Automatic Labeling of Chest Radiography Images

Fateme Sadat Haghpanah
Biomedical Engineering
Columbia University
New York, United States
fh2382@columbia.edu

***Abstract*—Chest radiography is the most common imaging examination globally, critical for screening, diagnosis and managing many life-threatening diseases. Recognizing different diseases is not easy, even for experts and computer-aided recognition system can be much beneficial. In this project, a deep neural network architecture proposed to automatically identify some of the most important chest diseases by looking at the X-ray image. The performance of the designed model compared with previous works and it shows comparable performance.**

***Index Terms*—Chest X-ray, Deep Learning, Convolutional Neural Networks, Automatic Diagnosis**

I. Introduction

Radiography using X-ray is one of the most common imaging techniques. It can be used to image lots of different parts of the body, including hands, arms, foot, and chest. Chest radiography is the most common imaging examination globally, critical for screening, diagnosis, and management of many life-threatening diseases [7]. Recognizing all different chest diseases is not an easy task, even if a human is observing. As a result, developing computer systems in order to assist in reading and interpreting chest images became a hot topic.

Recently, there is considerable interest in developing the different deep neural networks based on the applications. Convolutional Neural Networks (CNNs) that represent mid-level and high-level abstractions obtained from raw data (e.g. images) [9]. Recent results indicate that the generic descriptors extracted from CNNs are extremely effective in object recognition, and are currently the leading technology [13] [10]. In the medical field, such large datasets are usually not available. Initial studies can be found in the medical field that uses deep architecture methods [11] [2].

One important and different factor of previous deep learning based methods on chest X-rays is the data set that each study tries to work on. CheXNet is one of the previous works on X-ray data set of a chest, containing over 100,000 frontal view X-ray images with 14 diseases, using 121-layers CNN [12]. ChestX-rays8 is another dataset containing 108,948 frontal view X-ray images of 32,717 unique patients with the text-mined eight disease image labels [15]. The other study applying CNNS as automated classification on four deidentified HIPAA-compliant datasets [8]. Researchers from Stanford University provided a large dataset of chest radiography images, named CheXpert [7], which this study is based on this too. CheXpert is one of the biggest and most recent chest X-ray datasets. More details on this dataset are provided in section II.

II. Dataset

The CheXpert dataset has more than 224000 images gathered from 65240 patients. Images in the dataset might be from the lateral or frontal view. Two sample images from the dataset can be seen in figure 1.

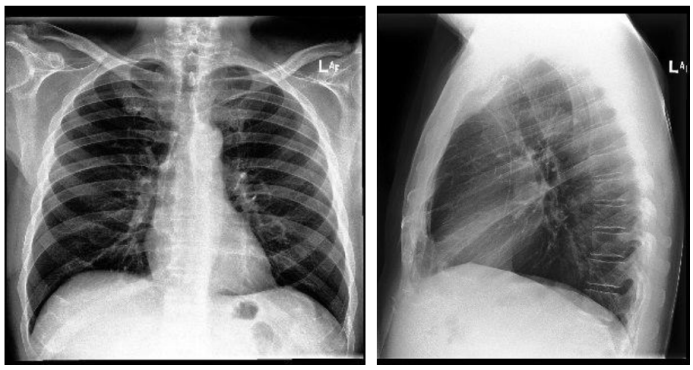

Fig. 1. Sample images from dataset. Left: Frontal, Right: Lateral

This dataset consists of 14 different labels. The labels are generated automatically from radiology reports using natural language processing techniques. Therefore, in the cases that the radiology report used vague phrases or was not sure about some phenomena, we have some uncertain labels in the dataset. The statistics of the dataset can be seen in figure 2.

Pathology	Positive (%)	Uncertain (%)	Negative (%)
No Finding	16627 (8.86)	0 (0.0)	171014 (91.14)
Enlarged Cardiom.	9020 (4.81)	10148 (5.41)	168473 (89.78)
Cardiomegaly	23002 (12.26)	6597 (3.52)	158042 (84.23)
Lung Lesion	6856 (3.65)	1071 (0.57)	179714 (95.78)
Lung Opacity	92669 (49.39)	4341 (2.31)	90631 (48.3)
Edema	48905 (26.06)	11571 (6.17)	127165 (67.77)
Consolidation	12730 (6.78)	23976 (12.78)	150935 (80.44)
Pneumonia	4576 (2.44)	15658 (8.34)	167407 (89.22)
Atelectasis	29333 (15.63)	29377 (15.66)	128931 (68.71)
Pneumothorax	17313 (9.23)	2663 (1.42)	167665 (89.35)
Pleural Effusion	75696 (40.34)	9419 (5.02)	102526 (54.64)
Pleural Other	2441 (1.3)	1771 (0.94)	183429 (97.76)
Fracture	7270 (3.87)	484 (0.26)	179887 (95.87)
Support Devices	105831 (56.4)	898 (0.48)	80912 (43.12)

Fig. 2. Statistics of labels in the dataset.

Moreover, CheXpert providers created a validation and test dataset which includes 200 and 500 studies respectively. These datasets labeled by expert radiologists to evaluate the models' performances more accurately. They did not publish the test set and we only have access to the validation set.

III. Method

The problem at hand is essentially a classification problem, therefore we can use developed machine learning techniques to solve this problem. More specifically, since each image in our problem might have assigned more than one label, our problem is a multi-label classification problem. Recently, CNNs have shown significant performances in the computer vision tasks, specially image classification problems. Therefore, I used a CNN architecture to solve the problem.

In general, I solved the problem with two different approaches. First, I used a pre-trained CNN to extract the features from the images and then classify them using a feed-forward neural network. The second approach is training the feature extractor and the classifier end to end. This approach helps the model to tune feature extractor models in a way to generate more convenient and beneficial features for the classifier. On the other hand, this approach obviously requires more training time and resource. Specific details of the tested architectures are provided in section IV.

In contrast to architectures for single class classification problems which use softmax over the final layer, I used a sigmoid function. Doing so, each node in the final layer shows the probability of each possible pathology problem in the image. Based on this approach, the loss function becomes the sigmoid cross entropy. I just used certain labels in the dataset and ignored the uncertain one. It is one of the approaches that the authors in [7] also used. Therefore, the loss of each sample becomes the sum of cross entropy losses over certain labels. Formally, if X would be the input image, y is the label set for the image, y' is the output of the model, and u shows the uncertain labels, we would have the loss function for one image as equation 1 where I is the identity function.

$$L(X,y) = \sum_i I(y[i] \neq u)[y_i log(p(Y_i = 1|X)) + (1-y_i) log(p(Y_i = 0|X))] \quad (1)$$

IV. Experiments

As said in the methods section, I used two different approaches to solve the problem, freezing the feature extractor or training it with the end to end model and the classifier. The general model architecture can be seen in figure 3.

Since we do not have access to the test dataset, I used the given validation set as my test dataset and split my train set to train and validation sets. In all of the experiments, I used Adam as my optimizer. I used an NVIDIA TITAN Xp GPU to train my models.

A. Freezing feature extractor

In the first set of experiments, I selected the five most frequent labels in my dataset and I only used them to train and test my model. I just used the images that have a certain label in at least one of these categories. The selected categories are Atelectasis, Cardiomegaly, Consolidation, Edema, and Pleural Effusion.

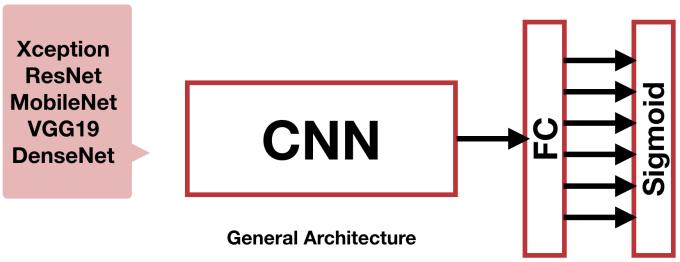

Fig. 3. General models architecture

Then, I extracted the features of images using a pretrained CNN over Imagenet [3] dataset. I used a set of different network architectures for the feature extractor including Xception [1], DenseNet121 [6], ResNet50 [4], VGG19 [14], and MobileNet [5]. Except for MobileNet where I scaled images to have size of 224×224, all other networks used images of size 220×220.

After extracting the features, I used a feed-forward neural network to classify images using the extracted features. Also here, I tested different network setups including a one-layer classifier, a three-layer classifier, and a three-layer classifier with dropout. Performance of each model over the validation set is given in table I. I have also calculated the performance metrics for each label separately, but the results are not included for brevity.

Model	Acc	Prec	Recall	F Score	AUROC
Xception_1	0.68	0.68	0.9	0.77	0.74
Xception_3	0.7	0.74	0.8	0.77	0.76
Xception_3_D	0.7	0.73	0.81	0.77	0.76
VGG19_3_D	0.69	0.74	0.78	0.75	0.74
ResNet50_3_D	0.7	0.74	0.79	0.77	0.76
DenseNet121_3_D	0.7	0.73	0.82	0.77	0.76
MobileNet_1	0.7	0.73	0.81	0.77	0.76

TABLE I
PERFORMANCE OF DIFFERENT ARCHITECTURES. "ACC" STANDS FOR ACCURACY AND "PREC" STANDS FOR PRECISION. THE FIRST PART OF THE MODEL NAMES SHOWING THE FEATURE EXTRACTOR NETWORK, THE SECOND PART SHOWS THE NUMBER OF LAYERS USED IN THE CLASSIFIER AND CONTAINING "D" SHOWS DROPOUT USED IN CLASSIFIER. THE RATE OF DROPOUT SET TO BE 0.5.

Comparing the results to the ones provided in [7] shows that the performance of our model is much worse than their proposed architecture. Therefore, I concluded that it might be necessary to fine-tune and train the feature extractor part of the network.

B. End to end training

Training all weights of a network in an end to end fashion requires a lot more training time and resources, specially when the network is big and has lots of parameters. Therefore, at first, I started the end to end training using the smallest

network, MobileNet. I just changed the final layer of the MobileNet to have the same dimension as a number of the selected classes. I also used the weights trained over Imagenet dataset to initialize the network. The performance metrics can be seen in table II. The results are better than the previous method but are not near to the [7] paper, specially over our test set (their validation set). So, I decided to follow their model and use DenseNet121 as my base architecture.

Data	Accuracy	Precision	Recall	F Score	AUROC
Validation	0.74	0.77	0.83	0.8	0.8
Test	0.66	0.44	0.72	0.55	0.77

TABLE II
PERFORMANCE OF FINETUNED MOBILENET NETWORK.

I did a couple of different experiments over DenseNet121. First, I used the same setup and it did not work well. Then, I tried to train it from scratch (not using Imagenet weights as initialization) and the model overfits a lot. Then, I changed the input image sizes to 320×320 and also trained my model overall categories in the training set, rather than just the 5 selected labels. I just excluded the "No Finding" category since it is essentially the case where all of the other labels is not present, so it is dependent on them. Finally, this setup gives me comparable results to the CheXpert paper. In CheXpert paper, they only provided AUROC on their validation set over 5 categories. Their results and our model's corresponding performance provided in table III. As can be seen, except for the Atelectasis class which needs further investigation, our result is comparable to them, even our result for Cardiomegaly is superior to them. It is worth mentioning that the results reported in the CheXpert paper are obtained by ensemble of 30 different generated model checkpoints, but we just used one model for our prediction. This works requires much more training time and resource. Also, in the case of having multiple images for a sample, they used the maximum probability of each finding in each sample, but our model uses one image and does not combine from multiple images. These differences might be the reason for their superior performance.

Model	Ate	Car	Con	Edema	PE
CheXpert	0.818	0.828	0.938	0.934	0.928
Ours	0.49	0.86	0.89	0.86	0.89

TABLE III
COMPARISION BETWEEN AUROC SCORES OF OUR MODEL AND CHEXPERT PAPER MODEL. ABBREVIATIONS: ATE=ATELECTASIS, CAR=CARDIOMEGALY, CON=CONSOLIDATION, PE=PLEURAL EFFUSION

The performance metrics of this model is provided in table IV. You can see that this model despite having similar AUROC scores to the CheXpert model and having better ones than the finetuned MobileNet, has lower overall performance metrics on the test set in comparison to the finetuned MobileNet model.

The loss curve can be seen in figure 4. In my previous experiments, I have increased the number of epochs which resulted in overfitting, therefore I reduced it to three. I used batch size equal to 16 and 0.0001 as my learning rate. The default beta values in Tensorflow for Adam optimizer used for training.

Data	Accuracy	Precision	Recall	F Score	AUROC
Validation	0.81	0.84	0.84	0.84	0.89
Test	0.58	0.29	0.65	0.4	0.71

TABLE IV
PERFORMANCE OF FINETUNED DENSENET121 NETWORK.

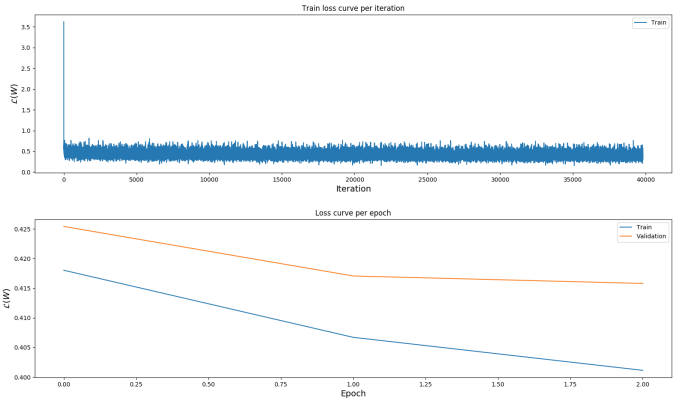

Fig. 4. Top: Training loss curve over each iteration. Bottom: Training and Validation loss curve over each epoch. The generalization error , gap between train and the validation curve, is too small and the model is not overfitted.

V. CONCLUSION

We tried a variety of different deep neural network architectures in order to solve our problem. Overall, it shows that finetuning all the weights of the model and not freezing the feature extractor can give us superior performance. Also, we could see that using some good initial parameters, like weights trained over Imagnet, can be of great importance. Moreover, I find out that adding the training sample of other classes and using all of the samples improved the performance of the model on previously selected classes.

It seems that analyzing the chest X-ray images automatically can be achieved using CNNs and neural networks, but as we can see in this report, it requires a lot of parameter tuning and experiments. I could continue working on this in order to improve my architecture and its parameters.

ACKNOWLEDGMENT

Really thankful of professors, Andrew Laine, and Paul Sajda and the TA, Arunesh Mittal for adding the course to the schedule of the department for the first time and sharing their knowledge and experience in deep learning with BME students.

Also, a big thanks to my best friend, Mohammad, who always believes in me and make me believe in myself.

REFERENCES

[1] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[2] Dan C. Cireşan, Alessandro Giusti, Luca M. Gambardella, and Jürgen Schmidhuber. Mitosis detection in breast cancer histology images with deep neural networks. In Kensaku Mori, Ichiro Sakuma, Yoshinobu Sato, Christian Barillot, and Nassir Navab, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013*, pages 411–418, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[5] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[6] Forrest Iandola, Matt Moskewicz, Sergey Karayev, Ross Girshick, Trevor Darrell, and Kurt Keutzer. Densenet: Implementing efficient convnet descriptor pyramids. *arXiv preprint arXiv:1404.1869*, 2014.

[7] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, et al. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *Thirty-Third AAAI Conference on Artificial Intelligence*, 2019.

[8] Paras Lakhani and Baskaran Sundaram. Deep learning at chest radiography: automated classification of pulmonary tuberculosis by using convolutional neural networks. *Radiology*, 284(2):574–582, 2017.

[9] Yann LeCun, Koray Kavukcuoglu, and Clément Farabet. Convolutional networks and applications in vision. *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pages 253–256, 2010.

[10] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 1717–1724, Washington, DC, USA, 2014. IEEE Computer Society.

[11] Adhish Prasoon, Kersten Petersen, Christian Igel, François Lauze, Erik Dam, and Mads Nielsen. Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network. In Kensaku Mori, Ichiro Sakuma, Yoshinobu Sato, Christian Barillot, and Nassir Navab, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013*, pages 246–253, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[12] Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, et al. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. *arXiv preprint arXiv:1711.05225*, 2017.

[13] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014.

[14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[15] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2097–2106, 2017.