Image Embedding and Model Ensembling for Automated Chest X-Ray Interpretation

Edoardo Giacomello
Dipartimento di Elettronica,
Informazione e Bioinformatica
Politecnico di Milano
edoardo.giacomello@polimi.it

Pier Luca Lanzi
Dipartimento di Elettronica,
Informazione e Bioinformatica
Politecnico di Milano
pierluca.lanzi@polimi.it

Daniele Loiacono
Dipartimento di Elettronica,
Informazione e Bioinformatica
Politecnico di Milano
daniele.loiacono@polimi.it

Luca Nassano
Dipartimento di Elettronica,
Informazione e Bioinformatica
Politecnico di Milano
luca.nassano@mail.polimi.it

Abstract—**Chest X-ray (CXR) is perhaps the most frequently-performed radiological investigation globally. In this work, we present and study several machine learning approaches to develop automated CXR diagnostic models. In particular, we trained several Convolutional Neural Networks (CNN) on the CheXpert dataset, a large collection of more than 200k CXR labeled images. Then, we used the trained CNNs to compute embeddings of the CXR images, in order to train two sets of tree-based classifiers from them. Finally, we described and compared three ensembling strategies to combine together the classifiers trained. Rather than expecting some performance-wise benefits, our goal in this work is showing that the above two methodologies, i.e., the extraction of image embeddings and models ensembling, can be effective and viable to solve tasks that require medical imaging understanding. Our results in that perspective are encouraging and worthy of further investigation.**

I. INTRODUCTION

The chest X-ray (CXR) is perhaps the most common imaging examination performed for screening, diagnostic purposes, and management of many life threatening diseases. Indeed, automated CXR interpretation could significantly benefit medical practices for tasks like the prioritizing of patients in emergency departments or the screening of a large population. Among the others, machine learning seems a very promising technology to tackle this problem, especially since the recent developments on Convolutional Neural Networks (CNNs) [1], that proved to be very successful when it comes at image understanding and classification [2]. Besides the algorithmic and computational improvements, the availability of very large datasets is arguably a key element behind the recent successes of these approaches. Unfortunately, the amount of medical data publicly available is rather limited, both for legal and practical reasons, making it very difficult to apply machine learning in this domain. However, many recent efforts have been made to provide the scientific community with quite large dataset of labeled medical images, that allowed the training of machine learning models, able to reach or even to outperform human experts. In particular, the *CheXpert* dataset [6] – which includes more than 200k CXR labeled images – has been recently made available for a scientific competition on automated CXR interpretation [16] that proved

how machine learning might be able to achieve outstanding results.

On the other hand, in the medical field we still lack of pre-trained machine learning models that could be easily applied to new tasks with a fine-tuning using a rather limited amount of data, as it already happens for language [24] and general-purpose images [4]. In particular, in the case of medical imaging, making available pre-trained models to extract *image embeddings* might be more practical solution. In fact, *image embeddings* – low-dimensional representations of the images as a continuous vectors – can be easily extracted using a Convolutional Neural Network (CNN) and used as input to train classifiers based on trees, kernels, Bayesian statistics, etc. Thus, the advantage of using embeddings lies in retaining the benefit of a CNN trained on a large corpus of images while designing a specific classifier for new data and, eventually, for a slightly different problem. In general, we envision the possibility of developing a library of embedding models trained by the research community for different kind of medical imaging and for different tasks. Such embedding models and the classifiers trained using them, could be also be combined and mixed together using ensembling strategies.

In this paper, we perform a preliminary investigation of these ideas: inspired by the work of Pham et al. [15], we trained seven different CNNs on the Chexpert dataset and used them also to extract image embeddings. Then, we used these embeddings to train a set of classifiers based on Random Forests [21] and eXtreme Gradient Boosting [22]. Finally, we also investigated three different ensembling strategies to combine all the model trained. Our results are promising and show that image embeddings, as expected, do contain the relevant information to train additional classifiers with a performance similar or better than the one achieved using the CNNs in the first place. We also showed that quite simple ensembling strategies could be used to effectively combine together classifiers, leading to better overall performances.

The paper is organized as follows. In Section II we provide an overview of the most relevant papers on the application of Deep Learning to automate chest X-Ray interpretation and in Section III we describe the CheXpert dataset used in this work. Then, in Section IV we describe in detail our approach: (i) how we dealt with labels uncertainty, (ii) how we exploited

the labels dependencies, (iii) the CNN classifiers trained, (iv) how we computed the image embeddings and used them to train additional classifiers, and (v) the ensembling strategies used to combine the trained classifiers. Thus the experimental design and experimental results are discussed respectively in Section V and Section VI. Finally, we draw our conclusions in Section VII.

II. RELATED WORK

Along with the availability of large datasets containing chest X-Rays labeled images [5]–[7], several successful approaches based on deep learning and convolutional neural networks have been proposed in the literature. In their seminal paper [8], Rajpurkar et al. trained a DenseNet-121 [18] model on the ChestX-ray14 dataset [7]; their model, dubbed as *CheXNet*, achieved state-of-the-art performance on the classification of the 14 major thoracic diseases and outperformed expert radiologists on the detection of pneumonia. In a later work [9], Rajpurkar et al. introduced *CheXNeXt* that improves the performance of *CheXNet* and achieves a performance similar to expert radiologist on 10 thoracic diseases. Notable works that focus on the ChestX-ray14 dataset are the work of Kumar et al. [10], who introduced a cascaded CNNs that can diagnose all the 14 thoracic diseases better than baseline, and the work of Lu et al. [11], who applied an evolutionary algorithm to search for the most suited CNN architecture to solve the classification task. A different approach was followed by Ye et al. [12] that introduced *Probabilistic-CAM* (PCAM), an extension of CAM [13], to perform the localization of thoracic diseases on the ChestX-ray14 dataset in a semi-supervised fashion; at the same time, the localization model trained can be successfully applied also to solve the image classification problem with a performance similar or better than some of the previous approaches introduced in the literature, such as *CheXNet* [7].

More recently, two very large datasets have been released: CheXpert [6] and MIMIC-CXR [5], which include respectively 224000 and 350000 images. Along with the publication of the dataset, Irvin et al. [6] also proposed a solution based on a 121-layer DenseNet trained with different approaches to deal with uncertainty that is present in the labels of CheXpert dataset. Their model was able to achieve performance similar or better than expert radiologists on the classification of 5 thoracic diseases, selected as the most representative of the dataset. Instead, Rubin et al. [14] introduced *DualNet*, consisting of two CNNs jointly trained on frontal and lateral chest radiographies, included in the very large MIMIC-CXR dataset [5]. Their results show that *DualNet* outperforms state-of-the-art classifiers trained separately on a single type of image (i.e., either frontal or lateral).

Finally, in a very recent work Pham et al. [15] trained several state-of-the-art CNN on the CheXpert dataset, showing the benefits of exploiting the conditional dependencies among the labels in the training as well as of employing an ensemble of classifiers with different architectures instead of a single one.
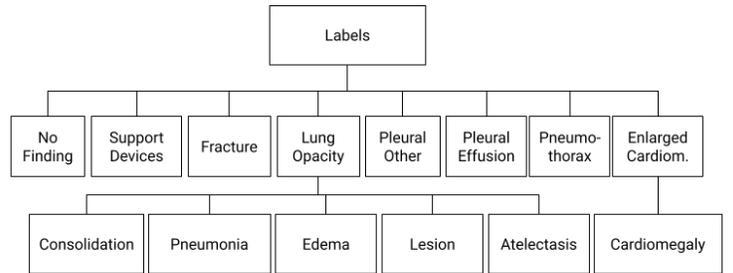
Fig. 1: Hierarchical structure of the findings. This is a slightly simplified version of the hierarchy described in [6].

III. CHEXPERT DATASET

From a Machine Learning point of view, CXR interpretation can be modeled as an image classification problem and, thus, requires a large dataset that is labeled with quality standards close to the ones provided by expert radiologists. In this work, we focused on the CheXpert dataset [16], that is composed of 223316 CXRs of 65240 patients, collected from the Stanford Hospital from October 2002 to July 2017. The dataset is provided in two different image formats: the high quality format is 16-bit PNG and the low-quality format is 8-bit PNG. Each image is annotated with a vector of 14 labels, corresponding to major findings in a CXR. The labels have been extracted from text radiology reports using an automatic rule-based labeler. In particular, the labeling process consisted of three different phases: (i) the *Impression* section – that generally summarizes the key finding of the exam – of each report is analyzed and a list of mentions is extracted, by matching a list of phrases designed by multiple expert radiologists; (ii) each mention is assigned to a label according to a level of confidence between *positive*, *negative* and *uncertain*; (iii) each image is encoded as a vector that include one element for each label: positive labels are encoded as 1, negative labels are encoded as 0, uncertain labels are encoded as *u*. CheXpert dataset includes two kinds of images: frontal and lateral X-Rays. Lateral images are available only for some patients, generally when the diagnosis is uncertain. For this reason, the amount of frontal images is much higher. In addition to the training set, the authors also provide a set of 200 images – annotated by human experts – that can be used to assess the performances of the machine learning approaches. TableI shows the data distribution among the 14 labels included in the dataset. Following what suggested literature [15], [16], in this work we focused only on five representative findings in CheXpert dataset: Atelectasis, Cardiomegaly, Consolidation, Edema, and Pleural Effusion. In addition, the 14 findings included in the dataset are not independent but they forms instead an hierarchy as showed in Figure 1. Accordingly, in order to be successful, any machine learning approach applied to CheXpert dataset would need to deal both with the uncertainty of the labeling process as well as with the dependency among the labels, that could be exploited to improve the performances.

Pathology	Positive (%)	Uncertain (%)	Negative (%)
No Finding	16974 (8.89)	0 (0.0)	174053 (91.11)
Enlarged Card.	30990 (16.22)	10017 (5.24)	150020 (78.53)
Cardiomegaly	23385 (12.24)	549 (0.29)	167093 (87.47)
Lung Opacity	137558 (72.01)	2522 (1.32)	50947 (26.67)
Lung Lesion	7040 (3.69)	841 (0.44)	183146 (95.87)
Edema	49675 (26.0)	9450 (4.95)	131902 (69.05)
Consolidation	16870 (8.83)	19584 (10.25)	154573 (80.92)
Pneumonia	4675 (2.45)	2984 (1.56)	183368 (95.99)
Atelectasis	29720 (15.56)	25967 (13.59)	135340 (70.85)
Pneumothorax	17693 (9.26)	2708 (1.42)	170626 (89.32)
Pleural Effusion	76899 (40.26)	9578 (5.01)	104550 (54.73)
Pleural Other	2505 (1.31)	1812 (0.95)	186710 (97.74)
Fracture	7436 (3.89)	499 (0.26)	183092 (95.85)
Support Devices	107170 (56.1)	915 (0.48)	82942 (43.42)

TABLE I: Number of Positive, Uncertain and Negative samples for each finding. In bold the five findings we focused on in this paper.

IV. METHODOLOGY

In this section, we provide an overview of the machine learning approaches we applied and compared on the CheXpert dataset. In particular, we first describe the approaches we used to deal with label uncertainty and dependency described in the previous section. Then, we provide an overview of all the machine learning models we compared on the CheXpert dataset: (i) several models based on convolutional neural networks and (ii) two models based on trees. Finally, we describe how these models can also be aggregated as an ensembling to compute a better final prediction.

A. Dealing with uncertain labels

As described in the previous section, the CheXpert dataset includes a significant number of samples that have been labeled as uncertain. The uncertainty could reflect both an unreliable diagnosis or an ambiguity in the report. In [6], Irvin et al. compared different policies to deal with uncertain labels, such as assuming uncertain labels as either positive or negative. On the other hand, Pham et al. [15] showed that these policies would possibly result in several wrong labels and, consequently, could be misleading for training a model. Accordingly, in this work we followed the *Label-Smoothing Regularization* (LSR) approach introduced by Szegedy et al. [17], that allows to exploit the large amount of uncertain labels in CheXpert dataset but prevents the model from becoming over confident on uncertain samples. This is achieved by replacing the uncertain label, *u*, with a random value drawn from a uniform distribution $U(a, b)$ (with $b > a > 0.5$).

B. Exploiting dependencies between labels

As described in Section III, the labels included in the CheXpert dataset have a hierarchical dependency. Thus, when training a classification model, such dependencies could be exploited to achieve better performances. To this purpose, inspired by the work of Pham et al. [15], we employed a *conditional training* approach that aims at learning from data the conditional probabilities distribution of labels. This approach relies an the hierarchical dependency model illustrated

in Figure 1 and involves a two-steps training process. First, we train our classifiers only with samples that have positive values (i.e., equal to 1) in labels that are not leaves in the label hierarchy (i.e., Lung Opacity and Enlarged Cardiomegaly as reported in Figure 1). Second, we perform an additional training of the classifiers on the whole dataset, to tune their prediction of labels at higher level in the hierarchy. As a result of this two-steps training process, the output of our classifiers can be viewed as the conditional probability that a label is positive assuming as positive its *parent* labels (if they exist in the hierarchy). Accordingly, when conditional training is employed, to predict the unconditional probability of unseen data we simply apply the Bayes rule: we compute the probability for each label as the product of the classifier outputs for that specific label and all the labels above in the hierarchy.

C. CXR Classification with CNNs

In this work we trained and compared seven different convolutional neural networks, that differ in terms of architectures, topology, and number of parameters. More specifically, the networks considered – along with the number of parameters – are DenseNet121 (7M), DenseNet169 (12,5M), DenseNet201 (18M) [18], InceptionResNetV2 (54M) [3], Xception (21M) [19] , VGG16 (15M) [2] , VGG19 (20M) [2]. We used different network architectures because, as reported also by other authors [15], each architecture has different performances on different labels. Indeed, since there are no prevailing architectures, an approach based on the aggregation of different models can be beneficial to the final performances as we will later discuss. In order to use the networks as classifiers, we removed the original dense layer and replaced it with a Global-Average-Pooling (GAP) [20] layer, followed by a Fully Connected Layer that matches the number of labels in our dataset.

D. CXR Classification with Trees

In the recent years, CNNs proved very successful in many image understanding tasks, including CXR interpretation. Arguably, the main reason behind these successes is the capability of CNNs to learn effective image representations directly from data, without the need of design task-specific features. To investigate better this idea, in this paper we combined CNNs with two well known classifiers based on trees: Random Forests [21] (RF) and eXtreme Gradient Boosting [22] (XGBoost). More specifically, we applied the same CNNs trained to classify the CXR images (as described in the Section IV-C) to extract also a compact image representation, usually called *image embeddings*. The underlying idea of this process, widely used nowadays in several application domain, is that a CNNs is composed by a a sequence of convolutional layer typically followed by one or more fully connected layers. While the convolutional part of a CNN can be basically seen as an universal features extractor, the fully connected layers are actually responsible to solve the specific tasks the CNN is applied to (i.e., either a classification or a regression problem). Thus, we

can easily apply a CNN to an image and extract the output of the convolutional part to use it as a large vector of features the describe the input image, i.e., an *image embedding*. In this work, we generated several image embeddings using the different CNNs trained on the CheXpert dataset and used such generated dataset to train RF and XGBoost classifiers.

E. Ensembling Strategies

As already mentioned, in a multi-label classification task, like the CheXpert one, it might be difficult to find a single classifiers that outperforms the others on each target and it might also happen that for some of the targets no strong classifiers are available. In this setting, we might rely on ensembling strategies that allow to combine several weak classifiers into a stronger one. In particular, in this paper we investigated three different approaches to combine together multiple classifiers:

Simple Averaging. The first approach we used simply consists in averaging the predictions made by the classifiers. If we call $y_i(\mathbf{x})$ the prediction vector provided as output by the classifier i for the input $(\mathbf{x})$, then the ensemble prediction $\tilde{y}(\mathbf{x})$ computed using N classifiers is:

$$\tilde{y}(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} y_i \qquad (1)$$

The major drawback of this approach is that it assigns the same weight to all the classifiers, without acknowledging that some classifiers may outperform others for specific labels or may be just more confident of their predictions for a specific input $(\mathbf{x})$.

Entropy-Weighted Average. An alternative approach to simple averaging is to weight each classifier by taking into account their confidence level. In particular, we developed an heuristic weighting approach based *Entropy*. In Information Theory, Entropy measures the level of uncertainty of the outcomes of some random variable. Accordingly, we might model the prediction of each classifier (for a specific label) as as random variable which follows a Bernoulli distribution with success probability equal to the actual output of the classifier for that label. We can thus use the Entropy value of such variable as a measure of the classifier confidence:

$$H_k(p_{k,i}) = -p_{k,i} \log_2 p_{k,i} - (1 - p_{k,i}) \log_2 (1 - p_{k,i}) \quad (2)$$

where $p_{k,i}$ is the prediction of the classifier k for label i. As a result, $H_k(p_i)$ measures the level of uncertainty of the classifier k, while $(1 - H(p_i))$ might be seen as the confidence of classifier k about its prediction on label i. In an ensemble of N classifiers that provide a prediction for L labels, for each input we will get a *prediction matrix* $P = (p_{k,i}) \in \mathbb{R}^{NxL}$. Applying Equation 2 we can combine for each label $i$the classifiers predictions as follows:

$$y_i = \sum_{k=1}^{N} (1 - H_k(p_{k,i})) \, p_{k,i} \qquad (3)$$

Stacking. The last aggregation approach we investigated involves Stacking [23]. This approach combines multiple classification models using a meta-classifier. Specifically, a train set is first used to train the base classifiers, then the predictions of the base models are used as features to train the meta-learner. The pseudo-code for the stacking algorithm is shown in Algorithm 1.

Algorithm 1 Stacking

1: **procedure** STACKEDGENERALIZATION
2:     $D \leftarrow \{(x_1, y_1), ..., (x_m, y_m)\}$ ▷ Training Dataset
3:     $\Phi \leftarrow \Phi_1, \Phi_2, ..., \Phi_T$ ▷ Base Classifier
4:     **for** $t \leftarrow 1, T$ **do**
5:         $h_y = \Phi_t(D)$
6:     **end for**
7:     $D' \leftarrow \emptyset$
8:     **for** $i = 1, 2, ..., m$ **do**
9:         **for** $t = 1, 2, ..., T$ **do**
10:            $z_{it} = h_t(x_i)$
11:        **end for**
12:        $D' \leftarrow D' \cup \{((z_{i1}, z_{i2}, ..., z_{iT}), y_i)\}$
13:    **end for**
14:    $h' \leftarrow \Phi(D')$ ▷ Learn meta-classifier
15:    **return** h'
16: **end procedure**

V. EXPERIMENTAL DESIGN

A. Preprocessing

In this work, for the sake of simplicity, we trained our classifiers only on the frontal images included in the CheXpert dataset, as they were present for every patient. Then, we further split the dataset into a training set (roughly 90% of samples corresponding to $N = 189116$ samples) and a validation set (roughly 90% of samples corresponding to $N = 1911$ samples) for tuning the model hyper-parameters. Thus, we kept the additional set of 202 samples included in CheXpert as test set to assess the final performances of our classifiers. We pre-processed the data by dropping additional information for each patient such as sex and age. The uncertain labels were mapped into values sampled from a uniform distribution $U(0.55, 0.85)$, following the LSR approach introduced in Section IV. Concerning the CXR images included in the dataset, we processed them to reduce as much as possible any noise, such as text or irregular borders, that could affect learning performances. Accordingly, we first resized the images to 256 x 256 pixels and then we applied a template matching algorithm in order to find a region of 224x224 containing a chest template. Moreover, to match the data shape with those of the network inputs, we converted the images from 1 to 3 channels (RGB), and we scaled their values in the range [0,1]. Finally, since the models have been pre-trained on ImageNet, we normalized all the images with respect to that dataset mean and standard deviation.

B. CNN Training

In our experimental analysis, we trained seven popular CNNs: DenseNet121, DenseNet169, DenseNet201, Inception-ResNetV2, Xception, VGG16, and VGG19. As described in

Model Name	Embedding Shape
DenseNet121	(1,1024)
DenseNet169	(1,1664)
DenseNet201	(1,1920)
InceptionResNetV2	(1,1536)
Xception	(1,2048)
VGG16	(1,512)
VGG19	(1,512)

TABLE II: Shape of the CXR image embeddings computed by each CNN.

Model Name	Max Depth	Min Sample Split	Min Sample Leaf
DenseNet121	15	2	10
DenseNet169	15	2	10
DenseNet201	30	10	10
InceptionResNetV2	30	10	1
Xception	30	10	1
VGG16	5	2	10
VGG19	15	50	1

TABLE III: Values of the hyper-parameters computed for each RF classifier

SectionIV-C, all these networks have been pre-trained on the ImageNet dataset [4] before training them on the CheXpert dataset: the networks are initialized using the weights provided by the Tensorflow 2.0 Keras module, discarding the classification layer while retaining the convolutional layers. To apply the conditional training approach described in SectionIV-B, in the first stage, we trained the networks only using the samples – 23526 samples out of 189116 ones – labeled as positive for all the findings that are not at the bottom of the label hierarchy (see Figure1); then, we froze all the layers except the last fully connected one and we fine-tuned the networks by training them on the whole training set. In both these training stages, binary cross-entropy was used as loss function and the learning rate was initially set to 1e-4, to be reduced by a factor of 10 after each epoch.

Besides training seven different CNNs to classify unlabeled CXRs images, we used them also to compute an image embedding for each sample of the CheXpert dataset. In particular, the embeddings are computing by extracting the output of the Global Average Pooling layer that is before the last fully connected layer used to classify the image. As a result, each trained CNN is able to map a CXR image into a large embedding vector, whose size actually depends on the topology of each CNN as illustrated in Table II.

C. Training Trees

Using the image embeddings as an input, we trained two sets of additional classifiers based on trees: for each embedding dataset (computed by using each one of the CNNs trained) a RF classifier and an XGBoost classifier were trained. Concerning the RF classifiers, we performed a grid search to optimize the classifier hyper-parameters based on the performances achieved on the validation set. More specifically we optimized the following hyper-parameters: (i) *Max Depth*, i.e., the largest tree depth allowed, that regulates the balance between accuracy and overfitting; (ii) *Min Sample Split*, i.e., the smallest number of samples to allow the split of an internal tree node; (iii) *Min Sample Leaf*, i.e., the smallest number of samples required for a node to become a leaf of the tree. Instead we set *Number of Estimators*, i.e., the number of trees in the forest, to 200 (due to computational reasons) and *Max Features*, i.e., the number of features considered when searching for the best split, was set to the square root of the embedding size. Such hyper-parameters optimization process was carried out for each RF classifier, corresponding to one of the seven sets of image embeddings computed. Table III

shows the results of the optimization and the final values of the parameters.

Concerning XGBoost, instead, we performed an hyper-parameters optimization focused on boosting rounds and the maximum depth of trees. Our analysis, showed that the best settings for all the classifiers resulted to be using maximum depth equal to 3 and 50 boosting rounds.

D. Stacking

As described in the SectionIV-E, in order to combine the output of the several classifiers trained, we compared three different ensembling strategies, including stacking (or stacked generalization). To apply stacking, it is required to train a sort of meta-classifier that receives as input the output of all the classifiers to combine and computes as output the final combined prediction. In principle, this meta-classifier can be trained used any machine learning method and should possibly not be trained using the same data used to train the classifiers combined together in order to avoid possible bias. In this paper, we used a RF classifier to apply stacking and trained it using the samples in the validation set. The RF parameters have been empirically set as follows: *Max Tree Depth* was set to 30, *Number of estimators* was set to 1400, *Maximum Tree Depth* was set to 30, *Minimum Sample Split* was set to 5, and *Minimum Samples per Leaf* was set to 1.

E. Performance Evaluation

To assess the performances of our classifiers we computed the *Area Under the Receiving Operating Characteristic* (AUROC), i.e., the area under a curve (*Receiving Operating Characteristic*) that is obtained plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) of the classifier, also respectively known as *sensitivity* and *specificity*. To compute TPR and FPR, the probability predicted by the network needs to be converted into a binary decision, using a threshold between 0 and 1, that affects the trade-off between the two metrics. As a reference, an AUROC value of 0.5 means no discriminative power, while – in medical field – a value between 0.7 and 0.8 is considered acceptable, a value between 0.8 and 0.9 is considered excellent, and values larger than 0.9 are considered outstanding [25]. Once we set a specific threshold value to use the classifier in practice, it is also possible to plot the confusion matrix of each classifier on the test.

Model	Atelectasis	Cardiomegaly	Consolidation	Edema	P. Effusion	Mean
DenseNet121	**0.854**	**0.800**	0.891	0.920	0.917	**0.876**
DenseNet169	0.850	0.795	0.882	**0.936**	0.915	**0.876**
DenseNet201	0.834	0.791	0.881	0.917	0.925	0.870
InceptionResNetV2	0.816	0.784	0.897	0.925	0.919	0.869
Xception	0.841	0.770	0.880	0.909	0.924	0.865
VGG16	0.843	0.772	0.898	0.932	0.919	0.873
VGG19	0.843	0.769	**0.900**	0.927	**0.933**	0.874

TABLE IV: Performances of the CNNs trained. The name of different CNN architecture is reported in the `Model` column. The performance is computed as the AUROC achieved on test set. The `Mean` column shows the average performance on all the five findings. We reported in bold the best performance for each finding and overall.

Method	Atelectasis	Cardiomegaly	Consolidation	Edema	P. Effusion	Mean
Best CNN	0.854	0.800	0.900	**0.936**	0.933	0.885
Simple Average	0.854	**0.811**	0.908	**0.936**	0.933	0.888
Entropy Weighted Avg.	**0.856**	**0.811**	**0.912**	**0.936**	0.930	**0.889**
Stacking	0.842	0.797	0.871	0.921	**0.937**	0.873

TABLE V: Comparison of the performances of the best CNN classifier for each finding (reported as `Best CNN`) and the three ensembling strategies considered. The performance is computed as the AUROC achieved on test set. We reported in bold the best performance for each finding and overall.

Model	Atelectasis	Cardiomegaly	Consolidation	Edema	P. Effusion	Mean
RF+DenseNet121	0.851	*0.818*	0.885	0.915	***0.945***	*0.883*
RF+DenseNet169	*0.855*	*0.814*	*0.893*	**0.922**	*0.933*	***0.884***
RF+DenseNet201	*0.863*	*0.814*	0.878	**0.922**	*0.936*	*0.882*
RF+InceptionResNetV2	0.830	0.779	*0.898*	0.918	*0.933*	*0.872*
RF+Xception	0.831	*0.810*	*0.907*	*0.913*	*0.932*	*0.879*
RF+VGG16	*0.858*	***0.822***	*0.913*	0.886	0.917	*0.879*
RF+VGG19	***0.873***	0.798	0.895	0.892	0.917	*0.875*

TABLE VI: Performances of the RF classifiers. In the `Model` column we reported the name of the CNN used to generated the image embeddings the RF classifier was trained from. The performance is computed as the AUROC achieved on test set. We reported in bold the best performance for each column and in italic the performances that are better than the corresponding one achieved by the CNN.

Method	Atelectasis	Cardiomegaly	Consolidation	Edema	P. Effusion	Mean
Best RF	0.855	0.814	0.893	0.922	0.933	0.884
Simple Average	0.859	**0.828**	0.918	0.921	**0.940**	0.893
Entropy Weighted Avg.	**0.872**	0.826	0.918	0.916	0.936	**0.897**
Stacking	0.840	0.761	0.883	0.908	0.937	0.866

TABLE VII: Comparison of the performances of the best RF classifier for each finding (reported as `Best RF`) and the three ensembling strategies considered. The performance is computed as the AUROC achieved on test set. We reported in bold the best performance for each finding and overall.

VI. RESULTS

In this section, we present and discuss of our experimental results. First, we show the results obtained using the classifiers based on CNNs, then the results obtained by classifiers based on trees (RF and XGBoost), and finally the performances achieved by combining all the best classifiers together.

A. Results of CNNs

Table IV shows the results achieved by each CNN trained for each one of the main five findings considered: Atelectasis, Cardiomegaly, Consolidation, Edema, and Pleural Effusion. The results show that all the CNNs achieve a similar average AUROC over the five findings and they also have very similar performances on each single finding: all the networks achieves outstanding performances on identifying Edema and Pleural Effusion, while they struggle at detecting Cardiomegaly. On the other hand, the results show that, as expected, there is no a single CNN that outperforms the others consistently for all the five findings: as an example, VGG19 achieves the best performance on Consolidation and Pleural Effusion but also the worst performance on Cardiomegaly. For this reason we applied three different ensembling strategies, described in Section IV-E, to combine all the seven CNNs trained.

Table V shows the results of such different ensembling strategies on each of the five findings along with the performance of the best CNNs for that specific finding. The results show that, except for the stacking approach, the ensembling strategies based on averaging allow to achieve overall better performances, exploiting the differences among the CNNs. In particular the strategy based on entropy-weighted average resulted to be the best achieving a slightly better performance

overall and for all findings except for Pleural Effusion. Interestingly, Pleural Effusion is the only target that benefit of a stacking approach, perhaps suggesting that it requires a more sophisticated ensembling strategy.

B. Results of RF

The second set of experiments we performed consisted of training RF classifiers using the image embeddings extracted by the previously trained CNNs. Table VI shows the performances achieved by each RF classifiers trained (in Table VI we reported in `Model` column the name of the CNN used to generate the image embeddings). The results show that the RF classifiers achieve in general a better performance than the CNN used to generate the image embeddings (reported in italic in Table VI) and this is always the case if we consider the mean performance. This confirms, as expected, that embeddings do actually encode all the relevant information to discriminate the findings and, more interestingly, that Random Forests are well suited to replace the last fully connected layer used in the network to perform classification. In addition, as for the classifiers based on CNNs, also in this case there is not a single classifier that outperforms consistently all the others, suggesting that there is the possibility of exploiting ensembling strategies to improve the performances. Indeed, Table VII compares the performances of the three ensembling strategies applied to RF classifiers along with the best performance achieved by a single RF classifier – which is not the same one for each finding. The results show that the ensembling strategies always outperform the single best RF classifiers and the entropy-weighted average achieved the best mean performance overall but simple average performs better in most of

Model	Atelectasis	Cardiomegaly	Consolidation	Edema	P. Effusion	Mean
XGB+DenseNet121	0.803	*0.837*	0.837	***0.944***	*0.935*	0.871
XGB+DenseNet169	0.812	*0.829*	0.866	0.916	*0.923*	0.869
XGB+DenseNet201	**0.824**	***0.867***	0.876	*0.920*	***0.938***	***0.885***
XGB+InceptionResNetV2	*0.820*	0.792	***0.911***	0.911	*0.922*	*0.871*
XGB+Xception	0.803	*0.804*	*0.901*	0.899	0.923	0.866
XGB+VGG16	0.800	*0.840*	0.849	0.922	*0.927*	0.868
XGB+VGG19	0.811	*0.819*	0.832	0.921	0.922	0.861

TABLE VIII: Performances of the XGBoost classifiers. In the `Model` column we reported the name of the CNN used to generated the image embeddings the XGBoost classifier was trained from. The performance is computed as the AUROC achieved on test set. We reported in bold the best performance for each column and in italic the performances that are better than the corresponding one achieved by the CNN.

Method	Atelectasis	Cardiomegaly	Consolidation	Edema	P. Effusion	Mean
Best XGBoost	0.824	**0.867**	**0.911**	**0.944**	0.938	0.885
Simple Average	0.829	0.863	0.902	0.933	0.939	0.893
Entropy Weighted Avg.	**0.839**	0.864	0.899	0.936	**0.940**	**0.896**

TABLE IX: Comparison of the performances of the best XG-Boost classifier for each finding (reported as `Best XGB`) and the two ensembling strategies considered. The performance is computed as the AUROC achieved on test set. We reported in bold the best performance for each finding and overall.

the findings. Despite the differences are very small, this can be easily explained by noting that entropy-weighted average outperforms simple average on identifying Atelectasis, where there is a single RF classifier much better than all the others (see Table VI): in this case, the entropy-weighted average can better exploit the most confident classifier weighting its prediction more than the others. Instead, in this case the stacking approach does not provide any improvement with respect to the most simple ensembling strategies.

C. Results of XGBoost

A set of experiments, very similar to the previous ones, have been performed for XGBoost. Table VIII shows the performance of the XGBoost classifiers trained with the image embeddings generated by CNNs. The results shows that, except for the `XGB+DenseNet201`, the mean performances achieved with XGBoost are slightly worse or very close to the ones achieved by the corresponding CNNs. The only notable exception are the performances achieved on Cardiomegaly, where the XGBoost classifiers outperforms both CNNs and the RF classifiers. This suggest that for that specific findings, the boosting mechanism has a larger impact on the performances. Similarly to what previously done, we applied ensembling strategies also to XGBoost classifiers. Table IX shows the performances achieved with the different ensembling strategies along with the one of the best XGBoost classifier – the performances of stacking strategy were worse than the one of simple and entropy-weighted averages and have not been reported in the Table. The results show that, despite ensembling strategies are outperformed on some of the findings by the best single XGBoost classifier, overall they achieve a better mean performance than single XGBoost classifiers. Also, the

Method	Atelectasis	Cardiomegaly	Consolidation	Edema	P. Effusion	Mean
DenseNet121	0.854	0.800	0.891	0.920	0.917	0.876
RF+DenseNet169	0.855	0.814	0.893	0.922	0.933	0.884
XGB+DenseNet201	0.824	**0.867**	0.876	0.920	0.938	0.885
CNN Ensemble	0.855	0.811	0.912	0.936	0.930	0.889
RF Ensemble	**0.872**	0.826	**0.918**	0.916	0.936	0.897
XGB Ensemble	0.839	0.864	0.899	**0.936**	**0.940**	0.896
Final Ensemble	0.860	0.860	0.917	0.934	0.939	**0.902**

TABLE X: Summary of the performances achieved by the best classifiers and ensembles developed in this work, along with the performances of the final ensemble. The performance is computed as the AUROC achieved on test set. We reported in bold the best performance for each finding and overall.

results show that entropy-weighted average is slightly better than simple average to combine classifiers, consistently with what we previously found.

D. Final Results

Finally, we combined together the three ensembles of classifiers presented so far, by applying the entropy-weighted average approach that resulted to be the most reliable one. Table X shows the performances of this final ensemble (dubbed `Final Ensemble` in the Table) along with the performances of the best classifiers trained for each method (CNN, Random Forest and XGBoost) and with the performances of the three ensembles previously presented. As expected, the results show that such final ensemble allows to combine the benefit of the different approaches and achieves the best overall performance (AUROC value of 0.902) with respect to the previously discussed solution.

Finally, we wanted also to provide an insight of the classification results that can be achieved in practice when using the final ensemble just presented. To this purpose we set, for each of the five findings, a threshold to discriminate among positive and negative when labeling unseen data. In particular, we set as a threshold the average model output on the validation set. In addition, to avoid mistakes on sample too close to the discriminant threshold, we labeled as *uncertain* the output within a specific range. Figure 2 shows the resulting confusion matrices for each finding computed on the test set. This provides a more immediate understanding of the final performances with respect to the AUROC values previously discussed. In particular we can notice that the number of uncertain samples are approximately among the 5% and the 15% of the total, which seems to be a reasonable amount of samples that would require a scrutiny instead of being labeled automatically. The results also shows that, in general, more false positives than false negative are generated by the model. Despite this was not achieved intentionally, it seems a desirable outcome for a diagnostic model – the only exception being the results on Cardiomegaly, suggesting that perhaps a less conservative threshold might be chosen.

VII. CONCLUSIONS

Chest X-ray (CXR) is a standard diagnostic tool, widely used in the clinical practice. Thus, reliable methods to au-
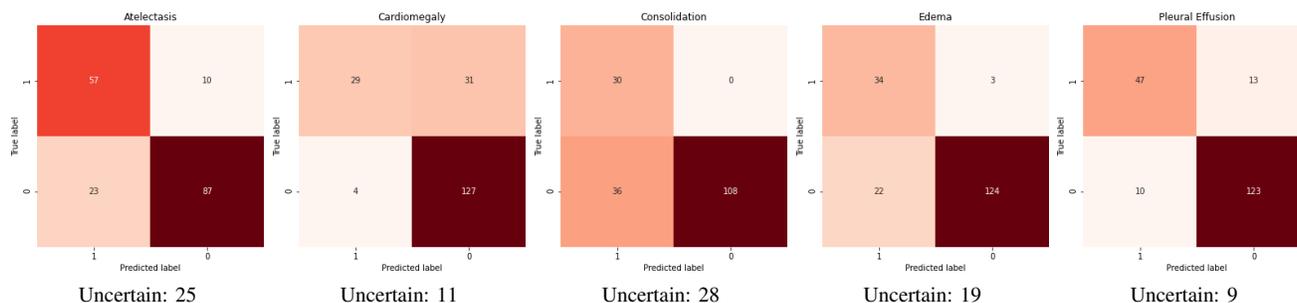
Fig. 2: Confusion matrices achieved classifying the 209 samples in the test set with the final ensemble model.

tomate CXR interpretation could benefit the work-flow of doctors. Machine learning is a promising technology to solve this task, as proved by the recent *CheXpert* competition [16] that made available a large dataset which includes more than 200k CXR labeled images. The goal of this paper is twofold: (i) investigate whether the embeddings of the CXR images extracted from CNNs might be used to train novel classifiers from scratch; (ii) study the benefit of model ensembling and comparing different ensembling strategies. To this purposes, we trained on the Chexpert dataset several CNNs: DenseNet121, DenseNet169, DenseNet201, Inception-ResNetV2, Xception, VGG16, and VGG19. Then, we used them to extract embeddings of the images in the dataset and trained from them two sets of classifiers with Random Forest and with eXtreme Gradient Boosting. Finally, we applied and compared three different ensembling strategies to combine all the models trained. Our results, although preliminary, are promising: the image embeddings do retain the enough information to train effective classifiers based on trees, achieving a final performance that is often even better than the one achieved by the CNN model used to extract the embeddings in the first place. Also, model ensembling resulted quite useful to combine classifiers, especially when none of the classifier is better than the others on all the labels. More specifically, our results showed that entropy-weighted averaging of the models predictions allow to achieve an overall better performance, by weighting more the most confident classifiers for each label.

However, further studies are needed to confirm our findings. In particular, in future works we plan to exploit the embedding models to train classifiers on other (public and private) CXR datasets to solve similar classification tasks, in order to compare the performances with the ones of classifiers trained from scratch. In addition, we will also investigate the application of convolutional autoencoder to extract more general and unbiased image embeddings, that could be possibly be used for many different kinds of tasks (e.g., prognosis prediction, target localization, etc.).

REFERENCES

[1] Y. LeCun, B. Boser, J. S. Denker *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, Dec 1989.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014.

[3] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," 2016.

[4] J. Deng, W. Dong, R. Socher *et al.*, "Imagenet: A large-scale hierarchical image database," in *IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[5] A. E. W. Johnson, T. J. Pollard, N. R. Greenbaum *et al.*, "Mimic-cxr-jpg, a large publicly available database of labeled chest radiographs," 2019.

[6] J. Irvin, P. Rajpurkar, M. Ko *et al.*, "Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison," 2019.

[7] X. Wang, Y. Peng, L. Lu *et al.*, "Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2017.369

[8] P. Rajpurkar, J. Irvin, K. Zhu *et al.*, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," 2017.

[9] P. Rajpurkar, J. Irvin, R. L. Ball *et al.*, "Deep learning for chest radiograph diagnosis: A retrospective comparison of the chexnext algorithm to practicing radiologists," *PLoS medicine*, vol. 15, no. 11, p. e1002686, 2018.

[10] P. Kumar, M. Grewal, and M. M. Srivastava, "Boosted cascaded convnets for multilabel classification of thoracic diseases in chest radiographs," in *Image Analysis and Recognition*, A. Campilho, F. Karray, and B. ter Haar Romeny, Eds. Cham: Springer International Publishing, 2018, pp. 546–552.

[11] Z. Lu, I. Whalen, Y. Dhebar *et al.*, "Multi-objective evolutionary design of deep convolutional neural networks for image classification," *IEEE Transactions on Evolutionary Computation*, pp. 1–1, 2020.

[12] W. Ye, J. Yao, H. Xue, and Y. Li, "Weakly supervised lesion localization with probabilistic-cam pooling," 2020.

[13] R. R. Selvaraju, M. Cogswell, A. Das *et al.*, "Grad-cam: Visual explanations from deep networks via gradient-based localization," *International Journal of Computer Vision*, vol. 128, no. 2, p. 336–359, Oct 2019. [Online]. Available: http://dx.doi.org/10.1007/s11263-019-01228-7

[14] J. Rubin, D. Sanghavi, C. Zhao *et al.*, "Large scale automated reading of frontal and lateral chest x-rays using dual convolutional neural networks," 2018.

[15] H. H. Pham, T. T. Le, D. Q. Tran *et al.*, "Interpreting chest x-rays via cnns that exploit disease dependencies and uncertainty labels," 2019.

[16] S. M. Group, "Chexpert competition," https://stanfordmlgroup.github.io/competitions/chexpert/.

[17] C. Szegedy, V. Vanhoucke, S. Ioffe *et al.*, "Rethinking the inception architecture for computer vision," 2015.

[18] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," 2016.

[19] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," 2016.

[20] M. Lin, Q. Chen, and S. Yan, "Network in network," 2013.

[21] Tin Kam Ho, "Random decision forests," in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 1, Aug 1995, pp. 278–282 vol.1.

[22] T. Chen and C. Guestrin, "Xgboost," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery*

and Data Mining - KDD '16, 2016. [Online]. Available: http://dx.doi.org/10.1145/2939672.2939785

[23] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608005800231

[24] T. B. Brown, B. Mann, N. Ryder *et al.*, "Language models are few-shot learners," 2020.

[25] J. N. Mandrekar, "Receiver operating characteristic curve in diagnostic test assessment," *Journal of Thoracic Oncology*, vol. 5, no. 9, pp. 1315 – 1316, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1556086415306043