Multi-source Transfer Learning with Convolutional Neural Networks for Lung Pattern Analysis

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

***Abstract*—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 (CAD) 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.**

***Index Terms*—Interstitial lung diseases, convolutional neural networks, texture classification, model ensemble, transfer learning, knowledge distillation, model compression**

I. Introduction

INTERSTITIAL lung diseases (ILDs) include more than 200 chronic lung disorders characterized by inflammation of the lung tissue, which often leads to pulmonary fibrosis. Fibrosis progressively reduces the ability of the air sacs to capture and carry oxygen into the bloodstream and eventually causes permanent loss of the ability to breathe. Early diagnosis of such diseases is crucial for making treatment decisions, while misdiagnosis may lead to life-threatening complications [1]. Although ILDs are histologically heterogeneous, they mostly have similar clinical manifestations, so that differential diagnosis is challenging even for experienced physicians. High resolution computed tomography (HRCT) is considered the most appropriate protocol for screening ILDs, due to the specific radiation attenuation properties of the lung tissue. The CT scans are interpreted by assessing the extent and distribution of the existing ILD pathologies in the lung. However, the inherent difficulty of the problem and the large quantity of radiological data that radiologists have to scrutinize result in low diagnostic accuracy and high inter- and intra-observer variability, which may be as great as 50% [2]. This ambiguity in the radiological assessment often leads to additional histological biopsies which increase both the risk and cost for patients. In order to assist the radiologist with the diagnosis and to avoid biopsies, a lot of research has been done towards computer-aided diagnosis (CAD) systems. The basic module of such systems is often a fixed scale texture classification scheme that detects the various ILD patterns in the CT scan and outputs a map of pathologies, which is later used to reach a final diagnosis. To this end, a great variety of image descriptors and classifiers have been proposed for recognizing lung patterns.

Deep learning techniques and especially convolutional neural networks (CNNs) have attracted much attention after the impressive results in the ImageNet Large Scale Visual Recognition Competition (ILSVRC) in 2012 [3]. Numerous studies followed that transformed the state of the art for many computer vision applications. Even though CNNs have existed for a couple of decades already [4], this breakthrough was only made possible thanks to the current processing capabilities and the large image databases available. The potential of deep learning in medical image analysis is already being investigated and the initial results are promising [5]. However, the adaptation of the existing deep learning tools from the domain of natural color images to medical images brings new challenges.

Firstly, 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

Manuscript received December 6, 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. S. Christodoulidis and M. Anthimopoulos contributed equally to this work. Asterisk indicates corresponding author.

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

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).

L. Ebner and A. Christe are with the Department of Diagnostic, Interventional and Pediatric Radiology, Bern University Hospital "Inselspital", 3010 Bern, Switzerland (e-mails: 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).

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 [5]. 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.

To deal with these issues, in [6] 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.

II. Related Work

In this section, we provide a brief overview of the previous studies on ILD pattern classification, followed by a short introduction to transfer learning using CNNs.

A. ILD Pattern Classification

A typical ILD pattern classification scheme takes as input a local region of interest (ROI) or volume of interest (VOI), depending on the available CT imaging modality, and is mainly characterized by the chosen feature set and classification method. The first proposed systems used handcrafted texture features such as first order statistics, gray level cooccurrence matrices, run-length matrices and fractal analysis [13]. Other systems utilized filter banks [14], [15], morphological operations [16], wavelet transformations [17] and local binary patterns [18]. Moreover, the ability of multiple detector computed tomography (MDCT) scanners to provide three-dimensional data has motivated researchers to expand existing 2D texture descriptors to three dimensions [19]–[21]. 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 [22], [23] and sparse representation models [24], [25]. Restricted Boltzmann machines (RBM) have also been used [26] to learn multi-scale filters with their responses being used as features. Once the feature vector of a ROI or VOI 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 [14] and Bayesian [13] classifiers, k-nearest neighbors [18], [20], fully-connected artificial neural networks [16], random forests [15] and support vector machines [22], [27].

Some attempts have recently been made to use deep learning techniques and especially CNNs for the classification of lung tissue patterns. Unlike the aforementioned feature learning methods, CNNs learn features in a supervised manner and train a classifier at the same time, by minimizing a cost function. Although the term deep learning refers to multiple learning layers, the first studies on the problem utilized rather shallow architectures. A modified RBM was proposed in [28] that resembles a CNN and performs both feature extraction and classification. Hidden nodes share weights and are densely connected to output nodes, while the whole network is trained in a supervised manner. In [29], a CNN with one convolutional layer and three fully-connected layers was used, but the rather shallow architecture of the network was unable to capture complex non-linear image characteristics. In our previous work [6], we designed and trained for the first time (to the best of our knowledge) a deep CNN for the problem of lung tissue classification, which outperformed shallower networks. The proposed CNN consists of 5 convolutional layers with 2 $\times$ 2 kernels and LeakyReLU activations, followed by global average pooling and three fully-connected layers. Other studies have used popular deep CNNs that exploit the discriminative power gained by pretraining on huge natural image datasets [30]. Although the architecture of these networks is far from optimal for lung tissue classification, they managed to achieve relatively good results by transferring knowledge from other tasks.

B. 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 [31]. 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 [32] . 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 [33]. 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 [34]. On the other hand, if there are no overfitting issues, the best strategy is to transfer and fine-tune every layer [33]. This way, the discovered features are

TABLE I: 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 [7]	Color	250	1	100	25000	98.304	257880	85870
DTD [8]	Color	47	120	1	5640	229.95 $\pm$ 89.14	180351	87485
FMD [9]	Color	10	100	1	1000	158.3 $\pm$ 43.2	18247	6285
KTB [10]	Grey	27	160	1	4480	331.776	207360	69120
KTH-TIPS-2b [11]	Color	11	4	108	4752	40	31481	10410
UIUC [12]	Grey	25	40	1	1000	307.2	47250	15750

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 [35]. 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 [36], 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 [37], 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 [33], 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 [30], [38]. 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.

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

A. Databases

Six texture benchmark databases were employed to serve as source domains for the multi-source transfer learning: the Amsterdam library of Textures (ALOT) [7], the Describable Textures Dataset (DTD) [8], the Flickr Material Database (FMD) [9], Kylberg Texture Database (KTB) [10], KTH-TIPS-2b [11] and the Ponce Research Group's Texture database (UIUC) [12]. 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) [39] and the Bern University Hospital, "Inselspital" (Insel) database [6].

1) 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 I summarizes the characteristics of the original source databases and the corresponding patch datasets.

2) Target Domain Dataset: The HUG database [39] 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

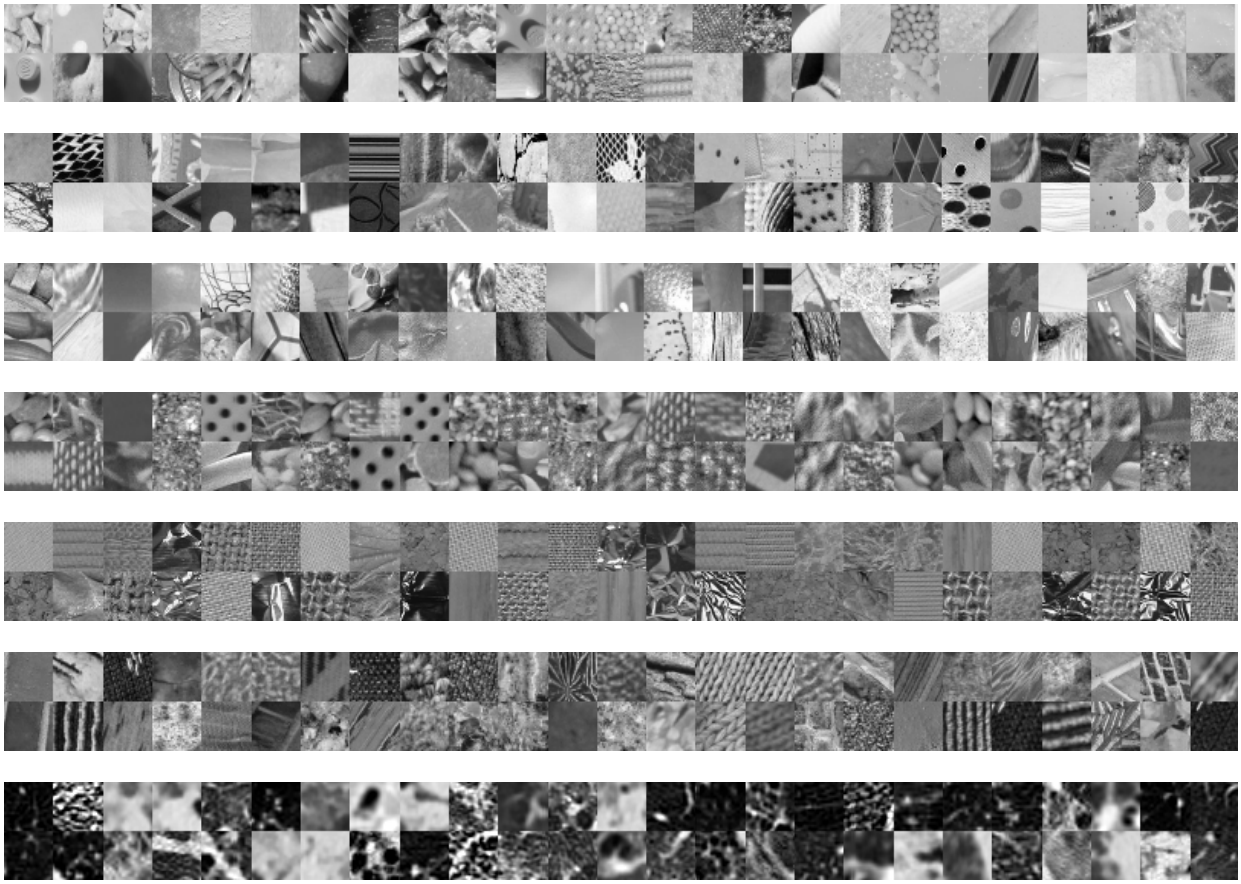

Fig. 1: Typical samples from each dataset. The color databases were converted to gray scale. From top to bottom: ALOT, DTD, FMD, KTB, KTH-TIPS-2b, UIUC, ILD

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, 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 $\times$ 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 [6].

B. 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 [6] throughout the different steps of the method. The input of the network is an image patch of 32 $\times$ 32 pixels. This patch is convolved by five subsequent convolutional layers with 2 $\times$ 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 [40] 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%.

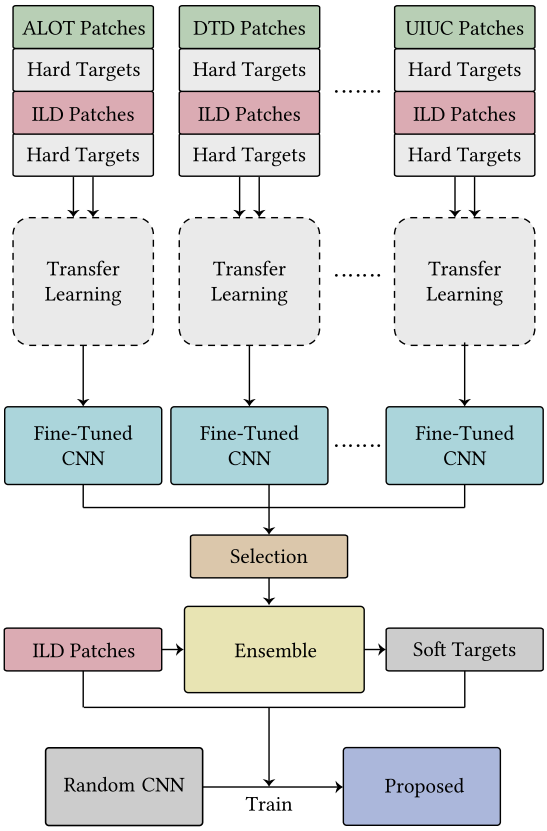

Fig. 2: Multi-source Transfer Learning: Knowledge is transferred from each source database to a different CNN. A selection process combines CNNs into an ensemble that is used to teach a single randomly initialized model.

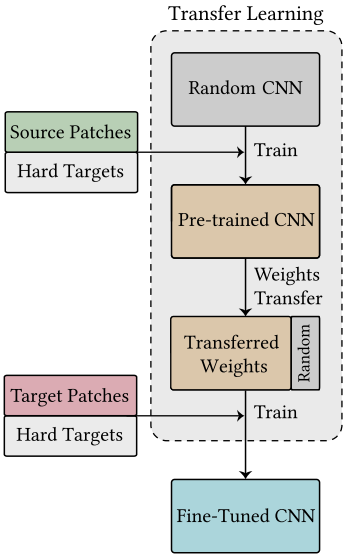

Fig. 3: Transfer Learning through weight transfer

C. Multi-source Transfer learning

The source datasets presented in Section III-A demonstrate a wide spectrum of different characteristics, as shown in Fig. 1 and Table I; 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. Fig. 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.

1) Single-Source Transfer Learning: Fig. 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 [6]. According to [33] 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.

2) 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 [41]. 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 [42]. 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. 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 $M = 5$.

3) 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 [43] [44]. 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 III-B. 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.

D. 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. Fig. 4 depicts an outline of our multi-task learning approach.

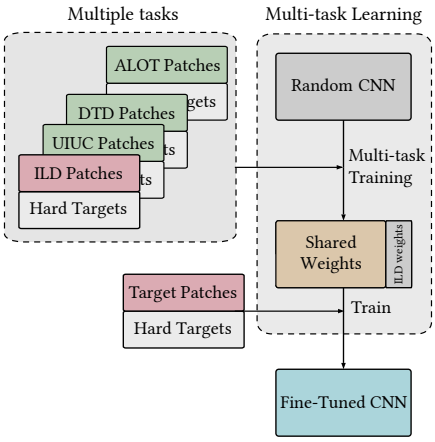

Fig. 4: Multi-task Learning

IV. Experimental Setup & Results

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

A. 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.

1) 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.

2) Implementation: The proposed method was implemented in Python using the Keras [45] framework with a Theano [46] 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.

B. 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 III-C. Finally, we analyze the performance of the proposed network and compare with other methods.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2016.2636929, IEEE Journal of Biomedical and Health Informatics

1) 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 III-B, 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 Fig. 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. 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 [33].

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.

2) Knowledge Fusion in an Ensemble: Fig. 5 also illustrates the performance of the ensemble that was built as described in Section III-C2. 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

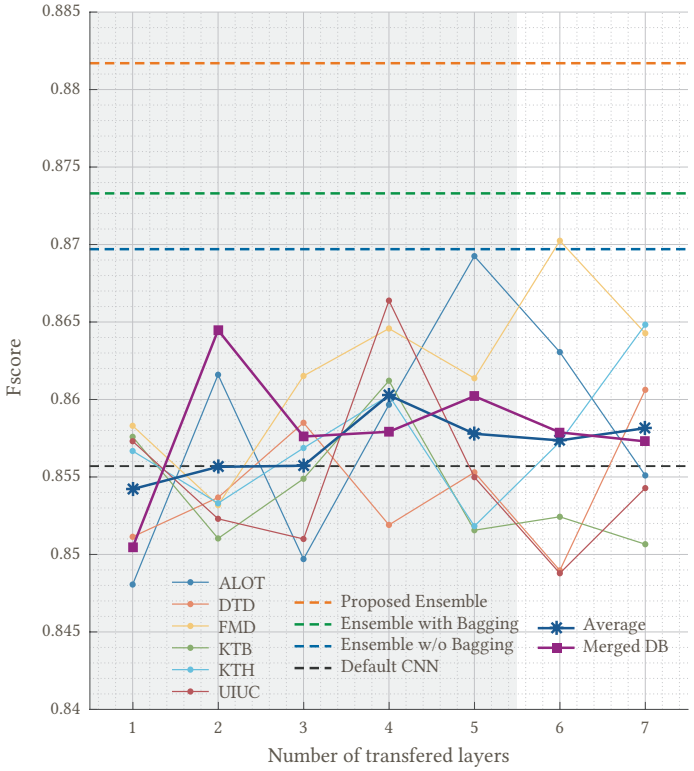

Fig. 5: The F_1-score produced by transferring of knowledge from single source domains for different number of transferred layers, averaged over three experiments. The horizontal lines correspond to the CNN without knowledge transfer or the different ensembles of CNNs.

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.

3) 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.

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

Study	Method	F_{avg}
Gangeh [22]	Local pixel textons - SVM-RBF	0.6942
Sorensen [18]	LBP, histogram - kNN	0.7420
Anthimopoulos [15]	Quantiles of local DCT, histogram - RF	0.8170
Li [29]	5-layer CNN	0.6657
LeNet [4]	7-layer CNN	0.6783
AlexNet [3]	8-layer CNN	0.7031
Pre-trained AlexNet	8-layer CNN	0.7582
VGG-Net [47]	16-layer CNN	0.7804
Anthimopoulos [6]	8-layer CNN	0.8557
Proposed Methods	**Multi-task Learning**	**0.8631**
	Compressed 8-layer CNN	**0.8751**
	Ensemble of CNNs	**0.8817**

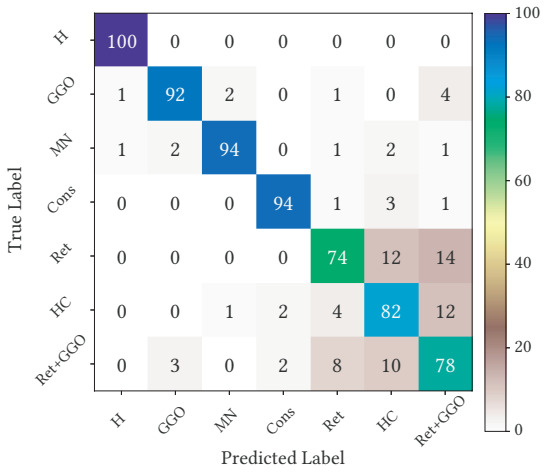

Fig. 6: Confusion matrix of the proposed compressed model.

4) 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 III-D. 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.

Table II 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 [6]. Finally, Fig. 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 [6].

V. Conclusion

In this paper 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.

References

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

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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, September 2011.

[11] P. Mallikarjuna, A. T. Targhi, M. Fritz, E. Hayman, B. Caputo, and J.-O. Eklundh, "The kth-tips2 database," 2006.

[12] 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.

[13] 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.

[14] 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.

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

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

[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] 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.

[20] 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," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 14, no. 3, pp. 675–680, 2010.

[21] A. Depeursinge, P. Pad, A. S. Chin, A. N. Leung, D. L. Rubin, H. Muller, and M. Unser, "Optimized steerable wavelets for texture analysis of lung tissue in 3-d ct: Classification of usual interstitial pneumonia," in *Biomedical Imaging (ISBI), 2015 IEEE 12th International Symposium on*. IEEE, 2015, pp. 403–406.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] G. van Tulder and M. de Bruijne, "Combining generative and discriminative representation learning for lung ct analysis with convolutional restricted boltzmann machines," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1262–1272, 2016.

[29] 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.

[30] 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, feb 2016.

[31] 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.

[32] 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.

[33] 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.

[34] L. Castrejon, Y. Aytar, C. Vondrick, H. Pirsiavash, and A. Torralba, "Learning aligned cross-modal representations from weakly aligned data," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[35] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, Jul. 1997.

[36] L. Zheng, Y. Zhao, S. Wang, J. Wang, and Q. Tian, "Good practice in cnn feature transfer," *arXiv preprint arXiv:1604.00133*, 2016.

[37] 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)*, June 2015, pp. 36–45.

[38] 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.

[39] 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.

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

[41] 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.

[42] 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.

[43] 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. New York, NY, USA: ACM, 2006, pp. 535–541.

[44] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," *ArXiv e-prints*, Mar. 2015.

[45] F. Chollet, "Keras," https://github.com/fchollet/keras, 2015.

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

[47] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.