Lung Pattern Classification for Interstitial Lung Diseases Using a Deep Convolutional Neural Network

Marios Anthimopoulos, *Member, IEEE*, Stergios Christodoulidis, *Member, IEEE*,
Lukas Ebner, Andreas Christe, and Stavroula Mougiakakou*, *Member, IEEE*

***Abstract*—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 paper, 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.**

Manuscript received January 15, 2016; revised February 15, 2016; accepted February 17, 2016. Date of publication February 29, 2016; date of current version April 29, 2016. This research was carried out within the framework of the INTACT research project, supported by Bern University Hospital, "Inselspital" and the Swiss National Science Foundation (SNSF) under Grant 156511. M. Anthimopoulos and S. Christodoulidis contributed equally to this work. *Asterisk indicates corresponding author.*

M. Anthimopoulos is with the ARTORG Center for Biomedical Engineering Research, University of Bern, 3008 Bern, Switzerland, and with the Department of Diagnostic, Interventional and Pediatric Radiology, Bern University Hospital "Inselspital", 3010 Bern, Switzerland, and also with the Department of Emergency Medicine, Bern University Hospital "Inselspital", 3010 Bern, Switzerland (e-mail: marios.anthimopoulos@artorg.unibe.ch).

S. Christodoulidis is with the ARTORG Center for Biomedical Engineering Research, University of Bern, 3008 Bern, Switzerland (e-mail: stergios.christodoulidis@artorg.unibe.ch).

L. Ebner and A. Christe are with the Department of Diagnostic, Interventional and Pediatric Radiology, Bern University Hospital "Inselspital", 3010 Bern, Switzerland (e-mail: lukas.ebner@insel.ch; andreas.christe@insel.ch).

*S. Mougiakakou is with the Department of Diagnostic, Interventional and Pediatric Radiology, Bern University Hospital "Inselspital", 3010 Bern, Switzerland, and also with the ARTORG Center for Biomedical Engineering Research, University of Bern, 3008 Bern Switzerland (e-mail: stavroula.mougiakakou@artorg.unibe.ch).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMI.2016.2535865

***Index Terms*—Convolutional neural networks, interstitial lung diseases, texture classification.**

I. Introduction

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 accounts for 15 percent of all cases seen by pulmonologists [1] and can be caused by autoimmune diseases, genetic abnormalities 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.

The diagnosis of an ILD involves questioning the patient about their clinical history, a thorough physical examination, pulmonary function testing, a chest X-ray and a CT scan. High resolution computed tomography (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 textural patterns in the lung CT scan. Typical ILD patterns in CT images are: reticulation, honeycombing, ground glass opacity (GGO), consolidation and micronodules (Fig. 1).

However, in some cases, the diagnosis cannot be confirmed radiologically. 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, explains the low diagnostic accuracy and the high inter- and intra- observer variability, which has been reported to be as great as 50% [3]. In ambiguous cases, additional invasive procedures are required, such as bronchoalveolar lavage and

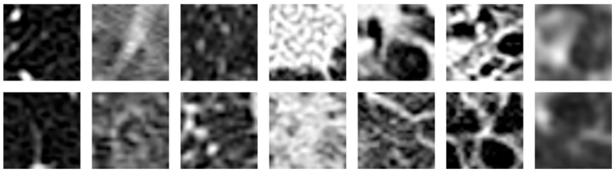


Fig. 1. Examples of healthy tissue and typical ILD patterns from left to right: healthy, GGO, micronodules, consolidation, reticulation, honeycombing, combination of GGO and reticulation.

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.

To avoid the dangerous 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 CT scan assessment typically consists of three stages: (a) lung segmentation, (b) lung disease quantification and (c) differential diagnosis. The first stage refers to the identification of the lung border, the separation of the lobes and in some cases the detection and removal of the bronchovascular tree. The second stage includes detection and recognition of the different tissue abnormalities and estimation of their extent in the lung. Finally, the third stage combines the previous results to suggest a probable differential diagnosis. In this study, we focus on the second stage and especially on the classification of lung tissue with different ILD abnormalities.

II. Related Work

In this section we provide an overview of the previous studies on ILD pattern classification, followed by a brief introduction to convolutional neural networks (CNN), which are employed in the proposed methodology.

A. ILD Pattern Classification

Since ILDs are generally manifested as textural alterations in the lung parenchyma, most of the proposed systems employ texture classification schemes on local regions of interest (ROIs) or volumes of interest (VOIs), depending on the 2D or 3D capabilities of the CT imaging modality employed. By sliding a fixed-scale classifier over pre-segmented lung fields, an ILD quantification map of the entire lung is generated. The latter can be used – either by physicians or CAD systems – to attempt the final diagnosis. The main characteristics of such a system are the chosen feature set and the classification method.

The first CAD systems for ILDs proposed classical feature extraction methods to describe 2D texture, such as first order gray level statistics, gray level co-occurrence matrices (GLCM), run-length matrices (RLM) and fractal analysis [4], [5]. These features were later merged and referred as the adaptive multiple feature method (AMFM) [6]. AMFM was generally accepted as the state of the art until new systems appeared that utilized more modern texture description techniques and provided a new perspective to the problem. Such systems employed filter banks [7]–[9], morphological operations followed by geometric measures [10], wavelet and contourlet transformations [11], [12], histograms of oriented gradients [8] and local binary patterns (LBP) [13]. Moreover, some systems exploited the ability of MDCT scanners to achieve almost isotropic 3D sub-millimeter resolution and expanded some of the already proposed 2D texture feature sets into three dimensions [14]–[17]. One of the latest studies on volumetric data proposed the use of multiscale 3D Riesz wavelet frames coupled with wavelet pyramids [18].

The previously presented systems have used hand-crafted features to describe lung tissue, which often fail to adapt to new data or patterns. More recent studies adopted learned schemes for feature extraction which customize the feature set to the training data and have achieved promising results. Most of these use unsupervised techniques, such as bag of features [19]–[21] and sparse representation models [22]–[24]. In these methods, a set of texture atoms or textons is identified by using k-means and k-SVD, on already described local patches. The resulting set of textons constitutes a problem-specific dictionary and every local structure in the image is represented by the closest texton or a linear combination of the entire set. The final global descriptor usually consists of the histogram of textons appearing in the image. Another tool which has be used for extracting learned features is the restricted Boltzmann machine (RBM). RBMs are generative artificial neural networks (ANNs) that are able to capture and reproduce the statistical structure of the input and were employed in [25] for learning multi-scale filters with their responses as the features.

Regardless of whether handcrafted or learned features are used, it is also crucial and challenging to choose an appropriate classifier that can optimally handle the properties of the created feature space. Many different approaches can be found in the literature. These use linear discriminant (LD) [5], [7], and Bayesian [6], [14], classifiers, k-nearest neighbors (kNN) [7], [13], [16], [20], ANN [10], random forest [9] and support vector machines (SVM) with linear [22], [25], polynomial [8] or radial basis function (RBF) [12], [19] kernels. Furthermore, multiple kernel learning classifier (m-MKL) was utilized in [11], while in [23], the minimum reconstruction error served as a classification criterion, after reconstructing the patch using class-specific dictionaries.

Some attempts have recently also been made to use deep learning (DL) techniques and especially CNNs, after their impressive performance in large scale color image classification [26]. 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 [27], 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 [28], 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 [26] (AlexNet) was used in [29] 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 Hounsfield unit (HU) windows. Moreover, 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.

B. 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 backpropagation 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.* [26] 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.

In this paper, 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.

III. 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.

A. 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 [30], 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 (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,

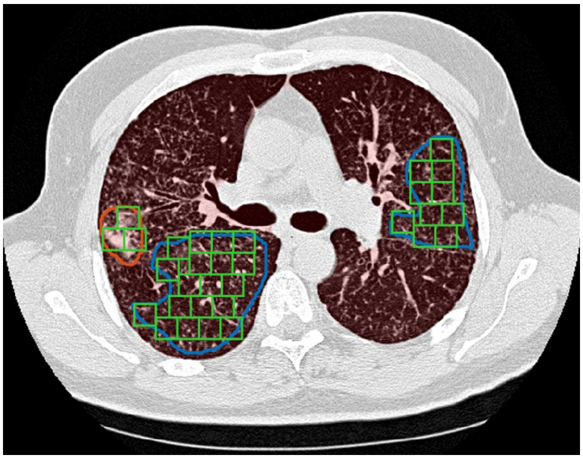


Fig. 2. Example of generating image patches through the annotations of a CT slice. The lung field is displayed with transparent red. The polygons are the ground truth areas with considered pathologies. The patches have 100% overlap with the lung, at least 80% overlap with the ground truth and 0% overlap with each other.

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. Fig. 2 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 I 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' 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 [26]. 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.

B. 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 [31]. 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.

TABLE I
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

1) Architecture: On the basis of these principles, we designed the network presented in Fig. 3. 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 [32], 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 to an area of $(L+1)^2$ for the $L_{\rm th}$ layer. Hence, the number of

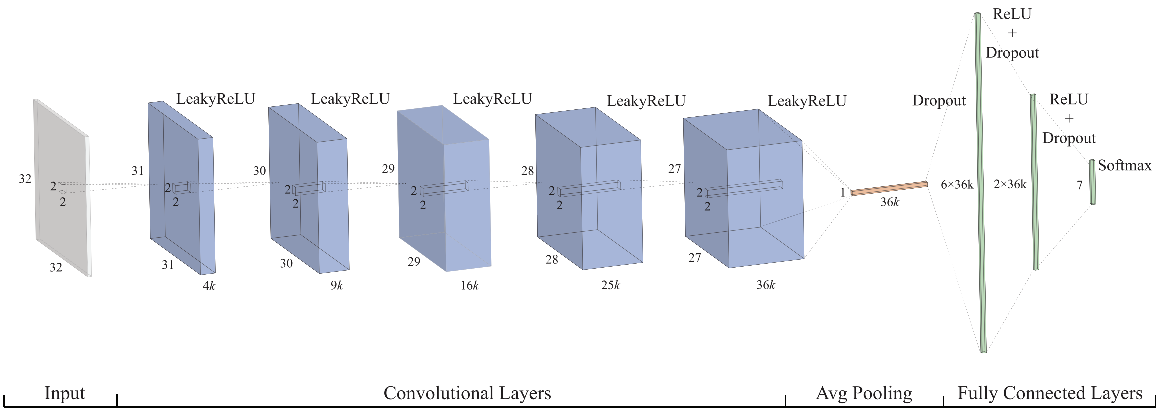


Fig. 3. Architecture of the proposed CNN for lung pattern classification. The value of parameter k was set to 4.

kernels we use for the L_{th} 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 [33].

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 [34], 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)).

$$f(x) = \begin{cases} x, & x > 0 \\ ax, & \text{else} \end{cases} \tag{1}$$

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., $\alpha = 0.01$) was originally proposed, here we increase its value to 0.3, which considerably improves performance. Similar observations have also been reported in other studies [35]. 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 [36] 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 [37]. 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 (F_{avg}) over the different classes ((2)) (see Section IV). An improvement is considered significant if the relative increase in performance is at least 0.5%.

IV. 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.

A. Experimental Setup

1) 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

TABLE II
PERFORMANCE OF THE CNN FOR DIFFERENT CONFIGURATIONS

Dropout fraction	Pooling type	Pooling percentage	Kernel number multiplier (k)	Number of kernels for L$_{th}$ layer	Number of conv layers	Kernel size	Input scale factor	Activation function	Testing F$_{avg}$	# Epochs × Epoch time
0	Avg	100%	4	$k(L+1)^2$	5	2 × 2	1	LReLU(0.3)	0.7908	90 × 11s
0.5	Max	100%	4	$k(L+1)^2$	5	2 × 2	1	LReLU(0.3)	0.8105	69 × 11s
0.5	Avg	50%	4	$k(L+1)^2$	5	2 × 2	1	LReLU(0.3)	0.7895	249 × 11s
0.5	Avg	25%	4	$k(L+1)^2$	5	2 × 2	1	LReLU(0.3)	0.7452	286 × 12s
0.5	Avg	100%	4	17	5	2 × 2	1	LReLU(0.3)	0.8446	300 × 12s
0.5	Avg	100%	4	36	5	2 × 2	1	LReLU(0.3)	0.8508	386 × 32s
0.5	Avg	100%	3	$k(L+1)^2$	5	2 × 2	1	LReLU(0.3)	0.8266	427 × 7s
0.5	Avg	100%	5	$k(L+1)^2$	5	2 × 2	1	LReLU(0.3)	0.8425	362 × 14s
0.5	Avg	100%	4	$k(L+1)^2$	7	2 × 2	1	LReLU(0.3)	0.8432	295 × 23s
0.5	Avg	100%	4	$k(L+1)^2$	6	2 × 2	1	LReLU(0.3)	0.8559	215 × 18s
0.5	Avg	100%	4	$k(L+1)^2$	4	2 × 2	1	LReLU(0.3)	0.8443	372 × 6s
0.5	Avg	100%	4	$k(L+1)^2$	5	2 × 2	1.5	LReLU(0.3)	0.8223	196 × 21s
0.5	Avg	100%	4	$k(L+1)^2$	5	3 × 3	1.5	LReLU(0.3)	0.8390	328 × 260s
0.5	Avg	100%	4	$k(L+1)^2$	5	3 × 3	1	LReLU(0.3)	0.8147	193 × 67s
0.5	Avg	100%	4	$k(L+1)^2$	5	2 × 2	1	ReLU	0.7871	90 × 11s
0.5	Avg	100%	4	$k(L+1)^2$	5	2 × 2	1	LReLU(0.01)	0.8094	110 × 12s
0.5	**Avg**	**100%**	**4**	$\boldsymbol{k(L+1)^2}$	**5**	**2 × 2**	**1**	**LReLU(0.3)**	**0.8547**	**386 × 12s**

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.

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

where

$$\begin{aligned}\text{recall}_c &= \frac{\text{samples correctly classified as } c}{\text{samples of class } c}\\ \text{precision}_c &= \frac{\text{samples correclty classified as } c}{\text{samples classified as } c}\\ \text{Accuracy} &= \frac{\text{correctly classified samples}}{\text{total number of samples}}.\end{aligned} \tag{3}$$

2) Implementation: The proposed method was implemented[1] using the Theano [38] framework, while for AlexNet and VGG-Net we used Caffe [39]. Methods which do not involve convolutional networks were coded in python and MATLAB. All experiments were 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.

B. 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.

1) Tuning of Hyper-Parameters: Here we demonstrate the effect of the most crucial choices for the architecture and the training procedure. Table II 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 ($\sim 5\times$). 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 6×6-7×7. 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.

[1] An implementation of the proposed method is provided by the authors as supplementary material, and can be downloaded at http://ieeexplore.ieee.org.

TABLE III
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**

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 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 F_{avg} 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 F_{avg} of more than 6%, an effect obviously due to overfitting.

Table III 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 Fig. 4, 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 III and its stable behavior independently from its parameters.

2) Comparison With the State of the Art: Table IV 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%.

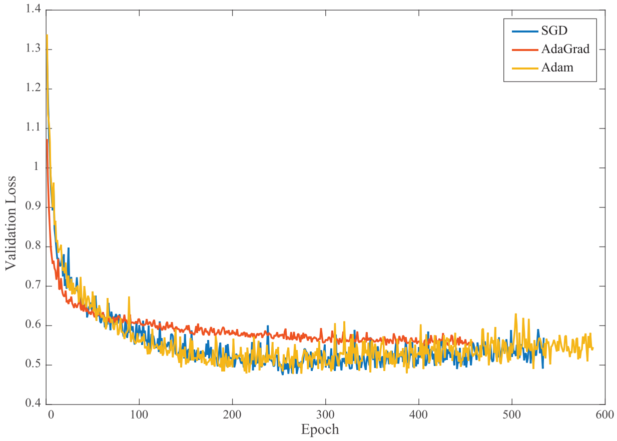


Fig. 4. Comparison of the convergence speed between three optimizers.

TABLE IV
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 [13]	LBP+histogram	kNN	0.7322	0.7333
Anthimopoulos [9]	Local DCT + histogram	RF	0.7786	0.7809
Proposed	**CNN**		**0.8547**	**0. 8561**

TABLE V
COMPARISON OF THE PROPOSED METHOD WITH OTHER CNNS

Method	F_{avg}	Accuracy
Li [28]	0.6657	0.6705
LeNet [40]	0.6783	0.6790
AlexNet [26]	0.7031	0.7104
Pre-trained AlexNet [26]	0.7582	0.7609
VGG-Net [32]	0.7804	0.7800
Proposed Method	**0.8547**	**0.8561**

Table V 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 [40], 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 [26] and VGG-Net-D [35], two networks

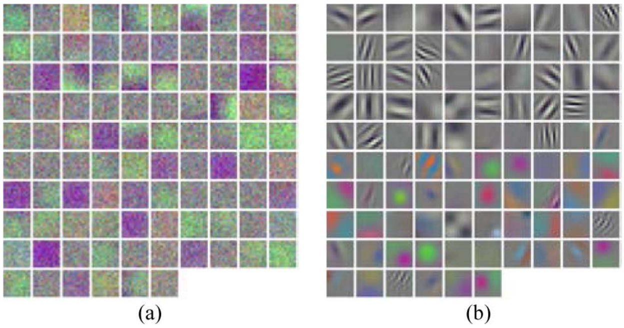


Fig. 5. Filter of the first layer of AlexNet by (a) training from scratch on our data, (b) fine-tuning the pre-trained on ImageNet version.

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 [32]. 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 (Fig. 5(a)) 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 (Fig. 5(b)) 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 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.* [13] and the proposed CNN. Fig. 6 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 Fig. 6. For each ROC, the area under the curve (AUC) was computed and the 95% confidence interval was plotted according to [41]. 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 [42] 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 Fig. 6, where large distance between curves correlates with statistically significant differences.

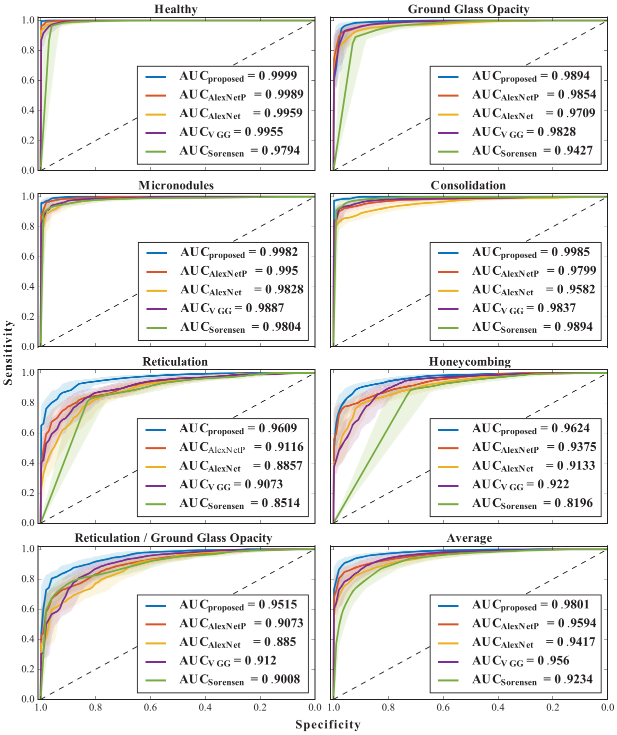


Fig. 6. ROC analysis for the proposed CNN and four previous methods: AlexNet, AlexNet pre-trained (AlexNetP), VGG-Net and the method by Sorensen *et al.* [13]. The analysis was performed per class (one-vs-all) while the average over all classes is also presented. For each ROC, the AUC is given and the 95% confidence interval is plotted.

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,

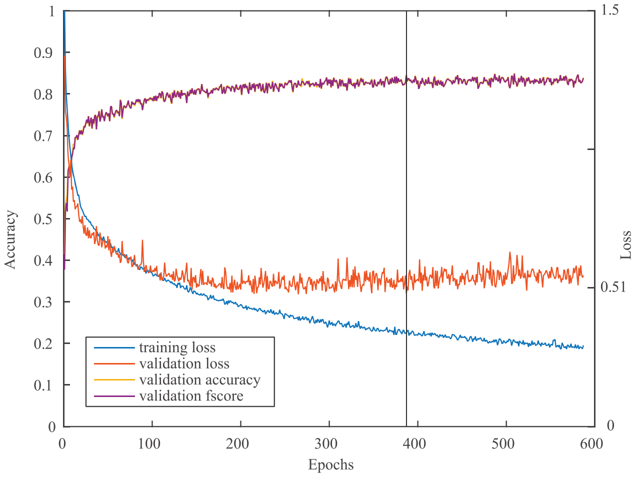


Fig. 7. Loss curves during the training of the proposed system.

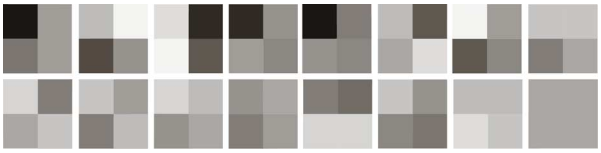
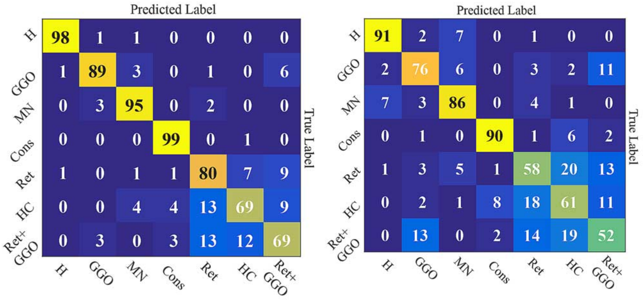


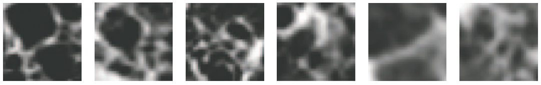
Fig. 8. The 2 × 2 kernels from the first layer of the proposed CNN.



(a)

(b)

Fig. 9. Confusion matrices of: (a) the proposed method, (b) the method by Sorensen *et al.* [13]. 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.



	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%

Fig. 10. Examples of misclassified patches by the proposed CNN. The output of the network is displayed below each patch.

other algorithmic choices, like the standard ReLU and the max pooling, may have affected the result, as shown in Table II, 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, making them prohibitive for the clinical practice.

3) Analysis of the System's Performance: In this paragraph, we provide additional insight into the performance of the proposed method. In Fig. 7, 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 F_{avg} 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 Fig. 8. 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.

Fig. 9(a) 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. Fig. 10 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 relevant to the discrimination between idiopathic pulmonary fibrosis (IPF) and non-specific interstitial pneumonia (NSIP), which are the most common ILDs. Fig. 9(b) presents the corresponding confusion matrix for the method by Sorensen *et al.* [13]. 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.

V. Conclusion

In this paper, 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 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.

References

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

[2] M. Demedts and U. Costabel, "ATS/ERS international multidisciplinary consensus classification of the idiopathic interstitial pneumonias," *Eur. Respiratory J.*, vol. 19, no. 5, pp. 794–796, 2002.

[3] I. Sluimer, A. Schilham, M. Prokop, and B. Van Ginneken, "Computer analysis of computed tomography scans of the lung: A survey," *IEEE Trans. Med. Imag.*, vol. 25, no. 4, pp. 385–405, Apr. 2006.

[4] K. R. Heitmann *et al.*, "Automatic detection of ground glass opacities on lung HRCT using multiple neural networks," *Eur. Radiol.*, vol. 7, no. 9, pp. 1463–1472, 1997.

[5] S. Delorme, M.-A. Keller-Reichenbecher, I. Zuna, W. Schlegel, and G. Van Kaick, "Usual interstitial pneumonia: Quantitative assessment of high-resolution computed tomography findings by computer-assisted texture-based image analysis," *Invest. Radiol.*, vol. 32, no. 9, pp. 566–574, 1997.

[6] R. Uppaluri *et al.*, "Computer recognition of regional lung disease patterns," *Am. J. Respir. Crit. Care Med.*, vol. 160, no. 2, pp. 648–654, 1999.

[7] C. Sluimer, P. F. van Waes, M. A. Viergever, and B. Van Ginneken, "Computer-aided diagnosis in high resolution CT of the lungs," *Med. Phys.*, vol. 30, no. 12, pp. 3081–3090, 2003.

[8] Y. Song, W. Cai, Y. Zhou, and D. D. Feng, "Feature-based image patch approximation for lung tissue classification," *IEEE Trans. Med. Imag.*, vol. 32, no. 4, pp. 797–808, Apr. 2013.

[9] M. Anthimopoulos, S. Christodoulidis, A. Christe, and S. Mougiakakou, "Classification of interstitial lung disease patterns using local DCT features and random forest," in *Proc. 36th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2014, pp. 6040–6043.

[10] Y. Uchiyama *et al.*, "Quantitative computerized analysis of diffuse lung disease in high-resolution computed tomography," *Med. Phys.*, vol. 30, no. 9, pp. 2440–2454, 2003.

[11] K. T. Vo and A. Sowmya, "Multiple kernel learning for classification of diffuse lung disease using HRCT lung images," in *Proc. IEEE Eng. Med. Biol. Soc. Conf.*, 2010, vol. 2010, pp. 3085–3088.

[12] A. Depeursinge *et al.*, "Near-affine-invariant texture learning for lung tissue analysis using isotropic wavelet frames," *IEEE Trans. Inf. Technol. Biomed.*, vol. 16, no. 4, pp. 665–675, Jul. 2012.

[13] L. Sørensen, S. B. Shaker, and M. De Bruijne, "Quantitative analysis of pulmonary emphysema using local binary patterns," *IEEE Trans. Med. Imag.*, vol. 29, no. 2, pp. 559–569, Feb. 2010.

[14] Y. Xu *et al.*, "Computer-aided classification of interstitial lung diseases via MDCT: 3D adaptive multiple feature method (3D AMFM)," *Acad. Radiol.*, vol. 13, no. 8, pp. 969–978, 2006.

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

[16] P. D. Korfiatis, A. N. Karahaliou, A. D. Kazantzi, C. Kalogeropoulou, and L. I. Costaridou, "Texture-based identification and characterization of interstitial pneumonia patterns in lung multidetector CT," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 3, pp. 675–680, May 2010.

[17] I. Mariolis *et al.*, "Investigation of 3D textural features' discriminating ability in diffuse lung disease quantification in MDCT," in *Proc. IEEE Int. Conf. Imag. Syst. Tech.*, 2010, pp. 135–138.

[18] A. Depeursinge *et al.*, "Optimized steerable wavelets for texture analysis of lung tissue in 3-D CT: Classification of usual interstitial pneumonia," in *Proc. 12th Int. Symp. Biomed. Imag.*, 2015, pp. 403–406.

[19] M. J. Gangeh *et al.*, "A texton-based approach for the classification of lung parenchyma in CT images," in *Proc. MICCAI*, 2010, pp. 595–602.

[20] A. Foncubierta-Rodríguez *et al.*, "Using multiscale visual words for lung texture classification and retrieval," in *Medical Content-Based Retrieval for Clinical Decision Support*. New York: Springer, 2012, vol. 7075, LNCS, pp. 69–79.

[21] R. Xu *et al.*, "Classification of diffuse lung disease patterns on high-resolution computed tomography by a bag of words approach," *Proc. MICCAI*, vol. 14, pt. 3, pp. 183–90, 2011.

[22] W. Zhao *et al.*, "Classification of diffuse lung diseases patterns by a sparse representation based method on HRCT images," in *Proc. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2013, pp. 5457–5460.

[23] K. T. Vo *et al.*, "Multiscale sparse representation of HRCT lung images for diffuse lung disease classification," in *Int. Conf. Image Process.*, 2012, pp. 441–444.

[24] M. Zhang *et al.*, "Pulmonary emphysema classification based on an improved texton learning model by sparse representation," *Proc. SPIE*, vol. 8670, 2013.

[25] Q. Li, W. Cai, and D. D. Feng, "Lung image patch classification with automatic feature learning," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2013, pp. 6079–6082.

[26] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," *Adv. Neural Inf. Process. Syst.*, p. 9, 2012.

[27] G. Van Tulder and M. de Bruijne, "Learning features for tissue classification with the classification restricted Boltzmann machine," *Med. Comput. Vis., Algorithms for Big Data*, pp. 47–58, 2014.

[28] Q. Li *et al.*, "Medical image classification with convolutional neural network," in *Proc. 13th Int. Conf. Control Automat. Robot. Vis.*, Dec. 2014, vol. 2014, pp. 844–848.

[29] M. Gao *et al.*, "Holistic classification of CT attenuation patterns for interstitial lung diseases via deep convolutional neural networks," in *1st Workshop Deep Learn. Med. Image Anal.*, 2015, pp. 41–48.

[30] A. Depeursinge *et al.*, "Building a reference multimedia database for interstitial lung diseases," *Comput. Med. Imag. Graph.*, vol. 36, no. 3, pp. 227–238, 2012.

[31] M. Tuceryan and A. K. Jain*, The Handbook of Pattern Recognition and Computer Vision*, C. H. Chen, L. F. Pau, and P. S. Wang, Eds. Singapore: Word Scientific, 1998.

[32] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Int. Conf. Learn. Represent.*, 2015.

[33] G. Hinton, "Dropout : A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.

[34] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *ICML Workshop on Deep Learning for Audio, Speech, and Language Processing (WDLASL 2013)*, 2013, vol. 28.

[35] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolution network," *ICML Deep Learn.*, pp. 1–5, 2015.

[36] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Int. Conf. Learn. Represent.*, 2015, pp. 1–13.

[37] K. He, X. Zhang, S. Ren, and J. Sun*, Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*, arXiv:1502.01852.

[38] F. Bastien *et al.*, "Theano: New features and speed improvements," *ArXiv Prepr*, pp. 1–10, 2012.

[39] Y. Jia *et al., Caffe: Convolutional Architecture for Fast Feature Embedding*, 2014, arXiv:1408.5093.

[40] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2323, Nov. 1998.

[41] J. Carpenter and J. Bithell, "Bootstrap confidence intervals: When, which, what? A practical guide for medical statisticians," *Stat. Med.*, vol. 19, no. 9, pp. 1141–1164, 2000.

[42] 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. 839–43, 1983.