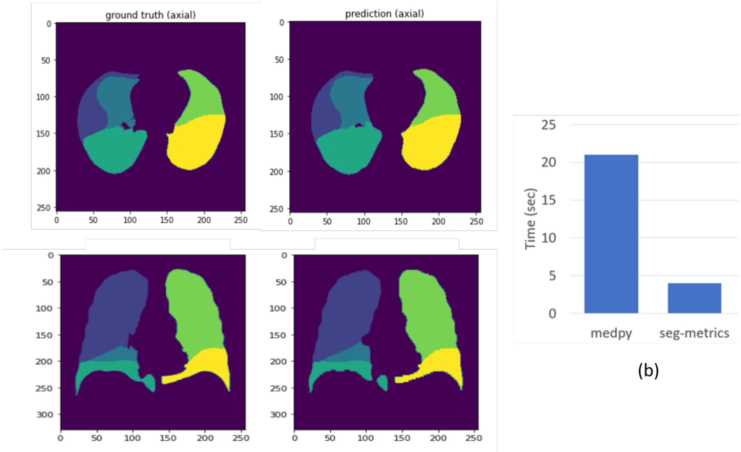
```
labels = [1]
gdth_file = 'data/gdth.mhd'
pred_file = 'data/pred.mhd'
csv_file = 'metrics.csv'
```

```
metrics = sg.write_metrics(labels, gdth_file, pred_file,
                           csv_file, metrics=['dice', 'hd95'])
dice = metrics['dice']
hd95 = metrics['hd95']
```

8.6 Comparison to other packages

`medpy` also provide functions to calculate metrics for medical images. Compared to it, our package `seg-metrics` has several advantages.

- **Faster.** `seg-metrics` is 5-10 times faster calculating distance based metrics (see Figure 8.2).



(a)

(b)

Figure 8.2: Performance comparison between `medpy` and `seg-metrics`. (a) Evaluated samples, a 3D lung lobe segmentation results (size: 256 × 256 × 256). Left: ground truth (manually annotated lobes), right: prediction (automatically predicted lobes). (b) Time comparison for the calculation of 'hd', 'hd95' and 'msd'.

- **More convenient.** `seg-metrics` can calculate all different metrics in once in one function (shown below)

```
gdth, pred = ...... # load two images
metrics = sg.write_metrics(labels=[1],
                  gdth_img=gdth,
                  pred_img=pred,
                  spacing=spacing,
                  metrics=['hd', 'hd95', 'msd']) # 3 outputs
```

while `medpy` needs to call different functions multiple times which cost more code and time, because the calculation of each 'hd', 'hd95', and 'msd' will always recalculate the distance map which cost much time.

```
hd = medpy.metric.binary.hd(result=pred, reference=gdth)
hd95 = medpy.metric.binary.hd95(result=pred, reference=gdth)
msd = medpy.metric.binary.asd(result=pred, reference=gdth)
```

- **More Powerful.** `seg-metrics` can calculate multi-label segmentation metrics and save results to .csv file in good manner, but `medpy` only provides **binary** segmentation metrics. For instance, if there are 5 labels for an image, our `seg-metrics` can calculate 5-label metrics by one-line command while `medpy` needs to at first convert 5-label image to five binary images, then calculate binary metrics one by one,

8.7 Limitation and future work

Because of time limitation for the development, there is still some space for package improvement.

- Package name. The package name is "seg-metrics" currently, as the abbreviation of "segmentation metrics". But the dash sign "-" in the name introduced some confusion during the installing and usage of the package. Duing the installation, `pip install seg-metrics` is used. However, users need to used it by `import seg_metrics`. The slight difference sometimes make new users confused and easy to make mistakes. This issue is because Python packaging system will automatically convert "_" to "-" during the installing. Because "segmetrics" has been used by other products, we may consider to change the package name to "metricseg", "metricsrater", "imagesegmetrics", etc. to avoid such issue in the future.

- Supported file type. Currently, the package supports most medical image formats with suffix of `.mhd, .mha, .nii, .nii.gz, .nrrd`, etc. Because we receive some users' requests, we will support more image formats (e.g. `.png, .jpg`) in the future.

- Usage guide. Currently, we just list the usage of different metrics, but we did not explain when to use which metrics. In the future, we hope to release a tutorial to users with some examples to which metrics are preferable in different scenarios.

8.8 Availability and requirements

- **Project name:** seg-metrics
- **Project home page:** `https://github.com/Jingnan-Jia/segmentation_metrics`
- **Operating system(s):** Platform independent
- **Programming language:** Python
- **License:** MIT license
- **Any restrictions to use by non-academics:** none

8.9 Acknowledgments

This author was supported by the China Scholarship Council No.202007720110 during the development of this package.

References

[1] H. Aung, A Sivakumar, S. Gholami, et al. “An overview of the anatomy and physiology of the lung”. In: *Nanotechnology-Based Targeted Drug Delivery Systems for Lung Cancer* (2019), pages 1–20.

[2] F. H. Martini and F. Martini. *Anatomy & physiology*. Rex Bookstore, Inc., 2005.

[3] C. P. Denton and D. Khanna. “Systemic sclerosis”. In: *The Lancet* 390.10103 (2017), pages 1685–1699.

[4] Y. Asano. “Systemic sclerosis”. In: *The Journal of dermatology* 45.2 (2018), pages 128–138.

[5] S. Acharya, S. Shukla, S. N. Mahajan, et al. “Interstitial lung disease in systemic sclerosis”. In: *Journal of Datta Meghe Institute of Medical Sciences University* 8.1 (Dec. 2013), pages 57–59.

[6] M. Elhai, C. Meune, M. Boubaya, et al. “Mapping and predicting mortality from systemic sclerosis”. In: *Annals of the rheumatic diseases* 76.11 (2017), pages 1897–1905.

[7] R. Steele, M. Hudson, E. Lo, et al. “Clinical decision rule to predict the presence of interstitial lung disease in systemic sclerosis”. In: *Arthritis care & research* 64.4 (2012), pages 519–524.

[8] H. Codes, A. A. Guler, C. Campochiaro, et al. “Systemic sclerosis and interstitial lung disease: From pathogenesis, to screening, diagnosis, and classification”. In: *Revista Colombiana de Reumatología* (2023).

[9] E. R. Volkmann, K. Andréasson, and V. Smith. “Systemic sclerosis”. In: *The Lancet* 401.10373 (2023), pages 304–318.

[10] Y. Allanore, R. Simms, O. Distler, et al. “Systemic sclerosis”. In: *Nature reviews Disease primers* 1.1 (2015), pages 1–21.

[11] E. J. Bernstein, S. Jaafar, S. Assassi, et al. “Performance characteristics of pulmonary function tests for the detection of interstitial lung disease in adults with early diffuse cutaneous systemic sclerosis”. In: *Arthritis & Rheumatology* 72.11 (2020), pages 1892–1896.

[12] O. Distler, S. Assassi, V. Cottin, et al. “Predictors of progression in systemic sclerosis patients with interstitial lung disease”. In: *European Respiratory Journal* 55.5 (2020).

[13] G. Raghu, M. Remy-Jardin, L. Richeldi, et al. “Idiopathic pulmonary fibrosis (an update) and progressive pulmonary fibrosis in adults: an official ATS/ERS/JRS/ALAT clinical practice guideline”. In: *American Journal of Respiratory and Critical Care Medicine* 205.9 (2022), e18–e47.

[14] J. Behr and D. E. Furst. “Pulmonary function tests”. In: *Rheumatology* 47.suppl_5 (2008), pages v65–v67.

[15] M. Caron, S. Hoa, M. Hudson, et al. “Pulmonary function tests as outcomes for systemic sclerosis interstitial lung disease”. In: *European Respiratory Review* 27.148 (June 2018).

[16] M. K. Ninaber, J. Stolk, J. Smit, et al. “Lung structure and function relation in systemic sclerosis: application of lung densitometry”. In: *European Journal of Radiology* 84.5 (2015), pages 975–979.

[17] K. P. Sylvester, N. Clayton, I. Cliff, et al. “ARTP statement on pulmonary function testing 2020”. In: *BMJ Open Respiratory Research* 7.1 (2020), e000575.

[18] B. L. Graham, V. Brusasco, F. Burgos, et al. “2017 ERS/ATS standards for single-breath carbon monoxide uptake in the lung”. In: *European Respiratory Journal* 49.1 (2017).

[19] R. Mehra and H. K. Walia. “Evaluation and Monitoring of Respiratory Function”. In: *Sleep Disorders Medicine: Basic Science, Technical Considerations and Clinical Aspects* (2017), pages 339–352.

[20] N Macintyre, R. Crapo, G Viegi, et al. “Standardisation of the single-breath determination of carbon monoxide uptake in the lung”. In: *European Respiratory Journal* 26.4 (2005), pages 720–735.

[21] Y. S. Choi, J. Oh, S. Ahn, et al. “Automated Pulmonary Function Measurements from Preoperative CT Scans with Deep Learning”. In: *2022 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*. IEEE. 2022, pages 01–04.

[22] A. McGowan, P. Laveneziana, S. Bayat, et al. “International consensus on lung function testing during the COVID-19 pandemic and beyond”. In: *ERJ open research* 8.1 (Jan. 2022).

[23] B. G. Cooper. “An update on contraindications for lung function testing”. In: *Thorax* 66.8 (2011), pages 714–723.

[24] H. Meng, Y. Liu, X. Xu, et al. “A machine learning approach for preoperatively assessing pulmonary function with computed tomography in patients with lung cancer”. In: *Quantitative Imaging in Medicine and Surgery* 13.3 (2023), page 1510.

[25] A.-M. Hoffmann-Vold, T. M. Maher, E. E. Philpot, et al. “The identification and management of interstitial lung disease in systemic sclerosis: evidence-based European consensus statements”. In: *The Lancet Rheumatology* 2.2 (2020), e71–e83.

[26] K. Showalter, A. Hoffmann, G. Rouleau, et al. “Performance of forced vital capacity and lung diffusion cutpoints for associated radiographic interstitial lung disease in systemic sclerosis”. In: *The Journal of rheumatology* 45.11 (2018), pages 1572–1576.

[27] K. C. Silver and R. M. Silver. “Management of systemic-sclerosis-associated interstitial lung disease”. In: *Rheumatic Disease Clinics* 41.3 (2015), pages 439–457.

[28] S. K. Zhou, H. Greenspan, C. Davatzikos, et al. “A review of deep learning in medical imaging: Imaging traits, technology trends, case studies with progress highlights, and future promises”. In: *Proceedings of the IEEE* 109.5 (2021), pages 820–838.

[29] S. Hussein, P. Kandel, C. W. Bolan, et al. "Lung and pancreatic tumor characterization in the deep learning era: novel supervised and unsupervised learning approaches". In: *IEEE transactions on medical imaging* 38.8 (2019), pages 1777–1787.

[30] K. Liang, L. Zhang, H. Yang, et al. "A model-based unsupervised deep learning method for low-dose CT reconstruction". In: *IEEE Access* 8 (2020), pages 159260–159273.

[31] Z.-H. Zhou. "A brief introduction to weakly supervised learning". In: *National science review* 5.1 (2018), pages 44–53.

[32] S.-J. Yoo, S. H. Yoon, J. H. Lee, et al. "Automated lung segmentation on chest computed tomography images with extensive lung parenchymal abnormalities using a deep neural network". In: *Korean Journal of Radiology* 22.3 (2021), page 476.

[33] M. Prokop, W. Van Everdingen, T. van Rees Vellinga, et al. "CO-RADS: a categorical CT assessment scheme for patients suspected of having COVID-19—definition and evaluation". In: *Radiology* 296.2 (2020), E97–E104.

[34] D Assayag, S Kaduri, M Hudson, et al. "High Resolution Computed Tomography Scoring Systems for Evaluating Interstitial Lung Disease in Systemic Sclerosis Patients". In: *Rheumatology, an open access journal Assayag et al. Rheumatology* 1 (2012), page 3.

[35] N. S. Goh, S. R. Desai, S. Veeraraghavan, et al. "Interstitial lung disease in systemic sclerosis: a simple staging system". In: *American journal of respiratory and critical care medicine* 177.11 (2008), pages 57–59.

[36] B. D. De Vos, J. M. Wolterink, P. A. de Jong, et al. "ConvNet-based localization of anatomical structures in 3-D medical images". In: *IEEE transactions on medical imaging* 36.7 (2017), pages 1470–1481.

[37] V. Proskurov, A. Kurmukov, M. Pisov, and M. Belyaev. "Fast lung localization in computed tomography by a 1D detection network". In: *2021 Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT)*. IEEE. 2021, pages 0173–0176.

[38] H. Chen, C. Shen, J. Qin, et al. "Automatic localization and identification of vertebrae in spine CT via a joint learning model with deep neural networks". In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part I 18*. Springer. 2015, pages 515–522.

[39] S. Belharbi, C. Chatelain, R. Hérault, et al. "Spotting L3 slice in CT scans using deep convolutional network and transfer learning". In: *Computers in Biology and Medicine* 87 (Aug. 2017), pages 95–103.

[40] P. Cheng, Y. Yang, H. Yu, and Y. He. "Automatic vertebrae localization and segmentation in CT with a two-stage Dense-U-Net". In: *Scientific Reports* 11.1 (2021), page 22156.

[41] G. E. Humpire-Mamani, A. A. A. Setio, B. Van Ginneken, and C. Jacobs. "Efficient organ localization using multi-label convolutional neural networks in thorax-abdomen CT scans". In: *Physics in Medicine & Biology* 63.8 (2018), page 085003.

[42] C. Cano-Espinosa, G. González, G. R. Washko, et al. "Automated Agatston score computation in non-ECG gated CT scans using deep learning". In: *Medical Imaging 2018: Image Processing*. Volume 10574. SPIE. 2018, pages 673–678.

[43] B. D. De Vos, J. M. Wolterink, T. Leiner, et al. "Direct automatic coronary calcium scoring in cardiac and chest CT". In: *IEEE transactions on medical imaging* 38.9 (2019), pages 2127–2138.

[44] D. Mu, J. Bai, W. Chen, et al. "Calcium scoring at coronary CT angiography using deep learning". In: *Radiology* 302.2 (2022), pages 309–316.

[45] G. González, G. R. Washko, and R. S. J. Estépar. "Deep learning for biomarker regression: application to osteoporosis and emphysema on chest CT scans". In: *Medical Imaging 2018: Image Processing*. Volume 10574. SPIE. 2018, pages 372–378.

[46] J. Park, J. Yun, N. Kim, et al. "Fully automated lung lobe segmentation in volumetric chest CT with 3D U-Net: validation with intra-and extra-datasets". In: *Journal of digital imaging* 33 (2020), pages 221–230.

[47] B. Lassen, E. M. van Rikxoort, M. Schmidt, et al. "Automatic segmentation of the pulmonary lobes from chest CT scans based on fissures, vessels, and bronchi". In: *IEEE transactions on medical imaging* 32.2 (2012), pages 210–222.

[48] S. E. Gerard, T. J. Patton, G. E. Christensen, et al. "FissureNet: a deep learning approach for pulmonary fissure detection in CT images". In: *IEEE transactions on medical imaging* 38.1 (2018), pages 156–166.

[49] K. George, A. P. Harrison, D. Jin, et al. "Pathological pulmonary lobe segmentation from CT images using progressive holistically nested neural networks and random walker". In: *International Workshop on Deep Learning in Medical Image Analysis*. Springer. 2017, pages 195–203.

[50] F. T. Ferreira, P. Sousa, A. Galdran, et al. "End-to-end supervised lung lobe segmentation". In: *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2018, pages 1–8.

[51] S. Chen, K. Ma, and Y. Zheng. "Med3d: Transfer learning for 3d medical image analysis". In: *arXiv preprint arXiv:1904.00625* (2019).

[52] S. Chen, G. Bortsova, A. García-Uceda Juárez, et al. "Multi-task attention-based semi-supervised learning for medical image segmentation". In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part III 22*. Springer. 2019, pages 457–465.

[53] F. Milletari, N. Navab, and S.-A. Ahmadi. "V-net: Fully convolutional neural networks for volumetric medical image segmentation". In: *2016 fourth international conference on 3D vision (3DV)*. Ieee. 2016, pages 565–571.

[54] J Jia. "A package to compute segmentation metrics: seg-metrics". In: *online at https://pypi.org/project/seg-metrics* (2020).

[55] C. Xiao, B. C. Stoel, M. E. Bakker, et al. "Pulmonary fissure detection in CT images using a derivative of stick filter". In: *IEEE transactions on medical imaging* 35.6 (2016), pages 1488–1500.

[56] Z. Zhai, M. Staring, I. Hernández Girón, et al. "Automatic quantitative analysis of pulmonary vascular morphology in CT images". In: *Medical physics* 46.9 (2019), pages 3985–3997.

[57] A. A. A. Setio, A. Traverso, T. De Bel, et al. "Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: the LUNA16 challenge". In: *Medical image analysis* 42 (2017), pages 1–13.

[58] Z. Zhai, M. Staring, and B. C. Stoel. "Lung vessel segmentation in CT images using graph-cuts". In: *Medical Imaging 2016: Image Processing*. Volume 9784. SPIE. 2016, pages 699–706.

[59] C. P. Denton and D. Khanna. *Systemic sclerosis*. Oct. 2017.

[60] A. U. Wells. "Interstitial lung disease in systemic sclerosis". In: *La Presse Médicale* 43.10 (Oct. 2014), e329–e343.

[61] G. Chassagnon, M. Vakalopoulou, A. Régent, et al. "Deep Learning – based Approach for Automated Assessment of Interstitial Lung Disease in Systemic Sclerosis on CT Images". In: *Radiology: Artificial Intelligence* 2.4 (July 2020), e190006.

[62] S. R. Desai, S. Veeraraghavan, D. M. Hansell, et al. "CT features of lung disease in patients with systemic scerosis: Comparison with idiopathic pulmonary fibrosis and nonspecific interstitial pneumonia". In: *Radiology* 232.2 (Aug. 2004), pages 560–567.

[63] L. Williamson. "New reference atlas for pulmonary fibrosis severity score in systemic sclerosis". In: *The Lancet Respiratory Medicine* 9.2 (Feb. 2021), pages 130–131.

[64] C. D. Collins, A. U. Wells, D. M. Hansell, et al. "Observer variation in pattern type and extent of disease in fibrosing alveolitis on thin section computed tomography and chest radiography". In: *Clinical Radiology* 49.4 (1994), pages 236–240.

[65] N. Sverzellati, A. Devaraj, S. R. Desai, et al. "Method for minimizing observer variation for the quantitation of high-resolution computed tomographic signs of lung disease". In: *Journal of computer assisted tomography* 35.5 (Sept. 2011), pages 596–601.

[66] G. Gonzalez Serrano, G. R. Washko, and R. San José Estépar. "Deep learning for biomarker regression: application to osteoporosis and emphysema on chest CT scans". In: *Proceedings of SPIE–the International Society for Optical Engineering*. Volume 10574. SPIE-Intl Soc Optical Eng, Mar. 2018, page 52.

[67] G. González, G. R. Washko, R. S. J. Estépar, et al. "Automated Agatston score computation in non-ECG gated CT scans using deep learning". In: *Proceedings of SPIE–the International Society for Optical Engineering*. Volume 10574. SPIE-Intl Soc Optical Eng, Mar. 2018, page 91.

[68] Y. Wang, Y. Qiu, T. Thai, et al. “A two-step convolutional neural network based computer-aided detection scheme for automatically segmenting adipose tissue volume depicting on CT images”. In: *Computer Methods and Programs in Biomedicine* 144 (June 2017), pages 97–104.

[69] S. Dabiri, K. Popuri, C. Ma, et al. “Deep learning method for localization and segmentation of abdominal CT”. In: *Computerized Medical Imaging and Graphics* 85 (Oct. 2020), page 101776.

[70] C. P. Bridge, M. Rosenthal, B. Wright, et al. “Fully-automated analysis of body composition from CT in cancer patients using convolutional neural networks”. In: *OR 2.0 Context-aware operating theaters, computer assisted robotic endoscopy, clinical image-based procedures, and skin image analysis*. Volume 11041 LNCS. Springer, Cham, 2018, pages 204–213. arXiv: `1808.03844`.

[71] R. Shadmi, V. Mazo, O. Bregman-Amitai, and E. Elnekave. “Fully-convolutional deep-learning based system for coronary calcium score prediction from non-contrast chest CT”. In: *Proceedings - International Symposium on Biomedical Imaging* 2018-April (May 2018), pages 24–28.

[72] D. Wei, S. Ahmad, J. Huo, et al. “SLIR: Synthesis, localization, inpainting, and registration for image-guided thermal ablation of liver tumors”. In: *Medical image analysis* 65 (2020), page 101763.

[73] M. H. Jafari, H. Girgis, N. Van Woudenberg, et al. “Automatic biplane left ventricular ejection fraction estimation with mobile point-of-care ultrasound using multi-task learning and adversarial training”. In: *International journal of computer assisted radiology and surgery* 14 (2019), pages 1027–1037.

[74] J. G. Nam, Y. Choi, S.-M. Lee, et al. “Prognostic value of deep learning–based fibrosis quantification on chest CT in idiopathic pulmonary fibrosis”. In: *European Radiology* 33.5 (2023), pages 3144–3155.

[75] K. Nagpal, D. Foote, Y. Liu, et al. “Development and validation of a deep learning algorithm for improving Gleason scoring of prostate cancer”. In: *NPJ digital medicine* 2.1 (2019), page 48.

[76] A. H. M. Linkon, M. M. Labib, T. Hasan, M. Hossain, et al. “Deep learning in prostate cancer diagnosis and Gleason grading in histopathology images: An extensive study”. In: *Informatics in Medicine Unlocked* 24 (2021), page 100582.

[77] W. Bulten, H. Pinckaers, H. van Boven, et al. “Automated deep-learning system for Gleason grading of prostate cancer using biopsies: a diagnostic study”. In: *The Lancet Oncology* 21.2 (2020), pages 233–241.

[78] R. W. Stidham, W. Liu, S. Bishu, et al. “Performance of a deep learning model vs human reviewers in grading endoscopic disease severity of patients with ulcerative colitis”. In: *JAMA network open* 2.5 (2019), e193963–e193963.

[79] B. Astuto, I. Flament, N. K. Namiri, et al. “Automatic deep learning–assisted detection and grading of abnormalities in knee MRI studies”. In: *Radiology: Artificial Intelligence* 3.3 (2021), e200165.

[80] T. Araujo, G. Aresta, L. Mendonça, et al. “DR| GRADUATE: Uncertainty-aware deep learning-based diabetic retinopathy grading in eye fundus images”. In: *Medical Image Analysis* 63 (2020), page 101715.

[81] P. Chen, L. Gao, X. Shi, et al. “Fully automatic knee osteoarthritis severity grading using deep neural networks with a novel ordinal loss”. In: *Computerized Medical Imaging and Graphics* 75 (2019), pages 84–92.

[82] S. L. Walsh, J. A. Mackintosh, L. Calandriello, et al. “Deep Learning-based Outcome Prediction in Progressive Fibrotic Lung Disease Using High-Resolution Computed Tomography”. In: *American journal of respiratory and critical care medicine* 206.7 (Oct. 2022), pages 883–891.

[83] G. Luo, S. Dong, W. Wang, et al. “Commensal correlation network between segmentation and direct area estimation for bi-ventricle quantification”. In: *Medical image analysis* 59 (2020), page 101591.

[84] N. Su, F. Hou, W. Zheng, et al. “Computed Tomography–Based Deep Learning Model for Assessing the Severity of Patients With Connective Tissue Disease–Associated Interstitial Lung Disease”. In: *Journal of computer assisted tomography* 47.5 (2023), pages 738–745.

[85] J. Meijs, A. A. Schouffoer, N. A. Marsan, et al. “Original article: Therapeutic and diagnostic outcomes of a standardised, comprehensive care pathway for patients with systemic sclerosis”. In: *RMD Open* 2.1 (2016).

[86] M. K. Ninaber, J. Stolk, J. Smit, et al. “Lung structure and function relation in systemic sclerosis: Application of lung densitometry”. In: *European Journal of Radiology* 84.5 (May 2015), pages 975–979.

[87] J. Jia, M. Staring, I. H. Girón, et al. “Prediction of lung CT scores of systemic sclerosis by cascaded regression neural networks”. In: *Medical Imaging 2022: Computer-Aided Diagnosis*. Volume 12033. SPIE, 2022, pages 837–843.

[88] K. Simonyan and A. Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings* (Sept. 2014). arXiv: `1409.1556`.

[89] K. He, X. Zhang, S. Ren, and J. Sun. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pages 770–778.

[90] F. N. Iandola, S. Han, M. W. Moskewicz, et al. “SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and$<$ 0.5 MB model size”. In: *arXiv preprint arXiv:1602.07360* (2016).

[91] S. Xie, R. Girshick, P. Dollár, et al. “Aggregated residual transformations for deep neural networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pages 1492–1500.

[92] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. “Densely Connected Convolutional Networks”. In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* 2017-Janua (Aug. 2016), pages 2261–2269. arXiv: `1608.06993`.

[93] X. Zhang, X. Zhou, M. Lin, and J. Sun. “Shufflenet: An extremely efficient convolutional neural network for mobile devices”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pages 6848–6856.

[94] Z. Liu, H. Mao, C.-Y. Wu, et al. “A ConvNet for the 2020s”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pages 11966–11976. arXiv: `2201.03545`.

[95] J. Zhang, L. Liu, P. Wang, and C. Shen. “To balance or not to balance: A simple-yet-effective approach for learning with long-tailed distributions”. In: *arXiv preprint arXiv:1912.04486* (2019).

[96] Z. Zhai, M. Staring, M. K. Ninaber, et al. “Pulmonary vascular morphology associated with gas exchange in systemic sclerosis without lung fibrosis”. In: *Journal of thoracic imaging* 34.6 (2019), pages 373–379.

[97] M. D. Zeiler and R. Fergus. “Visualizing and understanding convolutional networks”. In: *European conference on computer vision*. Springer. 2014, pages 818–833.

[98] T. K. Koo and M. Y. Li. “A guideline of selecting and reporting intraclass correlation coefficients for reliability research”. In: *Journal of chiropractic medicine* 15.2 (2016), pages 155–163.

[99] F. Pedregosa, V. Michel, O. Grisel, et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pages 2825–2830.

[100] R. Vallat. “Pingouin: statistics in Python.” In: *J. Open Source Softw.* 3.31 (2018), page 1026.

[101] J. G. Goldin, D. A. Lynch, D. C. Strollo, et al. “High-Resolution CT Scan Findings in Patients With Symptomatic Scleroderma-Related Interstitial Lung Disease”. In: *Chest* 134.2 (Aug. 2008), pages 358–367.

[102] A. K. Pandey, P. Wilcox, J. R. Mayo, et al. “Predictors of pulmonary hypertension on high-resolution computed tomography of the chest in systemic sclerosis: A retrospective analysis”. In: *Canadian Association of Radiologists Journal* 61.5 (Dec. 2010), pages 291–296.

[103] E. L. Herzog, A. Mathur, A. M. Tager, et al. “Interstitial Lung Disease Associated With Systemic Sclerosis and Idiopathic Pulmonary Fibrosis: How Similar and Distinct?” In: *Arthritis & rheumatology (Hoboken, N.J.)* 66.8 (2014), page 1967.

[104] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, et al. “Generative adversarial nets”. In: *Advances in Neural Information Processing Systems*. Volume 3. January. 2014, pages 2672–2680.

[105] B. L. Graham, I. Steenbruggen, M. R. Miller, et al. "Standardization of spirometry 2019 update. An official American thoracic society and European respiratory society technical statement". In: *American journal of respiratory and critical care medicine* 200.8 (2019), e70–e88.

[106] N. S. Goh, S. Veeraraghavan, S. R. Desai, et al. "Bronchoalveolar lavage cellular profiles in patients with systemic sclerosis–associated interstitial lung disease are not predictive of disease progression". In: *Arthritis & Rheumatism* 56.6 (2007), pages 2005–2012.

[107] H. Park, J. Yun, S. M. Lee, et al. "Deep Learning–based Approach to Predict Pulmonary Function at Chest CT". In: *Radiology* 307.2 (2023), e221488.

[108] M. R. Miller, J. Hankinson, V. Brusasco, et al. "Standardisation of spirometry". In: *European respiratory journal* 26.2 (2005), pages 319–338.

[109] G. L. Hall, N. Filipow, G. Ruppel, et al. "Official ERS technical standard: Global Lung Function Initiative reference values for static lung volumes in individuals of European ancestry". In: *European Respiratory Journal* 57.3 (2021).

[110] S. Stanojevic, B. L. Graham, B. G. Cooper, et al. "Official ERS technical standards: Global Lung Function Initiative reference values for the carbon monoxide transfer factor for Caucasians". In: *European Respiratory Journal* 50.3 (2017).

[111] P. H. Quanjer, S. Stanojevic, T. J. Cole, et al. "Multi-ethnic reference values for spirometry for the 3–95-yr age range: the global lung function 2012 equations". In: *European Respiratory Journal* 40.6 (June 2012), pages 1324–1343.

[112] C. Feichtenhofer. "X3d: Expanding architectures for efficient video recognition". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pages 203–213.

[113] V. Cheplygina. "Cats or CAT scans: Transfer learning from natural or medical image source data sets?" In: *Current Opinion in Biomedical Engineering* 9 (2019), pages 21–27.

[114] W. Kay, J. Carreira, K. Simonyan, et al. "The kinetics human action video dataset". In: *arXiv preprint arXiv:1705.06950* (2017).

[115] R. R. Selvaraju, M. Cogswell, A. Das, et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pages 618–626.

[116] Z. Wang and J. Yang. "Diabetic retinopathy detection via deep convolutional networks for discriminative localization and visual explanation". In: *arXiv preprint arXiv:1703.10757* (2017).

[117] P. Schober, C. Boer, and L. A. Schwarte. "Correlation coefficients: appropriate use and interpretation". In: *Anesthesia & analgesia* 126.5 (2018), pages 1763–1768.

[118] N. Le Gouellec, A. Duhamel, T. Perez, et al. "Predictors of lung function test severity and outcome in systemic sclerosis-associated interstitial lung disease". In: *PLoS One* 12.8 (2017), e0181692.

[119] S. Iwano, T. Okada, H. Satake, and S. Naganawa. “3D-CT Volumetry of the Lung Using Multidetector Row CT: Comparison with Pulmonary Function Tests”. In: *Academic Radiology* 16.3 (Mar. 2009), pages 250–256.

[120] E. A. Vallejos, J. Martinez, F. Cabrera, et al. *Association of FVC/DLCO with pulmonary hypertension risk and interstitial disease in systemic sclerosis patients*. 2020.

[121] S. J. Park. “Effects of Inspiratory Muscles Training Plus Rib Cage Mobilization on Chest Expansion, Inspiratory Accessory Muscles Activity and Pulmonary Function in Stroke Patients”. In: *Applied Sciences 2020, Vol. 10, Page 5178* 10.15 (July 2020), page 5178.

[122] H. Jia, Y. Xia, W. Cai, and H. Huang. *MICCAI 2020*. Volume 12264 LNCS. 2020, pages 480–490.

[123] M. E. Bakker, M. K. Ninaber, J. Stolk, et al. “Lung Density and Pulmonary Artery Diameter are Predictors of Pulmonary Hypertension in Systemic Sclerosis”. In: *Journal of Thoracic Imaging* 32.6 (2017), pages 391–397.

[124] J. Jia, E. R. Marges, J. K. De Vries-Bouwstra, et al. “Automatic Pulmonary Function Estimation From Chest CT Scans Using Deep Regression Neural Networks: The Relation Between Structure and Function in Systemic Sclerosis”. In: *IEEE Access* 11 (2023), pages 135272–135282.

[125] S. Siami-Namini, N. Tavakoli, and A. S. Namin. “The Performance of LSTM and BiLSTM in Forecasting Time Series”. In: *Proceedings - 2019 IEEE International Conference on Big Data, Big Data 2019* (Dec. 2019), pages 3285–3292.

[126] Y. Guo, H. Wang, Q. Hu, et al. “Deep Learning for 3D Point Clouds: A Survey”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.12 (Dec. 2021), pages 4338–4364. arXiv: `1912.12033`.

[127] A. Dai, A. X. Chang, M. Savva, et al. “ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes”. In: ().

[128] C. R. Qi, H. Su, M. Niebner, et al. “Volumetric and Multi-View CNNs for Object Classification on 3D Data”. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2016-December (Apr. 2016), pages 5648–5656. arXiv: `1604.03265`.

[129] Y. Li, R. Bu, M. Sun, et al. “PointCNN: Convolution On X-Transformed Points”. In: (). arXiv: `1801.07791v5`.

[130] C. R. Q. Li, Y. Hao, S. Leonidas, and J Guibas. “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space”. In: (). arXiv: `1706.02413v1`.

[131] A. Vaswani, N. Shazeer, N. Parmar, et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).

[132] H. Hu, Z. Zhang, Z. Xie, and S. Lin. “Local Relation Networks for Image Recognition”. In: *Proceedings of the IEEE International Conference on Computer Vision* 2019-October (Apr. 2019), pages 3463–3472. arXiv: `1904.11491`.

[133] H. Zhao, L. Jiang, J. Jia, et al. “Point Transformer”. In: *Proceedings of the IEEE International Conference on Computer Vision* (Dec. 2020), pages 16239–16248. arXiv: `2012.09164`.

[134] G. Qian, Y. Li, H. Peng, et al. “PointNeXt: Revisiting PointNet++ with Improved Training and Scaling Strategies”. In: (June 2022). arXiv: `2206.04670`.

[135] K. Zhou, X. Huang, Y. Li, et al. “Towards Deeper Graph Neural Networks with Differentiable Group Normalization”. In: *Advances in Neural Information Processing Systems* 2020-December (June 2020). arXiv: `2006.06972`.

[136] F. Wu, T. Zhang, A. H. de Souza, et al. “Simplifying Graph Convolutional Networks”. In: *36th International Conference on Machine Learning, ICML 2019* 2019-June (Feb. 2019), pages 11884–11894. arXiv: `1902.07153`.

[137] T. N. Kipf and M. Welling. “Semi-Supervised Classification with Graph Convolutional Networks”. In: *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings* (Sept. 2016). arXiv: `1609.02907`.

[138] C. Morris, M. Ritzert, M. Fey, et al. “Weisfeiler and leman go neural: Higher-order graph neural networks”. In: *Proceedings of the AAAI conference on artificial intelligence*. Volume 33. 01. 2019, pages 4602–4609.

[139] X. Xu, F. Zhou, B. Liu, et al. “Efficient Multiple Organ Localization in CT Image Using 3D Region Proposal Network”. In: *IEEE Transactions on Medical Imaging* 38.8 (Aug. 2019), pages 1885–1898.

[140] M. Defferrard, X. Bresson, and P. Vandergheynst. “Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering”. In: *Advances in Neural Information Processing Systems* (June 2016), pages 3844–3852. arXiv: `1606.09375`.

[141] P. Schweitzer PASCAL, E. Jan van Leeuwen EJVANLEEUWEN, N. Shervashidze, et al. “Weisfeiler-Lehman Graph Kernels Nino Shervashidze Kurt Mehlhorn Karsten M. Borgwardt”. In: *Journal of Machine Learning Research* 12 (2011), pages 2539–2561.

[142] P. Veličkoviťveličkoviťc, G. Cucurull, A. Casanova, et al. “GRAPH ATTENTION NETWORKS”. In: (). arXiv: `1710.10903v3`.

[143] M. Dörrich, M. Fan, and A. M. Kist. “Impact of Mixed Precision Techniques on Training and Inference Efficiency of Deep Neural Networks”. In: *IEEE Access* (2023).

[144] J. Byrd and Z. Lipton. “What is the effect of importance weighting in deep learning?” In: *International conference on machine learning*. PMLR. 2019, pages 872–881.

[145] I. Loshchilov and F. Hutter. “Sgdr: Stochastic gradient descent with warm restarts”. In: *arXiv preprint arXiv:1608.03983* (2016).

[146] I. Loshchilov and F. Hutter. “Decoupled weight decay regularization”. In: *arXiv preprint arXiv:1711.05101* (2017).

[147] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. “Algorithms for hyper-parameter optimization”. In: *Advances in neural information processing systems* 24 (2011).

[148] D. Ulyanov, A. Vedaldi, and V. Lempitsky. “Instance Normalization: The Missing Ingredient for Fast Stylization”. In: (July 2016). arXiv: `1607.08022`.

[149] S. Ioffe and C. Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: *32nd International Conference on Machine Learning, ICML 2015* 1 (Feb. 2015), pages 448–456. arXiv: `1502.03167`.

[150] J. L. Ba, J. R. Kiros, and G. E. Hinton. “Layer Normalization”. In: (July 2016). arXiv: `1607.06450`.

[151] J. C. Cai, Z. Akkus, K. A. Philbrick, et al. “Fully Automated Segmentation of Head CT Neuroanatomy Using Deep Learning”. In: *Radiology: Artificial Intelligence* 2.5 (Sept. 2020), e190183.

[152] A. T. Barroso, E. M. Martín, L. M. R. Romero, and F. O. Ruiz. “Factors affecting lung function: a review of the literature”. In: *Archivos de Bronconeumología (English Edition)* 54.6 (2018), pages 327–332.

[153] D. Müller, I. Soto-Rey, and F. Kramer. “Towards a guideline for evaluation metrics in medical image segmentation”. In: *BMC Research Notes* 15.1 (2022), page 210.

[154] B. C. Lowekamp, D. T. Chen, L. Ibáñez, and D. Blezek. “The design of SimpleITK”. In: *Frontiers in neuroinformatics* 7 (2013), page 45.

[155] *Medpy*. `https://pypi.org/project/MedPy/`. Accessed: 2015.

[156] L. Maier-Hein, B. Menze, et al. “Metrics reloaded: Pitfalls and recommendations for image analysis validation”. In: *arXiv. org* 2206.01653 (2022).

[157] A. A. Taha and A. Hanbury. “Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool”. In: *BMC medical imaging* 15.1 (2015), pages 1–28.

[158] *Segmentation Evaluation*. `http://insightsoftwareconsortium.github.io/SimpleITK-Notebooks/R_html/34_Segmentation_Evaluation.html`. Accessed: 2023-09-30.

[159] T. Heimann, B. Van Ginneken, M. A. Styner, et al. “Comparison and evaluation of methods for liver segmentation from CT datasets”. In: *IEEE transactions on medical imaging* 28.8 (2009), pages 1251–1265.

[160] V. Yeghiazaryan and I. Voiculescu. “Family of boundary overlap metrics for the evaluation of medical image segmentation”. In: *Journal of Medical Imaging* 5.1 (2018), pages 015006–015006.

List of publications

Journal articles

Jia, Jingnan, Emiel R. Marges, Jeska K. De Vries-Bouwstra, Maarten K. Ninaber, Lucia JM Kroft, Anne A. Schouffoer, Marius Staring, and Berend C. Stoel. "Automatic Pulmonary Function Estimation From Chest CT Scans Using Deep Regression Neural Networks: The Relation Between Structure and Function in Systemic Sclerosis." *IEEE Access* 11 (2023): 135272-135282.

Jia, Jingnan, Irene Hernández-Girón, Anne A. Schouffoer, Jeska K. De Vries-Bouwstra, Maarten K. Ninaber, Julie C. Korving, Marius Staring, Lucia JM Kroft, and Berend C. Stoel. "Explainable fully automated CT scoring of systemic sclerosis related interstitial lung disease by cascaded regression neural networks and its comparison with experts." (*submitted*).

Jia, Jingnan, Bo Yu, Prerak Mody, Maarten K. Ninaber, Lucia JM Kroft, Anne A. Schouffoer, Marius Staring, and Berend C. Stoel. "Using 3D point cloud and graph-based neural networks to improve the estimation of pulmonary function tests from chest CT." (*submitted*).

Wen, Jingxuan, YongChang Jiao, YiXuan Zhang, and **Jingnan Jia**. "Wideband circularly polarized dielectric resonator antenna loaded with partially reflective surface." *International Journal of RF and Microwave ComputerAided Engineering* 29, no. 12 (2019): e21962.

International conference proceedings

Jia, Jingnan, Zhiwei Zhai, M. Els Bakker, Irene Hernández-Girón, Marius Staring, and Berend C. Stoel. "Multi-task semi-supervised learning for pulmonary lobe segmentation." In *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, pp. 1329-1332. IEEE, 2021.

Jia, Jingnan, Marius Staring, Irene Hernández-Girón, Lucia JM Kroft, Anne A. Schouffoer, and Berend C. Stoel. "Prediction of lung CT scores of systemic sclerosis by cascaded regression neural networks." In *Medical Imaging 2022: ComputerAided Diagnosis*, vol. 12033, pp. 837-843. SPIE, 2022.

Preprint articles

Jia, Jingnan, Marius Staring, and Berend C. Stoel "seg-metrics: a Python package to compute segmentation metrics." *medRxiv*, 2024-02.

Open source software

Jia, Jingnan. "A package to compute segmentation metrics: seg-metrics." *online at https://pypi.org/project/seg-metrics* (2020).

Acknowledgements

The past few years in Leiden have been incredibly wonderful and I am very grateful for the many people and things I have encountered here.

Firstly, I would like to give my thanks to my previous promotor Prof. Boudewijn Lelieveldt, my current promotor Prof. Marius Staring and my supervisor Dr. Berend C. Stoel. Thank you for providing me with the position in LKEB. LKEB is a warm, international, and cooperative lab.

Berend, I feel very lucky to have you as my supervisor, not only for the very detailed academic guidance, but also for the emotional support. You are so kind, modest and gentle, which made the difficult PhD relatively easy.

Marius, you approach academics with great rigor, while remaining highly efficient and creative in different projects. I got many ideas from your constructive comments, especially when faced with technical challenges.

I appreciate that I met so many wonderful friendly colleagues in LKEB. Zhiwei, thank you for helping me gradually adapt to the life in the Netherlands. You also gave me valuable suggestions on my projects. Irene, thank you for your detailed discussion on weekly meeting. I also appreciated the several afternoons walking with you. Xiaowu, thank you for taking me in when I locked myself out so many times. Mody, thank you for your active invitations of various academic or non-academic activities. You initiated the new LKEB wiki and shared a lot of useful tools, which have benefited me a lot. Li-shin, thank you for helping me to switch from TensorFlow to PyTorch. You not only play table tennis very well but also make wonderful bubble tea. Yanli, Qiuyu and Xiaotong, thank you for your discussion about your or my projects which inspired me a lot. Xiaotong, I also thank you for helping me attend the courses in Amsterdam which I absolutely forgot. Michèle, thank you for the help on the development and maintenance of *shark_resource_usage* and other IT support. Bo, not only did you help me in my academic and job search process, but you also brought me a lot of happiness in daily life. Denis, thank you for your help on the use of MeVisLab. Victor, thank you for your help on the organization of Thursday's AI Meeting. Patrick, thank you for your sharing of MLflow and your daily lunch invitation. Silvia, thank you for your valuable suggestion about how to write a technical report, which lead to my first technical report about my open source package *seg-metrics*. Els, thank your for your work on the time-consuming pixel-wise annotation of lobe segmentation masks. Also, many thanks to Jouke, Rob, Faeze, Niels, Jeroen, Alexander, Baldur, Yunjie, Ruochen, Donghang, Chang, Soumyadeep, Jenia, Chinmay, Laurens, Vincent, Efe, Tahereh, Hessam, Mohamed, Oleh, Qian and Dennis.

I also would like to thank my cooperators from clinical domain. Lucia, Anne, Jeska and Julie, thank you for the support on the SSc scoring project. Emiel and Maarten, thank you for your support on the PFTs estimation project. I also thank my previous teacher in China, Prof. Yi Wang. I was very lucky to be your first research assistant. You helped me open the door to AI research.

I would also like to thank my dear friends. Kevin, thank your for guiding and supervising my workout. Brian and Jordi, it is a wonderful experience to travel to Ghent and Köln and enjoy bars there with you. Wensen, thank you for your help on a lot of daily things and emotional support. Thanks to Ling, Ningning, Chenhong, Ruilin, Youliang and ot her Chinese friends in Leiden. We cooked, traveled and played games together. Thanks to Zhixi, Shuangrong, Sipu, Jinlei and other friends in China. You gave me great emotional support remotely. Thanks to the China Scholarship Council for supporting me financially during my PhD study (感谢国家公派留学生奖学金对我博士研究的资金支持).

I would like to thank my families. 感谢妈妈和爸爸多年的辛苦付出, 感谢弟弟妹妹的关心以及对父母的照顾, 感谢豪豪, 胖胖和美美带给我的快乐时光. Thanks to my two cute cats, Pangpang and Meimei. You two have given me so much joy remotely.

I want to give my unique thanks to my girlfriend Wenqing. Thank you for your love. Meeting you in Leiden is the most beautiful thing that has happened in my PhD. I could not arrive here without your support.

Finally, I want to thank myself, Jingnan Jia. You always smiled at life. You did a great job ;).

Curriculum Vitae

Jingnan was born in Linfen City, Shanxi Province, China on August 29, 1993. In 2011, he started his bachelor in the major of Applied Physics at Taiyuan University of Technology. During his undergraduate study, he won the Bronze Medal (national level) of National College Student Business Plan Competition (The only one in his university) in 2014. He obtained the Outstanding Graduate in 2015. He began his master study from 2016 at Xidian University in the major of Electromagnetic Field and Microwave Technology. In the summer of 2018, he worked as a part-time research assistant in Dongguan University of Technology.

From 2019, he started his PhD study in the Division of Image Processing (Dutch abbreviation LKEB) under the Department of Radiology at Leiden University Medical Center (LUMC) in the Netherlands. His PhD research mainly focuses on automatic analysis of chest CT imaging in systemic sclerosis using deep learning.

From February 2024, he worked as a post-doctoral researcher in LKEB. His project is treatment response estimation from MRI images using deep learning.