Exploring Model Architectures and View-Specific Models for Chest Radiograph Diagnoses

Danny Takeuchi, Raymond Thai, Kevin Tran
{dtakeuch, raythai, ktran23}@stanford.edu

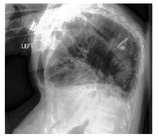
Motivation

Background

- Chest radiography is an imaging technique used in the diagnosis and management of many diseases.
- X-ray radiographs are inputted into a model and the diagnoses for the presence of medical conditions is outputted.
- Automated chest radiograph interpretation can improve clinical decision support and workflow.

Previous Work

- Many previous studies used a dataset called CheXpert, created by Irvin et al., a Stanford machine learning team
- Previous methods used one model to classify three potential X-ray view types (front, back, and lateral) and did not experiment with many model architectures.
- Despite the CheXpert dataset being open-source, previous work is largely closed-source.



Project Summary

Re-implement an open-source version of a Stanford Machine Learning Group's model on Chest Radiograph diagnosis and extend the model by applying view specific models and exploring different model architectures.

Data and Model Implementation

Dataset

- We used the CheXpert dataset of 224,316 chest X-rays of 65,240 patients. The ground-truth labels for each image was an x-ray interpretation from a panel of radiologists.
- Each image includes information about patient id, sex, age, study, X-ray view type, and 14 expert-labeled observations. We omitted age and sex features.
- We split our data for each view type into a training set, validation set, and test set with a ratio of 98:1:1.
- We scaled and cropped data into 225 x 224 images and converted images to RGB tensors.

Model Implementation

- We worked off the open-source github repository of a healthcare analytics company. The repository partially implemented Irvin et al's method by making predictions on individual scans with a Densenet.
- The repository did not incorporate any patient or study information and treated two X-rays from one patient as two unrelated scans.
- We reimplemented Irvin's architecture by making predictions grouped by patient and study instead of only on individual X-rays.
- By evaluating the original Github model centered on individual-scans compared to our patient-centric model, we found that there is a 2.5% increase when evaluating on individual patient instead of individual scan.

References

Bar, Yaniv, et al. "Chest pathology detection using deep learning with non-medical training." 2015 IEEE 12thInternational Symposium on Biomedical Imaging (ISBI). IEEE, 2015.
Huang, Gao, et al. "Densely connected convolutional networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
Irvin, Jeremy, et al. "Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison." Thirty-Third AAAI Conference on Artificial Intelligence. 2019.
Nair, Aditya, et al. "Detection of Diseases on Chest X-ray Using Deep Learning." 2019. Zhang, Quanshi, et al. "Interpreting CNNs via decision trees." arXiv preprint arXiv:1802.00121. 2018.
https://github.com/kshitij0987
https://stanfordmlgroup.github.io/competitions/chexpert

View-Specific Models

Method

- 3 view-types are available (front, back, lateral) so we parsed the original CheXpert dataset to create 3 datasets, each with views only from a certain viewtype.
- We trained and tested 6 view-specific models. A DenseNet and a VGG19 model trained only on data only from 1 of 3 view types.
- The baseline for each section used a single model trained on all view types.

Model	Trained by View	Trained by View with Classifier	Baseline
DenseNet 121	0.7313	0.7268	0.7648
VGG19	0.7150	0.7042	0.7465

View Classifier

- To classify views into one of three types, we tried three approaches: (1) Logistic regression (2) Feedforward Neural Network (3) Convolutional Neural Network.
- Analysis: As expected, our classification results steadily increased as our model complexity grew. More complex models could recognize more complicated features of scans.

Model	Accuracy
Logistic Regression	0.61
Feedforward NN	0.81
Convolutional NN	0.96

Results on Individual Views

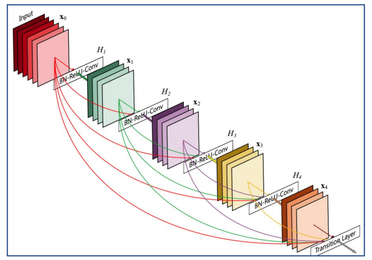
- We used the tested the view-specific models on only scans from their corresponding views and compared them to the baseline

Model	Baseline on PA	Specific on PA	Baseline on AP	Specific on AP	Baseline on Lat	Specific on Lat
DenseNet121	0.7100	0.6921	0.7543	0.7538	0.7043	0.6803
VGG19	0.7018	0.6837	0.7509	0.7489	0.6862	0.6608

Exploring Model Architectures

Baseline: DenseNet121 (using logistic classifier)

- Maximizes information flow through short-cuts between layers while using less VM capacity than a traditional deep CNN.
- First, the input tensor first flows through a convolutional feature layer to capture low-level features of the X-ray such as lines, colors, and boundaries.
- This is followed by a series of four dense blocks with transition layers in between.
- These dense blocks have 6, 12, 24, and 16 layers layers of alternating Batchnorm, ReLu, and 2D Convolutions.
- Each Transition Layer is composed of a 1x1 Convolution and a 2x2 average pooling layer with a stride of 2 in order to abstract out higher level features.



VGG16 and VGG19 (using logistic classifier)

- Well-suited for classifying chest radiographs. Nair et al.'s group drew promising results.
- Pretrained on over a million images from the ImageNet database.
- VGG16 is composed of 16 convolutional layers and VGG19 adds on another 3 fully connected layers to the end.
- Have more trainable parameters which slows down the training process considerably.

SVM (Linear, RBF, Polynomial) using DenseNet121

- Utilized DenseNet121 as a feature extractor for the x-ray images (converted 50176 rgb images to 1024 features).
- Passed these extracted features into kernelized linear SVM, radial basis function SVM, and polynomial SVM.
- Each SVM utilized a cost factor of 1 and a gamma of 1.
- More interpretable because the hidden layers of the DenseNet are used as weights

Decision Tree using DenseNet121

- Also utilized DenseNet121 to extract features.
- Passed extracted features from x-ray images into entropy & information gain decision tree.
- Utilize a max depth of 32, minimum sample leaves of 5 and 100 random states as hyperparameters.
- More interpretable because the hidden layers of the DenseNet are used as weights

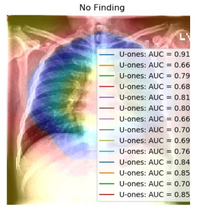
Results on Exploring Model Architectures

DenseNet121	VGG16	VGG19	Linear SVM using DenseNet Features	Poly. SVM using DenseNet Features	RBF SVM using Densenet Features	Decision Tree using Densenet Features
0.7648	0.7465	0.7409	0.7727	0.8479	0.8646	0.9316

Analysis

View Specific Models

- This baseline trained on all views outperformed our view-specific models on a data set with a variety of scans and individually outperforms every view-specific model.
- We hypothesize that the strong performance of the baseline is because one model can effectively learn and classify different view types (ie features learned from frontal scans can help in classification on lateral scans).
- Additionally, it seems that there lacked enough training data for the lateral and PA views. This hypothesis is supported by how 72% of all view data is of view-type PA and the view-specific model for AP was closest to performing at the baseline level.



View Comparison on the 14 Diseases

Densenet121	% Best	% Improve
AP	50	10.27
PA	29	3.75
Lateral	21	1

Exploring Model Architecture

- **VGG19/VGG16:** Using the mean AUROC score as our evaluation metric, DenseNet121 outperformed both VGG19 and VGG16 on all three view types. These gains are likely due to the Densenet's ability to directly leverage earlier layers which encourage feature reuse. This is crucial because disease diagnoses can stem from tiny changes within the images. It also avoids the vanishing gradient problem that deep VGG networks run into.
- **Linear SVM:** The linear SVM using DenseNet extracted features slightly outperforms the baseline DenseNet that uses a logistic classifier.. This improvement in performance is most likely due to their differences in loss functions. Linear SVM utilizes a hinge loss that is less sensitive to outliers present in the data. **Polynomial and RBF SVM:** The polynomial and radial basis function SVM outperformed the linear SVM and the baseline. Because all of these classifiers use the same features from the Densenet, improvement in performance is most likely because a non-linear decision boundary better separates out the data.
- **Decision Tree:** The decision tree was our best performing model overall. As with before, since the same extracted features are being used, the improvement in performance is likely due to the way the decision tree draws decision boundaries. The decision tree cuts the space based on information gain, in which the most expressive features are first used to separate out the data.

Future

SVM or Decision Tree with Other Feature Extraction Methods

Our most promising discovery was the success that we experienced using the DenseNet as a feature extractor for various SVM and decision tree classifiers. However, we only tried to extract features using the DenseNet121 net and may have experienced more success with other feature extraction methods like principal component analysis or other featurized neural networks.

View Specific Classification with More Data:

Our poor results with view-specific models may have been due to a shortage of training data for each view type. This hypothesis is supported by how we performed best on AP. Further research with more data and varying amounts of view-specific/mixed data could more conclusively determine whether view specific models could be helpful.

We would like to acknowledge Jeremy Irvin for providing guidance and potential research directions for this project.