Analysis of Machine Learning Algorithms for Diagnosis of Diffuse Lung Diseases

Isadora Cardoso¹ Eugenia Almeida¹ Hector Allende-Cid² Alejandro C. Frery¹ Rangaraj M. Rangayyan³ Paulo M. Azevedo-Marques⁴ Heitor S. Ramos¹

¹Instituto de Computação, Universidade Federal de Alagoas, Maceió, Brazil
²Escuela de Ingeniería Informática, Pontificia Universidad Católica de Valparaíso, Valparaíso, Chile
³Department of Electrical and Computer Engineering, Schulich School of Engineering, University of Calgary, Calgary, Alberta, Canada
⁴Department of Internal Medicine, Ribeirão Preto Medical School, University of São Paulo, Ribeirão Preto, Brazil

Address for correspondence Prof. Heitor S. Ramos, Universidade Federal de Alagoas, Institute of Computing, Av. Lourival Melo Mota, S/N, Cidade Universitária, Maceió, Alagoas, 57072900, Brazil (e-mail: heitor@laccan.ufal.br).

Methods Inf Med

Summary

Background Diffuse lung diseases (DLDs) are a diverse group of pulmonary disorders, characterized by inflammation of lung tissue, which may lead to permanent loss of the ability to breathe and death. Distinguishing among these diseases is challenging to physicians due to their wide variety and unknown causes. Computer-aided diagnosis (CAD) is a useful approach to improve diagnostic accuracy, by combining information provided by experts with Machine Learning (ML) methods.

Objectives Exploring the potential of dimensionality reduction combined with ML methods for diagnosis of DLDs; improving the classification accuracy over state-of-the-art methods.

Methods A data set composed of 3252 regions of interest (ROIs) was used, from which 28 features were extracted per ROI. We used Principal Component Analysis, Linear Discriminant Analysis, and Stepwise Selection – Forward, Backward, and Forward-Backward to reduce feature dimensionality. The feature subsets obtained were used as input to the following ML methods: Support Vector Machine, Gaussian Mixture Model, k-Nearest Neighbor, and Deep Feedforward Neural Network. We also applied a Deep Convolutional Neural Network directly to the ROIs.

Results We achieved the maximum reduction from 28 to 5 dimensions using LDA. The best classification results were obtained by DFNN, with 99.60% of overall accuracy.

Conclusions This work contributes to the analysis and selection of features that can efficiently characterize the DLDs studied.

Introduction

Diffuse lung diseases (DLDs), also called interstitial lung diseases, are a diverse group of chronic pulmonary diseases, encompassing over 200 disorders, characterized by inflammation of lung tissue and may progressively lead to permanent respiratory failure and death. For many of these diseases, the mean survival time is estimated to be from 2 to 5 years from the time of diagnosis.¹ Therefore, the diagnosis of these disorders should be done as soon as possible to help the patients survive; however, distinguishing among these diseases is a challenge to physicians due to their wide variety and unknown causes.

The diagnosis of a DLD is based on the complete history of the patient, as well as visual findings on chest X-ray images. High-Resolution Computed Tomography (HRCT) is generally...
considered to be the most appropriate method of imaging, since it minimizes the superposition of thoracic structures and produces a large number of slices to be analyzed. However, in some cases, the diagnosis cannot be confirmed radiologically, requiring more invasive procedures, but there is no guarantee of reliable diagnosis even after performing invasive biopsies.

Computer-aided diagnosis (CAD) is a useful approach to improve diagnostic accuracy, by combining information provided by experts with Machine Learning (ML) methods. Computers present several advantages over humans in image analysis: they do not suffer from boredom, fatigue, and distractions; they are consistent in their analyses; and they can combine the knowledge of several experts into a single analysis. However, it is worth mentioning that as the complete diagnosis is based on subjective information such as the patient’s history and socioeconomic factors, CAD cannot substitute the physician for giving diagnosis, but can act as a complementary tool for the physician, and possibly reduce the necessity of invasive procedures.

ML methods are among the best available methods to extract hidden information from DLD data. Several methods have been widely used over the years to classify various patterns related to DLDs. Pereyra et al. applied k-Nearest Neighbors (kNN), with k = 5, in texture and fractal features to classify the same six patterns presented in this work, achieving classification rates up to 82.62%. Using the same data set, Almeida et al. used Gaussian Mixture Models (GMM) and fuzzy membership functions to classify the most significant features, achieving an average classification accuracy of 63.50%. Artificial Neural Networks (ANN) is applied in several works, such as Kauczor et al., that compared segmentation and subsequent classification of HRCT containing ground-glass opacities (GG) patterns between ANN and a density mask, achieving a better accuracy when using ANN: 89.00% against 58.00% with the density mask; and Uchiyama et al. that extracted six features related to physical lung measures and classified them based on the seven following patterns: GG, reticular and linear opacities, nodular opacities, honeycombing (HC), emphysematous change, consolidation (CD), and normal lung tissue (NL). They achieved a sensitivity and specificity up to 97.40% and 88.00%, respectively. However, these methods need manual extraction of features from data in order to perform analysis.

In recent years, Deep Learning (DL) has set a new trend in ML, providing state-of-the-art performance in several fields. DL methods are based on ANN, but present better stability, generalization ability, and scalability. There are some recent works using DL methods, such as the Deep Convolutional Neural Network (DCNN) to classify DLDs. Anthimopoulos et al. proposed a DCNN architecture to classify GG, reticulation, CD, micronodules (NM), HC, and a combination of GG and reticulation patterns in DLDs, achieving an average F1-score of 85.47%. Christodoulidis et al. extended this work training a DCNN in six texture database and joined the learned knowledge in an ensemble to classify the patterns. It achieved an average F1-score of 88.17%, surpassing different methods that use hand crafted features and a range of different classifiers, as well as different CNN architectures. Shin et al. used the learning transferred from ImageNet, a database of world pictures for visual object recognition, to different CNN architectures to classify healthy, emphysema, GG, fibrosis, MN, and CD patterns in DLDs, achieving an accuracy of 79.00%. Hashimoto et al. built an ensemble of two deep DCNN architectures, which one responsible for learning a correct classification between CD and NL and GG and NL. They achieved a classification accuracy of 93.30%.

The aforementioned studies indicate that detecting DLDs from HRCT images using ML and DL methods is feasible, with the latter yielding to better results, especially applying DCNN with knowledge transferred from other data sets. Besides that, DCNNs are capable of learning information from the given images, avoiding the necessity of extracting features.

In the present work, we compare the discriminatory power of a few ML methods, including DL, when applied to DLD data.

**Objectives**

The objectives of this study are: 1) exploring the potential of feature dimensionality reduction by applying ML methods to HRCT images of DLDs; and 2) improving the classification accuracy over state-of-the-art methods.

**Methods**

**Data Set**

From the Radiology Information System (RIS) of the Clinical Hospital of the School of Medicine of Ribeirao Preto – University of Sao Paulo (HCFMRP-USP), Pereyra et al. retrieved 247 HRCT images from 108 different exams. Thoracic radiologists evaluated each image and grouped them into six patterns related to DLD, with approximately 35 images per class: pulmonary consolidation (PC), emphysematous areas (EA), septal thickening (ST), HC, GG, and NL.

For each image, the radiologists indicated the most representative examples of each evaluated pattern, which were extracted as non-overlapping slices. These HRCT slices were normalized to eight bits per pixel and processed with histogram equalization and centering to highlight high-attenuation (at −400 Hounsfield Units (HU)) and low-attenuation (at −800HU) patterns. The image slices were segmented using the mathematical morphology operator closing, binary thresholding (at −800HU), and object selection. The aforementioned process produced a total of 3252 regions of interest (ROIs), with 64 × 64 pixels each, which characterize each radiographic pattern: 450 ROIs for PC, 501 for EA, 589 for ST, 529 for HC, 594 for GG, and 589 for NL. Fig. 1 shows examples of the ROIs for each selected radiological pattern. After these procedures, the ROI labels were again reviewed and validated with radiologists for the sake of accuracy.

A set of 28 texture and fractals features was extracted from each ROI, which are:

- Low- and high-order statistics: histogram mean, histogram median, histogram standard deviation, histogram skewness, and histogram kurtosis;
• Haralick’s\textsuperscript{12} texture measures: energy, difference moment, correlation, variance, inverse difference moment, sum of squares: average, sum of squares: variance, sum of squares: entropy, entropy difference variance, difference entropy, information measures of correlation 1, information measures of correlation 2, maximal correlation coefficient (these features were calculated for four orientations (0°, 45°, 90°, and 135°), using one pixel as the reference distance);
• Laws’ measures of texture energy;\textsuperscript{13} Laws’ wave measures, Laws’ wave measures (rotation-invariant), Laws’ ripple measures, Laws’ ripple measures (rotation-invariant), and Laws’ level measures (these features were extracted using the appropriate convolution masks);
• Statistical measures of the power spectral density (PSD): standard deviation, mean, and median (obtained from the Discrete Fourier Transform (DFT)); and
• Fractal Dimension (FD), using the method described by Banik et al.\textsuperscript{14} Details about these features are available in Rangayyan.\textsuperscript{15}

**Dimensionality Reduction Methods**

Before applying the ML methods to the data set described earlier, we studied the use of dimensionality reduction methods. This is an important step since a high feature dimension may lead to a sparse space unable to represent the model appropriately – this phenomenon is called the Curse of Dimensionality. This can reduce the accuracy, increase the possibility of overfitting and, hence, produce poor results. Moreover, some ML methods cannot handle high-dimensional data sets.\textsuperscript{16} In this stage, we used Principal Component Analysis (PCA),\textsuperscript{16} Linear Discriminant Analysis (LDA),\textsuperscript{17} and Stepwise Selection – Forward (SSF), Backward (SSB), and Forward-Backward (SSFB),\textsuperscript{18} with the Bayesian Information Criterion as the penalty criterion.

LDA and PCA perform a transformation of the data into a new set. This property can be seen as a disadvantage if the goal is to reduce the number of feature dimensions, since it would still be necessary to extract all features of the data to apply these methods. In order to avoid this limitation, and also prevent overfitting while discarding irrelevant and/or redundant features, we use the feature selection methods mentioned above.

**Machine Learning Methods**

We use the ML methods kNN, GMM, SVM, Deep Feedforward Neural Network (DFNN), and DCNN to perform classification.

kNN is one of the simplest ML methods. It calculates the distance between each test point and the training points, and assigns to the test point the majority class among the k-nearest neighbors.

GMM is a probabilistic model of clustering that represents each class as a mixture of Gaussian distributions where the mean and covariance matrix are unknown parameters, estimated by maximizing the likelihood.\textsuperscript{17} As an extension, Fraley and Raftery\textsuperscript{19} proposed the use of the labels of the observations to determine the best parameterization of the covariance matrix, the cluster form, and the number of Gaussians in the mixture for each class, turning GMM into a supervised algorithm.

SVM is a supervised classification method that aims to find a hyperplane that maximizes the separation between classes. It is important that this hyperplane not only maximizes the separation between classes, but also that the margins between classes and the corresponding hyperplanes are as wide as possible. SVM works very well with high-dimensional data and diminishes the effect of the curse of dimensionality. It is possible to apply SVM to data that are not separated by linear functions using the kernel method, which consists of transforming the original space into a new one in such a way as to facilitate the use of a linear function to separate the classes.\textsuperscript{17}

The universal approximation theorem states that ANNs with linear output nodes and a single hidden layer can approximate any function by simply increasing the size of the hidden layer. However, the size of the hidden layer may become impractical. In addition, the ANN may fail to learn the represented function and, consequently, have poor generalization. A deeper architecture with a large number of hidden layers can, in many circumstances, reduce the number of nodes needed to represent the model, which can reduce generalization errors by having a greater ability to learn the represented function.\textsuperscript{6} A DFNN is nothing but an ANN with multiple hidden layers.

DCNNs are a type of ANN composed of multiple stages, divided into three parts (convolution, activation, and pooling) dedicated to extracting features, followed by traditional hidden layers and a final classification layer. The use of the convolution function to extract features brings some advantages, such as: the sparse interaction property improves the efficiency, since the learned filter is usually much smaller than the input vector, which needs less storage memory and fewer operations to perform; parameter sharing, i.e., the learning of a whole filter at a time instead of a single value, does not affect the time of knowledge propagation, but reduces the size of the storage required; and equivariant representation makes the input equivariant to translation.\textsuperscript{6}
The training process of a DCNN requires a massive amount of data in order to learn information through the features extracted by itself. This learning can come from architectures pre-trained with a different set of images, and the acquired knowledge is applied to the desired problem, an approach called Transfer Learning (TL). TL can save training time and is especially useful in problems with limited available training datasets, common in medical images. However, due to equivariant representation, it is possible to create synthetic observations derived from the real observations, a method called data augmentation: for each epoch, the method generates images randomly from the original ones, according to the selected parameters. Hence, the network almost never sees two exactly same training examples. This approach does not only reduces the cost of collecting a huge dataset, but by manipulating the training images, we increase the data variance, creating a more robust network, i.e., a network that avoids overfitting and generalizes better. In numbers, the data augmentation step generates a total of images equivalent to the number of epochs times the number of images in the training set.

Reproducibility

Programing Platform

Our platform of choice to implement the methods used in this work was the R language (version 3.3.2). We used the packages stats (for PCA and Stepwise selection), MASS (for LDA), class (for kNN), e1071 (for SVM), mclust (for GMM), and h2o (for DFNN). For DCNN, we used the Python language, with the Keras library as an interface for the TensorFlow library. We used 10-fold cross-validation to validate the ML algorithms.

All dimensionality reduction and ML experiments were performed in a Linux OS on a machine with two Nvidia GTX 1080 Ti GPUs, Intel R CoreTM i7–7700K CPU @ 4.20GHz, and 32 GB of RAM. For the sake of reproducibility, all seeds used in the experiments can be obtained from the first author.

Parameters of ML Methods

The parameters used in each ML method are presented in Table 1. Besides building a DCNN from the related fundamentals, we also used a pre-trained network, VGG-net, as proposed by Simonyan and Zisserman, using the weights of ImageNet.

The referred DCNN architecture built from fundamentals presented the best results and is exhibited in Fig. 2, where “Input” represents the first layer, i.e., the observations, and “Output” is the last layer.

Results

We applied the aforementioned dimensionality reduction methods and obtained, from the 28 features in the full set of features, 18 for SSB, 13 for PCA, 12 for SSFB, 11 for SSF, and 5 for LDA.

Using PCA, the 13 first principal components captured 99.09% of the data variance. The first principal component represents about 40% of the variance. Each component after the 8th represents less than 0.01%, and each one after the 17th component represents less than 0.001%. The variance present in the data is almost completely captured by eight components. We decided to preserve 13 components since they gave the best result when used with the ML methods.

LDA presents the smallest vector among the assessed techniques, with only five dimensions. Its first dimension represents about 40% of the variance, coincidently similar to the PCA method. Although similar amount of information is represented by both PCA and LDA in their first components, it is difficult to know which are the features that contribute to the information retained by both methods due to the different transformations each technique adopts.

With stepwise selection, it is easier to know which features to choose. Table 2 shows which features were chosen by using the stepwise selection methods; the symbol X represents that the feature is in the selected subset. There are some features chosen by the three stepwise selection methods used and others discarded by all of them. This is valuable information regarding which features are important or not, but it is outside the scope of this work to investigate the reason for their relevance.

The subsets obtained by applying dimensionality reduction and the original data set were used as input to the ML methods, except DCNN, which uses directly the extracted ROIs. With this approach, we are in position to verify the influence dimensionality reduction methods have on the studied case.

Fig. 3 shows the accuracy achieved by using the ML methods combined with dimensionality reduction. The figure does not include DCNN since it does not use the features data set, but the ROIs themselves. It is noteworthy that each ML method performs differently depending on the dimensionality reduction method used. For SVM, the best result, about 87%, was obtained by using the full data set of 28 features. This makes sense, since SVM is robust to the curse of dimensionality. kNN obtained about 83% using the LDA subset and GMM obtained about 85% using the SSF subset. With both methods, these results present better accuracy when compared to case with the full set of features; especially with GMM, which achieved an improvement of about 10%.

The DCNNs here presented similar results, but the pre-trained DCNN provided significantly greater results: 99.60% of accuracy against 99.24% when using the DCNN built from fundamental.

The best accuracy was provided by the DFNN, presenting the highest result of 99.60%, the same obtained using the pre-trained DCNN. The accuracy of the DFNN decreases with the feature subsets, possibly because they lose valuable information.

Table 3 presents the best results and their standard deviation obtained by the classification methods along with the dimensionality reduction procedures used to achieve them. To evaluate all methods together, including DCNN, we used the following macro-averaged measures, obtained from...
the confusion matrix: F1 score, specificity, sensitivity, and accuracy.

Table 4 shows the training time needed by the ML methods along with the Confidence Interval (CI), estimated by running 10 replications of the respective algorithms. While kNN, SVM, and GMM need less than 30 seconds to train their model, DCNN and DFNN require several minutes, especially when using DCNN with pre-trained weights. DFNN needs about 50 minutes and DCNN needs about 6 hours. However, the prediction time is quite similar for all methods: less than 2 seconds to classify an observation.

**Discussion**

The results obtained show that dimensionality reduction methods are effective in improving the quality of classification procedures. It is important to note that more information does not always lead to better results with ML methods; a judicious choice of features leads to the best results.

The choice of the best dimensionality reduction method depends on which ML method is applied. It is necessary to test different methods to obtain a good insight about their performance. Dimensionality reduction provides several benefits, such as better understanding about the data, better classification, and enhanced generalization.

ML methods require calibration, and since they have different properties, they behave differently according to the input data. There is no universal solution to the problems at hand; it is necessary to test the chosen methods to determine their parameters and obtain the best possible results. The solutions presented in this work are not optimal; it is likely that more experimentation and computational

---

**Table 1** Parameters used by the ML methods to obtain the results presented in this work

| Method       | Neighbors | Distance metric | SVM Kernel | SVM Penalization | DFNN Hidden layers | DFNN Activation function | DFNN Epochs | DFNN Learning function | DFNN Penalization | DCNN Convolution layers | DCNN Max pooling layers | DCNN Epochs | DCNN Batch size | DCNN Activation layers | DCNN Last layer | DCNN Learning function | DCNN Penalization | DCNN Data augmentation | DCNN Architecture | DCNN Learning Function | DCNN Batch size | DCNN Epochs | DCNN Last layer |
|--------------|-----------|-----------------|------------|------------------|-------------------|---------------------|-------------|------------------------|-------------------|------------------------|-----------------------|-------------|-----------------|----------------------|-----------------|----------------------|-------------------|------------------------|-------------------|-------------|-----------------|
| KNN          | k = 5,    | Euclidean       | Non-homogeneous polynomial, \((xy + b)^d\), with \(d = 2, \gamma = 0.5, b = 3\) \((x \text{ and } y \text{ are the vectors from training and test in the input space, respectively})\) | \(C_s = 1\)      | 3 layers with 230 nodes each, | Hyperbolic Tangent, | 300 epochs, | AdaDelta, with learning rate \(lr = 1\), and hyperparameters \(p = 0.99\) and \(\epsilon = 10^{-5}\) | Ridge and Lasso penalties, both with constraint values set to \(10^{-5}\) | all the filters are of size \(3 \times 3\), with a stride of 1, | all the filters are of size \(2 \times 2\), with a stride of 2, | 200 epochs, | sets of 20 samples, | ReLU (rectified linear unit) function: \(f_{\text{ReLU}}(x) = \max(0, x)\), | A fully-connected feed-forward layer using softmax as the activation function, | AdaDelta, with the same hyperparameters as those used in DFNN, | Dropout (that randomly removes nodes to avoid overfitting), with a probability equal to 3/10, | performs shear range, random zoom, and horizontal flip | VGGNet with 16-layer network, | Stochastic Gradient Descent (SGD), with \(lr = 0.001\) and Nesterov momentum of 0.9, | sets of 10 samples, | 300 epochs, | the same used when building from fundamentals |
power would have led to better results. The parameters used in the present work were obtained empirically, but in a reproducible manner.

The result presented here, as expected, show that DL methods improve in reference to state-of-the-art results. However, they need more time to train and are computationally more expensive. Another limitation is that they need a large amount of data to train appropriately the models. The data set used here, with 3252 observations, may not be large enough for this task, so the results obtained by the DFNN might be biased. It is difficult to know for sure if this is the case, since it would be necessary to use a large data set to test the model which, for a number of reasons, is hard to obtain in medical applications. We may overcome this limitation with the DCNN.

DCNN relies on TL and data augmentation to obtain more knowledge and to apply it in the chosen problem. Christodoulidis et al.\(^8\) and Anthimopoulos et al.\(^7\) demonstrated that it is possible to apply both approaches in DLDs. In our results, we achieved the same accuracy using a DCNN with pre-

Table 2 Comparison between the feature subsets obtained using stepwise selection

<table>
<thead>
<tr>
<th>Features</th>
<th>SSF</th>
<th>SSB</th>
<th>SSFB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Histogram Mean</td>
<td>―</td>
<td>―</td>
<td>―</td>
</tr>
<tr>
<td>Histogram Median</td>
<td>X</td>
<td>X</td>
<td>―</td>
</tr>
<tr>
<td>Histogram Standard Deviation</td>
<td>―</td>
<td>X</td>
<td>―</td>
</tr>
<tr>
<td>Histogram Skewness</td>
<td>―</td>
<td>X</td>
<td>―</td>
</tr>
<tr>
<td>Histogram Kurtosis</td>
<td>―</td>
<td>X</td>
<td>―</td>
</tr>
<tr>
<td>DFT Mean</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>DFT Median</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>DFT Standard Deviation</td>
<td>―</td>
<td>―</td>
<td>―</td>
</tr>
<tr>
<td>Fractal Dimension</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Laws’ Wave Measures</td>
<td>―</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Laws’ Wave Measures (Rotation-Invariant)</td>
<td>―</td>
<td>X</td>
<td>―</td>
</tr>
<tr>
<td>Laws’ Ripple Measures</td>
<td>―</td>
<td>―</td>
<td>―</td>
</tr>
<tr>
<td>Laws’ Ripple Measures (Rotation-Invariant)</td>
<td>―</td>
<td>―</td>
<td>―</td>
</tr>
<tr>
<td>Laws’ Level Measures</td>
<td>X</td>
<td>―</td>
<td>X</td>
</tr>
<tr>
<td>Haralick’s Energy</td>
<td>―</td>
<td>X</td>
<td>―</td>
</tr>
<tr>
<td>Haralick’s Difference Moment</td>
<td>―</td>
<td>X</td>
<td>―</td>
</tr>
<tr>
<td>Haralick’s Correlation</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Haralick’s Variance</td>
<td>―</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Haralick’s Inverse Difference Moment</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Haralick’s Sum of Squares: Average</td>
<td>―</td>
<td>―</td>
<td>―</td>
</tr>
<tr>
<td>Haralick’s Sum of Squares: Variance</td>
<td>―</td>
<td>―</td>
<td>―</td>
</tr>
<tr>
<td>Haralick’s Sum of Squares: Entropy</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Table 2 (Continued)

<table>
<thead>
<tr>
<th>Features</th>
<th>SSF</th>
<th>SSB</th>
<th>SSFB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Haralick’s Entropy</td>
<td>―</td>
<td>X</td>
<td>―</td>
</tr>
<tr>
<td>Haralick’s Difference Variance</td>
<td>X</td>
<td>―</td>
<td>X</td>
</tr>
<tr>
<td>Haralick’s Difference Entropy</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Haralick’s Information Measures of Correlation 1</td>
<td>―</td>
<td>―</td>
<td>―</td>
</tr>
<tr>
<td>Haralick’s Information Measures of Correlation 2</td>
<td>―</td>
<td>―</td>
<td>―</td>
</tr>
<tr>
<td>Haralick’s Maximal Correlation Coefficient</td>
<td>X</td>
<td>X</td>
<td>―</td>
</tr>
<tr>
<td>Total of chosen features</td>
<td>11</td>
<td>18</td>
<td>12</td>
</tr>
</tbody>
</table>
trained weights and a DFNN. The DCNN has the advantage of not needing feature extraction, operating on the images themselves. The DCNN built from fundamentals presented a slightly inferior result; perhaps, with more fine-tuning or more data, this network could achieve the same result or surpass the best results.

Furthermore, Christodoulidis et al. concluded something similar to our work about DCNNs: pre-trained DCNN presents better results than architectures built from fundamentals, which testifies to the importance of TL, especially in applications with limited available medical data.

It is worth mentioning that the overall accuracies obtained using kNN (83.12%) and GMM (84.71%) in the present work surpass the accuracy of classification obtained by Pereyra et al.2 (82.62%) and Almeida et al.23 (63.00%).

**Conclusions**

This work performed an analysis of six radiographic patterns related to DLDs. Before using the ML methods, we applied dimensionality reduction techniques to remove redundant and/or irrelevant data. The best reduction was provided by LDA, which reduced 28 features to 5. However, since LDA transforms the data into a new data set, it is hard to know which features contribute to improvement of the classification performance. Some features were not selected by any method (SSF, SSB, and SSFB). This gives clues regarding which features are important or not.

We used SVM, kNN, GMM, and ANN to classify the data. We also used two DL methods, DCNN and DFNN. The best results were achieved for both DFNN and DCNN (using pre-trained weights), outperforming the other techniques, with an overall accuracy of 99.60%. However, DFNN requires feature extraction, which is not necessary when using...
DCNN. On the other hand, after performing feature extraction using the available techniques, DFNN needs less time to execute than DCNN.

This work advances the analysis and selection of those features that can efficiently characterize the DLDs studied. Furthermore, the method presented can be generalized and may support the development of a CAD system to analyze any sufficiently large dataset of HRCT images, not only DLDs.

Funding
This work was partially funded by Fapeal, CNPq, and SEFAZ-AL. The work of Héctor Allende-Cid was supported by the project FONDECYT Initiation into Research 11150248.

References
15. Rangayyan RM. Biomedical Image Analysis. CRC Press; 2004