UNIVERSITÄT
BERN

Graduate School for Cellular and Biomedical Sciences
UNIVERSITY OF BERN

Lung Pattern Analysis using Artificial Intelligence for the Diagnosis Support of Interstitial Lung Diseases

PhD Thesis submitted by

CHRISTODOULIDIS Stergios

from **Greece**

for the degree of PhD in Biomedical Engineering

Supervisor
Prof. Dr. MOUGIAKAKOU Stavroula
ARTORG Center for Biomedical Engineering Research
Faculty of Medicine of the University of Bern
Bern, Switzerland

Co-advisor
Prof. Dr. PARAGIOS Nikos
Centre de Vision Numérique
Centrale Supélec
Paris, France

Copyright Notice

Accepted by the Faculty of Medicine, the Faculty of Science and the Vetsuisse Faculty of the University of Bern at the request of the Graduate School for Cellular and Biomedical Sciences

Bern, Dean of the Faculty of Medicine

Bern, Dean of the Faculty of Science

Bern, Dean of the Vetsuisse Faculty Bern

UNIVERSITY OF BERN
Graduate School for Cellular and Biomedical Sciences
Faculty of Medicine

Abstract

Doctor of Philosophy in Biomedical Engineering

Lung Pattern Analysis using Artificial Intelligence for the Diagnosis Support of Interstitial Lung Diseases

by CHRISTODOULIDIS Stergios

Interstitial lung diseases (ILDs) is a group of more than 200 chronic lung disorders characterized by inflammation and scarring of the lung tissue that leads to respiratory failure. Although ILD is a heterogeneous group of histologically distinct diseases, most of them exhibit similar clinical presentations and their diagnosis often presents a diagnostic dilemma. Early diagnosis is crucial for making treatment decisions, while misdiagnosis may lead to life-threatening complications. If a final diagnosis cannot be reached with the high resolution computed tomography scan, additional invasive procedures are required (e.g. bronchoalveolar lavage, surgical biopsy). The aim of this PhD thesis was to investigate the components of a computational system that will assist radiologists with the diagnosis of ILDs, while avoiding the dangerous, expensive and time-consuming invasive biopsies. The appropriate interpretation of the available radiological data combined with clinical/biochemical information can provide a reliable diagnosis, able to improve the diagnostic accuracy of the radiologists.

In this thesis, we introduce two convolutional neural networks particularly designed for ILDs and a training scheme that employs knowledge transfer from the similar domain of general texture classification for performance enhancement. Moreover, we investigate the clinical relevance of breathing information for disease classification. The breathing information is quantified as a deformation field between inhale-exhale lung images using a novel 3D convolutional neural network for medical image registration. Finally, we design and evaluate the final end-to-end computational system for ILD classification using lung anatomy segmentation algorithms from the literature and the proposed ILD quantification neural networks. Deep learning approaches have been mostly investigated for all the aforementioned steps, while the results demonstrated their potential in analyzing lung images.

Keywords – Interstitial Lung Diseases, Diffuse Lung Diseases, High Resolution Tomography, Magnetic Resonance Imaging, Computer Aided Diagnosis, Machine Learning, Deep Convolutional Neural Networks, Semantic Segmentation, Transfer Learning, Medical Image Registration, Breathing Quantification, Radiomics

Acknowledgements

There are many sayings about the PhD life most of which are somewhat accurate. If I had to describe it myself, I would go with: continuous and unending literature studying, a lot of coding and writing, personal commitment, stressful long submission nights, deadlines, reviewer #2, personal evolvement and communication with people. Since the first steps in this PhD path friendly colleagues seem that they want to make sure that you fully understand what you are going to encounter. It is however not until you experience it yourself that you understand what they meant and eventually become one of them. In one way or another each and everyone that walks this path will end up a different person on the other side. Distilling it down however, PhD life is mostly about overcoming obstacles that are densely placed on so many aspects of one's life. Having said that, I would like to point out that this thesis is not only the result of four years of hard work but also the result of constant support from all the people around me. I would therefore like to dedicate the following paragraphs to these unseen heroes pushing me over these obstacles as a way to express my gratitude.

First of all, nothing of these would be possible if some years ago Prof. Stavroula Mougiakakou had not answered my first email. I am really grateful that she gave me this opportunity in the first place, trusted my judgment in so many occasions and supported me with valuable advice in all kind of matters until now. She was a great supervisor that sincerely cared about me. Moreover, being a simple engineer, knowing nothing about interstitial lung diseases, I feel really glad that Prof. Dr. Med. Andreas Christe had the patience to explain to me all the details and procedures about ILDs. Also, most – if not all – of the results, are because Andreas together with PD Dr. med. Lukas Ebner spend a lot of time annotating HRCTs with ground truth polygons. I know how painful this can be and I am really thankful for their efforts. I would also like to thank all my colleagues in the lab for all the conversations, the jokes, the drinks and food we shared together. I would like however to express my gratitude in particular to Marios for being such an awesome postdoc and with whom we spent most of the long submission nights together, discussed all kinds of crazy research ideas and had so much fun along the way.

One of the memorable part of my PhD path was the collaboration with Prof. Nikos Paragios and my PhD visit in Centre de Vision Numérique (CVN) in Centrale Supélec in Paris. I met Prof. Paragios in Paris after he agreed to serve as a co-advisor for this thesis in a summer school he organized in 2015. At the beginning of 2018 I spend four months in Paris where I was a part of CVN. I am grateful for this chance and all the positive reinforcement and great advice. Moreover, I would like to especially thank Mihir and Maria for being so nice with me and making me feel like home from the very beginning. To conclude, I would like to add that I was really lucky during this visit in Paris to stay in Fondation Hellénique where I met so many great people which I would like to thank for all the in depth late night discussions on so many topics.

Finally, I will be forever grateful to my dear family, amazing friends and of course my caring life partner for so many reasons that I could not even express here in writing. You know who you are. Thank you for bearing with me all these years that passed, and for the ones to come.

Contents

List of Abbreviations

AI	Artificial Intelligence
AMFM	Adaptive Multiple Features Method
ASPP	Àtrous Spatial Pyramid Pooling
ATS	American Thoracic Society
AUC	Area Under the Curve
BN	Batch Normalization
CAD	Computer Aided Diagnosis
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CV	Cross Validation
DL	Deep Learning
DWM	Discrete Wavelet Metric
ERS	European Respiratory Society
GLCM	Gray-Level Co-occurrence Matrix
GPU	Graphical Processing Unit
HRCT	High resolution Computed Tomography
HU	Hounsfield Units
IIP	Idiopathic Interstitial Pneumonia
ILD	Interstitial Lung Diseases
IPF	Idiopathic Pulmonary Fibrosis
LBP	Local Binary Patterns
LD	Linear Discriminant
MI	Mutual Information
ML	Machine Learning
MLP	MultiLayer Perceptron
MRI	Magnetic Resonance Imaging
MSE	Mean Square Error
MTL	Multi-Task Learning
NCC	Normalized Cross Correlation
NSIP	Non-Specific Interstitial Pneumonia
PACS	Picture Archiving and Communication System
RBF	Radial Basis Function
RBM	Restricted Boltzmann Machine
RLM	Run Length Matrix
ROC	Receiver Operating Characteristic
ROI	Region Of Interest
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
UTE	Ultra-short Time of Echo

Dedicated to my family.

Chapter 1

Introduction

1.1 Interstitial Lung Diseases

The term interstitial lung disease (ILD) refers to a group of more than 200 chronic lung disorders characterized by inflammation of the lung tissue, which often leads to scarring - usually referred to as pulmonary fibrosis. Fibrosis may progressively cause lung stiffness, reducing the ability of the air sacs to capture and carry oxygen into the bloodstream and eventually leads to permanent loss of the ability to breathe. ILDs account for 15 percent of all cases seen by pulmonologists [1] and can be caused by autoimmune diseases, genetic abnormalities, infections and long-term exposures to hazardous materials. However, the cause of ILDs is mostly unknown and the lung manifestations are described as idiopathic interstitial pneumonia (IIP). In 2002, an international multidisciplinary consensus conference, including the American Thoracic Society (ATS) and the European Respiratory Society (ERS), proposed a classification for ILDs [2], in order to establish a uniform set of definitions and criteria for their diagnosis. These guidelines were updated and amended for diagnostics in 2013 [3] and 2017 [4].

The diagnosis of an ILD involves, questioning the patients about their clinical history, a thorough physical examination, pulmonary function testing, a chest X-ray and a high resolution computed tomography (HRCT) scan. HRCT is generally considered to be the most appropriate protocol, due to the specific radiation attenuation properties of the lung tissue. The imaging data are interpreted by assessing the extent and distribution of the various ILD pathological tissue types in the chest CT scan. Typical ILD pathological tissue types in CT images are: reticulation, honeycombing, ground glass opacity, consolidation and micronodules. A few examples of the different tissue types are presented in Figure 1.1.

Although ILDs are a histologically heterogeneous group of diseases, they mostly have rather similar clinical manifestations with each other, or even with different lung disorders, so that differential diagnosis is fairly difficult even for experienced physicians. This inherent property of ILDs, as well as the lack of strict clinical guidelines and the large quantity of radiological data that radiologists have to scrutinize, explain the low diagnostic accuracy and the high inter- and intra- observer variability, which has been reported to be as great as 50% [5]. In ambiguous cases, additional invasive procedures are required, such as bronchoalveolar lavage and histological confirmation. However, performing a surgical biopsy exposes the patient to a number of risks and increases the healthcare costs, while even such methods do not always provide a reliable diagnosis.

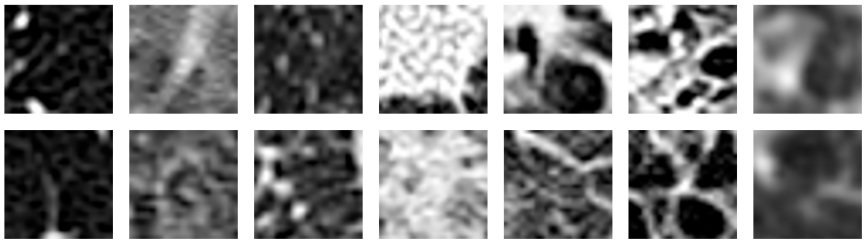


FIGURE 1.1: Examples of healthy tissue and typical ILD patterns from left to right: healthy, ground glass opacity, micronodules, consolidation, reticulation, honeycombing, combination of GGO and reticulation.

1.2 Computer Aided Diagnosis Support Systems

In order to minimize the dangerous and sometimes unreliable histological biopsies, much research has been conducted on computer aided diagnosis systems (CAD) which could assist radiologists and increase their diagnostic accuracy. A CAD system for lung HRCT scan assessment typically consists of three stages: (a) lung anatomy segmentation, (b) lung tissue characterization and (c) differential diagnosis. The first stage, refers to the identification of the lung border, the separation of the lobes and the detection and removal of the bronchovascular tree. The second stage, includes detection and recognition of the different tissue abnormalities (HRCT texture patterns). Finally, the third stage combines the previous results so as to estimate the extend and distribution of pathological tissue within the lung and outputs a probability based differential diagnosis. Such systems mostly utilize machine learning approaches in order to perform each task and use annotated data for training. In the following sections, the most important research studies for each of the aforementioned stages are presented.

1.2.1 Anatomy Segmentation

Lung tissue normally presents substantially lower density than its surrounding tissues, resulting in a large contrast in Hounsfield Units (HU) within thorax CT images. Therefore, many of the conventional lung segmentation methods rely on simple intensity thresholding techniques followed by morphological operations and connected component analysis for the refinement of the results [6], [7]. In the cases however where high intensity pathological tissue (e.g. nodules) is manifested near the borders of the lung cavity, simple morphological operations and filling techniques are not sometimes sufficient to obtain an accurate segmentation of lung field. For this reason, other geometric approaches were utilized, such us the "rolling ball" operation [8]. More recently, after the enchanted capabilities of the multi-detector CT scanners that can produce volumetric scans, 3D based approaches were also studied [9], [10]. These simple intensity methods however, become unreliable in cases containing pathologies with high density such as ILDs. The high density of the ILD patterns that corresponds to high attenuation values, along with their often peripheral and basal manifestations, can cause severe under-segmentation problems propagating the error to all subsequent steps of the CAD system. A typical approach for the segmentation of the lung fields therefore involves a couple of supplementary steps for the correction of the final result. Korfiatis et al. [11] applied k-means clustering followed by a filling operation to obtain an initial lung field estimation and then used iterative support vector machine (SVM) classification of border pixels based on gray level and wavelet coefficient statistics features. In order to exploit all available information, while avoiding the strong dependency from image intensity and special texture characteristics, additional thorax anatomical features were proposed in some works. Hua et al. [12] applied a graph search algorithm with a cost function that combines anatomical information, image intensity, and image gradient. Prasad et al. [13] proposed an adaptive thresholding technique that exploits the fact that the curvature of the ribs and the curvature of the lung boundary are closely matched.

1.2.2 Tissue Characterization

The term lung disease quantification includes the detection and recognition of the various ILD pathologies, as well as the identification of their extent in the lung. Since ILDs are generally manifested as texture alterations of the lung parenchyma, most of the proposed systems employ texture classification schemes on local regions of interest (ROI). A typical lung disease quantification scheme takes as input a local 2D or 3D ROI which is described by a chosen feature set and uses an artificial intelligent (AI) system to classify it. The first proposed systems used handcrafted texture features, in order to describe the ROIs, such as first order statistics, gray level co-occurrence matrices, run-length matrices, and fractal analysis [14]. Other systems, utilized filter banks [15], [16], morphological operations [6], wavelet transformations [17], and local binary patterns [18]. More recently, researchers proposed the use of feature sets learned from data, which are able to adapt to a given problem. Most of these methods rely on unsupervised techniques, such as bag of features [19], [20]

and sparse representation models [21], [22]. Restricted Boltzmann machines (RBM) have also been used [23] to learn multiscale filters with their responses being used as features. Once the feature vector of a ROI has been calculated, it is fed to a classifier that is trained to discriminate between the patterns. Many different approaches have been proposed for classification, including linear discriminant analysis [15] and Bayesian classifiers [14], k-nearest neighbors [11], [18], multi-layered perceptrons (MLP) [6], random forests [16], and support vector machines (SVM) [19], [24]. Some attempts have also been recently made to use deep learning (DL) techniques and especially convolutional neural networks (CNNs), after their impressive performance in large scale color image classification [25]. Unlike other feature learning methods that build data representation models in an unsupervised manner, CNNs learn features and train an ANN classifier at the same time, by minimizing the classification error. Although the term DL implies the use of many consecutive learning layers, the first attempts on lung CT images adopted shallow architectures. In [26], a modified RBM was used for both feature extraction and classification of lung tissue, incorporating some features of CNNs. Weight sharing was used among the hidden neurons, which were densely connected to label (output) neurons, while the whole network was trained in a supervised manner, using contrastive divergence and gradient descent. In [27], the authors designed a CNN with one convolutional layer and three dense layers and trained it from scratch. However, the shallow architecture of the network cannot leverage the descriptive ability of deep CNNs. The pre-trained deep CNN of [25] (AlexNet) was used in [28] to classify whole lung slices after fine-tuning with lung CT data. AlexNet was designed to classify natural color images with input size 224×224, so the authors had to resize the images and artificially generate three channels by applying different HU windows. However, the substantial differences in the domains of general color images and medical images raise doubts regarding the transfer of knowledge between them, while classifying whole slices may only provide very rough quantification of the disease.

1.2.3 Differential Diagnosis

Despite the extensive research that has been undertaken for the ILD quantification on CT images, there has not yet been proposed a system able to suggest automatically a final diagnostic decision for a case. Van Ginneken et al. [29] proposed an automatic method for the segmentation of lung fields into 42 regions followed by a classification step which assigns to each region a confidence value for being abnormal. The product of the individual confidence values provides a global diagnosis on the abnormality of the whole lung. Zheng et al. [30] proposed a system that segments lung areas, identifies suspicious volumetric ILD lesions, computes five global features for each of them (size, contrast, average local pixel value fluctuation, mean of stochastic fractal dimension, and geometric fractal dimension) and classifies the corresponding case into one of three categories of severity (mild, moderate, and severe) by using a distance-weighted k-NN algorithm. Fukushima et al. [31] proposed the use of an ANN that combines 10 clinical parameters with 23 HRCT features in order to provide a final differential diagnosis. However, the used HRCT features were not computed automatically, but rated manually by radiologists. Wang et al. [32] after classifying the VOIs of a case into normal/abnormal using run length matrix (RLM) and Gray-Level Co-Occurrence Matrix (GLCM) features, they used a simple rule to classify the case itself: If the number of VOIs reported as abnormal in a case is greater than a specified threshold, the case was considered as abnormal; otherwise it was considered as normal.

1.3 Hypothesis and Aims

Interstitial lung diseases is a group of rare, chronic, progressive diseases that affect the lungs and are difficult to diagnose. Without the appropriate treatment, the life expectancy is only two to five years after the diagnosis. Early diagnosis is of great importance, and therefore enhancing the diagnostic accuracy of the involved physicians can have a very large impact on the patients. The main scope of this PhD thesis is to investigate novel machine learning approaches for the automatic quantification and assessment of medical images from patients suffering with interstitial lung diseases, under the following hypothesis:

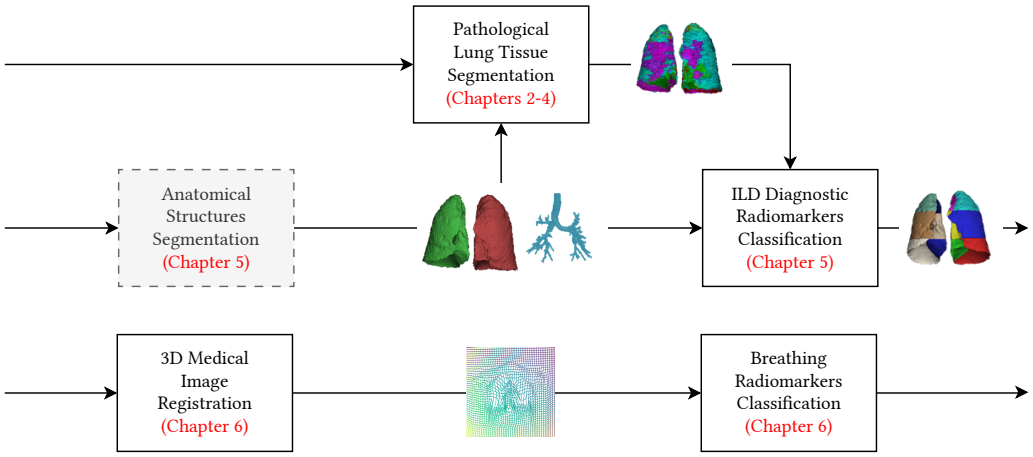


FIGURE 1.2: Overview of Thesis. Diagrams flow from left (inputs) to right (outputs). Top diagram presents the ILD diagnosis support and bottom diagram shows the breathing radiomarkers extraction and classification. Grayed out block implemented from literature.

Hypothesis: *A robust, accurate and fast automatic detection of pathological lung tissue and quantification of medical image findings, could enhance the diagnostic performance of radiologists and could be used for defining image radio-markers and diagnostic endpoints.*

This PhD thesis lies between the fields of machine learning and medical diagnosis. Throughout the years, more and more studies attempt to bridge the gap between the two fields. The aim is the design, development and evaluation of machine learning algorithms for the automatic quantification of clinically relevant information retrieved from chest HRCT and magnetic resonance imaging (MRI) scans, so that they can be used for assisting the diagnostic procedure.

1.4 Structure of the Thesis

In the following chapters, the main contributions of this PhD thesis are presented. In each chapter, a short introduction on the particular field of study is given along with a detailed description of the utilized data. A schematic presentation of the different contributions of this PhD Thesis in given in Figure 1.2. In detail:

- Chapter 2: We propose and evaluate a CNN, designed for the classification of pathological lung tissue image patches. A comparative analysis proved the effectiveness of the proposed CNN against established texture classification methods from the literature, in a challenging dataset with 120 unique HRCT scans. The classification performance ($\sim$ 85.5%) demonstrated the potential of CNNs in analyzing lung patterns.
- Chapter 3: Due to the sparsity of annotated medical data, we investigate the feasibility of utilizing knowledge from other texture classification tasks. Thus, in Chapter 3, we present an improved method for training the previously proposed network (Chapter 2) by transferring knowledge from the similar domain of general texture classification. Six publicly available texture databases are used to pre-train networks with the proposed architecture, which are then fine-tuned on the lung tissue data. The resulting CNNs are combined in an ensemble and their fused knowledge is compressed back to a network with the original architecture. The proposed training approach resulted in an absolute increase of about 2% (i.e $\sim$ 87.5%) in the performance of the already proposed CNN with no other modifications.

- Chapter 4: Image patch classification combined with sliding window schemes, such the ones presented in Chapters 2 and 3, could be highly inefficient as local image features have to be recalculated multiple times for adjacent positions of the input window. Luckily, the convolutional layers (with the appropriate padding) in a typical CNN produce feature maps that maintain spatial correspondence with the input image. We therefore propose, in Chapter 4, the use of a deep purely CNN for the semantic segmentation of ILD pathological tissue from whole HRCT slices, as the basic component of a CAD system for ILDs. The training was performed in an end-to-end and semi-supervised fashion, utilizing both labeled and non-labeled image regions. The experimental results show significant performance and time improvements with respect to the state of the art.

- Chapter 5: We introduce and evaluate an end-to-end CAD system for the automatic classification of HRCT images into four radiological diagnostic categories. The proposed CAD system consists of a sequential pipeline in which at first, the anatomical structures of the lung are segmented, then the pathological lung tissue is identified and finally by combining these information a final radiological diagnosis is reached using a random forest classifier. The experimental results show the potential of utilizing a CAD system for this task, while also sets a path for further development and investigation.

- Chapter 6: We propose a novel CNN architecture that couples linear and deformable registration within a unified architecture endowed with near real-time performance. We evaluate the performance of our network on the challenging problem of MRI lung registration and demonstrate superior performance with respect to state of the art elastic registration methods. The proposed deformation (between inspiration and expiration) was considered within a clinically relevant task of ILDs classification and showed promising results.

- Chapter 7: We conclude the thesis with an general discussion of the topics covered and with an outlook of future perspectives.

Chapter 2

Lung Tissue Classification

This chapter is a modified version of:
M. Anthimopoulos*, S. Christodoulidis*, L. Ebner, A. Christe and S. Mougiakakou, *"Lung Pattern Classification for Interstitial Lung Diseases Using a Deep Convolutional Neural Network,"* in IEEE Transactions on Medical Imaging, vol. 35, no. 5, pp. 1207-1216, May 2016.
DOI: 10.1109/TMI.2016.2535865

*This study was a highly collaborative effort of M. Anthimopoulos and S. Christodoulidis, who share the first authorship. All figures and the experiments were prepared and executed by S. Christodoulidis while the text was writer by S. Christodoulidis and M.Anthimopoulos. The technical research directions were chosen after long discussions between M. Anthimopoulos, S. Christodoulidis and S. Mougiakakou while the medical research directions were decided by A. Christe and L. Ebner.

Automated tissue characterization is one of the most crucial components of a computer aided diagnosis (CAD) system for interstitial lung diseases (ILDs). Although much research has been conducted in this field, the problem remains challenging. Deep learning techniques have recently achieved impressive results in a variety of computer vision problems, raising expectations that they might be applied in other domains, such as medical image analysis. In this chapter, we propose and evaluate a convolutional neural network (CNN), designed for the classification of ILD patterns. The proposed network consists of 5 convolutional layers with 2×2 kernels and LeakyReLU activations, followed by average pooling with size equal to the size of the final feature maps and three dense layers. The last dense layer has 7 outputs, equivalent to the classes considered: healthy, ground glass opacity (GGO), micronodules, consolidation, reticulation, honeycombing and a combination of GGO/reticulation. To train and evaluate the CNN, we used a dataset of 14696 image patches, derived by 120 CT scans from different scanners and hospitals. To the best of our knowledge, this is the first deep CNN designed for the specific problem. A comparative analysis proved the effectiveness of the proposed CNN against previous methods in a challenging dataset. The classification performance ($\sim 85.5\%$) demonstrated the potential of CNNs in analyzing lung patterns. Future work includes, extending the CNN to three-dimensional data provided by CT volume scans and integrating the proposed method into a CAD system that aims to provide differential diagnosis for ILDs as a supportive tool for radiologists.

2.1 Field of Study

2.1.1 Convolutional Neural Networks

CNNs are feed-forward ANN inspired by biological processes and designed to recognize patterns directly from pixel images (or other signals), by incorporating both feature extraction and classification. A typical CNN involves four types of layers: convolutional, activation, pooling and fully-connected (or dense) layers. A convolutional layer is characterized by sparse local connectivity and weight sharing. Each neuron of the layer is only connected to a small local area of the input, which resemble the receptive field in the human visual system. Different neurons respond to different local areas of the input, which overlap with each other to obtain a better representation of the image. In addition, the neurons of a convolutional layer are grouped in feature maps sharing the same weights, so the entire procedure becomes equivalent to convolution, with the shared weights being the filters for each map.

Weight sharing drastically reduces the number of parameters of the network and hence increases efficiency and prevents overfitting. Convolutional layers are often followed by a non-linear activation layer, in order to capture more complex properties of the input signal. Pooling layers are also used to subsample the previous layer, by aggregating small rectangular subsets of values. Max or average pooling is usually applied by replacing the input values with the maximum or the average value, respectively. The pooling layers reduce the sensitivity of the output to small input shifts. Finally, one or more dense layers are put in place, each followed by an activation layer, which produce the classification result. The training of CNNs is performed similarly to that of other ANNs, by minimizing a loss function using gradient descent based methods and back propagation of the error.

Although the concept of CNNs has existed for decades, training such deep networks with multiple stacked layers was achieved only recently. This is mainly due to their extensive parallelization properties, which have been coupled with massively parallel GPUs, the huge amounts of available data, and several design tricks, such as the rectified linear activation units (ReLU). In 2012, Krizhevsky et al. [25] won the ImageNet Large-Scale Visual Recognition Challenge, convincingly outperforming the competition on a challenging dataset with 1000 classes and 1.2 million images. The proposed deep CNN, also known as AlexNet, consists of five convolutional layers with ReLU activations, some of which are followed by max-pooling layers, and three dense layers with a final 1000-way softmax. The network was trained with stochastic gradient descent (SGD) with a momentum term, maximizing the multinomial logistic regression objective. Deep architectures permit learning of data representations in multiple levels of semantic abstraction, so even high-level visual structures like cars or faces can be recognized in the last layers by combining low-level features of the first, such as edges. Nevertheless, designing a deep CNN for a specific problem is not trivial, since a large number of mutually dependent parameter values and algorithmic choices have to be chosen. Although much research has been conducted in recent years on deep CNNs for color image classification, very little has been done on the problems of texture recognition and medical image analysis.

2.1.2 Contribution

In this study, we propose a deep CNN for the classification of ILD patterns that exploits the outstanding descriptive capability of deep neural networks. The method has been evaluated on a dataset of 120 cases from two hospitals and the results confirm its superiority compared to the state of the art. To the best of our knowledge, this is the first time a deep CNN has been designed and trained for lung tissue characterization. Finally, we provide empirical rules and principles on the design of CNN architectures for similar texture classification problems.

2.2 Materials and Methods

In this section, we first describe the dataset used in the study, followed by the proposed CNN. The definition of the input data and desired outputs prior to the actual methods provides a better definition of the problem and thus a better understanding of the methods.

2.2.1 Data

The dataset used for training and evaluating the proposed method was made using two databases of ILD CT scans from two different Swiss university hospitals:

The first is the publicly available multimedia database of ILDs from the University Hospital of Geneva [33], which consists of 109 HRCT scans of different ILD cases with 512×512 pixels per slice. Manual annotations for 17 different lung patterns are also provided, along with clinical parameters from patients with histologically proven diagnoses of ILDs.

The second database was provided by the Bern University Hospital, "Inselspital", and consists of 26 HRCT scans of ILD cases with resolution 512×512.

The scans were produced by different CT scanners with slightly different pixel spacing so a preprocessing step was applied, which rescaled all scans to match a specific spacing value

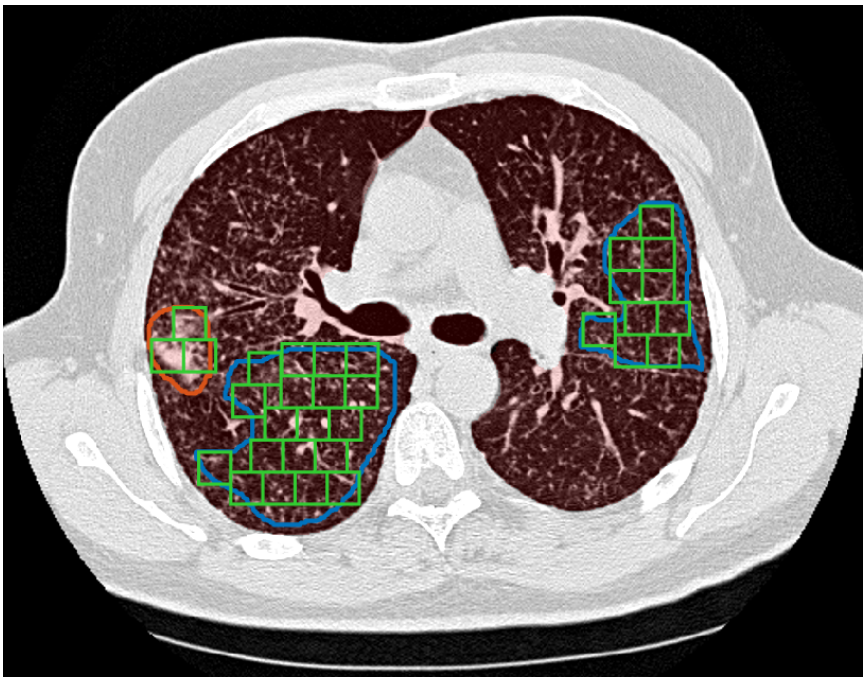


FIGURE 2.1: Example of generating image patches through the annotations of a CT slice. The lung field is displayed with transparent red. The polygons are the ground truth areas with considered pathologies. The patches have 100% overlap with the lung, at least 80% overlap with the ground truth and 0% overlap with each other.

(i.e., 0.4 mm). However, the use of different reconstruction kernels by the scanners, still remains an open issue that complicates the problem even further. The image intensity values were cropped within the window [-1000, 200] in HU and mapped to [0, 1]. Experienced radiologists from the "Inselspital" annotated (or re-annotated) both databases by manually drawing polygons around the six most relevant ILD patterns, namely GGO, reticulation, consolidation, micronodules, honeycombing and a combination of GGO and reticulation. Healthy tissue was also added, leading to 7 classes. The annotation focused on typical instances of the considered ILD patterns, excluding ambiguous tissue areas that even experienced radiologists find difficult to classify. Hence, tissue outside the polygons may belong to any pattern, including that considered. Moreover, the annotators tried to avoid the bronchovascular tree which (in a complete CAD system) should be segmented and removed, before applying the fixed-scale classifier. Annotation of the lung fields was also performed for all scans.

The considered classes appeared in the annotations of 94 out of the 109 scans of the Geneva database, to which the 26 cases from "Inselspital" were added, giving a total of 120 cases. On the basis of the ground truth polygons of these cases, we extracted in total 14696 non-overlapping image patches of size 32×32, unequally distributed across the 7 classes. Figure 2.1 presents an example of how patches are generated through the annotations of a CT slice. For each pattern, Table I provides the number of ground truth polygons, the average and standard deviation of their area, the number of cases in which it was annotated and the number of extracted patches. The healthy pattern was only annotated in 8 cases, which however proved to be enough, since its texture does not present large deviations. It has to be noted that one case may contain multiple types of pathologies, so the sum of cases in Table 2.1 is larger than 120. The patches are entirely included in the lung field and have an overlap with the ground truth polygons of at least 80%. For each class, 150 patches were randomly selected for the test and 150 for the validation set. The choice of 150 was made based on the patch number of the rarest class (i.e., honeycombing) leaving about 50% of the patches for training. On the remaining patches, data augmentation was employed in order to maximize the number of training samples and equalize, at the same time, the samples'

TABLE 2.1: Statistics of the database. (H: healthy, GGO: ground glass opacity, MN: micronodules, cons: consolidation, ret: reticulation, HC: honeycombing)

	H	GGO	MN	Cons	Ret	HC	Ret+GGO
#Polygons	105	823	317	1129	870	692	1593
Avg Area (10^3px)	39.8	11.7	58.4	9.5	11.7	13.7	24.1
Std Area (10^3px)	21.5	11.8	52.7	7.5	14.1	10.6	19.6
#Cases	8	44	19	25	38	22	55
#Patches	1142	1185	3192	2823	1056	613	4685

distribution across the classes. Data augmentation has often been employed in image classification, in order to increase the amount of training data and prevent over-fitting [25]. To this end, 15 label-preserving transformations were used, such as flip and rotation, as well as the combinations of the two. For each class, the necessary number of augmented samples was randomly selected, so all classes would reach the training set size of the rarest class, i.e., 5008, leading to 35056 equally distributed training patches.

2.2.2 Proposed CNN

In order to decide on the optimal architecture and configuration of a CNN, one should first comprehend the nature of the problem considered – in this case – the classification of ILD patterns. Unlike arbitrary objects in color images, which involve complex, high-level structures with specific orientation, ILD patterns in CT images are characterized by local textural features. Although texture is an intuitively easy concept for humans to perceive, formulating a formal definition is not trivial, which is the reason for the many available definitions in the literature [34]. Here, we define texture as a stochastic repetition of a few structures (textons) with relatively small size, compared to the whole region. Image convolution highlights small structures that resemble the convolution kernel throughout an image region, and in this way the analysis of filter bank responses has been successfully used in many texture analysis applications. This encourages the use of CNNs to recognize texture by identifying the optimal eproblem-specific kernels; however some key aspects stemming from our definition of texture have to considered: (i) The total receptive field of each convolutional neuron with respect to the input (i.e., the total area of the original input "seen" by a convolutional neuron) should not be larger than the characteristic local structures of texture, otherwise non-local information will be captured, which is irrelevant to the specific texture, (ii) since texture is characterized by fine grained low-level features, no pooling should be carried out between the convolutional layers, in order to prevent loss of information, (iii) each feature map outputted by the last convolutional layer should result in one single feature after pooling, in order to gain some invariance to spatial transformations like flip and rotation. Unlike color pictures that usually have high-level geometrical structure (e.g., the sky is up), a texture patch should still be a valid sample of the same class when flipped or rotated.

Architecture

On the basis of these principles, we designed the network presented in Figure 2.2. The input of the network is a 32×32 image patch, which is convolved by a series of 5 convolutional layers. The size of the kernels in each layer was chosen to be minimal, i.e., 2×2. The use of small kernels that lead to very deep networks was proposed in the VGG-net [35], which was ranked at the top of ILSVRC 2014 challenge by employing 3×3 kernels and up to 16 convolutional layers. Here, we go one step further by shrinking the kernel size even more to involve more non-linear activations, while keeping the total receptive field small enough (6×6) to capture only the relevant local structure of texture. Each layer has a number of kernels proportional to the receptive field of its neurons, so it can handle the increasing complexity of the described structures. The size of the rectangular receptive field is 2×2 for the first layer and is increased by 1 in each dimension, for each layer added, leading

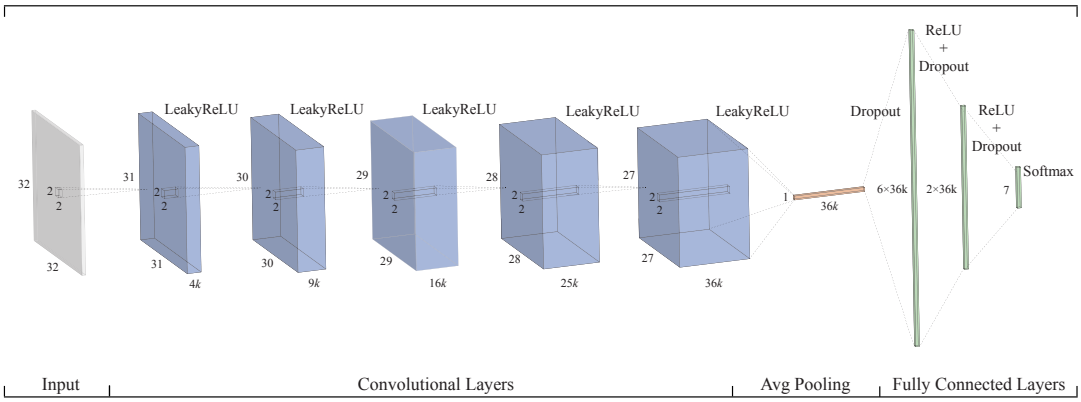


FIGURE 2.2: Architecture of the proposed CNN for lung pattern classification. The value of parameter k was set to 4.

to an area of $(L+1)^2$ for the Lth layer. Hence, the number of kernels we use for the Lth layer is $k(L+1)^2$, where the parameter k depends on the complexity of the input data and was set to 4 after relevant experiments. An average pooling layer follows, with size equal to the output of the last convolutional layer (i.e., 27×27). The resulting features, which are equal to the number of features maps of the last layer i.e., $f = 36k$, are fed to a series of 3 dense layers with sizes $6f$, $2f$ and 7, since 7 is the number of classes considered. The use of large dense layers accelerated convergence, while the problem of overfitting was solved by adding a dropout layer before each dense layer. Dropout can be seen as a form of bagging; it randomly sets a fraction of units to 0, at each training update, and thus prevents hidden units from relying on specific inputs [36].

Activations

It is well-known that the choice of the activation function significantly affects the speed of convergence. The use of the ReLU function f(x)=max(0,x) has been proven to speed up the training process many times compared to the classic sigmoid alternative. In this study, we also noticed that convolutional activations have a strong influence on the descriptive ability of the network. Driven by this observation and after experimenting with different rectified activations, we propose the use of LeakyReLU [37], a variant of ReLU, for activating every convolutional layer. Unlike ReLU, which totally suppresses negative values, leaky ReLU assigns a non-zero slope, thus allowing a small gradient when the unit is not active ((1)).

$$f(x) = \begin{cases} x, & x > 0 \\ \alpha x, & \text{else} \end{cases} \tag{2.1}$$

where α is a manually set coefficient.

LeakyReLU was proposed as a solution to the "dying ReLU" problem, i.e., the tendency of ReLU to keep a neuron constantly inactive as may happen after a large gradient update. Although a very low negative slope coefficient (i.e., α=0.01) was originally proposed, here we increase its value to 0.3, which considerably improves performance. Similar observations have also been reported in other studies [38]. A very leaky ReLU seems to be more resilient to overfitting when applied to convolutional layers, although the exact mechanism causing this behavior has to be further studied. For the dense part of the network, the standard ReLU activation was used for the first two layers and softmax on the last layer, to squash the 7-dimensional output into a categorical probability distribution.

Training Method

The training of an ANN can be viewed as a combination of two components, a loss function or training objective, and an optimization algorithm that minimizes this function. In this study, we use the Adam optimizer [39] to minimize the categorical cross entropy. The cross entropy represents the dissimilarity of the approximated output distribution (after softmax)

from the true distribution of labels. Adam is a first-order gradient-based algorithm, designed for the optimization of stochastic objective functions with adaptive weight updates based on lower-order moments. Three parameters are associated with Adam: one is the learning rate and the other two are exponential decay rates for the moving averages of the gradient and the squared gradient. After relevant experiments, we left the parameters to their default values namely, learning rate equal to 0.001 and the rest 0.9 and 0.999, respectively. The initialization of the convolutional layers was performed using orthogonal matrices multiplied with a scaling parameter equal to 1.1, while a uniform distribution was utilized for the dense layers, scaled by a factor proportional to the square root of the layer's number of inputs [40]. The weight updates are performed in mini-batches and the number of samples per batch was set to 128. The training ends when the network does not significantly improve its performance on the validation set for a predefined number of epochs. This number is set to 200 and the performance is assessed in terms of average f-score (Favg) over the different classes ((2)) (see Section IV). An improvement is considered significant if the relative increase in performance is at least 0.5%.

2.3 Experimental Setup and Results

This section focuses on the presentation and discussion of the results. Before that, we describe the experimental setup namely, the chosen evaluation strategy and some details on the implementation of the methods.

2.3.1 Experimental Setup

Evaluation

The evaluation of the different ILD patch classification approaches is based on a train-validation-test scheme. The actual training of the methods was carried-out on the training set, while the validation set was used for fine tuning the hyper-parameters; the overall performance of each system was assessed on the test set. As principle evaluation measure, we used the average F-score over the different classes (2), due to its increased sensitivity to imbalances among the classes; the overall accuracy is also computed (3). It has to be noted that the presented performances are not comparable to performances reported in the literature due to the use of different datasets and the consideration of different patterns. However, we trust that the difficulty of a dataset may only affect the absolute performance of methods and not their relative performance rank.

$$F_{avg} = \frac{2}{7} \sum_{c=1}^{7} \frac{\text{recall}_c * \text{precision}_c}{\text{recall}_c + \text{precision}_c} \tag{2.2}$$

where

$$\text{recall}_\text{c} = \frac{\text{samples correctly classified as c}}{\text{samples of class c}} \tag{2.3}$$

$$\text{precision}_\text{c} = \frac{\text{samples correctly classified as c}}{\text{samples classified as c}} \tag{2.4}$$

$$\text{Accuracy} = \frac{\text{correctly classified samples}}{\text{total number of samples}} \tag{2.5}$$

Implementation

The proposed method was implemented in Python[1] using the Keras[2] framework with Theano [41] back-end, while for AlexNet and VGG-Net we used Caffe [42]. Methods which do not involve convolutional networks were coded in python and MATLAB. All experiments were

[1] https://github.com/intact-project/ild-cnn
[2] https://github.com/keras-team/keras

TABLE 2.2: Performance of the CNN for different configurations

Dropout fraction	Pooling type	Pooling percentage	Kernel number multiplier (k)	Number of kernels for Lth layer	Number of conv layers	Kernel size	Input scale factor	Activation function	Testing Favg	# Epochs x Epoch time
0	Avg	100%	4	$k(L+1)^2$	5	2×2	1	LReLU(0.3)	0.7908	$90\times11s$
0.5	Max	100%	4	$k(L+1)^2$	5	2×2	1	LReLU(0.3)	0.8105	$69\times11s$
0.5	Avg	50%	4	$k(L+1)^2$	5	2×2	1	LReLU(0.3)	0.7895	$249\times11s$
0.5	Avg	25%	4	$k(L+1)^2$	5	2×2	1	LReLU(0.3)	0.7452	$286\times12s$
0.5	Avg	100%	4	17	5	2×2	1	LReLU(0.3)	0.8446	$300\times12s$
0.5	Avg	100%	4	36	5	2×2	1	LReLU(0.3)	0.8508	$386\times32s$
0.5	Avg	100%	3	$k(L+1)^2$	5	2×2	1	LReLU(0.3)	0.8266	$427\times7s$
0.5	Avg	100%	5	$k(L+1)^2$	5	2×2	1	LReLU(0.3)	0.8425	$362\times14s$
0.5	Avg	100%	4	$k(L+1)^2$	7	2×2	1	LReLU(0.3)	0.8432	$295\times23s$
0.5	Avg	100%	4	$k(L+1)^2$	6	2×2	1	LReLU(0.3)	0.8559	$215\times18s$
0.5	Avg	100%	4	$k(L+1)^2$	4	2×2	1	LReLU(0.3)	0.8443	$372\times6s$
0.5	Avg	100%	4	$k(L+1)^2$	5	2×2	1.5	LReLU(0.3)	0.8223	$196\times21s$
0.5	Avg	100%	4	$k(L+1)^2$	5	3×3	1.5	LReLU(0.3)	0.8390	$328\times260s$
0.5	Avg	100%	4	$k(L+1)^2$	5	3×3	1	LReLU(0.3)	0.8147	$193\times67s$
0.5	Avg	100%	4	$k(L+1)^2$	5	2×2	1	ReLU	0.7871	$90\times11s$
0.5	Avg	100%	4	$k(L+1)^2$	5	2×2	1	LReLU(0.01)d	0.8094	$110\times12s$
0.5	**Avg**	**100%**	**4**	$\mathbf{k(L+1)^2}$	**5**	$\mathbf{2\times2}$	**1**	**LReLU(0.3)**	**0.8547**	$\mathbf{386\times12s}$

performed under a Linux OS on a machine with CPU Intel Core i7-5960X @ 3.50 GHz, GPU NVIDIA GeForce Titan X, and 128 GB of RAM.

2.3.2 Results

This section presents the experimental results and is split into three parts. Firstly, we present a set of experiments that justify the choice of the different components and the tuning of the hyper-parameters. A comparison of the proposed method with previous studies follows and finally, an additional analysis of the system's performance is given.

Tuning of Hyper-Parameters

Here we demonstrate the effect of the most crucial choices for the architecture and the training procedure. Table 2.2 demonstrates the classification performance for different configurations of the network's architecture, as well as the training time needed. The proposed configuration, presented in bold, yielded an F_{avg} of 0.8547. Using the LeakyReLU with the originally proposed parameter, reduces the performance by roughly 5% and the use of standard ReLU by a further 2%. Increasing the size of the kernels to 3×3 also resulted in a drop by 4% in performance, accompanied by a significant increase in the epoch time ($\sim5\times$). The larger kernels increased the total receptive field of the network to 11×11, which proved to be too big for the characteristic local structures of the considered textures. By keeping the 3×3 kernels and increasing the image resolution by 50%, each training epoch became slower by more than $20\times$, but still without reaching the proposed performance. When we just upsampled the input image while using the 2×2 kernels, the result was again significantly inferior to that proposed, since the receptive field relatively to the input size was smaller than optimal. By altering the number of convolutional layers, we can infer that the optimal architecture will have 5-6 layers that correspond to a total receptive field of $6\times6-7\times7$. In this study, we propose the use of 5 convolutional layers, preferring efficiency to a small increase in performance.

To identify the optimal number of kernels, we experimented with the k multiplier. The corresponding results show that 4 is the optimal choice, both in terms of performance and efficiency. A couple of experiments were also conducted to study the effect of using a constant number of kernels in each convolutional layer. Firstly, we chose 17 kernels in order to match the epoch time of the proposed configuration, which resulted in a performance drop of about 1%. With 36 kernels per layer, the results were comparable to that proposed, having though an epoch time almost 3-fold longer. This experiment showed that the choice of the distribution of kernels in the convolutional layers is basically a matter of efficiency and

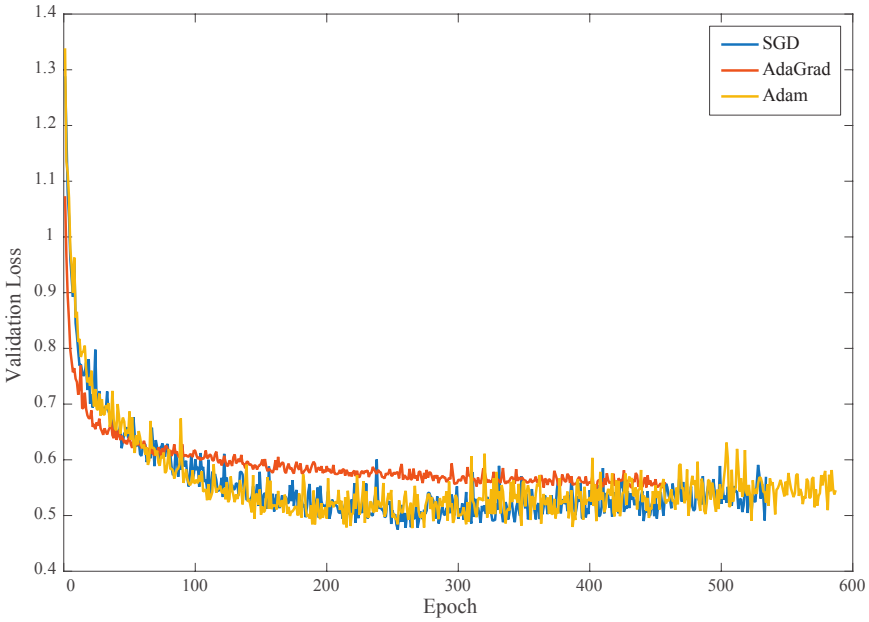


FIGURE 2.3: Comparison of the convergence speed between three optimizers.

does not so drastically affect the accuracy of the system, assuming that a sufficient number of kernels is used.

Changing the size of the pooling layer from 100% of the last feature map to 50% or 25%, resulted in a drop in Favg of more than 6% and 9%, respectively. By splitting the feature map in multiple pooled regions, different features are generated for the different areas of the image, so that the CNN is highly non-invariant to spatial transformations like flip and rotation. In another experiment, max pooling was employed instead of average, yielding a result that was inferior by nearly 4%. Although max pooling is the common choice for most CNNs and proved to be much faster in terms of convergence, in our problem average seems to be more effective. Finally, when we removed the dropout layers, we observed a decline in Favg of more than 6%, an effect obviously due to overfitting.

Table 2.3 demonstrates the effects of using different optimizers and loss functions for training the CNN. The parameters for each optimizer have been tuned accordingly on the validation set. For the SGD we used a learning rate of 0.01 with a momentum of 0.95, while for AdaGrad we used 0.001 learning rate. Minimizing the categorical cross-entropy by the Adam optimizer yielded the best results in a small number of iterations. SGD follows, with about 1% lower performance and AdaGrad with even higher drop in performance of 3%. Finally, we also employed Adam to minimize the mean squared error (MSE), which yielded comparable results.

In Figure 2.3, the convergence of the three different optimizers is illustrated in terms of the validation loss over the epochs. AdaGrad starts with a rapid descent, but soon stops improving probably due to the quickly reduced learning rate. Adam and SGD seem to perform almost equally, but here we chose Adam because of the slightly better performance as shown in Table 2.3 and its stable behavior independently from its parameters.

TABLE 2.3: Performance of the proposed CNN with different training options

Optimizer	Loss Function	F_{avg}	Accuracy	Epoch
SGD	Cross-Entropy	0.8434	0.8428	333
AdaGrad	Cross-Entropy	0.8219	0.8228	257
Adam	MSE	0.8499	0.8523	155
Adam	**Cross-Entropy**	**0.8547**	**0.8561**	**386**

TABLE 2.4: Comparison of the proposed with state-of-the-art methods using handcrafted features

Method	Features	Classifier	F_{avg}	Accuracy
Gangeh [19]	Intensity textons	SVM-RBF	0.7127	0.7152
Sorensen [18]	LBP + histogram	kNN	0.7322	0.7333
Anthimopoulos [16]	Local DCT + histogram	RF	0.7786	0.7809
Proposed	**CNN**		**0.8547**	**0.8561**

Comparison With the State of the Art

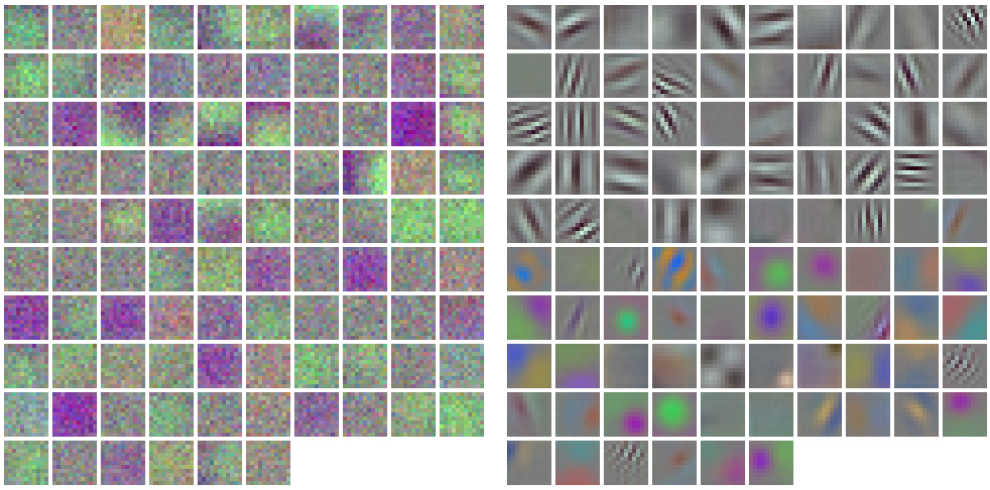
Table 2.4 provides a comparison of the proposed CNN with state-of-the-art methods using handcrafted features and different classifiers. All the methods were implemented by the authors and the parameters for each one were fine-tuned using a trial and error procedure on the validation set. The results prove the superior performance of the proposed scheme that outperformed the rest by 8% to 14%.

Table 2.5 provides a comparison with other CNNs. The first row corresponds to a shallow network with just one convolutional and three dense layers, which constitutes the first CNN-based approach to the problem, to the best of our knowledge. The fairly low results achieved by this network on our dataset, could be due to several reasons: (i) the 16 kernels used for the convolutional layer are not enough to capture the complexity of the problem, (ii) the use of a 2×2 max pooling results in 169 local features per feature map, that describe a high-level spatial distribution not relevant to the problem, and (iii) the shallow architecture prevents the network from learning highly non-linear features. The second CNN we test is the LeNet [43], a network designed for character classification. It has two convolutional layers, each followed by pooling and three dense layers. The first layer uses 6 kernels and the second 16, both with the same size 5×5. The results produced on our dataset are similar to the previous CNN for similar reasons.

Furthermore, we evaluated the performance of the well-known AlexNet [25] and VGG-Net-D [38], two networks much larger and deeper than the previous, with the first having 5 convolutional layers and the second 13. The two networks were designed for the classification of 224×224 color images, so in order to make our data fit, we rescaled the 32×32 patches to 224×224 and generated 3 channels by considering 3 different HU windows according to [35]. First, we tried training the AlexNet from scratch on our data. However, the size of this kind of networks requires very large amounts of data, in order to be trained properly. The achieved accuracy was in the order of 70% and the noisy and low-detailed filters obtained from the first convolutional layer (Figure 2.4a) show that the size, as well as the scale of the network, are too large for our problem. To overcome the problem of insufficient data we fine-tuned the already trained (on ImageNet) AlexNet, which is currently the most common technique for applying it to other problems. The results were improved by about 5% showing that for training large CNNs, the size of the used set can be more important than the type of data. However, by looking at the filters of the first layer (Figure 2.4b) one may notice that the scale of the edges does not match our problem, considering that the 11×11 filters correspond to less than 2×2 in our input image. Finally, we tested the pre-trained (on ImageNet) VGG-Net after fine-tuning it, since training a network with that size from scratch would need even more data than AlexNet. The network achieved an improvement of about

TABLE 2.5: Comparison of the proposed method with other CNNs

Method	F_{avg}	Accuracy
Li [27]	0.6657	0.6705
LeNet [43]	0.6783	0.6790
AlexNet [25]	0.7031	0.7104
Pre-trained AlexNet [25]	0.7582	0.7609
VGG-Net [35]	0.7804	0.7800
Proposed	**0.8547**	**0.8561**



(A) (B)

FIGURE 2.4: Filter of the first layer of alexnet by (A) training from scratch on our data, (B) fine-tuning the pre-trained on imagenet version.

2% compared to AlexNet probably due to the smaller size of kernels that permit the use of more convolutional layers, however the result is still inferior to that proposed.

For a more detailed comparison at different operating points we also performed a receiver operating characteristic (ROC) analysis for AlexNet, AlexNet pre-trained (AlexNetP), VGG-Net, the method by Sorensen et al. [18] and the proposed CNN. Figure 2.5 presents the ROC curves for each of the compared methods and each of the considered classes using a one-vs-all scheme. The average ROC curves over the different classes are presented in the last chart of Figure 2.5 For each ROC, the area under the curve (AUC) was computed and the 95% confidence interval was plotted according to [44]. The comparison showed that the proposed method achieved the highest AUC on each of the 7 classes. To test the statistical significance of the AUC differences, a statistical analysis was performed based on [45] and using 10,000 bootstraps. The results of the analysis confirmed the statistically significant ($p < 0.05$) superior performance of the proposed CNN against all methods, when comparing on the most difficult patterns i.e., consolidation, reticulation, honeycombing and reticulation/GGO. For the rest of the patterns (healthy, GGO and micronodules) the difference between the proposed method and the pre-trained AlexNet was not considered significant ($p = 0.058, 0.445, 0.056$), while for GGO the difference from VGG-Net was also non-significant ($p = 0.271$). Finally, the superiority of the proposed method after averaging over all considered classes was also found to be statistically significant ($p \ll 0.05$). These results are in line with the corresponding ROC curves of Figure 6, where large distance between curves correlates with statistically significant differences.

Furthermore, we conducted an experiment to estimate the efficiency of the different CNNs when used to recognize the pathologies of an entire scan by sliding the fixed-scale classifier on the images. By using the minimal step for sliding the window, i.e., 1, the proposed CNN needed 20 seconds to classify the whole lung area in the 30 slices of an average-sized HRCT scan. The corresponding time needed by AlexNet was 136 and by VGG-Net 160 seconds. By increasing the step to 2, which still produces a sufficiently precise pathology map – the time needed for any method is reduced by a factor of 4.

Concluding, the two tested deep CNNs showed inferior performance mainly because they do not comply with the principles described in Section III-B: (i) their overall receptive field relatively to the input image is larger than needed, (ii) the use of pooling between the convolutional layers results in loss of information, (iii) the use of small size for the last pooling makes the extracted features position dependent. Moreover, other algorithmic choices, like the standard ReLU and the max pooling, may have affected the result, as shown in Table 2.2, as well as the different input size. Finally, apart from the relatively low accuracy, the efficiency of these very large networks could also be an issue for using them in this kind of applications. The slower prediction will multiply the operating time by at least a factor of 7,

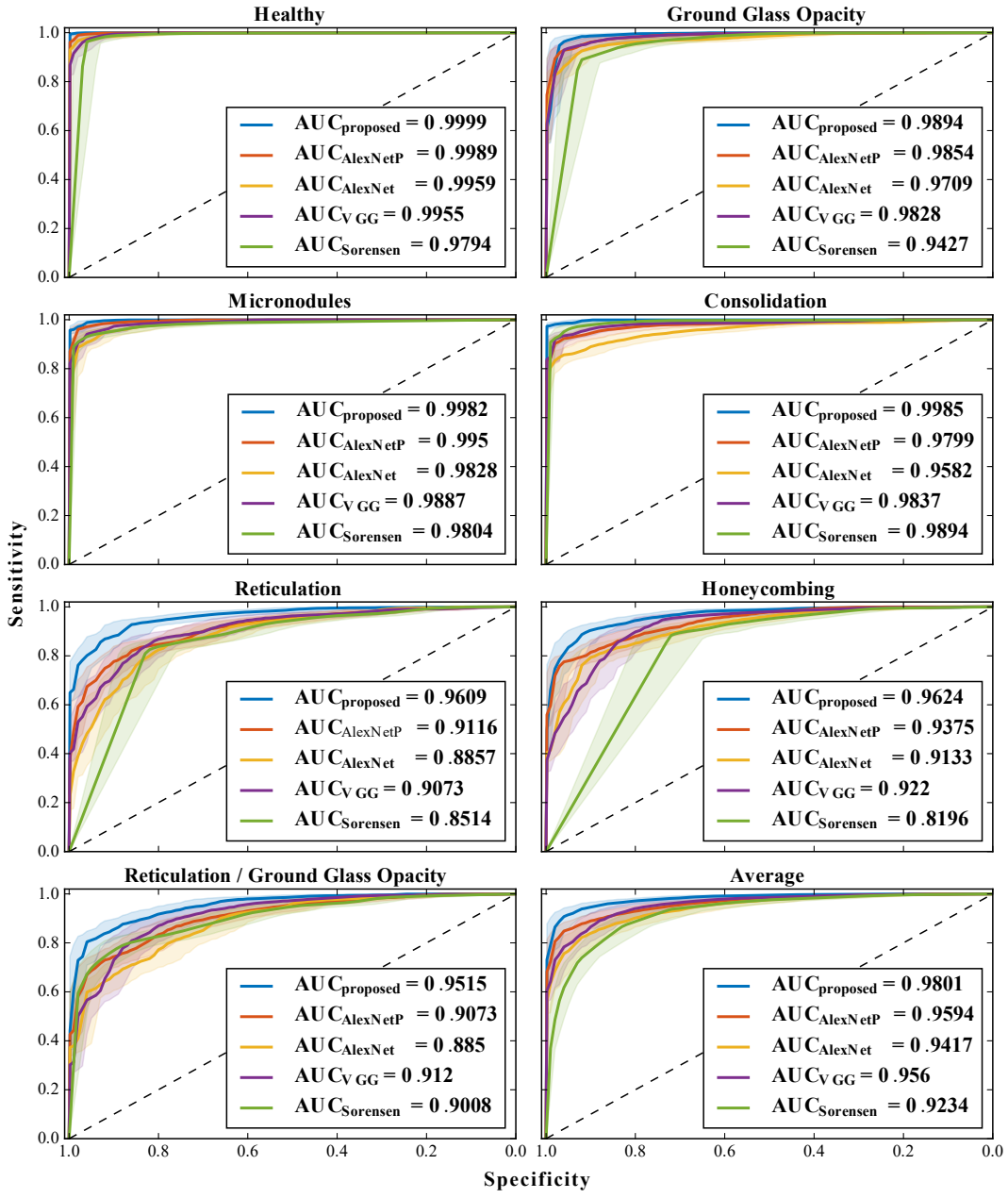


FIGURE 2.5: ROC analysis for the proposed CNN and four previous methods: alexnet, alexnet pre-trained (alexnetp), VGG-net and the method by sorensen et al. [18]. The analysis was performed per class (one-vs-all) while the average over all classes is also presented. For each roc, the AUC is given and the 95% confidence interval is plotted.

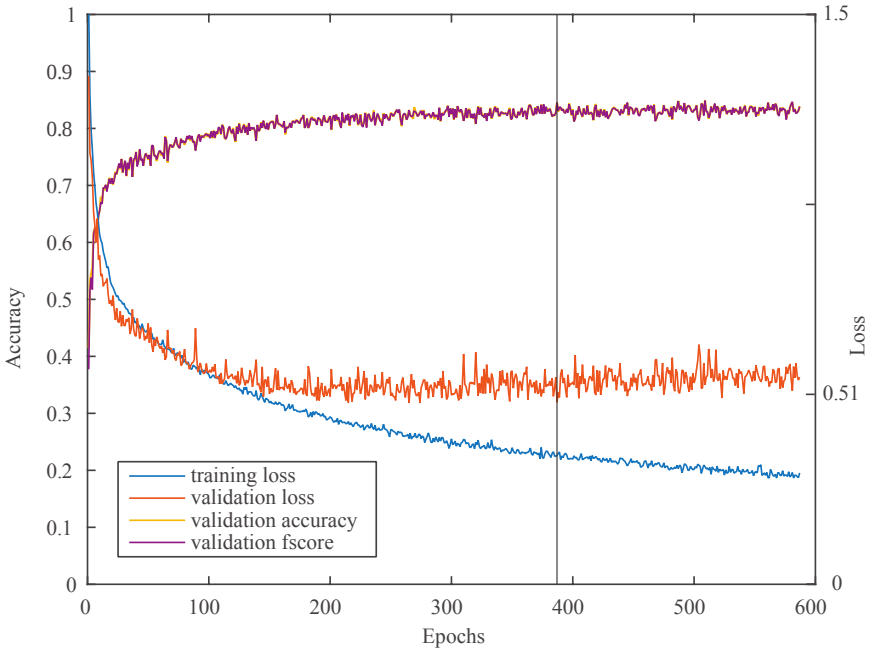


FIGURE 2.6: Loss curves during the training of the proposed system.

making them prohibitive for the clinical practice.

Analysis of the System's Performance

In this paragraph, we provide additional insight into the performance of the proposed method. In Figure 2.6, we show the loss and performance curves during the training of the system. The blue and orange descending curves correspond to the loss function values for the training and for the validation sets during training. The two curves start to diverge from one another after around 100 epochs; however, validation loss continues to descend slightly until roughly 200 epochs. The gray vertical line indicates the best model found. The yellow and purple curves represent the accuracy and Favg on the validation set and after a few epochs they overlap almost completely, showing that when the network gets sufficiently trained, it treats the classes fairly balanced.

The 16 kernels for the first convolutional layer of the best model are illustrated in Figure 2.7. Although the small number and size of the kernels do not permit much discussion, one may notice their differential nature that captures fundamental edge patterns. These patterns grow in size and complexity while passing through consecutive convolutional layers, so that the last layer describes the micro-structures that characterize texture.

Figure 2.8a shows the confusion matrix of the proposed method for the seven considered classes. The confusion between honeycombing and reticular patterns is due to their common fibrotic nature and contributes a major share to the overall error. Figure 2.9 presents some difficult cases of these patterns that were misclassified, together with the corresponding output of the network. The relatively high misclassification rate between the combined GGO/reticulation and the individual GGO and reticulation patterns could be justified by the fact that the former constitutes an overlap of the latter. This combinational pattern is particularly difficult for every classification scheme tested, and it has not been considered in most of the previous studies. We decided to include it here, because its presence is very

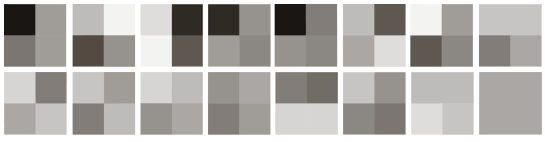


FIGURE 2.7: The 2×2 kernels from the first layer of the proposed CNN.

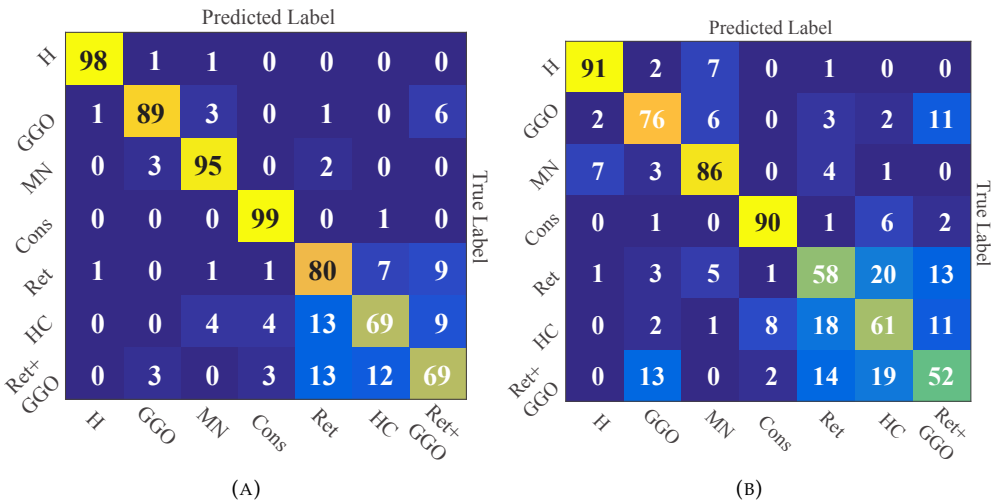
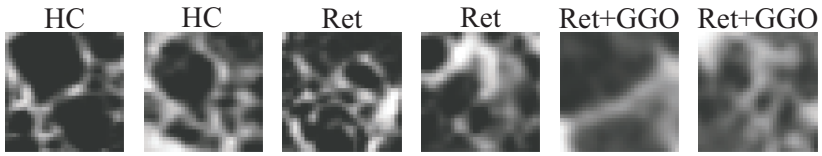


FIGURE 2.8: Confusion matrices of: (A) the proposed method, (B) the method by sorensen et al. [18]. The entry in the ith row and jth column corresponds to the percentage of samples from class i that were classified as class j. H: healthy tissue; MN: micronodules; GGO: ground glass opacity; cons: consolidation; ret: reticulation, HC: honeycombing.

relevant to the discrimination between idiopathic pulmonary fibrosis (IPF) and non-specific interstitial pneumonia (NSIP), which are the most common ILDs. Figure 2.8b presents the corresponding confusion matrix for the method by Sorensen et al. [18]. The results show that the higher misclassification rate is mainly caused by the reticular patterns, which require an accurate description of texture apart from the first-order description of intensity values.

2.4 Conclusions

In this chapter, we proposed a deep CNN to classify lung CT image patches into 7 classes, including 6 different ILD patterns and healthy tissue. A novel network architecture was designed that captures the low-level textural features of the lung tissue. The network consists of 5 convolutional layers with 2×2 kernels and LeakyReLU activations, followed by just one average pooling, with size equal to the size of final feature maps and three dense layers. The training was performed by minimizing the categorical cross entropy with the Adam optimizer. The proposed approach gave promising results, outperforming the state of the art on a very challenging dataset of 120 CT scans from different hospitals and scanners. The



	HC	HC	Ret	Ret	Ret+GGO	Ret+GGO
Healthy	< 1%	< 1%	< 1%	< 1%	< 1%	< 1%
GGO	< 1%	< 1%	< 1%	< 1%	< 1%	< 1%
MN	< 1%	< 1%	< 1%	< 1%	8.83%	< 1%
Cons	< 1%	20.23%	< 1%	< 1%	< 1%	< 1%
Ret	**50.58%**	17.60%	19.55%	49.02%	**57.36%**	< 1%
HC	43.02%	19.94%	**60.76%**	1.86%	1.25%	**94.21%**
Ret+GGO	6.38%	**41.41%**	19.38%	**49.06%**	31.92%	5.49%

FIGURE 2.9: Examples of misclassified patches by the proposed CNN. The output of the network is displayed below each patch.

method can be easily trained on additional textural lung patterns while performance could be further improved by a more extensive investigation of the involved parameters. The large number of parameters and the relatively slow training (typically a few hours) could be considered as a drawback of this kind of DL approaches, together with the slight fluctuation of the results, for the same input, due to the random initialization of the weights. In future studies, we plan to extend the method to consider three dimensional data from MDCT volume scans and finally to integrate it into a CAD system for differential diagnosis of ILDs.

Chapter 3

Multi-source Transfer Learning

This chapter is a modified version of:
S. Christodoulidis*, M. Anthimopoulos*, L. Ebner, A. Christe and S. Mougiakakou, *"Multisource Transfer Learning With Convolutional Neural Networks for Lung Pattern Analysis,"* in IEEE Journal of Biomedical and Health Informatics, vol. 21, no. 1, pp. 76-84, Jan. 2017.
DOI: 10.1109/JBHI.2016.2636929

*This study was a highly collaborative effort of M. Anthimopoulos and S. Christodoulidis, who share the first authorship. All figures and the experiments were prepared and executed by S. Christodoulidis while the text was writer by S. Christodoulidis and M.Anthimopoulos. The technical research directions were chosen after long discussions between M. Anthimopoulos, S. Christodoulidis and S. Mougiakakou while the medical research directions were decided by A. Christe and L. Ebner.

Early diagnosis of interstitial lung diseases is crucial for their treatment, but even experienced physicians find it difficult, as their clinical manifestations are similar. In order to assist with the diagnosis, computer-aided diagnosis systems have been developed. These commonly rely on a fixed scale classifier that scans CT images, recognizes textural lung patterns, and generates a map of pathologies. In a previous study, we proposed a method for classifying lung tissue patterns using a deep convolutional neural network (CNN), with an architecture designed for the specific problem. In this study, we present an improved method for training the proposed network by transferring knowledge from the similar domain of general texture classification. Six publicly available texture databases are used to pretrain networks with the proposed architecture, which are then fine-tuned on the lung tissue data. The resulting CNNs are combined in an ensemble and their fused knowledge is compressed back to a network with the original architecture. The proposed approach resulted in an absolute increase of about 2% in the performance of the proposed CNN. The results demonstrate the potential of transfer learning in the field of medical image analysis, indicate the textural nature of the problem and show that the method used for training a network can be as important as designing its architecture.

3.1 Field of Study

Medical imaging data are much more difficult to acquire compared to general imagery, which is freely available on the Internet. On top of that, their annotation has to be performed by multiple specialists to ensure its validity, whereas in natural image recognition anyone could serve as annotator. This lack of data makes the training on medical images very difficult or even impossible for many of the huge networks proposed in computer vision. A common way to overcome this problem is to pretrain the networks on large color image databases like ImageNet, and then fine-tune them on medical imaging data, a method often referred to as transfer learning. This approach has yielded adequately good results for many applications and has demonstrated the effectiveness of transfer learning between rather different image classification tasks [46]. Secondly, the architecture of popular CNNs from the field of computer vision, is generally suboptimal for problems encountered in medical imaging such as texture analysis, while their input size is fixed and often not suitable.

3.1.1 Transfer Learning

Transfer learning is generally defined as the ability of a system to utilize knowledge learned from one task, to another task that shares some common characteristics. Formal definitions and a survey on transfer learning can be found in [47]. In this study, we focus on supervised transfer learning with CNNs. Deep CNNs have shown remarkable abilities in transferring knowledge between apparently different image classification tasks or even between imaging modalities for the same task. In most cases, this is done by weight transferring. A network is pretrained on a source task and then the weights of some of its layers are transferred to a second network that is used for another task. In some cases, the activations of this second network are just used as "off-the-shelf" features which can then be fed to any classifier [48] . In other cases, the non-transferred weights of the network are randomly initialized and a second training phase follows, this time on the target task [49]. During this training, the transferred weights could be kept frozen at their initial values or trained together with the random weights, a process usually called "fine-tuning". When the target dataset is too small with respect to the capacity of the network, fine-tuning may result in overfitting, so the features are often left frozen. Finding which and how many layers to transfer depends on the proximity of the two tasks but also on the proximity of the corresponding imaging modalities. It has been shown that the last layers of the network are task specific while the earlier layers of the network are modality specific [39]. On the other hand, if there are no overfitting issues, the best strategy is to transfer and fine-tune every layer [49]. This way, the discovered features are adapted on the target task, while keeping the useful common knowledge. Another type of transfer learning is the multi-task learning (MTL) approach that trains on multiple related tasks simultaneously, using a shared representation [50]. Such process may increase the performance for all these tasks and It is typically applied when training data for some tasks are limited.

Transfer learning has been extensively studied over the past few years, especially in the field of computer vision, with several interesting findings. In [51], pretrained CNNs such as VGG-Net and AlexNet are used to extract "off-the-shelf" CNN features for image search and classification. The authors demonstrate that fusing features extracted from multiple CNN layers improves the performance on different benchmark databases. In [52], the factors that influence the transferability of knowledge in a fine-tuning framework are investigated. These factors include the network's architecture, the resemblance between source and target tasks and the training framework. In a similar study [49], the effects of different fine-tuning procedures on the transferability of knowledge are investigated, while a procedure is proposed to quantify the generality or specificity of a particular layer. A number of studies have also utilized transfer learning techniques, in order to adapt well-known networks to classify medical images. In most of the cases, the network used is the VGG, AlexNet or GoogleNet pretrained on ImageNet [53], [54]. However, these networks are designed with a fixed input size usually of $224 \times 224 \times 3$, so that images have to be resized and their channels artificially extended to three, before being fed to the network. This procedure is inefficient and may also impair the descriptive ability of the network.

3.1.2 Contribution

In Chapter 2 we proposed a novel CNN that achieved significant improvement with respect to the state of the art. The network's architecture was especially designed to extract the textural characteristics of ILD patterns, while its much smaller size allowed it to be successfully trained on solely medical data without transfer learning. In this study, we propose a novel training approach that improves the performance of the newly introduced CNN, by additionally exploiting relevant knowledge, transferred from multiple general texture databases.

3.2 Materials and Methods

In this section we present a method for transferring knowledge from multiple source databases to a CNN, ultimately used for ILD pattern classification. Prior to this, we describe the databases that were utilized for the purposes of this study as well as the architecture of

TABLE 3.1: Description of the source domain databases

Database	Type	Number of classes	Number of instances per class	Number of images per instance	Total number of images	Area per image (10^3px)	Number of training patches	Number of validation patches
ALOT [55]	Color	250	1	100	25000	98.304	257880	85870
DTD [56]	Color	47	120	1	5640	229.95 ± 89.14	180351	87485
FMD [57]	Color	10	100	1	1000	158.3 ± 43.2	18247	6285
KTB [58]	Grey	27	160	1	4480	331.776	207360	69120
KTH-TIPS-2b [59]	Color	11	4	108	4752	40	31481	10410
UIUC [60]	Grey	25	40	1	1000	307.2	47250	15750

the newly proposed CNN, in order to provide a better foundation for the description of the methodology.

3.2.1 Databases

Six texture benchmark databases were employed to serve as source domains for the multi-source transfer learning: the Amsterdam library of Textures (ALOT) [55], the Describable Textures Dataset (DTD) [56], the Flickr Material Database (FMD) [57], Kylberg Texture Database (KTB) [58], KTH-TIPS-2b [59] and the Ponce Research Group's Texture database (UIUC) [60]. Moreover, the concatenation of all aforementioned databases was also used. As target domain, we used two databases of ILD CT scans from two Swiss university hospitals: the Multimedia database of ILD by the University Hospital of Geneva (HUG) [33] and the Bern University Hospital, "Inselspital" (Insel) database [61].

Source Domain Datasets

All the source domain databases are publicly available texture classification benchmarks. Each class corresponds to a specific texture (e.g. fabric, wood, metal, foliage) and is represented by pictures of one or more instances of the texture. Two of the databases – ALOT and KTH-TIPS-2b – also contain multiple pictures for each instance under different angles, illumination and scales. The image size is fixed for all databases apart from DTD, while FMD also provides texture masks.

For the creation of the training-validation dataset, all the color databases (i.e. ALOT, DTD, FMD, KTH-TIPS-2b) were converted to gray-scale and non overlapping patches were extracted with a size equal to the input of the proposed CNN namely, 32×32. When not provided, partitioning between training and validation sets was performed at the instance level, except for ALOT, where the number of instance is equal to the number of classes. No testing set was created for the source domain databases, since the ultimate goal is to test the system only on the target domain. In the case of DTD, where training, validation and test sets are provided, the test set was added to the training set. Table 3.1 summarizes the characteristics of the original source databases and the corresponding patch datasets.

Target Domain Dataset

The HUG database [33] consists of 109 HRCT scans of different ILD cases with 512×512 pixels per slice and an average of 25 slices per case. The average pixel spacing is 0.68mm, and the slice thickness is 1-2mm. Manual annotations for 17 different lung patterns are also provided, along with clinical parameters from patients with histologically proven diagnoses of ILDs. The Insel database consists of 26 HRCT scans of ILD cases with resolution 512×512 and an average of 30 slices per case. Average pixel spacing is 0.62mm and slice thickness is 1-2mm.

A number of preprocessing steps was applied to the CT scans before creating the final ILD patch dataset. The axial slices were rescaled to match a certain x,y-spacing value that was set to 0.4mm, while no rescaling was applied on the z-axis. The image intensity values were cropped within the window [-1000, 200] in Hounsfield units (HU) and mapped to [0,

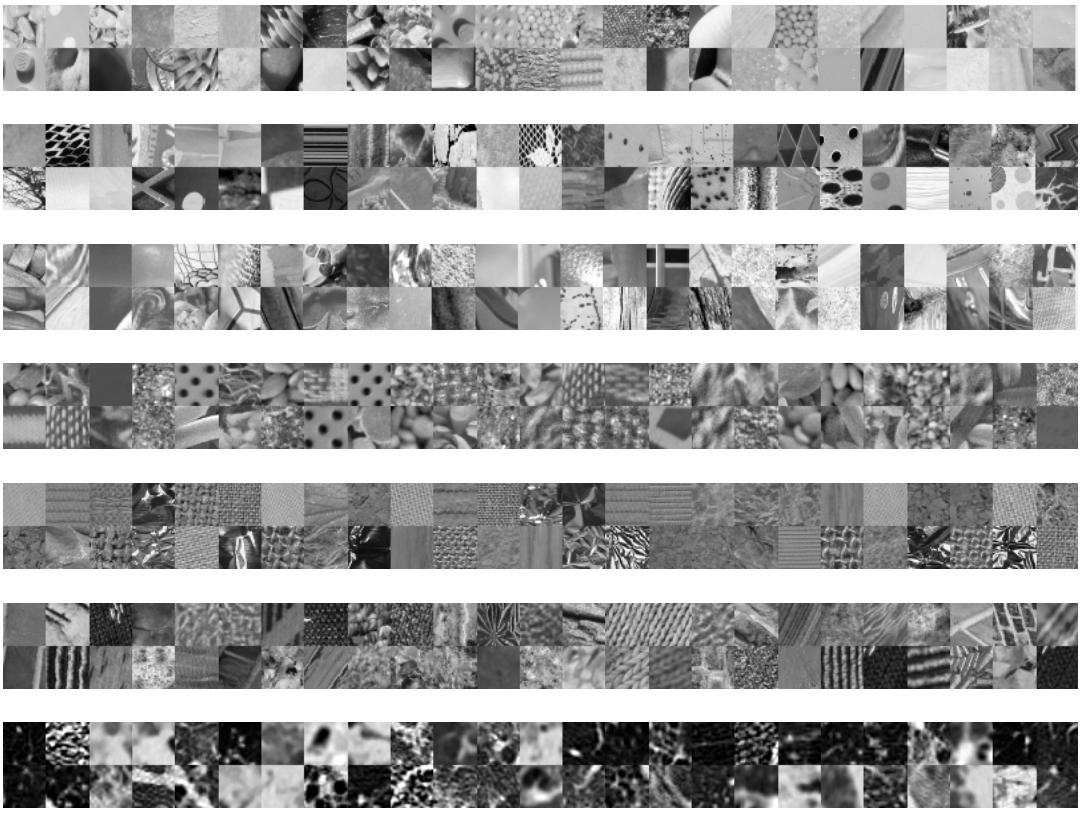


FIGURE 3.1: Typical samples from each dataset. The color databases were converted to gray scale. From top to bottom: ALOT, DTD, FMD, KTB, KTH-TIPS-2b, UIUC, ILD

1]. Experienced radiologists from Bern University hospital annotated (or re-annotated) both databases by manually drawing polygons around seven different patterns including healthy tissue and the six most relevant ILD patterns, namely ground glass, reticulation, consolidation, micronodules, honeycombing and a combination of ground glass and reticulation. In total 5529 ground truth polygons were annotated, out of which 14696 non-overlapping image patches of size 32×32 were extracted, unequally distributed across the 7 classes. The patches are entirely included in the lung field and they have an overlap with the corresponding ground truth polygons of at least 80%. From this patch dataset, 150 patches were randomly chosen from each class for the validation and 150 for the test set. The remaining patches were used as the training set, which was artificially augmented to increase the amount of training data and prevent over-fitting. Label-preserving transformations were applied, such as flip and rotation, as well as combinations of the two. In total, 7 transformations were used while duplicates were also added for the classes with few samples. The final number of training samples was constrained by the rarest class and the condition of equal class representation that led to 5008 training patches for each class. In total, the training set consists of 35056 patches while the validation and test sets contain of 1050 patches each. More details about this dataset can be found in [61].

3.2.2 CNN Architecture

In order to minimize the parameters involved and focus only on the aspects of transfer learning, we used the same CNN architecture as proposed in [61] throughout the different steps of the method. The input of the network is an image patch of 32×32 pixels. This patch is convolved by five subsequent convolutional layers with 2×2 kernels, while the number of kernels is proportional to the receptive field of each layer with respect to the input. The number of kernels we used for the L_{th} layer is $k(L+1)^2$, where the parameter k depends on the complexity of the input data and was chosen to be 4. The output of the final convolutional layer is globally pooled, thus passing the average value of each feature map to a

series of three dense layers. A rectified linear unit (ReLU) is used as the activation function for the dense layers, while the convolutional layers employ very leaky ReLU activations with $\alpha = 0.3$. Finally, Dropout is used before each dense layer dropping 50% of its units. For training the network, the Adam optimizer [39] was used with the default values for its hyper-parameters. The training ends when none of 200 consecutive epochs improves the network's performance on the validation set by at least 0.5%.

3.2.3 Multi-source Transfer learning

The source datasets presented in Section 3.2.1 demonstrate a wide spectrum of different characteristics, as shown in Figure 3.1 and Table 3.1; hence, we expect that they will also contribute a range of diverse and complementary features. If this assumption holds, the parallel transfer learning from all datasets into one model will improve its performance more than any individual dataset would. However, the standard transfer learning approach by transferring weights can only utilize one source dataset. To tackle this problem, we transfer knowledge from each source to a different CNN and then fuse them into an ensemble that is expected to have performance superior to any of the individual models but also a larger computational complexity. We then transfer the fused knowledge back to a network with the original architecture, in order to reduce the complexity while keeping the desirable performance. Simple weight transferring is again not possible here, since it requires models with the same architecture. We therefore use model compression, a technique that transfers knowledge between arbitrary models for the same task. Figure 3.2 depicts the full pipeline of the proposed multi-source transfer learning method while in the next paragraphs, we describe its three basic components in more detail.

Single-Source Transfer Learning

Figure 3.3 illustrates the used weight transfer scheme from a source task to the target task, namely the ILD classification. Starting from the first layer, a number of consecutive layers are transferred from the pretrained network to initialize its counterpart network. The rest of the network is randomly initialized, while the last layer changes size to match the number of classes in the target dataset (i.e. 7). The transferred layers are then fine-tuned along with the training of the randomly initialized ones. We decided to fine-tune the layers instead of freezing them since the proposed network is relatively small and has been previously trained on the target dataset without overfitting [61]. According to [49] weight freezing should only be used to avoid overfitting problems. In order to investigate the effects of transferring different number of layers, we have performed a set of experiments for each of the source datasets.

Knowledge Fusion in an Ensemble

Ensembles are systems that use multiple predictors, statistically independent to some extent, in order to attain an aggregated prediction. Using ensembles to achieve a better performance is a well established technique and has been successfully exploited in many applications [62]. Such systems usually perform better than each of the predictors alone, while they also gain stability. This performance gain arises from the fact that the different prediction models that form the ensemble, capture different characteristics of the function to be approximated.

In order to build a strong ensemble, instead of manually selecting the models, we implemented an ensemble selection approach similar to the one presented in [63]. The employed algorithm is a forward selection procedure which selects models from a pool and iteratively adds them to the ensemble following a specific criterion. Moreover, some additions to prevent over-fitting were also implemented. The pool from which the algorithm selects models includes all the networks that were pretrained on the source datasets and fine-tuned on the ILD dataset, snapshots of these networks during training, as well as a few randomly initialized networks trained from scratch on the target data. After creating the CNN model pool, a subset is randomly sampled from it with half of its size. Then, the models in the subset are ranked by their performance and the best N models are chosen to initialize the ensemble.

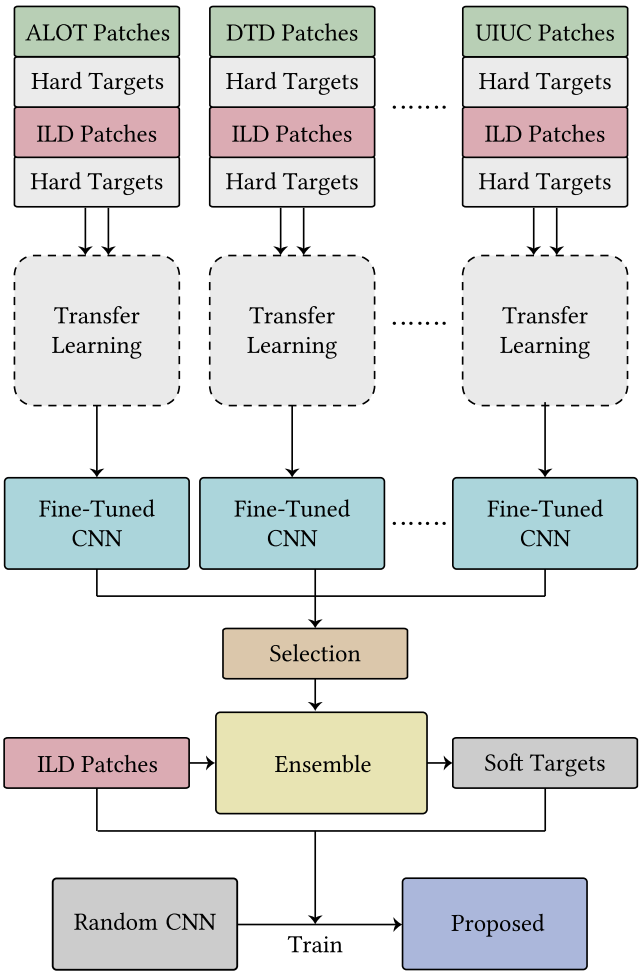


FIGURE 3.2: Multi-source Transfer Learning: Knowledge is transferred from each source database to a different CNN. A selection process combines CNNs into an ensemble that is used to teach a single randomly initialized model.

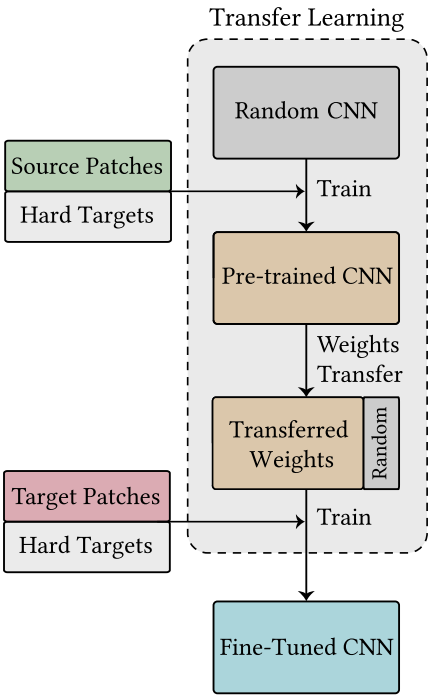


FIGURE 3.3: Transfer Learning through weight transfer

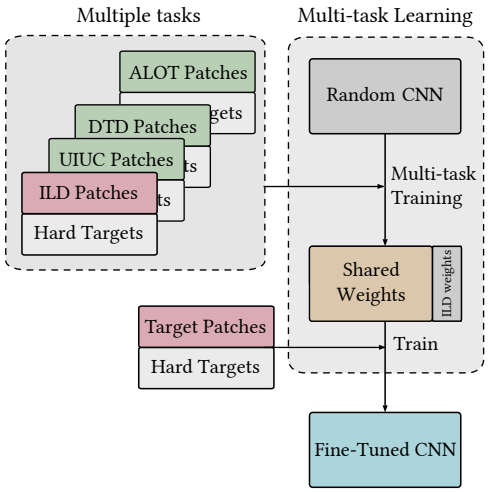


FIGURE 3.4: Multi-task Learning

From the rest of the subset's models, we add the one that increases the ensemble performance the most, and continue adding models until no more gain can be achieved. Model selection is performed with replacement, meaning that the same model can be included more than once. The whole procedure is repeated for M subsets generating M ensembles which are then aggregated into one, by averaging their outputs. The selection of the models is based on the average F-score of the validation set while the involved parameters have been tested on a grid with $N = \{1, 2, \ldots, 25\}$ and $M = \{1, 2, \ldots, 15\}$. For each position of the parameter grid the selection was repeated 100 times and finally the best ensemble was found for $N = 2$ and $M = 5$.

Model Compression

Model compression is used as a final step, to compress the knowledge of the huge ensemble created by the previous procedure, into a single network with the original architecture. Model compression, also known as knowledge distillation, is the procedure for training a model using "soft targets" that have been produced by another, usually more complex model [64] [65]. As soft targets one can use the class probabilities produced by the complex model or the logits namely, the output of the model before the softmax activation. The model that produces the soft targets is often called the teacher, while the model to which the knowledge is distilled plays the role of the student. The soft targets carry additional knowledge discovered by the teacher, regarding the resemblance of every sample to all classes. This procedure can be considered as another type of knowledge transfer which is performed for the same task, yet between different models. In our case, the ensemble is employed as a teacher while the student is a single, randomly initialized CNN with the original architecture described in Section 3.2.2. After being trained on the soft targets the student model will approximate the behavior of the ensemble model and will even learn to make similar mistakes. However, these are mistakes that the student would have probably made by training on the hard targets, considering its relatively inferior capacity.

3.2.4 Multi-task Learning

MTL is another way to fuse knowledge from multiple sources into multiple models. In this study we used it as a baseline method. The method simultaneously trains models for each of the tasks, with some of the weights shared among all models. In our implementation, we train seven networks, one for each of the source datasets and one for the target dataset. These CNNs share all the weights apart from the last layer, the size of which depends on the number of classes for that particular task. The parallel training was achieved by alternating every epoch the task between the target and one of the source tasks. In other words,

odd epochs train on the target task while even epochs train on source tasks in a sequential manner. Although MTL fuses knowledge from all involved tasks, it does not use tasks exclusively as source or target like the standard transfer learning approach. Since our final goal is to improve the performance of the target task, we further fine-tune the resulting model on the ILD dataset. Figure 3.4 depicts an outline of our multi-task learning approach.

3.3 Experimental Setup and Results

In this section we describe the setup of the conducted experiments, followed by the corresponding results with the related discussion.

3.3.1 Experimental Setup

For all the experiments presented in this section, a train-validation-test scheme was utilized. The presented results were calculated on the test set while the selection of hyper-parameters and the best resulting models was made over the validation set. In the rest of this section, we describe the chosen evaluation protocol and some implementation details.

Evaluation

As a principle evaluation metric we used the average F_1-score over the different classes, due to its increased sensitivity to imbalances among the classes. The F_1-score is calculated as follows:

$$F_{avg} = \frac{2}{7} \sum_{c=1}^{7} \frac{recall_c \cdot precision_c}{recall_c + precision_c}$$

where $recall_c$ is the fraction of samples correctly classified as c over the total number of samples of class c, and the $precision_c$ is the fraction of samples correctly classified as c over all the samples classified as c.

Implementation

The proposed method was implemented in Python using the Keras [66] framework with a Theano [41] back-end. All experiments were performed under Linux OS on a machine with CPU Intel Core i7-5960X @ 3.50GHz, GPU NVIDIA GeForce Titan X, and 128GB of RAM.

3.3.2 Results

In this section, we present the results of the performed experiments, grouped according to the three basic components of the system as presented in Section 3.2.3. Finally, we analyze the performance of the proposed network and compare with other methods.

Single-Source Transfer Learning

In this first series of experiments we investigate the performance gain by transferring knowledge from individual source datasets to the target task, i.e. the classification of ILD patterns. A CNN model was pretrained on each of the six source datasets and then fine-tuned on the ILD data. A seventh source dataset was added that consists of all six datasets merged in one. As described in Section 3.2.2, the proposed network has five convolutional and three dense layers. Starting from the first, we transfer one to seven layers for each of the pretrained networks. The rest of the layers are randomly initialized and the entire CNN is fine-tuned on the ILD task. Different random initializations may result in deviations of the results so to minimize this effect, we repeated each experiment three times and reported the mean values.

The results of this experiment are depicted in Figure 3.5, where the region of the light gray background denotes the convolutional layers, while the rest denote the first two dense layers. The horizontal dashed line at 0.855 represents the performance of the network trained from scratch (with random initialization). The best results were achieved when six layers (i.e.

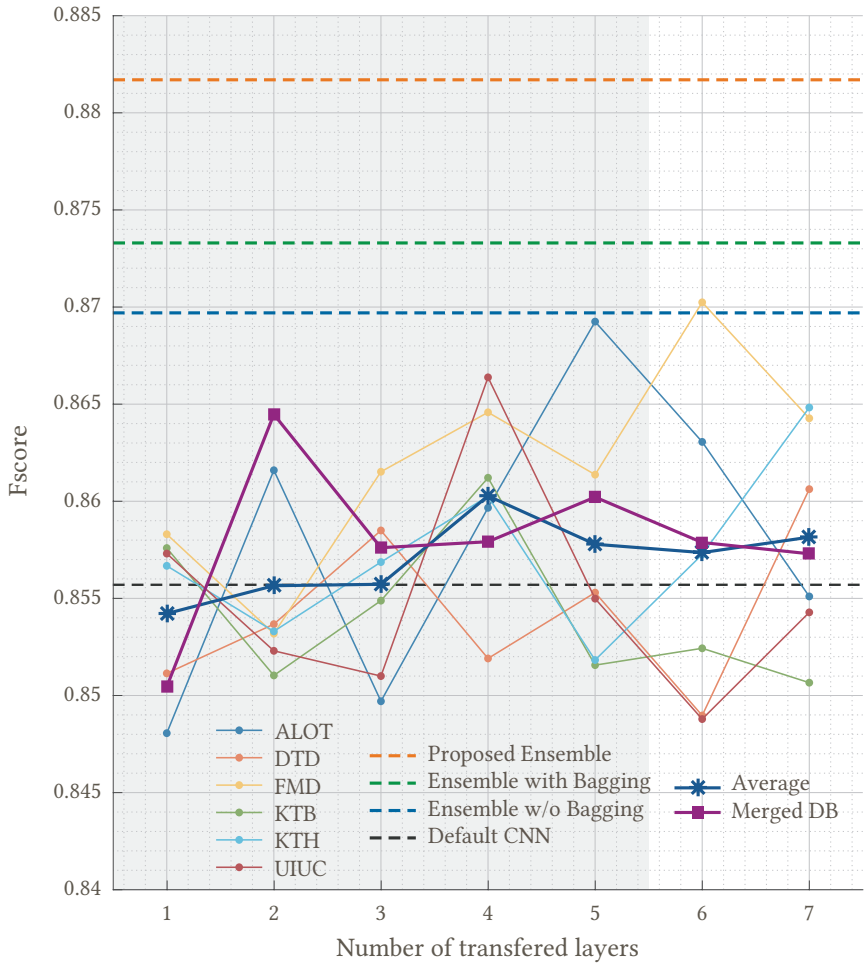


FIGURE 3.5: The F_1-score produced by transferring of knowledge from single source domains for different number of transferred layers, averaged over three experiments. The horizontal lines correspond to the CNN without knowledge transfer or the different ensembles of CNNs.

five convolutional layers and one dense) were transferred from the CNN that was pretrained on the FMD dataset. However, no optimal weight transferring strategy can be inferred for every pretrained network, due to their relative different behavior. An additional line with the average performance over all source datasets is also shown. According to this line, the contribution of weight transferring increases, on average, when transferring at least four layers. Weight transferring seems to help even when transferring all layers. This is probably due to the ability of fine-tuning to adapt even the most task-specific features to the target task, an observation which is inline with the conclusions of [49].

As for the runtime of the experiments, one could expect a faster training for a pretrained network since its initial state is closer to a good solution than a randomly initialized network. Indeed, the average number of epochs for the pretrained is 426 instead of 479 for the random, with each epoch taking about 12 seconds. However, this difference is small and statistically non-significant ($p \approx 0.11$) probably due to the fact that loss drops with a lower rate while approaching the end of training, so the starting point does not significantly affect the number of required epochs.

The conducted experiments have demonstrated that the random initializations before pretraining or fine-tuning, as well as the different source datasets may introduce a significant variance between the network's results. This unstable behavior of single-source transfer learning combined with the assumption of reduced correlation among the resulting models, motivated us to build an ensemble model to fuse the extracted knowledge and reduce the aforementioned variance.

TABLE 3.2: Comparison of the proposed method with methods from the literature

Study	Method	F_{avg}
Gangeh [19]	Local pixel textons - SVM-RBF	0.6942
Sorensen [18]	LBP, histogram - kNN	0.7420
Anthimopoulos[16]	Quantiles of local DCT, histogram - RF	0.8170
Li [27]	5-layer CNN	0.6657
LeNet [43]	7-layer CNN	0.6783
AlexNet [25]	8-layer CNN	0.7031
Pre-trained AlexNet	8-layer CNN	0.7582
VGG-Net [35]	16-layer CNN	0.7804
Anthimopoulos [61]	8-layer CNN	0.8557
Proposed Methods	**Multi-task Learning**	**0.8631**
	Compressed 8-layer CNN	**0.8751**
	Ensemble of CNNs	**0.8817**

Knowledge Fusion in an Ensemble

Figure 3.5 also illustrates the performance of the ensemble that was built as described in Section 3.2.3. The ensemble clearly outperforms the rest of the models by reducing their variance (through output averaging) and by transferring multi-source knowledge, at the same time. In order to investigate the contribution of ensemble averaging alone, we also built an ensemble from a pool of randomly initialized models. The output of this ensemble reached a performance of 0.8697 which is better than the single randomly initialized CNN but still inferior to the multi-source ensemble. In addition, we used bootstrap aggregating (bagging) to boost the performance even more by reducing the correlation between the models. To this end, we trained each CNN of the ensemble on samples randomly sampled from the training set with replacement. The performance was slightly improved reaching 0.8733 which was however still inferior to the proposed ensemble. These results showed that although the ensemble by itself may increase the accuracy of stochastic models, the transferred knowledge also contributes to the final result.

Model Compression

For this last part, the ensemble was employed as a teacher producing soft targets for the ILD training dataset that were then used to train CNNs. We experimented with a number of different choices for the student networks choosing between the pretrained and fine-tuned networks from the previous steps as well as randomly initialized ones. All of the different students reached similar levels of performance, so we finally chose as student the one with the random initialization, for simplicity. The achieved performance after teaching the chosen student was 0.87518 in the test set. This result lies below the ensemble's performance yet above all the previously presented results.

Performance and Comparison with Previous Work

As a baseline method for comparison in multi-source transfer learning we used an MTL approach as described in Section 3.2.4. The performance on the ILD task while training along with the other tasks only reached the value of 0.8110. After a fine-tuning step, the performance reached the value of 0.8631, which is not much better than the network trained from scratch and similar to a number of single source pretrained networks. These results could be due to the limited capacity of the network that attempts to solve multiple problems at the same time. Modifications in the MTL scheme such as weighting the contributions of the different tasks or sharing different parts of the network could yield better results, however this would increase the complexity of the scheme and would require a large number of experiments on different strategies.

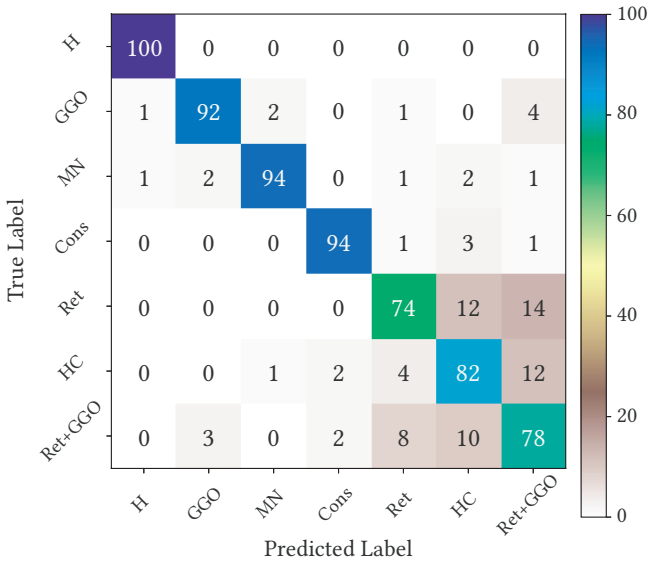


FIGURE 3.6: Confusion matrix of the proposed compressed model.

Table 3.2 provides a comparison with other methods from the literature. The first three rows correspond to methods that use hand crafted features and a range of different classifiers. The rest correspond to methods that utilize CNNs. All the results were reproduced by the authors by implementing the different methods and using the same data and framework to test them. The proposed multi-source transfer learning technique improved the performance of the proposed network by an absolute 2% compared to the previous performance 0.8557 of the same CNN in [61]. Finally, Figure 3.6 shows the confusion matrix of the proposed approach. As shown, the confusion is basically between the fibrotic classes (i.e. reticulation, honeycombing and the combination of ground glass and reticulation) which was expected. One may also notice that the matrix is more balanced than the one presented in [61].

3.4 Conclusions

In this chapter we presented a training method that improves the accuracy and stability of a CNN on the task of lung tissue pattern classification. The performance gain was achieved by the multiple transfer of knowledge from six general texture databases. To this end, a network was pretrained on each of the source databases and then fine-tuned on the target database after transferring different numbers of layers. The networks obtained were combined in an ensemble using a model selection process, which was then employed to teach a network with the original size. The resulting CNN achieved a gain in performance of about 2% compared to the same network when trained on the hard targets. This result proves the potential of transfer learning from natural to medical images that could be beneficial for many applications with limited available medical data and/or annotations. We believe that more challenging datasets, with additional classes and/or higher diversity, may benefit even more from similar approaches. Considering that even experienced radiologists would not achieve a perfect classification, especially on a patch level, the reported performances could have reached a peak. Finally, the reported increase in accuracy comes at the expense of increased training time since multiple models have to be trained. However, the inference time is still exactly the same and the additional training time required can be considered as a fair compromise for improving the performance, in cases of data shortage. Our future research plans in the topic include the use of the ensemble teacher for labeling unlabeled samples that will augment the training set of the student model. Such an approach could partially assist with the common problem of limited annotated data in the field of medical image analysis.

Chapter 4

Semantic Segmentation of Pathological Lung Tissue

This chapter is a modified version of:
M. Anthimopoulos*, S. Christodoulidis*, L. Ebner, T. Geiser, A. Christe and S. G. Mougiakakou,
"Semantic Segmentation of Pathological Lung Tissue with Dilated Fully Convolutional Networks," in IEEE Journal of Biomedical and Health Informatics.
DOI: 10.1109/JBHI.2018.2818620

*This study was a highly collaborative effort of M. Anthimopoulos and S. Christodoulidis, who share the first authorship. All figures and the experiments were prepared and executed by S. Christodoulidis while the text was writer by S. Christodoulidis and M.Anthimopoulos. The technical research directions were chosen after long discussions between M. Anthimopoulos, S. Christodoulidis and S. Mougiakakou while the medical research directions were decided by A. Christe and L. Ebner.

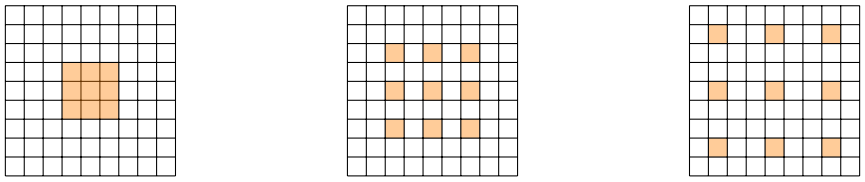
Early and accurate diagnosis of interstitial lung diseases (ILDs) is crucial for making treatment decisions, but can be challenging even for experienced radiologists. The diagnostic procedure is based on the detection and recognition of the different ILD pathologies in thoracic CT scans, yet their manifestation often appears similar. In this study, we propose the use of a deep purely convolutional neural network for the semantic segmentation of ILD patterns, as the basic component of a computer aided diagnosis (CAD) system for ILDs. The proposed CNN, which consists of convolutional layers with dilated filters, takes as input a lung CT image of arbitrary size and outputs the corresponding label map. We trained and tested the network on a dataset of 172 sparsely annotated CT scans, within a cross-validation scheme. The training was performed in an end-to-end and semi-supervised fashion, utilizing both labeled and non-labeled image regions. The experimental results show significant performance improvement with respect to the state of the art.

4.1 Field of Study

In this section, we provide a short review of the recent advances in deep learning for computer vision.

4.1.1 Semantic Segmentation

Although CNNs have existed for decades [67], they only became widely popular after their remarkable success in the ImageNet challenge of 2012 [68]. The winning approach of the competition [25], also known as AlexNet, was a deep CNN with five convolutional and three dense layers, each followed by a rectified linear unit (ReLU), while increased strides were used for the max pooling and convolution operations to gradually down-sample the feature maps. Dropout [36] and data augmentation were also utilized, in order to prevent overfitting. Since then, the proposed deep CNNs have led to continuous improvements in the results on ImageNet and other datasets, mainly by enhancing their architecture and by increasing their depth and width. The VGG network [35] reduced the size of the kernels to 3×3, while increasing the number of layers to 19. GoogleNet [69] used consecutive inception modules, where different convolutional and pooling operations are performed in



(A) D = 1 (B) D = 2 (C) D = 3

FIGURE 4.1: Dilated convolution kernels. D denotes the dilation rate

parallel with their outputs merged. This approach drastically reduced the number of parameters while improving the results. ResNet [70] introduced skip connections between layers which permitted the training of networks with hundreds of layers and pushed the limits of deep CNNs even further. The batch normalization (BN) technique [71] also supported these developments, by regularizing and accelerating the training procedure.

Many of the already proposed CNNs have recently been adapted to perform semantic segmentation, rather than just image classification. The term semantic segmentation refers to the task of assigning a class label to every pixel of an image. A simple approach to this task is to use any fixed-scale classification method under a sliding window scheme and then to aggregate the results to build a label map. However, this could be highly inefficient as local image features would have to be recalculated multiple times for adjacent positions of the input window. Luckily, the convolutional layers (with the appropriate padding) in a typical CNN produce feature maps that maintain spatial correspondence with the input image. Therefore, input images of any size can be fed to the network and each pixel can be classified, on the basis of the values of the respective feature map position. This can be achieved by utilizing convolutional layers of size 1×1 that serve as local dense layers and, these networks are therefore often referred to as fully convolutional (FCNs).

However, the spatial correspondence between the input and output of a CNN, can be disrupted by the use of down-sampling operations such, as strided pooling and convolution. Down-sampling of the feature maps is often used to increase the receptive field of the network with respect to the input, as well as to reduce the amount of computational load. In order to restore the original size of the input, researchers have used encoder-decoder architectures, where the encoder usually adopts a well-known architecture such as VGG [35] and the decoder reverses the process by mapping the feature representation back to the input data space. To this end, upsampling operations and transposed convolution [72] (also known as fractionally strided convolution or "deconvolution") have been used for semantic segmentation [73]. Alternatively, in [74] and [75], max unpooling has been used as the inverse operation of each max pooling layer, where the pairs of pooling/unpooling layers are coupled by transferring the max indices from the encoder to the decoder. In [76], a similar architecture was proposed for biomedical image segmentation, with additional skip connections that concatenate the feature maps of an encoding layer to the feature maps of the same-scale decoding layer.

Recently, some CNNs for semantic segmentation have been proposed that use dilated convolutions to increase the receptive field, instead of downsampling the feature maps. Dilated convolution, also called àtrous, is the convolution with kernels that have been dilated by inserting zero holes (àtrous in French) between the non-zero values of a kernel. This was originally proposed for efficient wavelet decomposition in a scheme also known as "algorithme àtrous" [77]. Figure 4.1 shows examples of kernels with different dilation rates. Dilated convolution can increase the receptive field without increasing the number of parameters, as opposed to normal convolution. Moreover, feature maps are densely computed on the original image resolution without the need for downsampling. In [78], a CNN module with dilated convolutions was designed to aggregate multiscale contextual information and improve the performance of state-of-the-art semantic segmentation systems. The module has eight convolutional layers with exponentially increasing dilation rates (i.e. 1, 1, 2, 4, 8, 16), resulting in an exponential increase in the receptive field, while the number of parameters is only grown linearly. Similarly, expansion of the receptive field was achieved in [79] by integrating dilated convolutions in a bottleneck module that was designed for efficiency. In [80], the àtrous spatial pyramid pooling (ASPP) scheme is proposed that uses multiple

TABLE 4.1: Data statistics across the considered classes i.e. Healthy (H), Ground Glass Opacity (GGO), Micronodules (MN), Consolidation (Cons), Reticulation (Ret) and Honeycombing (HC).

	H	GGO	MN	Cons	Ret	HC	Totals
#Pixels$\times 10^5$	92.5	27.7	35.8	7.08	28.2	20.1	211.4
#Cases	66	82	15	46	81	47	172

parallel dilated convolutional layers, in order to capture information from multiple scales.

4.1.2 Contribution

In this study, we propose the use of a deep fully-convolutional network for the problem of ILD pattern recognition that uses dilated convolutions and is trained in an end-to-end and semi-supervised manner. The proposed CNN takes as input a lung HRCT image of arbitrary size and outputs the corresponding label map, thus avoiding the limitations of a sliding window model. Additionally, the utilization of non-labeled image regions in the learning procedure, permits robust training of larger models and proves to be particularly useful when using databases with sparse annotations.

4.2 Materials and Methods

This section presents the proposed fully convolutional neural network for semantic lung tissue segmentation. Prior to this, we describe the materials used for training and testing the network.

4.2.1 Materials

For the purposes of this study, we compiled a dataset of 172 HRCT scans, each corresponding to a unique ILD or healthy subject. The dataset contains 109 cases from the publicly available multimedia database of interstitial lung diseases [33] by the Geneva University Hospital (HUG), along with 63 cases from Bern University Hospital - "Inselspital" (INSEL), as collected by the authors. The scans were acquired between 2003 and 2015 using different scanners and acquisition protocols. The INSEL scans are volumetric, while the HUG scans have a 10-15mm spacing. The slice thickness is 1-2mm for both datasets.

Two experienced radiologists from INSEL annotated or re-annotated ILD typical pathological patterns, as well as healthy tissue in both databases[1]. A lung field segmentation mask was also provided for each case. In total six types of tissue were considered: normal, ground glass opacity, micronodules, consolidation, reticulation and honeycombing. It should be emphasized that these annotations do not cover the entire lung field, but only the most typical manifestations of the listed ILD patterns. This protocol was followed in both databases since it permits the annotation of more scans for the same effort and thus increases data diversity. On the other hand, sparse annotations also introduce challenges. Non-annotated lung areas have to be excluded from both supervised training and evaluation. Another challenging characteristic of the databases is the uneven distribution of the considered classes across the cases. Table 4.1 provides statistics for the entire dataset, while Figure 4.2 presents a sample CT lung slice along with the given annotations.

4.2.2 Methods

In this study, we propose the use of a deep purely convolutional network for the problem of lung tissue semantic segmentation. The network is inspired by [78] and consists of solely convolutional layers that use dilated kernels to increase the receptive field, instead of downsampling the feature maps. This kind of network has been shown to be suitable for similar dense prediction problems that require high resolution precision. The proposed network

[1]ITK-snap was used for the annotation process, http://www.itksnap.org

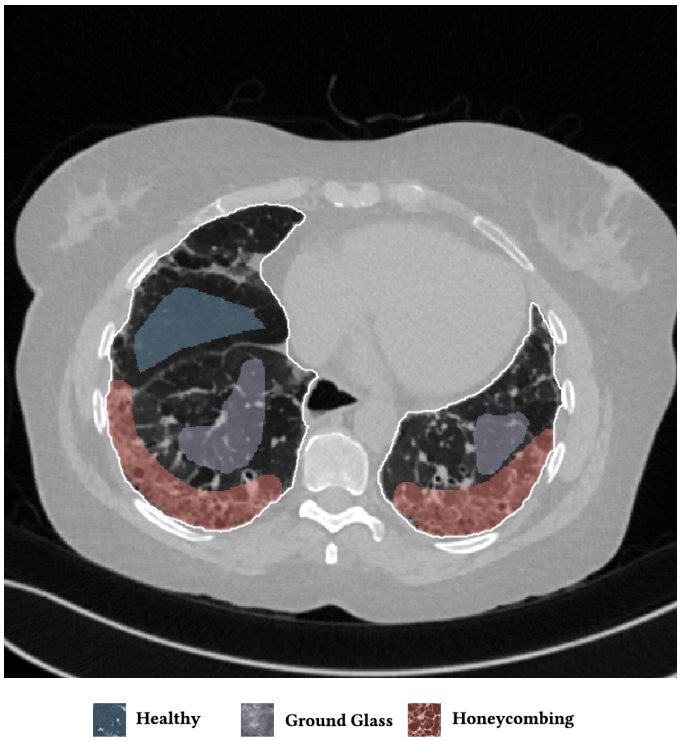


FIGURE 4.2: A typical slice with annotations. The white border line denotes the lung field segmentation, the blue denotes healthy tissue, the purple micronodules and the red the honeycombing pattern.

(Figure 4.3) has 13 convolutional layers and a total receptive field of 287×287. Specifically, each of the first ten layers has 32 kernels of size 3×3 and dilation rates 1, 1, 2, 3, 5, 8, 13, 21, 34 and 55, respectively. We chose not to increase the dilation rates exponentially, as is commonly done, in order to avoid extreme gridding problems that have been reported in several studies [81], [82]. Instead, we use the first terms of the Fibonacci sequence as dilation rates; this mitigates the gridding problem by providing a less steep dilation rate increase and thus denser sampling.

The output of the first 10 layers, as well as the input of the network, are concatenated, thus leading to 1+10×32 = 321 feature maps, which are passed through a dropout layer with a rate of 0.5 and fed to the rest of the network. This concatenation is allowed by the lack of pooling layers and the appropriate zero padding for each convolution and brings several benefits. It permits the aggregation of features from all different scales and levels of abstraction, while it also facilitates the flow of gradients thought the network and therefore allows faster training. The last three layers have 1×1 kernels and play the role of locally dense layers that reduce the feature dimensionality for each pixel from 321 to 128, 32 and finally 6, which is the number of classes considered. The output is converted into a probability distribution by the softmax function.

A BN layer follows each convolution and is based on the batch statistics in both training and test time. This is permitted, as the batch size is one, so there is always a full batch during inference. This approach has been proposed before, under the term instance normalization (InstanceNorm) [83], and has exhibited good performance in texture synthesis, image stylization and image to image translation [84]. InstanceNorm provides invariance to intensity and contrast shifts, which makes the features adaptive for each slice and could mitigate problems caused by different CT scanners and reconstruction kernels. We also found that adding the normalized activations to the non-normalized ones, before passing them through the ReLU function (Figure 4.4) substantially improves the results. This instance normalization skip connection cancels the mean normalization of activations (when the trainable parameters have not been trained), while it performs a kind of feature contrast enhancement which reduces the importance of variance shift without providing complete invariance to the latter.

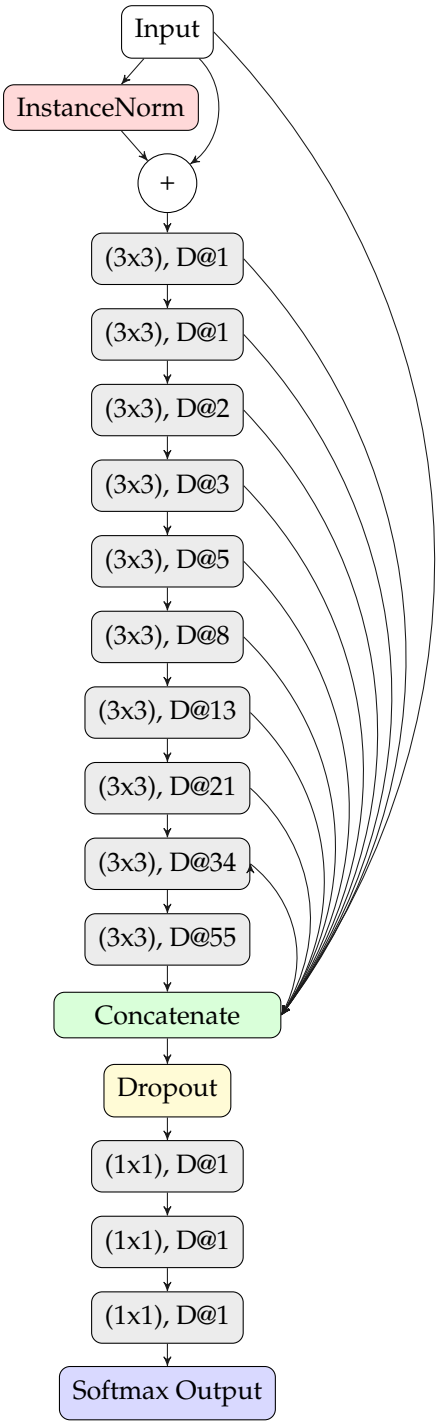


FIGURE 4.3: The architecture of the proposed network. Each gray box corresponds to a block like the one presented in Figure 4.4

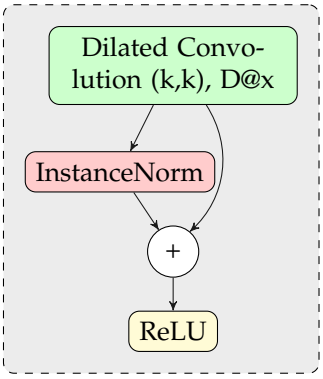


FIGURE 4.4: The block function of the proposed architecture. (k, k) is the size of the convolution kernel and x is the dilation rate.

The network was trained by minimizing the categorical cross entropy using the Adam optimizer [39] with a learning rate of 0.0001. The dense nature of the considered classification problem combined with the sparse available annotations, resulted in two issues. Firstly, large parts of the dataset were not annotated, and so could not be used for either supervised training or testing. Secondly, the distribution of the considered classes in the dataset was highly imbalanced, a fact that can be challenging for any classification method. We tackled both problems by scaling the considered loss and accuracy with appropriate weighting schemes computed for each set. All pixels corresponding to annotated areas were assigned a weight inversely proportional to the number of samples of its class in the specific set. In this way, all classes contributed equally to the considered metrics. Furthermore, we employed a semi-supervised learning technique to additionally exploit non-labeled areas of the data. We added an extra term to the supervised loss function, which corresponds to the entropy of the network's output on the areas that do not participate in the supervised learning. This entropy minimization technique has been used in different applications such as in [85] yielding significant improvements in performance. Similarly in [86] the technique of pseudo-labelling was introduced, where the network classifies non-annotated regions and then uses them as ground truth for fine-tuning. Semi-supervised learning techniques of this kind are based on the cluster assumption i.e. samples from the same class tend to form compact clusters. By minimizing the entropy of the network's output, the decision boundaries are driven away from areas densely populated by learning samples. If the cluster assumption holds and there is no large overlap between the classes this method may increase the network's generalization ability. It acts equivalently to manifold learning and includes self-learning as a special case, as it increases the confidence of the classifier. The influence of the semi-supervised term is controlled by an appropriate weight, which is scaled relatively with the proportion of the unlabeled regions versus the annotated ones. Hence, the loss for a pixel x with output $\hat{\mathbf{y}}$ is:

$$\mathcal{L}(x, \hat{\mathbf{y}}) = \begin{cases} -\sum_{i=1}^{C} w_s^i y_i \log(\hat{y}_i), & \text{when} \mathbf{y} \text{is given} \\ -\alpha w_u \sum_{i=1}^{C} \hat{y}_i \log(\hat{y}_i), & \text{otherwise} \end{cases} \tag{4.1}$$

where $\mathbf{y}$ is the true label in one-hot encoding, C is the number of classes, w_s^i is the supervised weight for class i (which is inversely proportional to the number of samples of the class), α is a scaler and w_u the unsupervised weight.

The training procedure stops when the network does not significantly improve its performance on the validation set for 50 epochs. The performance is assessed in terms of weighted (balanced) accuracy, while an improvement is considered significant if the relative increase in performance is at least 0.5%. In order to artificially increase the volume of training data and avoid overfitting, we transformed the images using flips and rotations, which are considered label-preserving in this domain. The augmentation was performed online i.e. for each training image in each epoch, one operation out of all eight combinations of flip and rotate is randomly selected and applied.

4.3 Experimental Setup and Results

In this section, we first present the setup of the experiments conducted, followed by the corresponding results that justify the algorithmic choices of the proposed method and compare it to the state of the art.

4.3.1 Experimental Setup

Given the relatively small size of the dataset with respect to the diversity of the problem, we adopted a 5-fold cross validation (CV) scheme to ensure the validity of the results. The data splitting was performed per scan, so tissue from one case was never present in more than one set. Specifically, the 172 scans of the dataset were divided into five non-overlapping sets, with one of them having 36 and the rest 34 scans. Every time a model was tested on a specific set, the rest of the data were used for training. On average over all folds, the number of slices was 2060 for training and 515 for testing. As principal performance metric, we used the balanced accuracy (Equation 4.2), averaged over the five folds.

$$BACC = \frac{1}{N} \sum_{i=1}^{N} \frac{c_i}{n_i} \tag{4.2}$$

where N is the number of classes, c_i is the number of correctly classified samples of class i and n_i is the total number of samples of class i. Since the slices were only sparsely annotated, the accuracy was calculated over the areas of the scans where a ground truth was available.

In order to avoid extreme class imbalances between the different sets, data splitting was performed using a simple hill climbing technique that maximizes the entropy of the class distribution for the five sets. The methods started from an arbitrary split and then randomly swapped two cases between two sets in an attempt to find a more balanced solution. If the new solution has a higher class distribution entropy (averaged over the 5 sets), we retained it and repeated the procedure until no further improvement was possible.

To minimize the number of computations and memory requirements, we discarded part of the data that lack annotations. Hence, we cropped the left and right lung on each slice and used only the ones with relevant annotations as inputs of the networks. This is permitted by the fully-convolutional nature of the tested networks that do not require a fixed input size. For the cropping, we utilized the available lung mask, while a margin of 32 pixels was added on each side to provide context that could be useful to the networks.

The proposed method was implemented in Python[2] using the Keras framework[3] with the Theano [87] back-end. All experiments were performed under Linux OS on a machine with CPU Intel Core i7-5960X @ 3.50GHz, GPU NVIDIA GeForce Titan X, and 128 GB of RAM.

4.3.2 Results

Table 4.2 presents a comparison between different network configurations. The bold line corresponds to the proposed CNN, while the rest correspond to models that differ from the proposed in only one aspect, as specified in the first column. The rest of the columns provide the number of model parameters, the average inference time per (single-lung) slice, and the average balanced accuracy across the five validation sets.

The proposed model achieved top performance with accuracy nearly equal to 82% and inference time 58ms. The use of 64 kernels per layer instead of 32 did indeed improve the results, yet not significantly enough and with higher inference times, whereas the network with 16 kernels performed notably worse. On reducing the dilated convolutional layers from 10 to 9, we observed a relatively small reduction in the accuracy. However, we chose to keep 10 layers, since the difference in memory and time requirements was also small and because the resulting receptive field was comparable with that of the state of the art networks used for comparison. The use of semi-supervised learning yielded an improvement of nearly 1.5%, with no additional requirements in computational resources. We also

[2]https://github.com/intact-project/LungNet
[3]https://github.com/fchollet/keras

TABLE 4.2: Comparison of the different network configurations

Network configuration	Number of parameters $\times 10^5$	Average inference time ms	CV balanced accuracy %
w/o dilated convolutions	1.30	51	68.0
w/o concatenation	0.93	53	72.6
w/o InstanceNorm	1.29	38	77.9
w/o InstanceNorm skip	1.30	57	78.6
16 kernels/layer	0.47	51	79.2
Exponential dilation [78]	1.03	48	79.5
Purely supervised	1.30	58	80.6
9 dilated layers	1.18	53	81.3
Proposed	**1.30**	**58**	**81.8**
64 kernels/layer	4.23	82	82.1

performed an experiment with exponential increase in the dilation rates of the consecutive layers, similarly to [78] i.e. 1, 1, 2, 4, 8, 16, 32 and 64. The resulting model was smaller and faster, since 2 fewer layers were required to achieve similar receptive field, however the accuracy decreased by almost 3%. In the case where the convolutions were not dilated, the network performed poorly, because of the radical decrease of the receptive field. The accuracy of the proposed model without any normalization was substantially poorer, probably because it could not properly handle the contrast differences among the scans caused by different CT scanners and reconstruction kernels. The use of instance normalization improved the performance by adaptively normalizing the feature contrast for each input. However, this kind of normalization also normalizes the mean intensity that could be a useful feature. By adding the InstanceNorm skip connection (Figure 4.4), the accuracy improved even further. We speculate this is because the mean normalization is diminished, while the resulting variance normalization is only partially invariant to contrast shifts. Finally, omitting the concatenation of the first 10 layers also resulted in significant impairment of the results, which was expected since only 32 features are considered.

In Figure 4.5 the accuracy curves for different values of w_u are presented. These curves are generated by averaging over the five folds the best accuracies achieved this far by each model in each epoch. The curve for the model without the unsupervised learning was also included for comparison. The best performing configuration proved to be the one with $w_u = 0.1$, which we utilized for the training of the proposed model.

Table 4.3 presents a comparison between the proposed network and three previous studies. It has to be noted that all models used the same unsupervised weight ($w_u = 0.1$) and whenever batch normalization was performed, this was based on batch statistics (instance normalization) since this yielded the best results. Figure 4.6 illustrates a few segmentation results for each of the models in Table 4.3.

The first line of the table refers to our previous work [61], which has been converted into a fully convolutional network so it can accept arbitrarily sized images for input. Its low accuracy is probably due to the small receptive field (33×33) and the extensive pooling. This architecture was sufficient to describe the local texture of the 32×32 single-class patches

TABLE 4.3: Comparison with previous studies

Network	Number of parameters $\times 10^5$	Average inference time ms	CV balanced accuracy %
ILD-CNN [61]	0.9	237	72.2
Segnet [74]	335	111	73.6
U-net [76]	310	88	77.5
Proposed	**1.3**	**58**	**81.8**

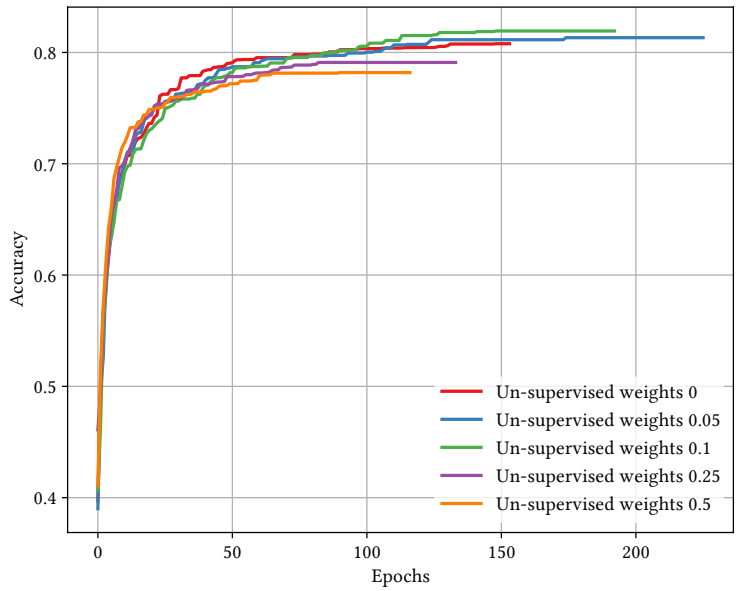


FIGURE 4.5: Accuracy curves for different values of w_u.

in [61], but could not capture higher level structure that is present in the whole-lung dataset of this study. The results of the model in Figure 4.6 show its noisy output near the lung boundaries or between patterns, where context information could be useful. Segnet [74] and U-net [76] yielded better results, with the latter being slightly faster and substantially more accurate. Both models have a very high number of parameters and large enough receptive fields to capture any relevant information. The superior performance of U-net could be attributed to its skip connections that allow features from the lower scales to directly contribute to its output. Indeed, Figure 4.6 illustrates the more detailed results of U-net as opposed to the overly smoothed areas produced by Segnet. Finally, the proposed network yielded the best results, while being faster and having far fewer parameters. The output examples in Figure 4.6 indicate that the proposed model manages to keep a better balance between fine details and smooth border among the different classes. Even thought it is really difficult to visually assess the performance of the system for the different classes, there are a few examples in Figure 4.6 with wrong classifications on which we can comment. Firstly, parts of the broncho-vascular tree in the third row were recognized as consolidation because of their similar densities, while accentuated terminal bronchial parts, that might be physiological as well, caused the erroneous classification of healthy areas into reticulation, in the first row. Some mistakes however are also attributed in the limited number of annotated classes. For example in row 6, there are emphysematic areas (dark area in the center of the lung) that have been annotated as healthy due to their similar density. Figure 4.7 shows the confusion matrix of the proposed model. As expected, many of the misclassifications occur between reticulation and honeycombing due to their similar textural appearance. Moreover, healthy tissue is often confused with reticulation probably because of the 2D sections of the bronchovascular tree that could resemble reticular patterns.

4.4 Conclusions

In this study, we proposed and evaluated a deep CNN for the semantic segmentation of pathological lung tissue on HRCT slices. The CNN is designed under a fully convolutional scheme and thus can handle variable input sizes, while it was trained in an end-to-end and semi-supervised fashion. The main characteristic of the proposed network is the use of dilated convolutions along with an instance variance normalization scheme, and multi-scale

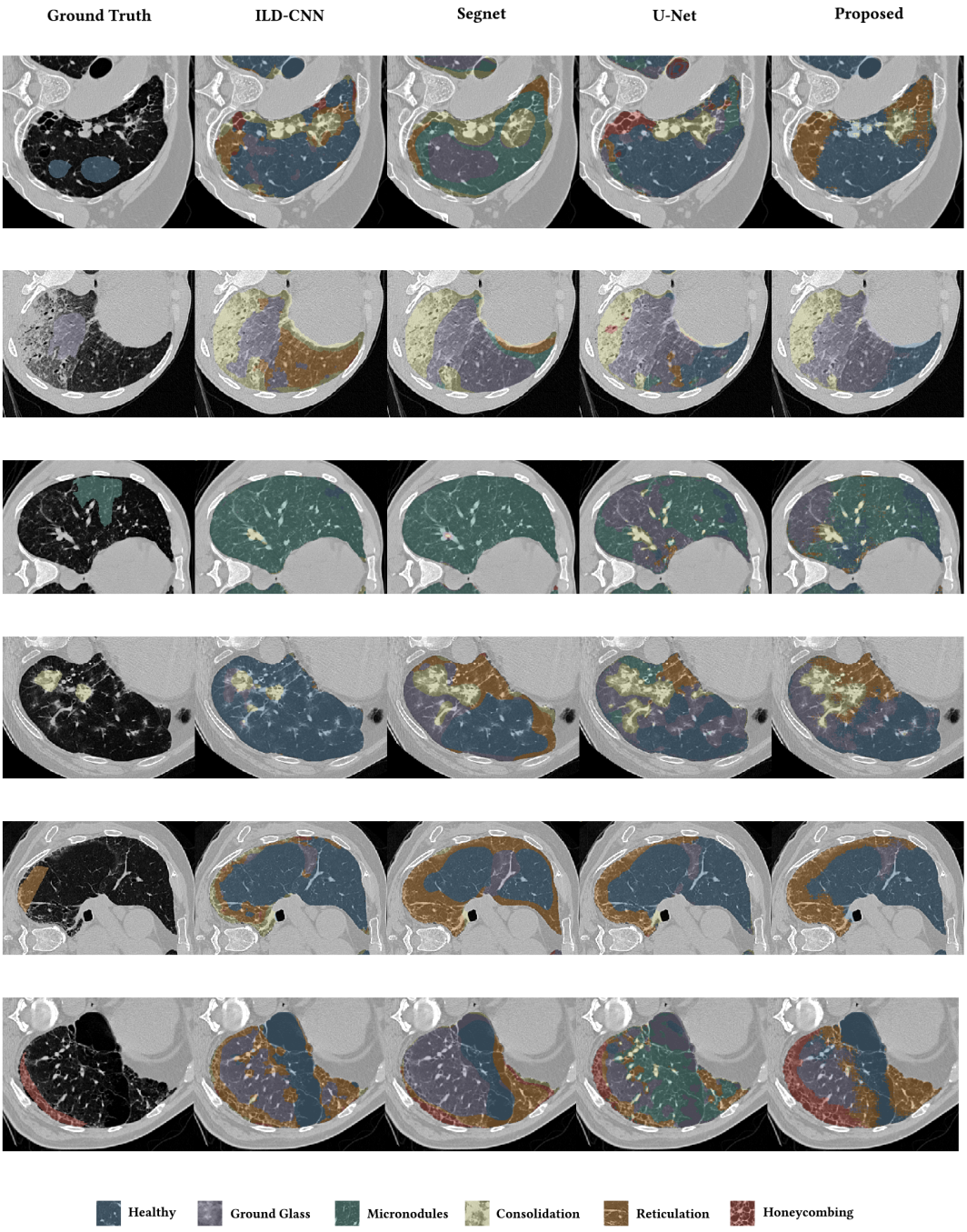


FIGURE 4.6: Output examples for the models of Table 4.3. From left to right: Ground Truth, ILD-CNN, Segnet, U-net, Proposed. Each example has a different pattern annotated. From top to bottom: Healthy (Blue), Ground Glass Opacity (Purple), Micronodules (Green), Consolidation (Yellow), Reticulation (Orange) and Honeycombing (Red).

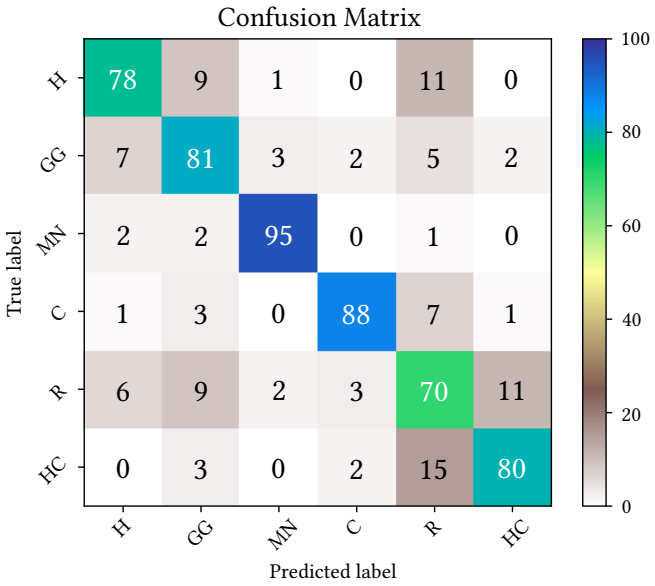


FIGURE 4.7: Confusion matrix of the proposed model as calculated over the cross validation scheme. The numbers represent percentages of pixels across all validation images.

feature fusion. The training and testing of the network was performed using a cross validation scheme on a dataset of 172 cases, whereas the split of the dataset into folds was performed per case. The proposed network surpassed the highest performance in previous studies, and is much more efficient in terms of memory and computation. Future work includes the modification of the model to consider the 3D nature of lung patterns, and to account for the bronchovascular tree. The former could be achieved by a direct extension of the architecture to 3D, similarly to 3D U-Net [88] and V-Net [89] or by employing a multi-planar view aggregation scheme, also referred to as 2.5D, [90]. Alternatively, a 3D post processing scheme could be used to refine the 2D segmentation output using conditional random fields or deformation models [91]–[93]. Finally, the result of a bronchovascular segmentation method could be utilized by the network to reduce false alarms.

Chapter 5

Computer Aided Diagnosis System for Idiopathic Pulmonary Fibrosis

This chapter borrows parts from:
Andreas Christe, Alan A Peters, Dionysios Drakopoulos, Thomai Stathopoulou, Stergios Christodoulidis, Marios M Anthimopoulos, Stavroula G Mougiakakou, Lukas Ebner, *"Computer-aided diagnosis of pulmonary fibrosis using deep learning and CT images,"* accepted for publication at Investigative Radiology.

It has been modified such that it highlights the contribution of the author. In detail, the main contribution of the author lies on the segmentation components, integration and CAD output generation. Ms Thomai Stathopoulou assisted in the anatomy segmentation component of the pipeline while the experimental setup and results have been performed by the rest of the authors.

Computer-aided diagnosis (CAD) systems could enhance the diagnostic performance of radiologists for idiopathic pulmonary fibrosis. In this study, we introduce and evaluate an end-to-end system for the automatic classification of high resolution computed tomography images into four radiological diagnostic categories. The proposed CAD system consists of a sequential pipeline in which at first the anatomical structures of the lung are segmented, then the pathological lung tissue is identified and finally by combining these information a final radiological diagnosis is reached using a random forest classifier. The experimental results show the potential of utilizing a CAD system for this task, while also sets a path for further development and investigation.

5.1 Field of Study

5.1.1 Diagnosis of Idiopathic Pulmonary Fibrosis

Idiopathic interstitial pneumonias (IIPs) is a sub group of ILDs with causes unknown to the medical community. Idiopathic pulmonary fibrosis or usual interstitial pneumonia (IPF/UIP) along with the non specific interstitial pneumonia (NSIP) are the most prevalent IIPs and together they account for the 80% of IIPs (Figure 5.1). Any patient with suspected ILD is questioned about his/her clinical history (first symptoms, environmental exposures, family history) and undergoes thorough physical examination (presence of crackles, finger clubbing, joint swelling, or tight skin), chest radiography, pulmonary function testing and usually high resolution computed tomography (HRCT). If the association with systemic diseases or causative agents such as: (i) long-term exposure to hazardous materials (e.g. asbestos, fumes, and gases), (ii) certain drugs or medications, (iii) infections, (iv) genetic abnormalities and (v) autoimmune diseases is ruled out, then a differential diagnosis among the IIP categories is carried out.

The differential diagnosis of IIPs is greatly based on a number of uniform criteria and guidelines that have been proposed by the American Thoracic Society (ATS) and the European Respiratory Society (ERS) in [3] and have been recently updated by the Fleischener Society recommendations [4]. Table 5.1 holds the radiological patterns and how these associate with the IPF diagnosis. Typicaly, radiologists screen the patient's HRCT for these UIP patterns (i.e. Typical UIP, Probable UIP, indeterminate for UIP and non-IPF) and along

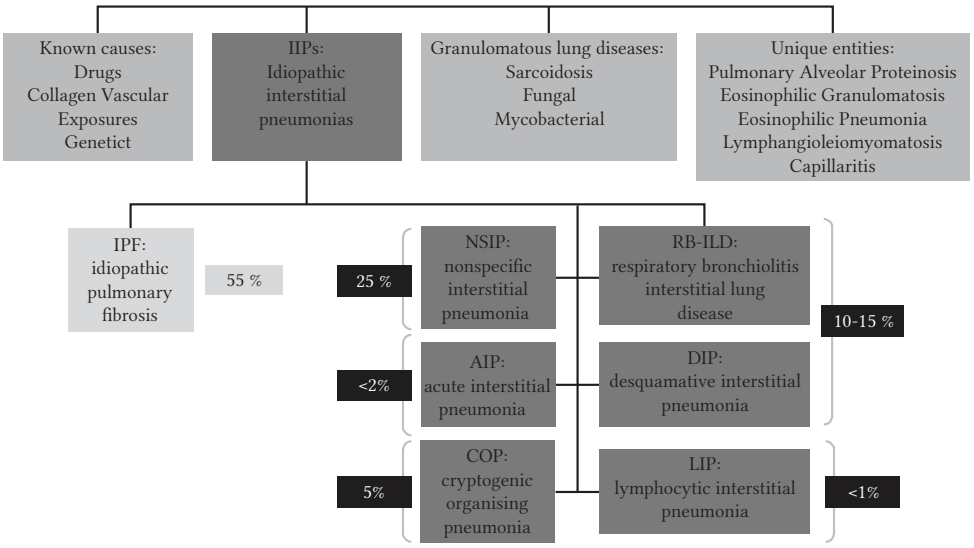


FIGURE 5.1: Classification of interstitial lung diseases [94].

with a clinical board of pneumonologists and histopathologists they decide on the diagnosis. In most of the cases, if any other than typical UIP pattern is identified, additional invasive procedures, such as transbronchial or surgical biopsy are required for the final diagnosis.

It is important to note here that the term CT patterns as presented in the IPF diagnosis guidelines (Table 5.1) should not be confused with textural patterns of pathological tissue that have been mentioned also in the previous chapters. These diagnostic CT patterns holds not only information about the distribution of the pathological tissue types (Features), but also their spatial distribution in the lung cavity (Characteristics). Identifying such CT patterns in HRCTs is considered to be a difficult task among radiologists and high inter-reader variability is reported [5].

The correct pattern identification on CT images plays a central role in the diagnosis and further treatment of patients with ILD, even more in conjunction with the recently proposed Fleischner recommendations. The aforementioned guidelines aim to expand the role of CT for the diagnosis of IPFs. However, the pattern recognition on CT can be challenging, even for subspecialized chest radiologists. Thus, supporting tools, such as CAD system for CT pattern detection and disease classification, have the potential to improve the radiological diagnosis of IPF. In particular, the combination of advanced segmentation algorithms, clinical background data and machine learning provides the opportunity for the development of a lung fibrosis CAD system.

5.1.2 Contribution

The purpose of this study was to assess the performance of a CAD system for the automatic classification IPF cases into radiological diagnostic CT patterns, based on HRCT chest images and clinical markers.

5.2 Materials and Methods

5.2.1 Databases

For the purposes of this study multiple databases were used for the training and evaluation of the different components of the system. More details about each database follow in the next paragraphs.

TABLE 5.1: Diagnostic categories of UIP, based on CT patterns [4]

	Features	Characteristics
Typical UIP CT pattern	Reticulation, Honeycombing, Absence of features suggesting a non-IPF diagnosis	Basal and subpleural
Probable UIP CT pattern	Reticulation, Absence of features suggesting a non-IPF diagnosis	Basal and subpleural
CT pattern **indeterminate** for UIP	Reticulation with inconspicuous features suggestive of a non-UIP pattern	Viariable or diffuse
CT features most consistent with a **non-IPF** diagnosis	Any of the following: predominant consolidation, extensive pure ground glass opacity (without acute exacerbation), extensive mosaic attenuation with extensive sharply defined lobular air trapping on expiration, diffuse nodules or cysts	Upper-, mid-lung or peribronchovascular predominant or subpleural sparing

Lung Tissue Research Consortium database

The Lung Tissue Research Consortium database (LTRC-DB)[1] is a resource program of the National Heart, Lung, and Blood Institute (NHLBI) that provides CT scans, as well as biospecimens to qualified investigators for use in their research. The LTRC was originally created in 2005 by the National Institutes of Health and is composed of four clinical centers from around the United States: Mayo Clinic Rochester, University of Michigan – Ann Arbor, University of Pittsburgh, and Temple University. Each center contributes to the recruitment and enrollment of protocol-eligible participants, as well as the procurement of the data. This library contains cases with different lung diseases along with annotations of the lung parenchyma, the airways and pathological tissue. The cases with proven ILD diagnosis are more than 100.

Multimedia database of ILDs

The multimedia database for ILDs (MD-ILD) [33] [2] was developed within the framework of the Talisman project at the University Hospital of Geneva and is made publicly available. The database consists of HRCT image series of 10mm slice spacing with annotated regions of pathological lung tissue and the lung parenchyma along with clinical parameters from patients with pathologically proven diagnoses of ILDs. Specifically, the library contains cases from 128 patients affected with one of the 13 histological diagnoses of ILDs, 108 image series with more than 41 liters of annotated lung tissue patterns as well as a comprehensive set of 99 clinical parameters related to ILDs.

Inselspital ILDs database

The Insel spital ILD database (INSEL-DB) have been created within the framework of this thesis and consists of 105 HRCT lung scans provided by the ILD board of the Bern university hospital, where pneumologists, radiologists and pathologists have diagnosed the

[1] https://ltrcpublic.com/
[2] http://medgift.hevs.ch/wordpress/databases/ild-database/

patients according to the international guidelines [4]. All diagnoses were ILD board consent diagnoses (radiologically or histologically proven cases). The cases consisted of 54 NSIP and 51 IPF cases. CT scans were retrospectively collected with irreversible data anonymization from October 2015 to June 2017. Images have been acquired with patients in the supine position, from the apex of the lung to the costodiaphragmatic recess with a slice thickness of 1mm. Moreover, any associated clinical and biochemical data (e.g. gender, age, smoking history, duration of illness, lung function tests, results of blood tests) were gathered in order to investigate whether they correlate with the actual diagnosis of each case, so they can provide information additional to the radiological data. Institutional board approval for the diagnosis was waived due to the retrospective collection of the patients.

The median age in the UIP group was 70 years (range, 49 to 84 years), and the group consisted of 11 female and 40 male patients. This group included 38 IPF cases, 8 cases of rheumatoid arthritis and 5 connective tissue disease patients, with accompanying pulmonary fibrosis and UIP patterns. In the NSIP group, the median age was 64 years (range, 38 to 83 years), with 20 female and 34 male patients, consisting of 5 idiopathic NSIP patients and 49 cases with the known etiology of NSIP: 24 hypersensitivity pneumonitis, 7 anti-synthetase syndrome, 6 medication related, 4 rheumatoid arthritis, 3 systemic sclerosis, 3 sarcoidosis and 2 Sjögren patients

Two chest radiology specialists classified the cases into the 4 UIP CT patterns of Table 5.1 through consensus, according to the Fleischner Society recommendations [4], to establish the ground truth. The radiologists first reviewed and classified all cases independently and then met to discuss the cases without agreement to determine the classification through consensus. The inter-reader agreement was substantial as shown in the results section. This radiological consensus represented the ground truth for further calculations.

Moreover, a radiologist with 10 years (Reader 1) experience in chest imaging and a chest fellow with 4 years (Reader 2) of experience read the images on a Picture Archiving and Communication System (PACS). Lung windows settings were used to read the hard kernel reconstructions (I70f). Both radiologists were blinded to the diagnoses and had to classify the cases into the 4 categories.

5.2.2 Anatomy Segmentation

For the anatomy segmentation a well established algorithm from the literature were utilized to segment the airways and the lung parenchyma. This algorithm have been implementled in house and its hyper-parameters were defined using the LTRC-DB. This algorithm does not include a training component and it is based on simple region growing, thresholding and morphological operations. In more detail the pipeline is based on the publications [10], [95], [96] and consists of the following steps: (1) Extraction of large airways, (2) Segmentation of lung regions; (3) Separation of the left and right lungs; (4) Smoothing.

As far as the airway segmentation is concerned, a seed point from the top axial slices of the scan initializes a region growing technique. The seed point is picked from a region within the center of the top axial slices that have average hounsfield units (HU) below a threshold. The seed point for the region growing process is the voxel with the lowest HU in the aforementioned region. The region growing process will iteratively build the trachea and the main stem bronchi while it will stop based on an explosion detection scheme [97]. After the large airways have been detected, the lung are segmented using also a region growing technique. As seed point for this process the voxel with the lowest HU within the airways is used while the optimal threshold value is set using the Otsu thresholding technique. In order to seperate the left and right lungs the airways are removed from the lung segmentation and connected component analysis is used to find the two lungs. Left and right lungs are identified using the center of gravity and the orientation of the scan. In some cases, the left and right lungs are really close in some parts, in such cases it is particularly difficult for the gray-scale thresholding to separate the two lungs. For this reason we utilize a dynamic programming approach similar to the one described in [10]. For the last step, each lung is smoothed separately using 3D morphological operations. A few examples of the output of the anatomy segmentation algorithm are presented in Figure 5.2.

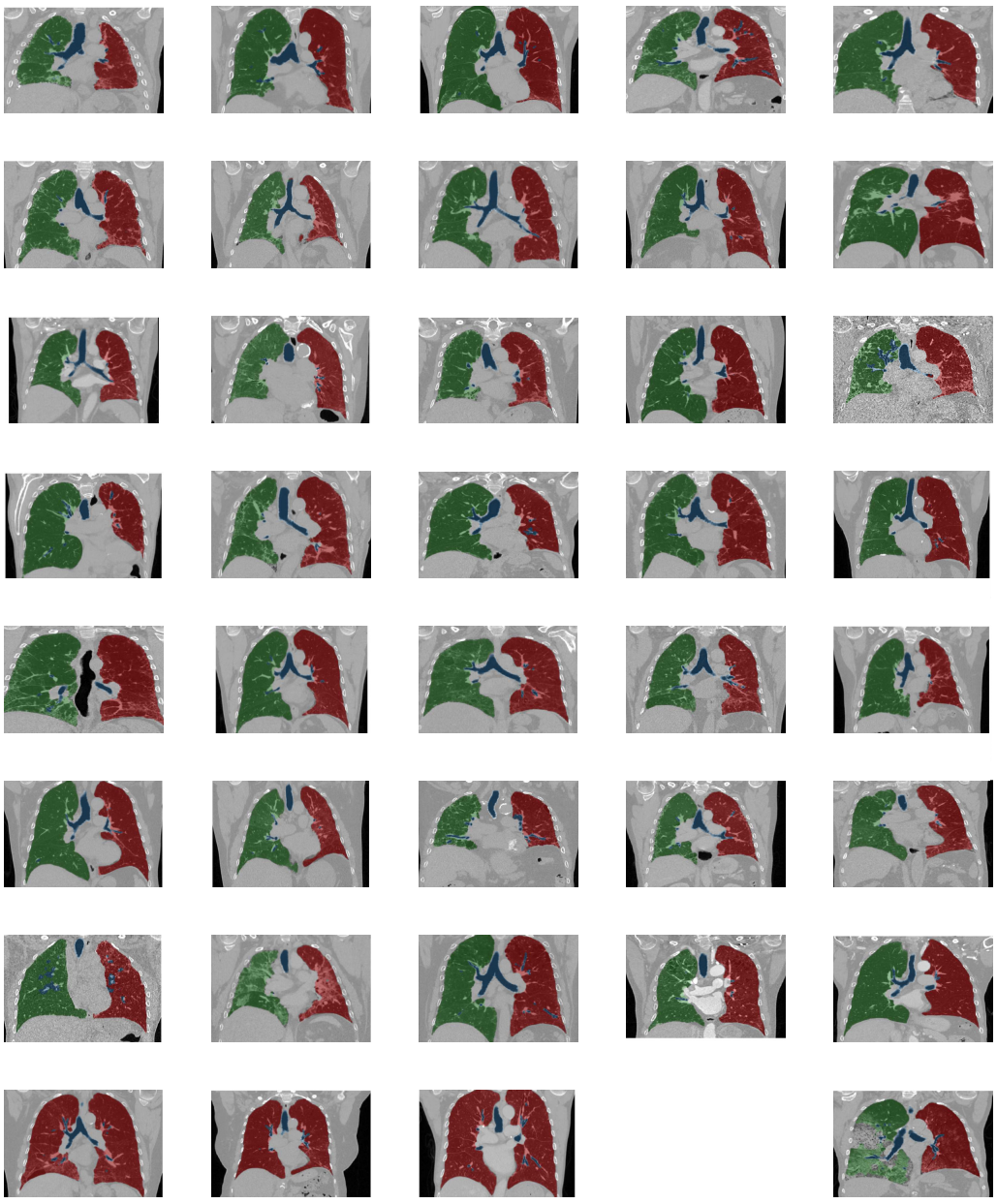


FIGURE 5.2: Examples of the output of the anatomy segmentation algorithms. Green denotes right lung, red denotes left lung and blue denotes the airways. In the last row a few examples where the left-right split did not succeed are presented, while also an under segmentation example is shown.

5.2.3 Tissue Characterization

After having acquired the segmentation of the lung parenchyma a convolutional neural network (CNN) for the quantification of the pathological lung tissue is utilized. For the purposes of implementation, training and evaluation of this step a mixture of the MD-ILD and INSEL-DB was used.

ILD consist of an admixture of the following basic tissue pathologies: reticulation, honeycombing, GGO, consolidation and normal lung. Our proposed system employs texture classification schemes to detect, classify and calculate the extent of pathological tissue on CT images. The suggested lung disease quantification scheme takes as input a section of a 2D CT slice of interest and uses a purely CNN scheme to calculate a corresponding label map with a single tissue class for each pixel. The proposed architecture is designed in such a way that the pixels are the training samples instead of the CT images [98]. Thus, the number of training samples is of the order of 106 and therefore the training of such deep network is possible. We adopted a 5-fold cross validation (CV) scheme to ensure the validity of the results stratified on apatient level. The balanced accuracy of the proposed CNN averaged over all folds was 81.8%. More details on the implementation of this step are already presented with more detail in Chapter 4.

5.2.4 Diagnosis Support

The diagnosis support module is the final step of the pipeline in which all previous outputs are aggregated to achieve a final diagnosis. The INSEL-DB along with the associated clinical parameters (age, gender, sex, etc.) was used for the evaluation of this step.

In order to calculate the distribution of the different pathological tissue types in the different areas of the lung an additional step was employed that divided each lung in 6 regions (upper, middle, lower, basal, peripheral). For this step a volume based split was utilized for the upper, middle and lower segmentation, while for the basal and peripheral segments a k-means clustering was utilized that was applied on the distances from the center of mass. The intersection of the aforementioned segments produces a total of 12 segments in both lungs. The distribution of pathological tissue as estimated by the tissue characterization CNN is calculated over each segment and these are used as features to train multiple one-vs-all random forest classifiers to classify the lung fibrosis for each case into 1) a typical UIP CT pattern, 2) a probable UIP CT pattern, 3) a CT pattern indeterminate for UIP, and 4) CT features that are most consistent with a nonIPF diagnosis (Table 1, [4]).

The different steps of the CAD system required approximately 6 minutes per case for the scan to be properly processed and for a diagnosis to be available. In Figure 5.3, a radial histogram visualization is presented. For all calculations, we used a CPU implementation running on an Intel Core i7-5960X CPU, except for the tissue characterization, for which we used a NVIDIA GeForce GTX TITAN GPU. More specifically, the steps needed on average the following times:

- lung and airways segmentation, 47.1 seconds,
- ILD pathology quantification, 192.4 seconds and
- diagnosis support, 124.4 seconds.

5.3 Experimental Setup and Results

5.3.1 Statistical Analysis Tools

Sensitivity, specificity, accuracy and positive predictive values were calculated for the readers and the proposed CAD system, using the independent chest radiology experts' consensus classification (1-4) as the ground truth. Positive predictive value (precision) and sensitivity (recall) were used to calculate the F-score (harmonic mean for precision and recall).

McNemar's test was used to compare the sensitivity, specificity and accuracy between the readers and the CAD system. Comparison of proportions was used to compare the

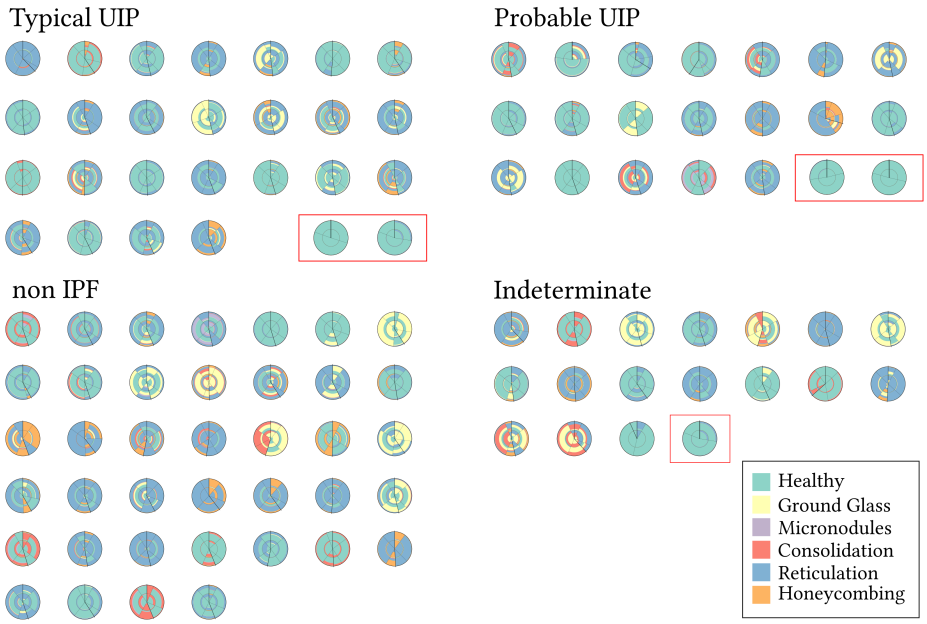


FIGURE 5.3: Interstitial lung disease visualization using radial histograms similar to the one used in [99]. Each sector denotes a region of the lung, and it is split in two parts, one for the basal (inner) and one for peripheral (outer). The color scheme denotes the pathological tissue. Solid lines denote the division of left and right lungs. The radial histograms with the red border, are cases where the anatomy segmentation failed to split the left and right lungs.

Fscores. The significance level was set to 0.05. MedCalc® version 15.0 (MedCalc Software, Ostend, Belgium) was utilized. The 4 categories were first analyzed in an un-pooled fashion for all entries. Then, the 4 groups were pooled into 2 categories: a) cases needing a biopsy for further diagnostics, according to the Fleischner Society white paper [4] (groups 3 and 4), and b) cases without further need of diagnostics (groups 1 and 2). A separate analysis of the correct classification into these two groups was performed for the findings of the readers and the machine. Furthermore, interobserver agreement between the radiologists and between the radiologists and the machine was performed individually by calculating the weighted Kappa, as follows: slight (0-0.2), fair (0.21-0.4), moderate (0.41-0.6), substantial (0.61-0.8) and almost perfect agreement (0.81-1).

5.3.2 Results

Fleischner Classification

Reader 1, Reader 2 and CAD demonstrated the same accuracy for classifying the pulmonary fibrosis, according to the Fleischner society guidelines [4]: 0.6, 0.54 and 0.56, respectively, with p-values >0.45. The CAD system achieved an F-score (harmonic mean for precision and recall) of 0.56, while the two readers in average achieved 0.57 (pvalue=0.991).

Fleischner Pooled Classification

When the 4 classification groups were pooled into the group requiring further work-up (groups 3 and 4) and into the group for which the diagnosis was clear without biopsy (UIP group 1 and 2), the accuracy increased. Reader 1, Reader 2 and the CAD scored similar accuracies, 0.81, 0.70 and 0.81, respectively. Reader 2 was slightly inferior (without reaching the level of statistical significance) to Reader 1 and CAD with p-values of 0.059 and 0.189, respectively. The sensitivities for choosing the cases that needed further work-ups were similar for Reader 1, Reader 2 and CAD at 0.86, 0.84 and 0.79, respectively, and p values >0.39. The CAD system demonstrated the best specificity; the specificities for Reader 1, Reader 2 and

CAD were 0.63, 0.44 and 0.67, respectively. The CAD system performed significantly better than Reader 2 (p-value=0.012) and equally well as Reader 1 (pvalue=0.773). The specificity of Reader 1 was significantly higher than that of Reader 2 (pvalue=0.037). The F-score was again similar between the CAD system and the readers. The CAD and the average reader F score was 0.80 and 0.79 (p-value=0.898), respectively.

Inter-observer Agreement

Reader 1 vs. Reader 2 demonstrated fair inter-reader agreements, with a weighted kappa ($\pm$standard error) of 0.30 ± 0.08 for group 4 (unpooled) and 0.3 ± 0.1 for group 2 (pooled) classifications. The CAD system vs. Reader 1 demonstrated moderate classification agreement, with a weighted kappa of 0.47 ± 0.08 and 0.54 ± 0.1, respectively, for the unpooled and pooled classifications. Compared with Reader 2, the agreement was only fair: 0.33 ± 0.08 and 0.3 ± 0.09, respectively. The inter-reader agreement of the two chest experts, who set the ground truth, was substantial: weighted kappa was 0.63 ± 0.08 and 0.66 ± 0.1 for the unpooled and pooled classifications, respectively

5.4 Discussion

The accuracy of the proposed CAD system for dichotomous classification into the group needing further intervention and the group without the need for further work-ups was as good as that of the radiologists. The automated system even outperformed the unexperienced radiologist, in terms of the specificity for patient identification requiring subsequent intervention. Demonstrating the best specificity means having the lowest false positive rate in the group undergoing further work-ups. This rate is important, considering the high mortality and morbidity of surgical wedge resection of the lung on the one hand, and the low operability of this group on the other hand [100]. Our results support the beneficial implementation of a computer-aided diagnostic algorithm. As dedicated chest radiologists are scarce, and interstitial pulmonary fibrosis, in particular IPF, is almost considered to be an orphan disease, it is difficult to build the required expertise in this field. Under these circumstances, the importance of CAD solutions becomes evident, highlighting the importance of such a CAD system.

There has been much discussion of whether the Fleischner Society white paper on classifying pulmonary fibrosis into 4 groups and recommending interventions in only two of these groups should be accepted. Some even feel that biopsy is needed for 3 groups (including the group with a probable UIP CT pattern). The authors believe that the number of biopsies can be even further reduced in a proper setting of multidisciplinary ILD boards, with follow-up of these patients. The pattern recognition method is certainly the right approach for classifying the probability of IPF today, however, with increasing precision and recall of CAD for ILD, it may be possible to skip this classification to directly call the disease by its name.

The accuracy of all readers (CAD inclusive) between 0.5 and 0.6 for the unpooled classification is not particularly high, but is still substantially above the level chance, as 4 classes were considered. In practice, the differentiation between groups 1 and 2 is less important than the differentiation between group 2 and 3, as the diseases are labeled IPF for both group 1 and 2. Apparently, the positive predictive value for IPF is 80% in group 2, therefore, a biopsy is not recommended for the first two groups [4]. For all other cases (groups 3 and 4), a biopsy would be needed. Therefore, it is important to differentiate between those patients with and without the need for biopsy. To appreciate the performance of the readers and the CAD system, one has to examine the results for the dichotomous differentiation into the group with and without the need for biopsy; the accuracy rate of 0.81 more appropriately represents the performance of both the radiologists and the machine. This accuracy is comparable with the results published by Depeursinge [101].

In addition, the inter-reader agreement was only fair for the 4x4 table classification, but it increased to moderate in the 2×2 classification pooling, which is acceptable and comparable to previous publications [101]. As radiologists and the CAD system did not have the same

false positive and false negative cases, there is also the potential of CAD to help radiologists classify the UIP pattern.

Note that our study suffers from several limitations. The number of cases was low. For more significant results, we are currently conducting a larger scale study with CAD improvements. However, given the low prevalence of interstitial lung fibrosis, these results are of considerable impact. Furthermore, only cases of pulmonary fibrosis were covered in this study due to the fine-tuning of our CAD towards fibrosis; a pattern extension for the algorithm will be implemented in later versions. Furthermore, we included the idiopathic forms of pulmonary fibrosis, and we also allowed for diseases with associated pulmonary fibrosis to be included; which may have confounded these results.

5.5 Conclusions

In this study, we present a preliminary evaluation of the integrated pipeline for the automatic classification of IPF. A multi-step approach is employed for the segmentation of anatomical structures of the lung cavity, the segmentation of lung pathological tissue and finaly the classification into a radiological diagnostic CT pattern. In conclusion, we found that a machine learning-supported computer aided detection algorithm was able to classify idiopathic pulmonary fibrosis with similar accuracy as a human reader. Moreover, the computer algorithm delivered results comparable to those of radiologists when grouping fibrosis patterns according to the Fleischner Society's newest recommendation.

Chapter 6

Linear and Deformable Medical Image Registration

This chapter is a modified version of:
Christodoulidis Stergios, Sahasrabudhe Mihir, Vakalopoulou Maria, Chassagnon Guillaume, Revel Marie-Pierre, Mougiakakou Stavroula, Paragios Nikos, *"Linear and Deformable Image Registration with 3D Convolutional Neural Networks,"* in Reconstruction and Analysis of Moving Body Organs 2018, MICCAI Workshop.
DOI: 10.1007/978-3-030-00946-5_2

This work was conducted during a short research visit in the Centre de Vision Numérique in Centrale Supélec in Paris, France. The main contribution of the author lies on the implementation and evaluation of the 3D registration convolutional neural network. Dr Maria Vakalopoulou assisted with the baseline comparisons and Mr. Mihir Sahasrabudhe with the evaluation of the clinical relevance of the deformation.

Image registration and in particular deformable registration methods are pillars of medical imaging. Inspired by the recent advances in deep learning, we propose in this chapter, a novel convolutional neural network architecture that couples linear and deformable registration within a unified architecture endowed with near real-time performance. Our framework is modular with respect to the global transformation component, as well as with respect to the similarity function while it guarantees smooth displacement fields. We evaluate the performance of our network on the challenging problem of MRI lung registration, and demonstrate superior performance with respect to state of the art elastic registration methods. The proposed deformation (between inspiration & expiration) was considered within a clinically relevant task of interstitial lung disease (ILD) classification and showed promising results.

6.1 Field of Study

6.1.1 Medical Image Registration

Image registration is the process of aligning two or more sources of data to the same coordinate system. Through all the different registration methods used in medical applications, deformable registration is the one most commonly used due to its richness of description [102]. The goal of deformable registration is to calculate the optimal non-linear dense transformation G to align in the best possible way, a source (moving) image S to a reference (target) image R [103], [104]. Existing literature considers the mapping once the local alignment has been performed and therefore is often biased towards the linear component. Furthermore, state of the art methods are sensitive to the application setting, involve multiple hyper-parameters (optimization strategy, smoothness term, deformation model, similarity metric) and are computationally expensive.

Medical image registration is a really important process for a variety of tasks in the clinical practice such as comparing multi-modal images, follow-up examinations, differences between anatomical structures between different patients and more. There is an extended literature on how deformation information can be used either for diagnostic or therapeutic

reasons (Reviewed in [105]–[107]). In [108] the authors utilize an elastic registration algorithm in pulmonary MRIs for the assessment of pulmonary fibrosis in patients with systemic sclerosis. Moreover, a number of studies (e.g. [109], [110]) utilize registration in order to estimate lung ventilation as a biomarker.

Recently, deep learning methods have gained a lot of attention due to their state of the art performance on a variety of problems and applications [111], [112]. In computer vision, optical flow estimation—a problem highly similar to deformable registration—has been successfully addressed with numerous deep neural network architectures [113]. In medical imaging, some methods in literature propose the use of convolutional neural networks (CNNs) as robust methods for image registration [114], [115]. More recently, adversarial losses have been introduced with impressive performance [116]. The majority of these methods share two limitations: *(i)* dependency on the linear component of the transformation and *(ii)* dependency on ground truth displacement which is used for supervised training.

6.1.2 Contribution

In this chapter, we address the previous limitations of traditional deformable registration methods and at the same time propose an unsupervised method for efficient and accurate registration of 3D medical volumes that determines the linear and deformable parts in a single forward pass. The proposed solution outperforms conventional multi-metric deformable registration methods and demonstrates evidence of clinical relevance that can be used for the classification of patients with ILD using the transformation between the extreme moments of the respiration circle. Moreover, we utilize the estimated deformations as a radio-marker and we evaluate its clinical relevance.

In summary, the main contributions of the study are fourfold: *(i)* coupling linear and deformable registration within a single optimization step / architecture, *(ii)* creating a modular, parameter-free implementation which is independent of the different similarity metrics, *(iii)* reducing considerably the computational time needed for registration allowing real-time applications, *(iv)* associating deformations with clinical information.

6.2 Materials and Methods

In this study, we propose the use of an unsupervised CNN for the registration of pairs of medical images. A source image S and a reference image R are presented as inputs to the CNN while the output is the deformation G along with the registered source image D. This section presents details of the proposed architecture as well as the dataset that we utilized for our experiments. Please note that henceforth, we will use the terms *deformation*, *grid*, and *transformation* interchangeably.

6.2.1 Linear and Deformable 3D Transformer

One of the main components of the proposed CNN is the 3D transformer layer. This layer is part of the CNN and is used to warp its input under a deformation G. The forward pass for this layer is given by

$$D = \mathcal{W}(S, G), \tag{6.1}$$

where $\mathcal{W}(\cdot, G)$ indicates a sampling operation $\mathcal{W}$ under the deformation G. G is a dense deformation which can be thought of as an image of the same size as D, and which is constructed by assigning for every output voxel in D, a sampling coordinate in the input S.

In order to allow gradients to flow backwards though this warping operation and facilitate back-propagation training, the gradients with respect to the input image as well as the deformation should be defined. Similar to [117], such gradients can be calculated for a backward trilinear interpolation sampling. The deformation is hence fed to the transformer layer as sampling coordinates for backward warping. The sampling process is illustrated by

$$D(\vec{p}) = \mathcal{W}(S, G)(\vec{p}) = \sum_{\vec{q}} S(\vec{q}) \prod_{d} \max\left(0, 1 - \left\| [G(\vec{p})]_d - \vec{q}_d \right\|\right) \tag{6.2}$$

where $\vec{p}$ and $\vec{q}$ denote pixel locations, $d \in \{x, y, z\}$ denotes an axis, and $[G(\vec{p})]_d$ denotes the d-component of $G(\vec{p})$.

Our modeling of the deformation G offers a choice of the type of deformation we wish to use—linear, deformable, or both. The linear (or affine) part of the deformation requires the prediction of a 3×4 affine transformation matrix A according to the relation $[\hat{x}, \hat{y}, \hat{z}]^T = A[x, y, z, 1]^T$, where $[x, y, z, 1]^T$ represents the augmented points to be deformed, whereas $[\hat{x}, \hat{y}, \hat{z}]^T$ represents their locations in the deformed image. The matrix A can then be used to build a grid, G_A, which is the affine component of the deformation G.

To model the deformable part G_N, a simple and straightforward approach is to generate sampling coordinates for each output voxel ($G_N(\vec{p})$). We can let the network calculate these sampling points directly. Such a choice would however require the network to produce feature maps with large value ranges which complicates training. Moreover without appropriate regularization, non-smooth and even unconnected deformations could be produced. In order to circumvent this problem, we adopt the approach proposed by [118] and predict spatial gradients Φ of the deformation along each dimension instead of the deformation itself. This quantity measures the displacements of consecutive pixels. By enforcing these displacements to have positive values and subsequently applying an integration operation along each dimension, the spatial sampling coordinates can be retrieved. This integration operation could be approximated by simply applying a cumulative sum along each dimension of the input (i.e integral image). In such a case, for example, when $\Phi_{\vec{p}_d} = 1$ there is no change in the distance between the pixels $\vec{p}$ and $\vec{p} + 1$ in the deformed image along the axis d. On the other hand, when $\Phi_{\vec{p}_d} < 1$, the distance between these consecutive pixels along d will decrease, while it will increase when $\Phi_{\vec{p}_d} > 1$. Such an approach ensures the generation of smooth deformations that avoid self-crossings, while allows the control of maximum displacements among consecutive pixels.

Finally, to compose the two parts we apply the deformable component to a moving image, followed by the linear component. When operating on a fixed image S, this step can be written as

$$\mathcal{W}(S, G) = \mathcal{W}\left(\mathcal{W}(S, G_N), G_A\right). \tag{6.3}$$

During training, the optimization of the decoders of A and G_N is done jointly, as the network is trained end-to-end. We also impose regularization constraints on both these components. We elaborate on the importance of this regularization for the joint training in Section 6.2.3.

6.2.2 Architecture

The architecture of the CNN is based on an encoder-decoder framework presented in [98] (Figure 6.1). The encoder adopts dilated convolutional kernels along with multi-resolution feature merging, while the decoder employs non-dilated convolutional layers and up-sampling operations. Specifically, a kernel size of $3 \times 3 \times 3$ was set for the convolutional layers while LeakyReLU activation was employed for all convolutional layers except the last two. Instance normalization was included before most of the activation functions. In total five layers are used in the encoder and their outputs are merged along with the input pair of image to form a feature map of 290 features with a total receptive field of $25 \times 25 \times 25$. In the decoder, two branches were implemented—one for the spatial deformation gradients and the other for the affine matrix. As far as the former is concerned, a squeeze-excitation block [119] was added in order to weigh the most important features for the spatial gradients calculation while for the latter a simple global average operation was used to reduce the spatial dimensions to one. For the affine parameters and the spatial deformation gradients, a linear layer and sigmoid activation were respectively used. Finally to retrieve Φ, the output of the sigmoid function should be scaled by a factor of 2 in order to fall in the range $[0, 2]$ and hence allow for consecutive pixels to have larger distance than the initial.

6.2.3 Training

The network was trained by minimizing the mean squared error (MSE) between the R and D image intensities as well as the regularization terms of the affine transformation parameters

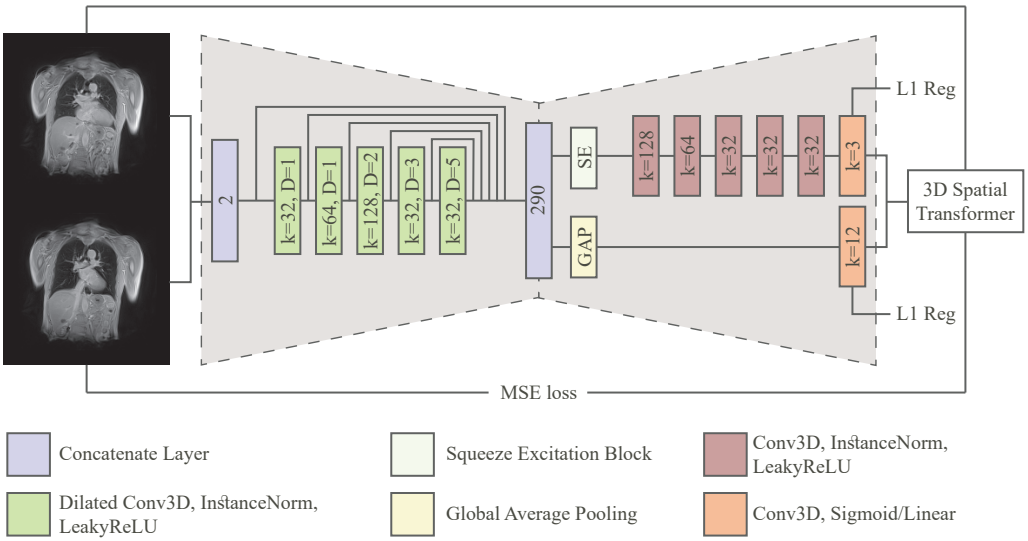


FIGURE 6.1: The overall CNN architecture. The network uses a pair of 3D images and calculates the optimal deformations from the one image to the other.

and the spatial deformation gradients using the Adam optimizer [39]. Our loss is defined as

$$\text{Loss} = \|R - \mathcal{W}(S, G)\|^2 + \alpha \|A - A_I\|_1 + \beta \|\Phi - \Phi_I\|_1 , \tag{6.4}$$

where A_I represents the identity affine transformation matrix, Φ_I is the spatial gradient of the identity deformation, and α and β are regularization weights. As mentioned before, regularization is essential to the joint optimization. To elaborate, without the L1 regularization on A, the network might get stuck in a local minimum where it aligns only high-level features using the affine transformation. This will result in a high reconstruction error. On the other hand, without the smoothness regularizer on Φ, the spatial gradients decoder network can predict very non-smooth grids which again makes it prone to fall in a local minimum. Having both linear and deformable components is helpful to the network because these two components now share the work. This hypothesis aligns with [118] and is also evaluated in Section 6.3.

The initial learning rate is 10^{-3} and subdued by a factor of 10 if the performance on the validation set does not improve for 50 epochs while the training procedure stops when there is no improvement for 100 epochs. The regularization weights α and β were set to 10^{-6} so that neither of the two components has an unreasonably large contribution to the final loss. As training samples, random pairs among all cases were selected with a batch size of 2 due to the limited memory resources on the GPU. The performance of the network was evaluated every 100 batches, and both proposed models—with and without affine components—converged after nearly 300 epochs. The overall training time was calculated to $\sim$ 16 hours.

6.2.4 Dataset

MRI exams were acquired as a part of a prospective study aiming to evaluate the feasibility of pulmonary fibrosis detection in systemic sclerosis patients by using magnetic resonance imaging (MRI) and an elastic registration-driven biomarker. This study received institutional review board approval and all patients gave their written consent. The study population consisted of 41 patients (29 patients with systemic sclerosis and 12 healthy volunteers). Experienced radiologists annotated the lung field for the total of the 82 images and provided information about the pathology of each patient (healthy or not). Additionally, eleven characteristic landmarks inside the lung area had been provided by two experienced radiologists.

All MRI examinations were acquired on a 3T-MRI unit (SKYRA magneton, Siemens Healthineers) using an 18-phased array body coil. All subjects were positioned in the supine position with their arms along the body. Inspiratory and expiratory MRI images were acquired using an ultrashort time of echo (UTE) sequence, the spiral VIBE sequence, with the same acquisition parameters (repetition time 2.73 ms, echo time 0.05 ms, flip angle 5°, field-of-view 620 × 620 mm, slice thickness 2.5 mm, matrix 188 × 188, with an in-plane resolution of 2.14 × 2.14 mm).

As a pre-processing step, the image intensity values were cropped within the window $[0, 1300]$ and mapped to $[0, 1]$. Moreover, all the images were scaled down along all dimensions by a factor of $2/3$ with cubic interpolation resulting to an image size of $64 \times 192 \times 192$ to compensate GPU memory constraints. A random split was performed and 28 patients (56 pairs of images) were selected for the training set, resulting to 3136 training pairs, while the rest 13 were used for validation.

6.3 Experimental Setup and Results

6.3.1 Evaluation

We evaluated the performance of our method against two different state-of-the-art methods, namely, Symmetric Normalization (SyN) [104], using its implementation on the ANTs package [120] and the deformable method presented in [103], [121] for a variety of similarity metrics (normalized cross correlation (NCC), mutual information (MI) and discrete wavelet metric (DWM), and their combination). For the evaluation we calculated the Dice coefficient metric, measured on the lung masks, after we applied the calculated deformation on the lung mask of the moving image. Moreover, we evaluate our method using the provided landmark locations. For comparison reasons we report the approximate computational time each of these methods needed to register a pair of images. For all the implementations we used a GeForce GTX 1080 GPU except for SyN implementation where we used a CPU implementation running on 4 cores of an i7-4700HQ CPU.

6.3.2 Results and Discussion

Starting with the quantitative evaluation, in Table 6.1 the mean Dice coefficient values along with their standard deviations are presented for different methods. We performed two different types of tests. In the first set of experiments (Table 6.1: Inhale-Exhale), we tested the performance of the different methods for the registration of the MRI images, between the inhale and exhale images, for the 13 validation patients. The SyN implementation reports the lowest Dice scores while at the same time, it is computationally quite expensive due to its CPU implementation. Moreover, we tested three different similarity metrics along with their combinations using the method proposed in [103] as described earlier. In this specific setup, the MI metric seam to report the best Dice scores. However, the scores reported by the proposed architecture are superior by at least $\sim 2.5\%$ to the ones reported by the other methods. For the proposed method, the addition of a linear component to the transformation layer does not change the performance of the network significantly in this experiment. Finally, we calculated the errors over all axes in predicted locations for eleven different manually annotated landmark points on inhale volumes after they were deformed using the decoded deformation for each patient. We compare the performance of our method against the inter-observer (two different medical experts) distance and the method presented in [103] in Table 6.2. We observe that both methods perform very well considering the inter-observer variability, with the proposed one reporting slitly better average euclidean distances.

For the second set of experiments (Table 6.1: All combinations), we report the Dice scores for all combinations of the 13 different patients, resulting on 169 validation pairs. Due to the large number of combinations, this problem is more challenging since the size of the lungs in the extreme moments of the respiratory circles can vary significantly. Again, the performance of the proposed architecture is superior to the tested baselines, highlighting its very promising results. In this experimental setup, the linear component plays a more important part by boosting the performance by $\sim 0.5\%$.

TABLE 6.1: Dice coefficient scores (%) calculated over the deformed lung masks and the ground truth.

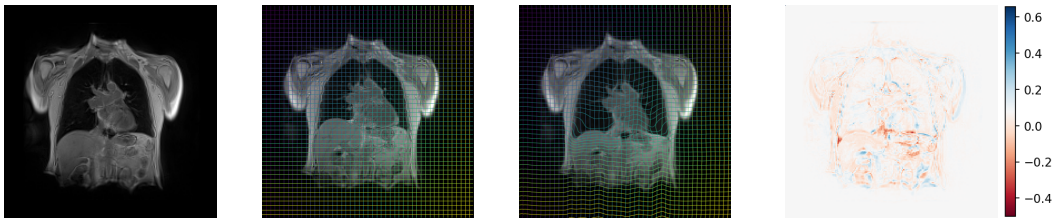
Method	Inhale-Exhale	All Combinations	Time/subject (s)
Unregistered	75.62±10.89	57.22±12.90	–
Deformable with NCC [103]	84.25±6.89	76.10±7.92	∼1 (GPU)
Deformable with DWM [103]	88.63±4.67	75.92±8.81	∼2 (GPU)
Deformable with MI [103]	88.86±5.13	76.33±8.74	∼2 (GPU)
Deformable with all above [103]	88.81±5.85	78.71±8.56	∼2 (GPU)
SyN [104]	83.86±6.04	–	∼2500 (CPU)
Proposed w/o Affine	91.28±2.47	81.75±7.88	∼0.5 (GPU)
Proposed	**91.48±2.33**	**82.34±7.68**	∼0.5 (GPU)

TABLE 6.2: Errors measured as average euclidean distances between estimated landmark locations and ground truth marked by two medical experts. We also report as *inter-observer*, the average euclidean distance between same landmark locations marked by the two experts. dx, dy, and dz denote distances along x-, y-, and z- axes, respectively, while ds denotes the average error along all axes.

Method	dx	dy	dz	ds
Inter-observer	1.664	2.545	1.555	3.905
Deformable with NCC, DWM, and MI [103]	1.855	3.169	2.229	4.699
Proposed w/o Affine	2.014	2.947	1.858	4.569
Proposed	**1.793**	**2.904**	**1.822**	**4.358**

Concerning the computation time, both [103] and the proposed method report very low inference time, due to their GPU implementations, with the proposed method reaching ~ 0.5 seconds per subject. On the other hand, [104] is computationally quite expensive, making it difficult to test it for all the possible combinations on the validation set.

Finally, in Figure 6.2, we present the deformed image produced by the proposed method on coronal view for a single patient in the two different moments of the respiratory cyrcle. The grids were superimposed on the images, indicating the displacements calculated by the network. The last column shows the difference between the reference and deformed image. One can observe that the majority of the errors occur on the boundaries, as the network fails to capture large local displacements.



(A) Reference image (B) Moving image (C) Deformed image (D) Difference

FIGURE 6.2: A visualized registration of a pair of images, generated by the proposed architecture. The initial and deformed grids are superimposed on the images.

6.3.3 Evaluation of the Clinical Relevance of the Deformation

To asses the relevance of the decoded transformations in a clinical setting, we trained a small classifier on top of the obtained residual deformations to classify patients as healthy or unhealthy. The residual deformation associated with a pair of images indicates voxel displacements, written as $G_\delta = G - G_I$, where G is the deduced deformation between the two images, and G_I is the identity deformation.

We trained a downsampling convolutional kernel followed by a multi-layer perceptron (MLP) to be able to predict whether a case is healthy or not. The network architecture is shown in Figure 6.3. The model includes batch normalization layers, to avoid overfitting, as we have few training examples at our disposal. Further, a Tanh activation function is used in the MLP. The downsampling kernel is of size $3 \times 3 \times 3$, with a stride of 2 and a padding of 1. The number of units in the hidden layer of the MLP was set to 100. We trained with binary cross entropy loss, with an initial learning rate of 10^{-4}, which is halved every fifty epochs. Training five models in parallel took about 2 hours on two GeForce GTX 1080 GPUs.

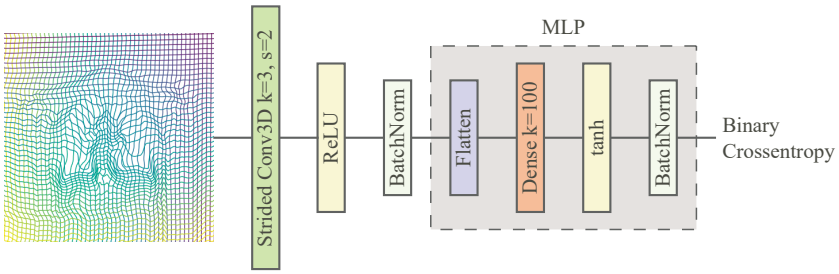


FIGURE 6.3: The neural network trained as a classifier on top of the transformations.

We cross-validate five models on the training set of 28 patients, and report the average response of these models on the rest 13 patients. We conduct the same experiment for deformations obtained using [103] and all similarity measures (NCC, DWM, MI). The results on the test set using a threshold of 0.5 on the predicted probability are reported in Table 6.3, suggesting that indeed the deformations between inhale and exhale carry information about lung diseases.

TABLE 6.3: Results on disease prediction using deformations on the test set. The reported accuracy is in percentage points.

Method	Accuracy
Deformable with NCC, DWM, and MI [103]	69.23
Proposed	**84.62**

6.4 Conclusion

In this chapter, we proposed a novel method which exploits the 3D CNNs to calculate the optimal transformation (combining a linear and a deformable component within a coupled framework) between pair of images that is modular with respect to the similarity function, and the nature of transformation. The proposed method generates deformations with no self-crossings due to the way the deformation layer is defined, efficient due to the GPU implementation of the inference and reports high promising results compared to other unsupervised registration methods. Currently, the proposed network was tested on the challenging problem of lung registration, however, its evaluation on the registration of other modalities, and other organs is one of the potential directions of our method.

Chapter 7

Concluding Remarks

7.1 Summary

The aim of this thesis was to develop a computational system that will assist radiologists with the diagnosis of ILDs, while minimizing the dangerous, expensive and time-consuming invasive biopsies. The system provides a computerized differential diagnosis, based on radiological data and clinical/biochemical markers and is focused on the discrimination between idiopathic interstitial pneumonias, in particular, idiopathic pulmorary fibrosis, while keeping a generic architecture that could be expanded to most types of ILDs. The appropriate interpretation of the available radiological data combined with clinical/biochemical information can provide reliable diagnostic radiomarkers, able to improve the diagnostic accuracy of the physicians. In all the implemented algorithms, experimental configurations and evaluation schemes, the main focus was the bridging of the machine learning and medical diagnosis. Between the rapidly evolving field of machine learning and the slow and rigorous evaluated field of automated medical diagnosis, really careful steps should be followed.

During this thesis, a number of novel algorithms have been proposed for the computerized assessment of HRCTs with suspected ILD. In particular, three major contributions can be identified. First, a number machine learning approaches have been designed and evaluated for the automatic detection and delineation of diffuse pathological tissue. Specifically, in Chapters 2-4, two convolutional neural network architectures have been proposed along with a training scheme that utilized the properties of knowledge transferring from multiple sources. Second, in Chapter 5, an end-to-end pipeline for the quantification and classification of idiopathic pulmonary fibrosis has been developed and evaluated. Last, in Chapter 6, an unsupervised method that employs a convolutional neural network was used for 3D medical image registration and lung breathing radio-markers extraction.

7.2 General Discussion

Studies of machine learning (ML) based CAD systems that could perform similarly to physicians at specific tasks are lately presented with increasing frequency. In some cases where a large amount of data and the necessary computational power are available, properly trained models can even outperform experienced physicians in particular tasks [122], [123]. In most health care use cases however, it is not possible to fully define tasks to be automatically processed, considering the heterogeneity of patients' population and of the diseases. It is therefore of great importance that CAD systems provide interpretable results and designed as assistive tools for physicians.

Provided that the right choices are made in the design and training of such CAD systems, their generalization performance is heavily depended on the size and quality of the training and validation datasets. Possible population biases could be propagated to the results of a study and therefore, particular care should be taken when designing such evaluation schemes. In order to tackle such issues, interdisciplinary teams of engineers, physicians and their patients from multiple institutes should work together to minimize possible cohort biases. When all the aforementioned criteria are met, CAD systems can provide fast and accurate results, while they could also be used for diagnostic standardization and thus benefit longitudinal studies.

The importance of well-curated data cannot be over stressed for data driven machine learning approaches in medicine. The last years, after the repeated successes of deep learning algorithms, a number of software frameworks has been made publicly available, making long strides towards fast prototyping, while more and more attention is drawn towards data gathering and pre-processing. On the other hand, the picture archiving and communication systems (PACS) in hospitals hold years worth of data that are kept unused. This is not an unknown issue in the community since a number of initiatives worldwide are for decades trying to compile registries of particular patient cohorts[1]. By expanding such registries to easier facilitate machine learning approaches, a huge impact could be achieved in health care. It is well known that the quality of medical images is getting better year after year. These improvements enable more and more diagnostic procedures to be based on the assessment of such images. Health care providers worldwide take new images every year either to monitor the progress of a patient or for diagnostic purposes. Appropriate frameworks for the effortless utilization of these prospective gathered images could further improve ML driven systems.

A recurrent topic on radiological societies revolves around the raise of radiology robots that will replace radiologists. Geoffrey Hinton, one of the acclaimed "fathers" of Deep Learning, claimed in one of his talks that medical schools should stop training radiologists since in five to ten years radiologists will have been replaced by DL. Considering the recent coordinated efforts from multiple involved parties, it is clear that in a couple of years more and more accurate, fast and robust CAD systems will be introduced. However, the medical decision making is a vastly complicated task, which in some cases is still under research, while it is often based on years of experience and other factors. Making sure that radiologists are trained to utilize interpretable AI based CAD systems, could definitely enhance their diagnostic performance. Thus, one could instead claim that in the years to come CAD assisted radiologists will replace radiologists who do not utilize such systems.

Last but not least, one really important challenge that is worth further investigation and discussion, is the ethic frame around machine learning in medicine. In order to avoid unnecessary risks, AI based CAD systems should conform with data privacy regulations, interpretability issues and achieve transparency.

7.3 Perspectives

This thesis lies in the cross-section of machine learning and medical diagnosis. A number of novel algorithms for the assessment of HRCT images of ILD patients have been suggested. The clinical relevance of an end-to-end ILD quantification scheme has been investigated, as well as the diagnostic information of breathing radiomarkers. For the best part of these four years and with the collaboration of multiple physicians and engineers, a number of key aspects have been identified for the further improvement of such systems, so that they can be used in clinical practice:

- Algorithms that generalize well.
- Better error detection and false positive reduction.
- Better interfaces between implemented tools and physicians.
- Integration of more contextual information.
- Publicly available databases.

Generalization is one of the most well known problems of CAD systems. Most of the times, due to the limited available data and the huge variability of pathology expression, CAD systems tend to over-fit on the population they were trained on. It is often the case that a CAD system achieves performance with really high specificity and sensitivity, yet when tested on data from a different cohort these metrics drop. physicians often do not trust the results of automatic methods, since there is a certain amount of errors. Trying to build algorithms that generalize well and that are enhanced with better error detection schemes

[1] e.g. http://www.cancerimagingarchive.net/

could further improve the performance of CAD systems, rendering them more useful in a health care environment.

In medical images - depending on the acquisition configuration - structures are often incomplete, missing or barely visible so that automated systems could not be reliably used. Human experts however, have the ability to easily infer the position of particular structures due to their experience. Interactive systems that request the feedback of physicians in all the involved steps and allow their input, could ensure a much better error management. Moreover, physicians, in order to make a diagnosis, are almost always using more than one sources of information. In some cases, multi-disciplinary boards are formed, in order to decide a diagnosis and treatment course. CAD systems could achieve much better results if all the relative contextual information is provided.

Apart from methodological improvements and better error management, the availability of public databases with medical data are of great importance towards better performing CAD systems. Such databases should include all the necessary information physicians use for the diagnosis and should come from a variety of equipment, patient groups and institutions. Large and well curated databases that follow the aforementioned criteria could be used for directly comparing different methods as well as for longitudinal studies of particular diseases.

Bibliography

[1] B. SOCIETY, "The diagnosis, assessment and treatment of diffuse parenchymal lung disease in adults," *Thorax*, vol. 54, no. Suppl 1, S1, 1999.

[2] M Demedts and U Costabel, *Ats/ers international multidisciplinary consensus classification of the idiopathic interstitial pneumonias*, 2002.

[3] W. D. Travis, U. Costabel, D. M. Hansell, T. E. King, D. A. Lynch, A. G. Nicholson, C. J. Ryerson, J. H. Ryu, M. Selman, A. U. Wells, J. Behr, D. Bouros, K. K. Brown, T. V. Colby, H. R. Collard, C. R. Cordeiro, V. Cottin, B. Crestani, M. Drent, R. F. Dudden, J. Egan, K. Flaherty, C. Hogaboam, Y. Inoue, T. Johkoh, D. S. Kim, M. Kitaichi, J. Loyd, F. J. Martinez, J. Myers, S. Protzko, G. Raghu, L. Richeldi, N. Sverzellati, J. Swigris, and D. Valeyre, "An official american thoracic society/european respiratory society statement: Update of the international multidisciplinary classification of the idiopathic interstitial pneumonias," *American Journal of Respiratory and Critical Care Medicine*, vol. 188, no. 6, pp. 733–748, 2013.

[4] D. A. Lynch, N. Sverzellati, W. D. Travis, K. K. Brown, T. V. Colby, J. R. Galvin, J. G. Goldin, D. M. Hansell, Y. Inoue, T. Johkoh, *et al.*, "Diagnostic criteria for idiopathic pulmonary fibrosis: A fleischner society white paper," *The lancet Respiratory medicine*, 2017.

[5] I. Sluimer, A. Schilham, M. Prokop, and B. Van Ginneken, "Computer analysis of computed tomography scans of the lung: A survey," *IEEE Transactions on Medical Imaging*, vol. 25, no. 4, pp. 385–405, 2006, ISSN: 02780062.

[6] Y. Uchiyama, S. Katsuragawa, H. Abe, J. Shiraishi, and Li, "Quantitative computerized analysis of diffuse lung disease in high-resolution computed tomography," *Medical Physics*, vol. 30, no. 9, pp. 2440–2454, 2003.

[7] K. G. Kim, J. M. Goo, J. H. Kim, H. J. Lee, B. G. Min, K. T. Bae, and J.-G. Im, "Computer-aided diagnosis of localized ground-glass opacity in the lung at ct: Initial experience," *Radiology*, vol. 237, no. 2, pp. 657–661, 2005.

[8] I. C. Sluimer, M. Prokop, I. Hartmann, and B. van Ginneken, "Automated classification of hyperlucency, fibrosis, ground glass, solid, and focal lesions in high-resolution ct of the lung," *Medical Physics*, vol. 33, no. 7Part1, pp. 2610–2620,

[9] V. A. Zavaletta, B. J. Bartholmai, and R. A. Robb, "High resolution multidetector ct-aided tissue analysis and quantification of lung fibrosis," *Academic radiology*, vol. 14, no. 7, pp. 772–787, 2007.

[10] S. Hu, E. A. Hoffman, and J. M. Reinhardt, "Automatic lung segmentation for accurate quantitation of volumetric x-ray ct images," *IEEE Transactions on Medical Imaging*, 2001.

[11] P. Korfiatis, C. Kalogeropoulou, A. Karahaliou, A. Kazantzi, S. Skiadopoulos, and L. Costaridou, "Texture classification-based segmentation of lung affected by interstitial pneumonia in high-resolution ct," *Medical Physics*, vol. 35, no. 12, pp. 5290–5302,

[12] P. Hua, Q. Song, M. Sonka, E. A. Hoffman, and J. M. Reinhardt, "Segmentation of pathological and diseased lung tissue in ct images using a graph-search algorithm," in *2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2011, pp. 2072–2075.

[13] M. N. Prasad, M. S. Brown, S. Ahmad, F. Abtin, J. Allen, I. da Costa, H. J. Kim, M. F. McNitt-Gray, and J. G. Goldin, "Automatic segmentation of lung parenchyma in the presence of diseases based on curvature of ribs," *Academic radiology*, vol. 15, no. 9, pp. 1173–1180, 2008.

[14] R. Uppaluri, E. A. Hoffman, M. Sonka, P. G. Hartley, G. W. Hunninghake, and G. McLennan, "Computer recognition of regional lung disease patterns," *American Journal of Respiratory and Critical Care Medicine*, vol. 160, no. 2, pp. 648–654, 1999.

[15] I. C. Sluimer, P. F. van Waes, M. A. Viergever, and B. van Ginneken, "Computer-aided diagnosis in high resolution ct of the lungs," *Medical physics*, vol. 30, no. 12, pp. 3081–3090, 2003.

[16] M. Anthimopoulos, S. Christodoulidis, A. Christe, and S. Mougiakakou, "Classification of interstitial lung disease patterns using local dct features and random forest," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Aug. 2014, pp. 6040–6043.

[17] K. T. Vo and A. Sowmya, "Multiple kernel learning for classification of diffuse lung disease using hrct lung images," in *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, IEEE, 2010, pp. 3085–3088.

[18] L. Sørensen, S. B. Shaker, and M. De Bruijne, "Quantitative analysis of pulmonary emphysema using local binary patterns," *Medical Imaging, IEEE Transactions on*, vol. 29, no. 2, pp. 559–569, 2010.

[19] M. J. Gangeh, L. Sørensen, S. B. Shaker, M. S. Kamel, M. De Bruijne, and M. Loog, "A texton-based approach for the classification of lung parenchyma in ct images," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2010*, Springer, 2010, pp. 595–602.

[20] A. Foncubierta-Rodríguez, A. Depeursinge, and H. Müller, "Using multiscale visual words for lung texture classification and retrieval," in *Medical Content-Based Retrieval for Clinical Decision Support*, Springer, 2011, pp. 69–79.

[21] W. Zhao, R. Xu, Y. Hirano, R. Tachibana, and S. Kido, "Classification of diffuse lung diseases patterns by a sparse representation based method on hrct images," in *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, IEEE, 2013, pp. 5457–5460.

[22] K. T. Vo and A. Sowmya, "Multiscale sparse representation of high-resolution computed tomography (hrct) lung images for diffuse lung disease classification," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*, IEEE, 2011, pp. 441–444.

[23] Q. Li, W. Cai, and D. D. Feng, "Lung image patch classification with automatic feature learning," in *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*, IEEE, 2013, pp. 6079–6082.

[24] A. Depeursinge, D. Van de Ville, A. Platon, A. Geissbuhler, P.-A. Poletti, and H. Müller, "Near-affine-invariant texture learning for lung tissue analysis using isotropic wavelet frames," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 16, no. 4, pp. 665–675, 2012.

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances In Neural Information Processing Systems*, pp. 1–9, 2012, ISSN: 10495258.

[26] G. van Tulder and M. de Bruijne, "Learning features for tissue classification with the classification restricted boltzmann machine," in *International MICCAI Workshop on Medical Computer Vision*, Springer, 2014, pp. 47–58.

[27] Q. Li, W. Cai, X. Wang, Y. Zhou, D. D. Feng, and M. Chen, "Medical image classification with convolutional neural network," in *Control Automation Robotics & Vision (ICARCV), 2014 13th International Conference on*, IEEE, 2014, pp. 844–848.

[28] M. Gao, U. Bagci, L. Lu, A. Wu, M. Buty, H.-C. Shin, H. Roth, G. Z. Papadakis, A. Depeursinge, R. Summers, Z. Xu, and D. J. Mollura, "Holistic classification of ct attenuation patterns for interstitial lung diseases via deep convolutional neural networks," in *1st Workshop on Deep Learning in Medical Image Analysis*, ser. DLMIA 2015, Münich, Germany, Oct. 2015, pp. 41–48.

[29] B. van Ginneken, S. Katsuragawa, B. M. ter Haar Romeny, K. Doi, and M. A. Viergever, "Automatic detection of abnormalities in chest radiographs using local texture analysis," *IEEE Transactions on Medical Imaging*, vol. 21, no. 2, pp. 139–149, 2002.

[30] B. Zheng, J. K. Leader, C. R. Fuhrman, F. C. Sciurba, and D. Gur, "Automated detection and classification of interstitial lung diseases from low-dose ct images," in *Medical Imaging 2004: Image Processing*, International Society for Optics and Photonics, vol. 5370, 2004, pp. 849–857.

[31] A. Fukushima, K. Ashizawa, T. Yamaguchi, N. Matsuyama, H. Hayashi, I. Kida, Y. Imafuku, A. Egawa, S. Kimura, K. Nagaoki, *et al.*, "Application of an artificial neural network to high-resolution ct: Usefulness in differential diagnosis of diffuse lung disease," *American Journal of Roentgenology*, vol. 183, no. 2, pp. 297–305, 2004.

[32] J. Wang, F. Li, K. Doi, and Q. Li, "Computerized detection of diffuse lung disease in mdct: The usefulness of statistical texture features," *Physics in Medicine & Biology*, vol. 54, no. 22, p. 6881, 2009.

[33] A. Depeursinge, A. Vargas, A. Platon, A. Geissbuhler, P.-A. Poletti, and H. Müller, "Building a reference multimedia database for interstitial lung diseases," *Computerized medical imaging and graphics*, vol. 36, no. 3, pp. 227–238, 2012.

[34] C. C. Hau, *Handbook of pattern recognition and computer vision*. World Scientific, 2015.

[35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[36] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014, ISSN: 1532-4435.

[37] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, 2013, p. 3.

[38] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015.

[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[41] R. Al-Rfou, G. Alain, A. Almahairi, and et al., "Theano: A python framework for fast computation of mathematical expressions," *CoRR*, vol. abs/1605.02688, 2016.

[42] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*, ACM, 2014, pp. 675–678.

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

[44] J. Carpenter and J. Bithell, "Bootstrap confidence intervals: When, which, what? a practical guide for medical statisticians," *Statistics in medicine*, vol. 19, no. 9, pp. 1141–1164, 2000.

[45] J. A. Hanley and B. J. McNeil, "A method of comparing the areas under receiver operating characteristic curves derived from the same cases.," *Radiology*, vol. 148, no. 3, pp. 839–843, 1983.

[46] H. Greenspan, B. Van Ginneken, and R. Summers, "Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1153–1159, 2016.

[47] S. J. Pan and Q. Yang, "A survey on transfer learning," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 22, no. 10, pp. 1345–1359, 2010.

[48] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: An astounding baseline for recognition," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, ser. CVPRW '14, Washington, DC, USA: IEEE Computer Society, 2014, pp. 512–519, ISBN: 978-1-4799-4308-1.

[49] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" *Advances in Neural Information Processing Systems 27 (Proceedings of NIPS)*, vol. 27, pp. 1–9, Nov. 2014.

[50] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, Jul. 1997, ISSN: 0885-6125.

[51] L. Zheng, Y. Zhao, S. Wang, J. Wang, and Q. Tian, "Good practice in CNN feature transfer," *CoRR*, vol. abs/1604.00133, 2016.

[52] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, "From generic to specific deep representations for visual recognition," in *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2015, pp. 36–45.

[53] H. C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, "Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1285–1298, May 2016, ISSN: 0278-0062.

[54] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, and J. Liang, "Convolutional neural networks for medical image analysis: Full training or fine tuning?" *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1299–1312, May 2016, ISSN: 0278-0062.

[55] G. J. Burghouts and J.-M. Geusebroek, "Material-specific adaptation of color invariant features," *Pattern Recognition Letters*, vol. 30, no. 3, pp. 306–313, 2009.

[56] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[57] L. Sharan, R. Rosenholtz, and E. Adelson, "Material perception: What can you see in a brief glance?" *Journal of Vision*, vol. 9, no. 8, pp. 784–784, 2009.

[58] G. Kylberg, "The kylberg texture dataset v. 1.0," Centre for Image Analysis, Swedish University of Agricultural Sciences and Uppsala University, Uppsala, Sweden, External report (Blue series) 35, 2011.

[59] B. Caputo, E. Hayman, and P. Mallikarjuna, *Class-specific material categorisation*, Oct. 2005.

[60] S. Lazebnik, C. Schmid, and J. Ponce, "A sparse texture representation using local affine regions," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 8, pp. 1265–1278, 2005.

[61] M. Anthimopoulos, S. Christodoulidis, L. Ebner, A. Christe, and S. Mougiakakou, "Lung pattern classification for interstitial lung diseases using a deep convolutional neural network," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1207–1216, May 2016, ISSN: 0278-0062.

[62] T. G. Dietterich, "Ensemble methods in machine learning," in *Proceedings of the First International Workshop on Multiple Classifier Systems*, ser. MCS '00, London, UK, UK: Springer-Verlag, 2000, pp. 1–15, ISBN: 3-540-67704-6.

[63] R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes, "Ensemble selection from libraries of models," in *Proceedings of the twenty-first international conference on Machine learning*, ACM, 2004, p. 18.

[64] C. Buciluǎ, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '06, Philadelphia, PA, USA: ACM, 2006, pp. 535–541, ISBN: 1-59593-339-5.

[65] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *CoRR*, vol. abs/1503.02531, 2015.

[66] F. Chollet, *Keras*, 2015.

[67] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, ISSN: 0018-9219.

[68] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[69] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

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

[71] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015.

[72] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*, Springer, 2014, pp. 818–833.

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

[74] V. Badrinarayanan, A. Handa, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling," *arXiv preprint arXiv: 1505.07293*, 2015.

[75] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1520–1528.

[76] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pp. 234–241, 2015.

[77] M. Holschneider, R. Kronland-Martinet, J. Morlet, and P. Tchamitchian, "A real-time algorithm for signal analysis with the help of the wavelet transform," in *Wavelets*, Springer, 1990, pp. 286–297.

[78] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *CoRR*, vol. abs/1511.07122, 2015.

[79] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: A deep neural network architecture for real-time semantic segmentation," *CoRR*, vol. abs/1606.02147, 2016.

[80] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *arXiv preprint arXiv:1606.00915*, 2016.

[81] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. W. Cottrell, "Understanding convolution for semantic segmentation," *CoRR*, vol. abs/1702.08502, 2017.

[82] F. Yu, V. Koltun, and T. A. Funkhouser, "Dilated residual networks," *CoRR*, vol. abs/1705.09914, 2017.

[83] D. Ulyanov, A. Vedaldi, and V. S. Lempitsky, "Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis," *CoRR*, vol. abs/1701.02096, 2017.

[84] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *CoRR*, vol. abs/1611.07004, 2016.

[85] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *Proceedings of the 17th International Conference on Neural Information Processing Systems*, ser. NIPS'04, Vancouver, British Columbia, Canada: MIT Press, 2004, pp. 529–536.

[86] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on Challenges in Representation Learning, ICML*, vol. 3, 2013, p. 2.

[87] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, vol. abs/1605.02688, May 2016.

[88] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: Learning dense volumetric segmentation from sparse annotation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2016, pp. 424–432.

[89] F. Milletari, N. Navab, and S. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," *CoRR*, vol. abs/1606.04797, 2016.

[90] H. R. Roth, L. Lu, J. Liu, J. Yao, A. Seff, K. Cherry, L. Kim, and R. M. Summers, "Improving computer-aided detection using convolutional neural networks and random view aggregation," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1170–1181, 2016.

[91] F. Liu, Z. Zhou, H. Jang, A. Samsonov, G. Zhao, and R. Kijowski, "Deep convolutional neural network and 3d deformable approach for tissue segmentation in musculoskeletal magnetic resonance imaging," *Magnetic resonance in medicine*, vol. 79, no. 4, pp. 2379–2391, 2018.

[92] P. F. Christ, M. E. A. Elshaer, F. Ettlinger, S. Tatavarty, M. Bickel, P. Bilic, M. Rempfler, M. Armbruster, F. Hofmann, M. D'Anastasi, *et al.*, "Automatic liver and lesion segmentation in ct using cascaded fully convolutional neural networks and 3d conditional random fields," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2016, pp. 415–423.

[93] K. Kamnitsas, C. Ledig, V. F. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, D. Rueckert, and B. Glocker, "Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation," *Medical image analysis*, vol. 36, pp. 61–78, 2017.

[94] A. U. Wells, "Managing diagnostic procedures in idiopathic pulmonary fibrosis," *European Respiratory Review*, vol. 22, no. 128, pp. 158–162, 2013.

[95] I. Sluimer, M. Prokop, and B. Van Ginneken, "Toward automated segmentation of the pathological lung in ct," *IEEE transactions on medical imaging*, vol. 24, no. 8, pp. 1025–1038, 2005.

[96] E. M. van Rikxoort, B. de Hoop, M. A. Viergever, M. Prokop, and B. van Ginneken, "Automatic lung segmentation from thoracic computed tomography scans using a hybrid approach with error detection," *Medical physics*, vol. 36, no. 7, pp. 2934–2947, 2009.

[97] K. Mori, J.-i. Hasegawa, J.-i. Toriwaki, H. Anno, and K. Katada, "Recognition of bronchus in three-dimensional x-ray ct images with applications to virtualized bronchoscopy system," in *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, IEEE, vol. 3, 1996, pp. 528–532.

[98] M. Anthimopoulos, S. Christodoulidis, L. Ebner, T. Geiser, A. Christe, and S. G. Mougiakakou, "Semantic segmentation of pathological lung tissue with dilated fully convolutional networks," *IEEE Journal of Biomedical and Health Informatics*, 2018.

[99] B. J. Bartholmai, S. Raghunath, R. A. Karwoski, T. Moua, S. Rajagopalan, F. Maldonado, P. A. Decker, and R. A. Robb, "Quantitative ct imaging of interstitial lung diseases," *Journal of thoracic imaging*, vol. 28, no. 5, 2013.

[100] N. L. S. T. R. Team, "Reduced lung-cancer mortality with low-dose computed tomographic screening," *New England Journal of Medicine*, vol. 365, no. 5, pp. 395–409, 2011.

[101] A. Depeursinge, A. S. Chin, A. N. Leung, D. Terrone, M. Bristow, G. Rosen, and D. L. Rubin, "Automated classification of usual interstitial pneumonia using regional volumetric texture analysis in high-resolution ct," *Investigative radiology*, vol. 50, no. 4, p. 261, 2015.

[102] A. Sotiras, C. Davatzikos, and N. Paragios, "Deformable medical image registration: A survey," *IEEE Transactions on Medical Imaging*, pp. 1153–1190, 2013.

[103] E. Ferrante, P. K. Dokania, R. Marini, and N. Paragios, "Deformable registration through learning of context-specific metric aggregation," in *Machine Learning in Medical Imaging: 8th International Workshp, MLMI, 2017, Held in Conjunction with MICCAI 2017, Quebec City, QC, Canada, September 10, 2017, Proceedings*. Springer International Publishing, 2017, pp. 256–265.

[104] B. Avants, C. Epstein, M. Grossman, and J. Gee, "Symmetric diffeomorphic image registration with cross-correlation: Evaluating automated labeling of elderly and neurodegenerative brain," *Medical Image Analysis*, 2008, ISSN: 1361-8415.

[105] S. Oh and S. Kim, "Deformable image registration in radiation therapy," *Radiation oncology journal*, vol. 35, no. 2, p. 101, 2017.

[106] D. Sarrut, T. Baudier, M. Ayadi, R. Tanguy, and S. Rit, "Deformable image registration applied to lung sbrt: Usefulness and limitations," *Physica Medica*, vol. 44, pp. 108–112, 2017.

[107] G. Cazoulat, D. Owen, M. M. Matuszak, J. M. Balter, and K. K. Brock, "Biomechanical deformable image registration of longitudinal lung ct images using vessel information," *Physics in Medicine & Biology*, vol. 61, no. 13, p. 4826, 2016.

[108] G. Chassagnon, C. Martin, R. Marini, M. Vakalopolou, A. Régent, L. Mouthon, N. Paragios, and M.-P. Revel, "Use of elastic registration in pulmonary mri for the assessment of pulmonary fibrosis in patients with systemic sclerosis," *Radiology*, vol. 0, no. 0, p. 182 099, 0, PMID: 30835186. DOI: 10.1148/radiol.2019182099.

[109] K. Ding, K. Cao, M. K. Fuld, K. Du, G. E. Christensen, E. A. Hoffman, and J. M. Reinhardt, "Comparison of image registration based measures of regional lung ventilation from dynamic spiral ct with xe-ct," *Medical Physics*, vol. 39, no. 8, pp. 5084–5098, 2012. DOI: 10.1118/1.4736808.

[110] J. M. Reinhardt, K. Ding, K. Cao, G. E. Christensen, E. A. Hoffman, and S. V. Bodas, "Registration-based estimates of local lung tissue expansion compared to xenon ct measures of specific ventilation," *Medical image analysis*, vol. 12, no. 6, pp. 752–763, 2008.

[111] S. Chandra, N. Usunier, and I. Kokkinos, "Dense and low-rank gaussian crfs using deep embeddings," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

[112] A. Riza, N. Neverova, and I. Kokkinos, "Densepose: Dense human pose estimation in the wild," *arXiv*, 2018.

[113] T.-W. Hui, X. Tang, and C. C. Loy, "Liteflownet: A lightweight convolutional neural network for optical flow estimation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[114] M. Simonovsky, B. Gutiérrez-Becker, D. Mateus, N. Navab, and N. Komodakis, "A deep metric for multimodal registration," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, Cham: Springer International Publishing, 2016.

[115] X. Cheng, L. Zhang, and Y. Zheng, "Deep similarity learning for multimodal medical images," 2016.

[116] P. Yan, S. Xu, A. R. Rastinehad, and B. J. Wood, "Adversarial image registration with application for MR and TRUS image fusion," 2018.

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

[118] Z. Shu, M. Sahasrabudhe, R. A. Güler, D. Samaras, N. Paragios, and I. Kokkinos, "Deforming Autoencoders: Unsupervised Disentangling of Shape and Appearance," *2018 IEEE European Conference on Computer Vision (ECCV)*, 2018.

[119] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[120] B. B. Avants, N. J. Tustison, G. Song, P. A. Cook, A. Klein, and J. C. Gee, "A reproducible evaluation of ants similarity metric performance in brain image registration," *NeuroImage*, 2011, ISSN: 1053-8119.

[121] B. Glocker, A. Sotiras, N. Komodakis, and N. Paragios, "Deformable medical image registration: Setting the state of the art with discrete methods," *Annual Review of Biomedical Engineering*, pp. 219–244, 2011.

[122] F. Ciompi, K. Chung, S. J. Van Riel, A. A. A. Setio, P. K. Gerke, C. Jacobs, E. T. Scholten, C. Schaefer-Prokop, M. M. Wille, A. Marchianò, *et al.*, "Towards automatic pulmonary nodule management in lung cancer screening with deep learning," *Scientific reports*, vol. 7, p. 46 479, 2017.

[123] J. De Fauw, J. R. Ledsam, B. Romera-Paredes, S. Nikolov, N. Tomasev, S. Blackwell, H. Askham, X. Glorot, B. O'Donoghue, D. Visentin, *et al.*, "Clinically applicable deep learning for diagnosis and referral in retinal disease," *Nature medicine*, vol. 24, no. 9, p. 1342, 2018.

Declaration of Originality

Last name, first name: CHRISTODOULIDIS Stergios

Matriculation number: 14-139-109

I hereby declare that this thesis represents my original work and that I have used no other sources except as noted by citations.
All data, tables, figures and text citations which have been reproduced from any other source, including the internet, have been explicitly acknowledged as such.
I am aware that in case of non-compliance, the Senate is entitled to withdraw the doctorate degree awarded to me on the basis of the present thesis, in accordance with the "Statut der Universität Bern (Universitätsstatut; UniSt)", Art. 69, of 7 June 2011.

Place, Date:

Signature: