DeepBreath: An Automated Chest X-Ray Diagnostic Tool

Lisa Ishigame
Department of Mechanical Engineering
Stanford University
lih11@stanford.edu

Andrea Oviedo
Department of Mechanical Engineering
Stanford University
aoviedob@stanford.edu

Abstract

Chest x-rays are one of the most widely used radiological examinations for diagnosing and detecting various thoracic diseases. Developing a tool that can help radiologists achieve more accurate diagnoses could have tremendous benefits for under-resourced hospitals and clinics. In this paper, we present various deep learning models trained to detect for the presence of 14 different observations from both lateral and frontal chest x-ray scans. From the explored architectures, the `DenseNet-169` [1] model performed the best, achieving a test accurracy of 86%.

1 Introduction

Chest radiography, or chest x-ray, is one of the most commonly used medical imaging examinations in the world. It can be used to diagnose multiple conditions such as heart failure or lung cancer, and is often used to monitor the progression of diseases or to track patient outcome after medical treatments [2]. X-rays are analyzed by radiologists who are highly specialized doctors trained in reading and providing a diagnosis from radiology exams. Radiologists then communicate their findings to the patient's physician who decides on the next course of treatments.

Developing a tool that can help classify a variety of different conditions from chest x-rays would provide immense support to understaffed hospitals or help physicians in clinical decision making. This tool could even be deployed in clinics and hospitals in developing countries that do not have trained radiologists on site. More than 150 million chest x-rays are performed in the US every year [3], which opens the door to creating robust algorithms using deep learning networks.

In this project, we used both single and lateral chest x-ray images, and trained three different deep learning models, `DenseNet, ResNet` and `VGG`, to automatically predict the probability or presence of 14 different thoracic diseases previously identified from radiology reports.

2 Related work

In the past, several groups have been able to train very accurate models to solve this particular chest x-ray classification problem. One of the first attempts was led by Wang et al. [4], who trained four different networks `AlexNet, GoogLeNet, VGGNet-16,` and `ResNet-50` to classify eight common chest conditions using pre-trained models. Next, Rajpurkar et al. [5] developed a 121-layer convolutional neural network, named CheXNet, to predict the probability of 14 different thoracic diseases as well as output a heatmap localizing the areas of the input image most indicative of pneumonia. The group reported a higher F1 score than the radiologist average score. This particular implementation used a `DenseNet`, developed by Huang et al. [6], and weights pre-trained

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

on ImageNet. More recently, Irvin et al. [7] also used a `DenseNet-121` model which predicted the probability of 14 thoracic diseases based off of a single-view radiograph (either lateral or frontal, outputting the maximum probability). This paper focused more specifically on different uncertainty approaches, and tested whether classifying uncertain cases as positives, negatives, or allowing for semi-supervised learning, led to better outcomes. A limitation that is common to all three papers is that the models trained only considered one x-ray view to train and make predictions. In our project, we were curious to see how a training dataset containing both frontal and lateral images affects the model's accuracy and performance. Following the footsteps of these papers, we also decided to explore similar models, `DenseNet`, `ResNet` and `VGG`, using similar hyperparameters.

3 Dataset and Features

For this project, we utilized a chest radiograph dataset published by Stanford's Machine Learning group [8], which was collected from Stanford Hospital between October 2002 and July 2017. The dataset consists of 223,648 chest x-rays of both front and lateral views of 65,240 patients labeled with the presence of 14 common observations: No Finding, Enlarged Cardiomediastinum, Cardiomegaly, Lung Lesion, Lung Opacity, Edema, Consolidation, Pneumonia, Atelectasis, Pleural Effusion, Pleural Other, Fracture, and Support Devices. The labels for each condition can be 0 for negative, 1 for positive, -1 for uncertain, or left blank for unmentioned.

To train our models, we first combined all of the available data into a single dataset. In some of our models, we eliminated lateral images from the dataset to only consider frontal images, in others we considered both frontal and lateral images. Following implementation suggested by [7], we considered all uncertain labels (-1) as positive (1), since in application it is better to assume more false positives that risk false negatives. Next, we randomized our dataset and divided it into 80%/10%10% training, validation and testing sets, which resulted in 152,983 training examples, and 19,123 validation and testing examples in the models where we only considered frontal images, and 178,918 training examples, and 22,365 validation and testing examples in the models where we considered both frontal and lateral images. Following implementation outlined in [9], we loaded the data using the `ImageDataGenerator` class and pre-processed the images by re-scaling by a factor of (1/255) and set the target size to be 224x224 as suggested by [5]. Figure 1 below shows an example image from our dataset.

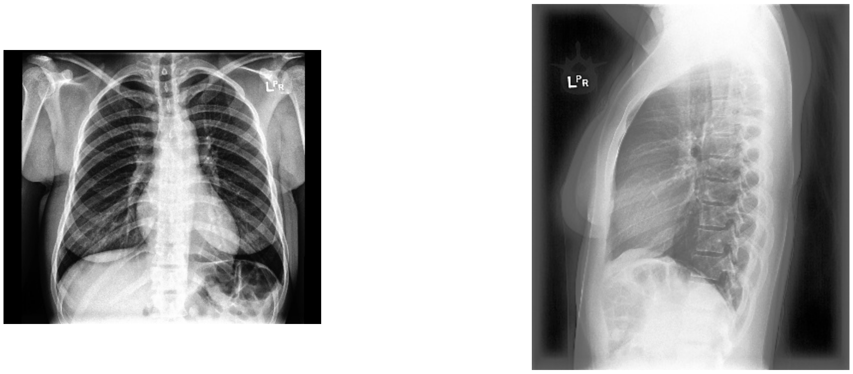


Figure 1: Frontal (left) and lateral (right) chest radiographs from Stanford's CheXpert dataset [8].

4 Methods

4.1 Baseline Model:

Our initial or baseline model consisted of a three-layer ConvNet in Tensorflow implemented using minibatch (t=64) gradient descent, following the implementation learned in class. The ConvNet consists of an initial Conv2D layer, followed by a maxpool layer, then another Conv2D layer, then another maxpool layer, then flattened into a fully connected layer with a sigmoid activation function at the output. Our network utilized Xavier initialization to initialize the weights, W1 and W2, of the two convolutional layers. The cost function minimized is the softmax cross entropy loss between the prediction and the true label. Figure 2 below better captures the architecture of our model.

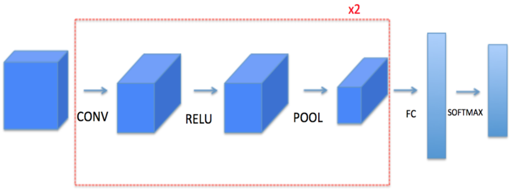


Figure 2: Baseline model architecture

4.2 Deep Models:

A Keras implementation was used to train three deep learning architectures, namely a `DenseNet-169` [6], `VGG-19` [10], and a `ResNet-50` [11]. The VGG architecture developed by [10] is built using only convolutional layers with 3x3 kernel size and max pooling layers of kernels 2x2 with a stride of 2. The advantage of this fixed size allows for a reduced number of trainable variables which leads to a deeper network, faster learning and less over-fitting. However, bigger networks usually experience the problem of vanishing gradients. In order to address this problem, [11] developed a convolutional neural network using Residual blocks, which allow for shortcut connections between layers, resulting in either an identify connection when the two shortcut connections have the same dimensions or a projection connection when the dimensions differ. Along a similar vein, [6] developed a dense convolutional network, or `DenseNet`, where each layer in the network is connected to every over layer in a feed-forward fashion. Traditional neural networks with L layers have L connections, but `DenseNets` allow for $\frac{L(L+1)}{2}$ direct connections between layers. This dense block allows each layer to share its feature-maps as inputs into subsequent layers, and takes the feature-maps from all preceding layer as inputs. The advantages to using this architecture are numerous: it strengthens feature propagation, encourages feature reuse, and reduces the number of trainable parameters. In all three models, we used Adam optimizer with the following parameters: $\beta_1 = 0.9, \beta_2 = 0.999$ and learning rate $1 * 10^{-4}$ as suggested by [7]. For each model trained, we optimized the binary cross entropy loss,

$$L(X, y) = -\Sigma y_o \log p(Y_o = 1|X) + (1 - y_o) \log p(Y_o = 0|X) \tag{1}$$

and trained for 3 epochs. In all models, we modified the last layers of the Keras pre-defined models and included a global spatial average pooling layer, three densely connected layers with a ReLu activation, and finally a logistics layer with 14 outputs using a sigmoid activation to output the probability of the 14 different thoracic diseases. Figure 3 below is a visual schematic of our deep learning models.

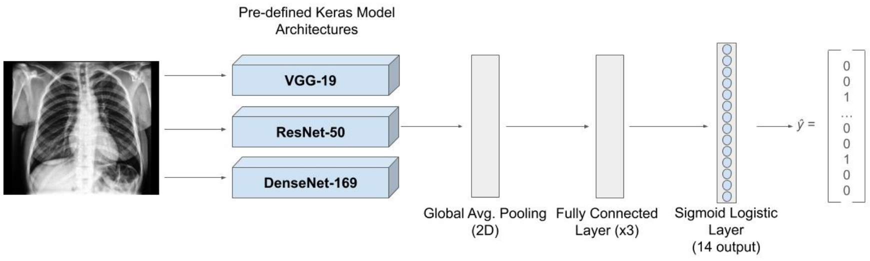


Figure 3: Deep model architectures, highlighting modified final layers for 14-class prediction

5 Results

5.1 Baseline Model:

We ran our baseline model for 100 iterations, first using a learning rate of 0.009, which did not result in good performance since the loss function did not decrease per number of iterations. After modifying the learning rate to $9 * 10^{-6}$ we achieved better results with the loss function decreasing as seen in Figure 4. However, this initial baseline model only considered 1000 training examples and 234 testing examples because we were limited by running the model on our computers. After reviewing these results, we decided to take inspiration from previous implementations [5] [7] and moved on to train deeper networks on AWS using Keras.

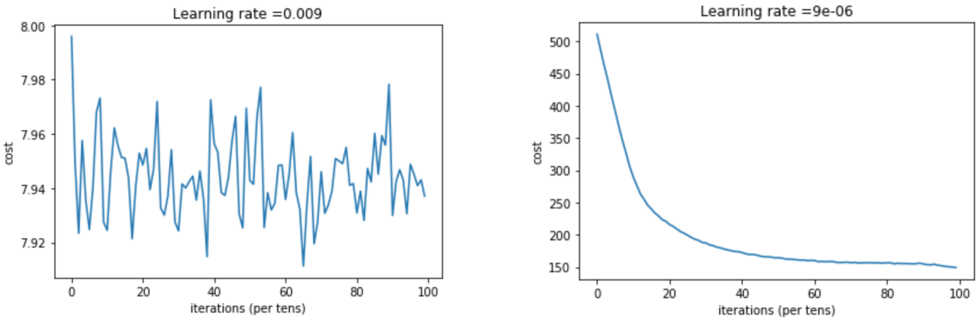


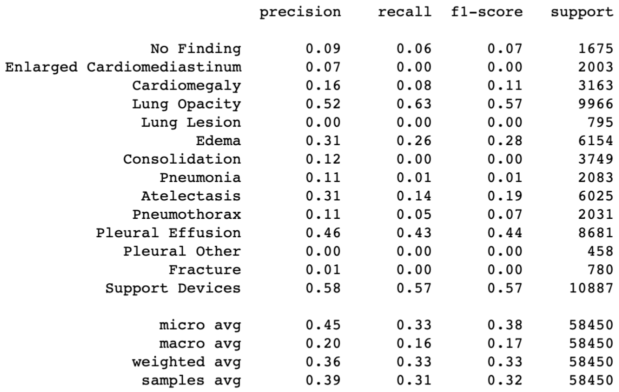
Figure 4: Loss curves for baseline model with lr = 0.009 (left) and lr = 9e-06 (right)

5.2 Deep Models:

The results for our three deep model architectures, `VGG-19`, `ResNet-50`, and `DenseNet-169`, trained on both frontal and lateral images, are summarized below in Table 1. As we expected, the `DenseNet` architecture resulted in the best performance, with an accuracy level of 86% in the frontal-only model, and an 85% accuracy with both frontal and lateral images.

Table 1: Performance assessment of Deep Models

Model	Test Accuracy	Training Ex (#)	Testing Ex (#)
VGG-19 (F)	0.84870	152,983	19,123
ResNet-50 (F)	0.84386	152,983	19,123
DenseNet-169 (F)	0.86099	152,983	19,123
DenseNet-169 (F&L)	0.85976	178,918	22,365



```
                            precision    recall  f1-score   support

                No Finding       0.09      0.06      0.07      1675
Enlarged Cardiomediastinum       0.07      0.00      0.00      2003
              Cardiomegaly       0.16      0.08      0.11      3163
              Lung Opacity       0.52      0.63      0.57      9966
               Lung Lesion       0.00      0.00      0.00       795
                     Edema       0.31      0.26      0.28      6154
             Consolidation       0.12      0.00      0.00      3749
                 Pneumonia       0.11      0.01      0.01      2083
               Atelectasis       0.31      0.14      0.19      6025
              Pneumothorax       0.11      0.05      0.07      2031
          Pleural Effusion       0.46      0.43      0.44      8681
             Pleural Other       0.00      0.00      0.00       458
                  Fracture       0.01      0.00      0.00       780
           Support Devices       0.58      0.57      0.57     10887

                 micro avg       0.45      0.33      0.38     58450
                 macro avg       0.20      0.16      0.17     58450
              weighted avg       0.36      0.33      0.33     58450
               samples avg       0.39      0.31      0.32     58450
```

Figure 5: Precision, Recall, and F1 scores for test set on DenseNet-169 (F)

Figure 5 above summarizes the precision, recall and F1 score for the DenseNet-169 trained on frontal images only. The model performed better at predicting certain conditions than others. We hypothesize that this is due to a data imbalance in the number of positive training examples for certain thoracic diseases. We plotted the loss curve for both the training and test set for this same model, and noticed a decreasing loss function for both datasets. We hypothesize that our model was able to achieve high accuracy on the test set because it was usually predicting 0, or not present, on most cases.

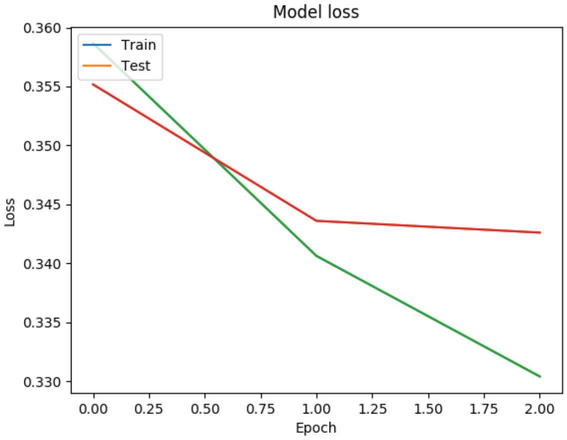


Figure 6: Loss curve for DenseNet-169 (F) model

6 Conclusion and Future Work

Of the model architectures explored in this paper, the DenseNet-169 had the highest performance in terms of accuracy and F1 scores, with a 86% accuracy on the test set, exhibiting a decaying loss function as depicted in Figure 6. The same model trained on both lateral and frontal images did not exhibit a higher performance. Because there were a lot fewer lateral images than frontal images (32,419 lateral images as opposed to 191,229 frontal images), we suspect that the model was unable to learn from this smaller dataset of lateral images. Possible avenues for future work could include only training models using frontal images, or perhaps performing data augmentation on the lateral image dataset to create a more balanced training set. In the future, we would like to address why our models exhibited a lower F1 score for certain categories, for example "Pleural Other" or "Enlarged Cardiomediastinum", but high for other categories, like "Lung Opacity" or "Support Devices." Again, we hypothesize that there were not enough data points in our training set that exhibited some of these 14 thoracic diseases. Therefore, future work could focus on data augmentation to expand the number of positive findings for these low frequency conditions to create a more balanced dataset. Training for more epochs, hyperparamter tuning, and using images with higher resolutions could lead to better performance.

7 Contributions

Both team members, Lisa Ishigame and Andrea Oviedo, contributed equally to the completion of this project. Both members worked on researching previous work, developing the models, writing the report, as well as creating the poster. We would like to thank Dr. Ng and the rest of the CS 230 teaching team, especially our project TA Sarah Ciresi, in helping us learn the power of deep learning and how it can be used to advance healthcare and supplement human performance to achieve better patient outcomes.

References

[1] Project github. https://github.com/lih11/DeepBreath.

[2] Chest x-ray, national heart lung and blood institute. https://www.nhlbi.nih.gov/health-topics/chest-x-ray.

[3] Information for patients: Chest x-rays. http://www.chestx-ray.com/index.php/education/informationforpatients.

[4] 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.

[5] 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.

[6] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[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. *arXiv preprint arXiv:1901.07031*, 2019.

[8] Chexpert: A large dataset of chest x-rays and competition for automated chest x-ray interpretation. https://stanfordmlgroup.github.io/competitions/chexpert/.

[9] Image preprocessing: Imagedatagenerator class. https://keras.io/preprocessing/image/#imagegenerator-class.

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

[11] 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.