YouTube link: https://www.youtube.com/watch?v=lRbiS4NVvaE

Chest X-ray Disease Diagnosis

Qixuan Huang	Zongyi Li	Weixing Tang	Zhiquan Zhang
Georgia Institute of Technology	Georgia Institute of Technology	Georgia Institute of Technology	Georgia Institute of Technology
1st Year Master	2nd Year Master	1st Year Master	1st Year Master

1. Abstract

Xray serves as the main diagnostic tool for variants of diseases but training experienced radiologist is time-consuming. So some machine learning approaches of classifying x-ray images have been proposed and tested. A recently released x-ray dataset - CheXpert - from Stanford Machine Learning lab which is much bigger than previously used dataset which can be served as better training and validating dataset. We use the CheXpert dataset and implemented the baseline model described in Jeremy Irvin's paper. Then we will seek for any potential improvements mentioned in other papers and test on the dataset. Our main goal is to improve the baseline model or find another model which can exceed the baseline scores presented in Jeremy Irvin's paper.

2. Introduction

X-Ray serves as the main diagnostic tool for variants of diseases in the US as well as around the world. However, it takes a long time and lots of practice for training a radiologist and the work of classifying diseases using X-Ray is time-consuming and prone to errors. A solution using machine learning models for x-ray disease diagnosis can be really helpful.

The currently proposed methods are evaluated on some small-to-middle scale problem due to the size of the dataset.[1] However, as the increasing construction of the hospital-scale chest X-ray Database, it is possible to build a truly large-scale and fully-automated high precision medical model with a clinically meaningful application nowadays[2]. Also, as the development of computer vision technique like VQA, the ImageNet pre-trained deep CNN models already perform very well in a large number of object classes and this will be suitable for the project by providing a good baseline for further tuning.

In general, a precise model of Chest X-ray Disease implementation can be realized and will be a good start for image diagnosis domain and expected be extend to other disease fields. Jeremy Irvine[3] introduced Chest X-ray14 dataset and used a labeler, which is set up in three stages: Mention Extraction, Mention Classification, Mention Aggregation. They tested on the different network model, like ResNet152, DenseNet121, Inception-v4, etc. H.Liu in paper[4] proposed a novel approach Segmentation-based Deep Fusion Network using Chest X-ray images, which fused the features from both the entire CXR images and local lung region images. Rajpurkar[5] developed a 121-layer convolutional neural network named CheXNet, which performed better than pre-existing models based on AUROC. X.Wang in paper[1] introduced Chest X-ray8 dataset and finished common thoracic Disease Detection and Localization by introducing a unified DCNN Framework with 8-dimensional label vector for multi-label setup, transition layer, multi-label classification loss layer. Recently in 2019, Baltruschat[6] used ResNet-50 architecture and a network integrating non-image data(patient age, gender, and acquisition type) in the classification process, and found that the X-ray-specific ResNet-38, integrating non-image data has the best prediction results. In Jeremy Irvin's paper, they found that DenseNet121 architecture produced the best results among several CNN architectures.

In this project, we followed the method proposed by Jeremy Irvin's paper and implemented a similar neural network using DenseNet121 as the pre-trained model. As DenseNet121 takes input image in 224*224, we first

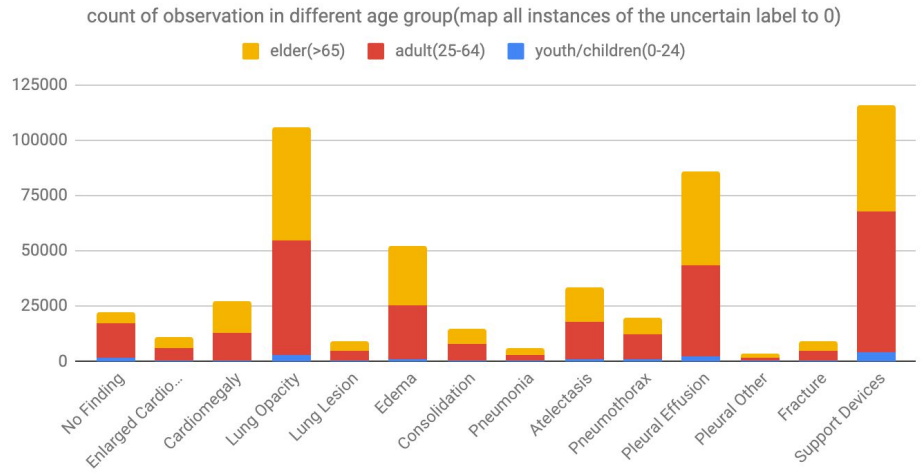
normalize the input images from the dataset to 224*224. The main focus of this project is implementing and comparing different approaches of handling uncertain labels in the dataset, namely u_zeros, u_ones, u_ignore, u_multiclass and u_selftrained, where u_zeros means mapping all uncertain labels to negative, u_ones means mapping all uncertain labels to positive, u_multiclass means treating uncertain labels as an independent class and u_selftrained means using semi-unsupervised learning strategy for filling uncertain labels.

3. Methods

3.1 Dataset Introduction

We used CheXpert dataset to do experiments. The complete dataset consists of 224,316 chest radiographs of 65,240 patients. And it consists of 14 labeled observations as positive, negative, or uncertain in the paper[3]. The authors did data collection and label selection for the original data from Stanford Hospital and the labeling model is introduced in the paper[3]. They developed an automated rule-based labeler to extract observations from the free text radiology reports to be used as structured labels for the images. After three stages of label extraction: mention extraction, mention classification and mention aggregation, each observation was mentioned as confidently present(1), confidently absent(0), uncertainty present(-1), or not mentioned(None).

Instead, We used the small train sample dataset with 178,726 numbers of image for 15,768 patients and the validation sets with 235 images. The size of each image is varied in range (300 x 300 ~ 400 x 400) due to the different resources, frontal/lateral and scale. we downsampled the image to 224 x 224 as our input in training experiments. We use pySpark with Databricks to do data visualization and preprocess the dataset by dropping the null label, grouped by gender(male and female), age (0-24 considered as youth/children, 25-64 as adults and >65 as seniors). Here is a bar chart of the count for each observation in the assigned group with U-Zeroes uncertainty approach. Briefly speaking, this dataset is biased: the number for each observation is quite different and it has limited information for youth/children which indicates the accuracy for that age group might not be as reliable as the other two age groups. Also, the size of data for the male is bigger than the female.



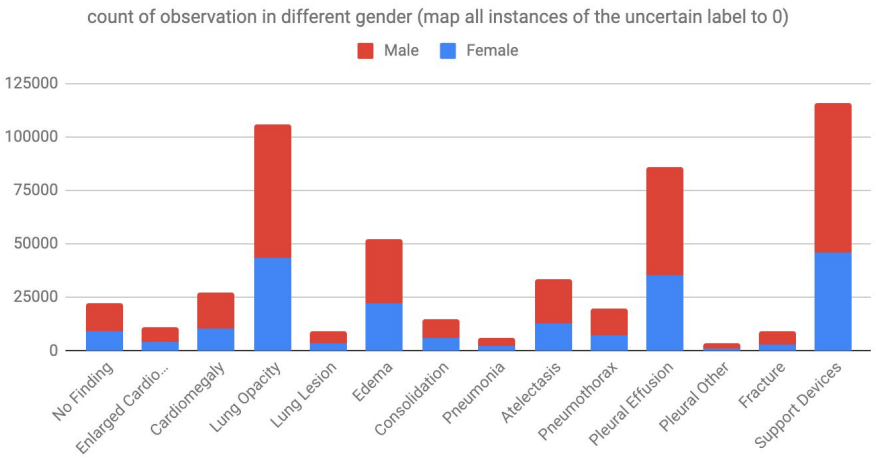


Figure1: Dataset visualization

3.2 Experimental Setup

After the use of pySpark in databricks to process original dataset(train.csv and valid.csv), we have tried to implement the baseline model using two different deep learning tools - Keras and Fastai. They both provide easily-use python packages for faster developing of deep learning models. However, we have further chosen Fastai as it provides strong support for using Pandas as data input and output which yields an advantage for doing data processing and result evaluation.

For shorter training and validating process and for the convenience of developing and testing models, we choose to use the downsampled dataset (each image with size 224*224) with total dataset size of approximate 11GB. The downsize image is obtained by resize_method in Fastai

Our models were trained and tested using a single GeForce GTX1080 GPU on local and a single Tesla P40 on Georgia Tech cluster.

3.3 Uncertainty Approaches

Irvin proposed four approaches for handling uncertain labels in the dataset[3] - ignoring, binary-mapping, self-training and 3-class classification.

In ignoring the process, we simply remove the label of uncertainty present instances in the training process. In order to simplify the process, we added two labels. The first is for previous label 1, and the other is for previous label 0. For binary-mapping, we mapped all instances of uncertainty present to 0 (u-zero model) or all to 1 (u-one model). In self-training, we first trained a model using ignoring to convergence, then use the model to make predictions that re-label each of the uncertainty labels with the probability prediction outputted by the model. In 3-class classification, we treated label -1 as its own class for each of the observations and set up the loss as the mean of the multi-class cross-entropy losses over the observations.

For consistency, we will use "u-zero" and “u-one” for mapping all uncertainty to 0 and 1 accordingly, and "u-multiclass" for 3-class classification. Furthermore, for "u-ignore", we set each label of observation (-1, 0, 1) to two labels, [positive, negative]. Therefore, the uncertainty can be considered as [0, 0] as u-ignore.

3.4 Model and Evaluation Metrics

For the model, we finally choose DenseNet[7], which shows great performance in for this dataset. The structure of DenseNet is like below.

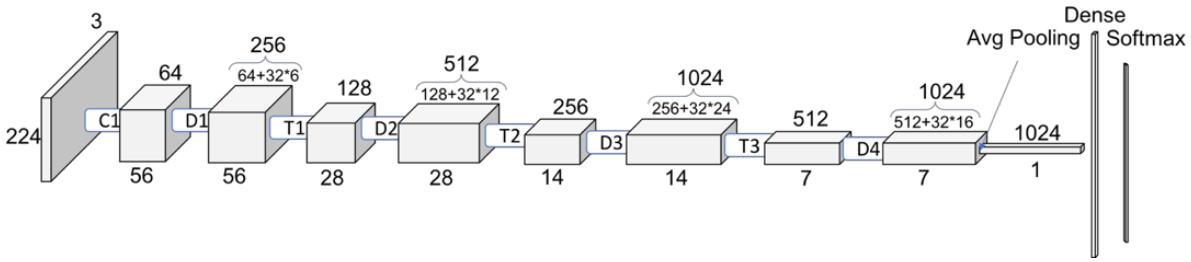


Figure2: Structure of DenseNet

It has many dense blocks that are made up of several convolutional layers. In general, the dense block with more convolutional layers is better but require more training time. It provides a direct connections between any two layers with the same feature-map size. Because of its compact internal representations and reduced feature redundancy, DenseNets will be good feature extractors for various computer vision tasks that build on convolutional features.[8]

Implementation of u-ignore is straightforward. We simply remove the -1's in the dataset as NAN values. For the implementation of u-zero and u-one, we utilized the built-in feature and label arguments from Fastai by setting -1 as the same group of 0 and 1 accordingly. The implementation of u-multiclass is much harder and details are missing in the baseline model introduction. We use the strategy of encoding each feature into three features as feature_u, feature_p and feature_n for uncertainty, positive and negative accordingly. For positive case, we set feature_p as 1, feature_n as 0 and feature_u as 0. Similarly, for the negative case, we set feature_p as 0, feature_n as 1 and feature_u as 0, and for the uncertain case, we set feature_p as 0, feature_n as 0 and feature_u as 1. For NAN label in the dataset, we set 0 for all three encoded features. After that, we use the simplified one hot encoding method to convert the preprocessed data for ML algorithm to reach a better performance of prediction.

As the preparation of training. we first use Fastai[9] built-in callback system with Leslie Smith's learning rate finder to locate a most suitable learning rate before training.

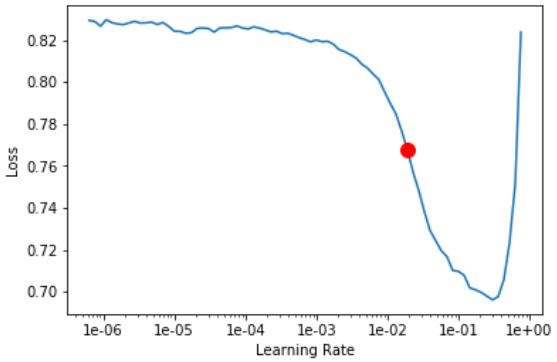


Figure3: Sample Learning rate finder

Then we make a model predicting for each of five cases for each feature of every image. This encoding method can produce results which we can easily validate using the same evaluation metrics as other binary classification approaches.

We use the AUC ROC score (Area Under the Receiver Operating Characteristic Curve) [10] to be consistent with other papers for easy comparison. Also, the AUC ROC score calculation is implemented and can be directly used from sklearn.metrics library.

4. Experimental Results

	Atelectasis	Cardiomegaly	Consolidation	Edema	Pleural Effusion
U-Ignore(Paper)	0.818	0.828	0.938	0.934	0.928
U-Ignore(Our)	0.771	0.795	0.886	0.864	0.902
U-Zeros(Paper)	0.811	0.840	0.932	0.929	0.931
U-Zeros(Our)	0.752	0.826	0.895	0.905	0.915
U-Ones(Paper)	0.858	0.832	0.899	0.941	0.934
U-Ones(Our)	0.808	0.785	0.856	0.925	0.918
U-MultiClass(Paper)	0.821	0.854	0.937	0.928	0.936
U-MultiClass(Our)	**0.848**	0.850	0.914	0.904	0.928
U-SelfTrained(Paper)	0.833	0.831	0.939	0.935	0.932
U-SelfTrained(Our)	0.641	0.670	0.755	0.781	0.762

Table 1: Different uncertainty handling methods on whole dataset (The bolded scores are where our model outperforms the baseline model for the same feature and same uncertainty approach. The blued scores are the highest score achieved in the paper for each feature and the yellowed scores are the highest score our implementation achieved for each feature.)

	Atelectasis	Cardiomegaly	Consolidation	Edema	Pleural Effusion
Female	0.807	0.759	0.855	0.92	0.873
Male	0.823	0.752	0.956	0.886	0.875
Average	0.815	0.756	0.906	0.903	0.874

Table 2: Gender sperated dataset for U-Zeros

	Atelectasis	Cardiomegaly	Consolidation	Edema	Pleural Effusion
Female	0.82	0.708	0.751	0.914	0.893
Male	0.777	0.813	0.943	0.913	0.869
Average	0.799	0.761	0.847	0.914	0.881

Table 3: Gender separate dataset for U-Ones

5. Discussion

For other DenseNet models, they tend to be overfit easily but DenseNet121, on the contrary, performs really good in different diseases. overfit different ways to process the uncertainty labels and found there was no such a way that can be best for all diseases. More experiments can be done to verify the effect of U-ignore on the predictions because the it seems that there are different explanations over the U-ignore and in our experiments, different implement of U-ignore will directly influence the performance of U-selftrained. Our u-ignore model performs relatively worse compared to other models and the u-ignore result given in paper, so that as expected, our u-selftrained failed to achieve an acceptable result.

We also do some experiments over the performance of models based on different gender and find that perhaps gender will affect the results and similarly, the age of the patient will also have some indirect influence. The result difference is not very significant. However, we do can notice some improvement within one gender for certain feature has been improved, such as female-Atelectasis. We can further purpose that training different models for different age segmentations and genders might possibly improve the overall model quality. (We are lack of time and resource for further experiment in this project.)

According to the visualization of the training dataset, it's obvious that the numbers of sample vary greatly from disease to disease and more or less, this would cause some unexpected negative influence on the models' performance known as unbalanced dataset. If possible, to balance the number of different types of disease is very necessary and will benefit the performance of the model.

6. Conclusions

We choose DenseNet121 in our experiments as DenseNet121 is identified as the best pre-trained model in Jeremy Irvin's paper[3]. From our experiment result, we can conclude the same result as Jeremy Irvin as for different types of diseases, each of them has its own best way to process uncertainty label accordingly. For example, U-One is best for Atelectasis, U-MultiClass is best for Cardiomegaly. To deal with different diseases, different preprocess and training tricks should be used regardingly. In general, from our experiment's result, **U-Multiclass** should be chosen for any new classification class which has not been experimented before as best potential handling method.

7. Challenges

7.1 Size of dataset

The large size of the dataset requires longer running time and, hence, results in slower development and debugging process in building the model. A single run requires around thirty minutes on the resource available to us (both locally and on cluster) which prevents us from doing multi-run experiments or trying more assumptions such as how grouping by age would affect the classification results.

7.2 Multi-class AUC ROC

Given pre-trained DenseNet121 in pytorch, we do not need to put too much effort into developing the actual neural network step by step. However, developing the evaluation metrics for each uncertainty handling methods accordingly still remain a big challenge for us. U-zeros and u-ones are straight-forward as they are binary classifications and skit-learn provides AUC ROC evaluation metrics for binary cases. U-multiclass, u-ignore and u-selftrained require method for calculating AUC ROC for multi-classes cases. We applied similar strategy as 'one-hot encoding' where we encode 'positive', 'negative' and 'uncertain' for a single class into two features - 'class_p' and 'class_n' and 'class_u'. For example, if a patient is positive for feature 1, he or she will have 1 for 'feature1_p', 0 for 'feature1_n' and 'feature_u'. Then we can treat every encoded feature as a binary classification case.

8. Contribution

	Draft/Literature Review	Data Pre-proccessing and Visualization	Building Neural Network	Final Report	Final Presentation
Qixuan Huang	4	5	3	4	3
Zongyi Li	4	3	3	4	5
Weixing Tang	4	3	5	4	3
Zhiquan Zhang	4	4	3	5	3

*0: not participated; 5: contributed a lot;

References:

[1] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3462–3471. IEEE, 2017.
[2] Brestel, Chen, et al. "RadBot-CXR: Classification of Four Clinical Finding Categories in Chest X-Ray Using Deep Learning." (2018).
[3] J. Irvin, P. Rajpurkar, M. Ko, Y. Yu, S. Ciurea-Ilcus, C. Chute, H. Marklund, B. Haghgoo, R. Ball, K. Shpanskaya, J. Seekins, D. A. Mong, S. S.Halabi, J. K. Sandberg, R. Jones, D. B. Larson, C. P. Langlotz, B. N.Patel, M. P. Lungren, and A. Y. Ng. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. arXiv preprint arXiv:1901.07031, 2019
[4] H. Liu, L. Wang, Y. Nan, F. Jin, and J. Pu. Sdfn: Segmentation-based deep fusion network for thoracic disease classification in chest x-ray images. arXiv preprint arXiv:1810.12959, 2018.
[5] P. Rajpurkar, J. Irvin, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C. Langlotz, K. Shpanskaya, M. P. Lungren, and A. Y. Ng. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. arXiv preprint arXiv:1711.05225, 2017.
[6] Baltruschat, Ivo M., et al. "Comparison of deep learning approaches for multi-label chest X-ray classification." Scientific Reports 9.1 (2019): 6381.
[7] Huang, Gao, Liu, et al. Densely Connected Convolutional Networks. arXiv.org. 2018. https://arxiv.org/abs/1608.06993 (accessed 15 Apr2019).
[8] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. arXiv preprint arXiv:1608.06993, 2016a.
[9] Open-i: An open access biomedical search engine. https://openi.nlm.nih.gov
[10] Fawcett, Tom. "An introduction to ROC analysis." Pattern recognition letters 27.8 (2006): 861-874.