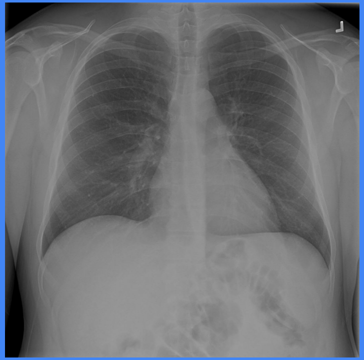
Chest X-ray Diagnosis via Supervised Machine Learning

Stanley Yu, Geoff Korb, Akshay Kulkarni



Introduction

Chest X-ray exams are one of the most common and inexpensive medical imaging options available.

Clinical diagnosis skill for chest X-rays is not available everywhere.

Prior to the NIH Chest X-Ray Dataset it was very difficult, if not impossible, to achieve clinically relevant computer-aided detection and diagnosis with chest X-rays.

Dataset

NIH Chest X-ray Dataset

112,120 images, 30,805 unique patients

The labels, expected to be >90% accurate, were created via text-mining of disease classifications from the associated radiological reports.

Our Dataset

10,000 images, 6,002 unique patients

Randomly sampled equal number of the top 10 most common findings

Approach and Methodology

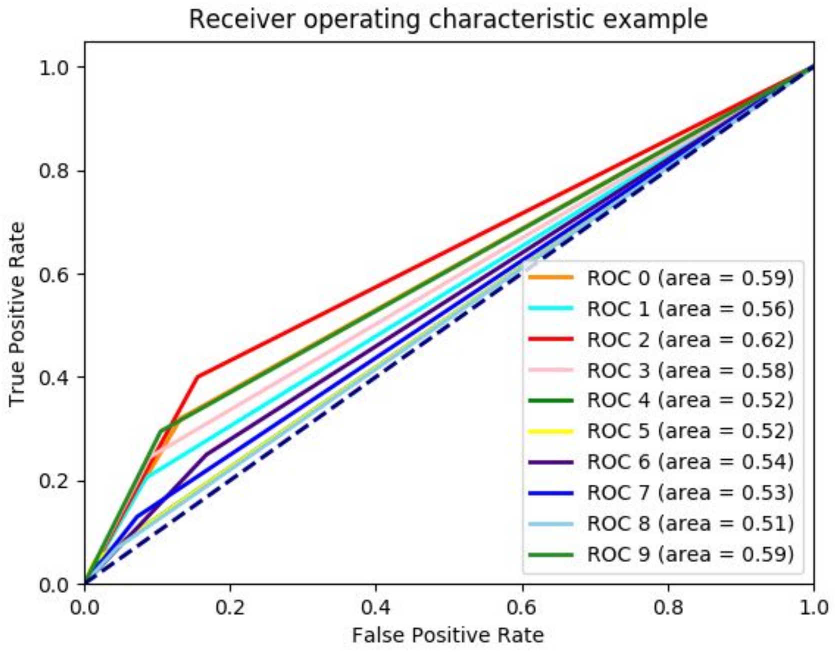
Pre-Processor

- grayscales, resizes, converts image to matrix, flattens, attaches labels
- normalizes that data

Models

- Logistic Regression
- Random Forest
- SVM
- ResNet18, 34 (pre-trained CNN)
- Keras 10 layer (CNN)

Logistic Regression (Multi-classification)



- Logistic Regression serves as a base model.
- Multi-classification of all labels leads to predictably reduced performance.

Logistic Regression (Binary Classification)

- Balancing the dataset benefited binary classification, but not multi-classification.

- The most accurate labels in binary classification are:
  - Consolidation, Accuracy: 0.7375
  - Effusion, Accuracy: 0.735
  - Cardiomegaly, Accuracy: 0.705

Random Forest

Why Random Forests?

- Improved accuracy and reduced model variance.
- Minimal parameter tuning

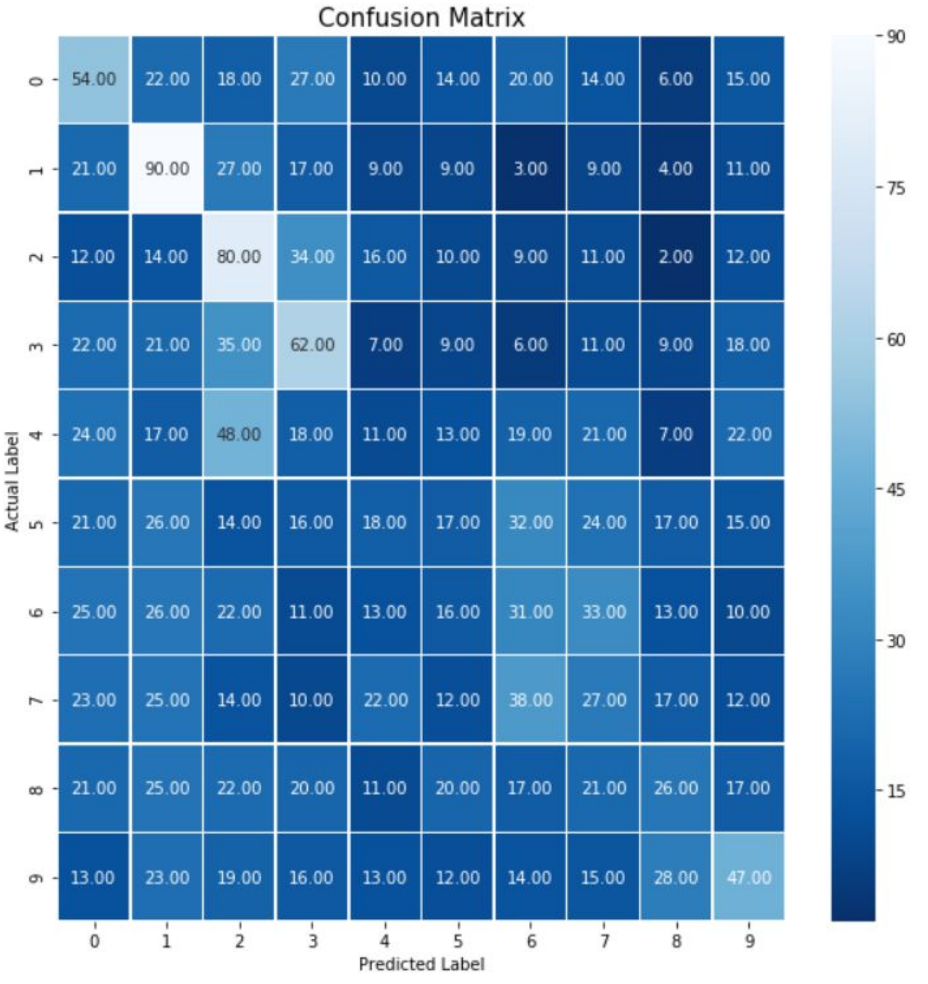
Modest (at best) Returns:

- Accuracy: 22.5%
- Precision: 22.3%
- Recall: 22.3%

Random Forests

Great at predicting:

1. Cardiomegaly [1]
2. Consolidation [2]
3. Maybe Effusion [3]
4. And that’s about it.

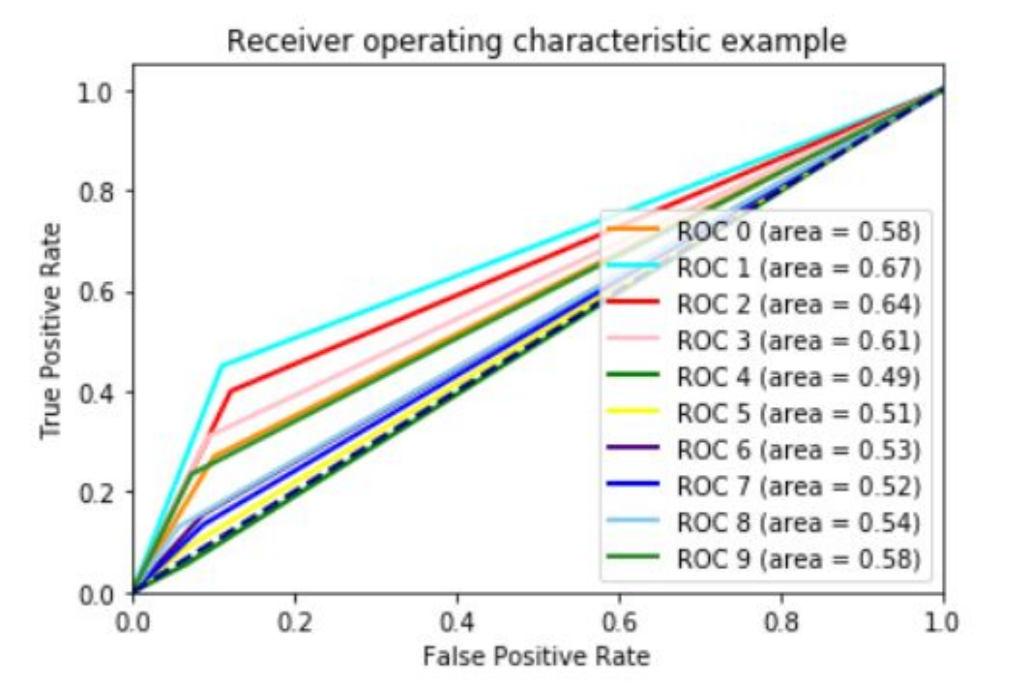


Random Forests

AUC: 0.5681

Possible improvements?

- More estimators?
- Different normalizations?
- Is the model appropriate?



ResNet with Pytorch and fast.ai

We decided to train ResNet18 and 34 layered architectures as the first of our CNN models since they are stable, well-performing architectures. The training was done from scratch with all layers unfrozen and using starting weights trained on the Imagenet dataset.

The choice of architecture was based on the resource and time constraints ResNet34 offers very good performance vs its size (which impacts training time) and allows for bigger batch size and so was chosen as a baseline model for this project.

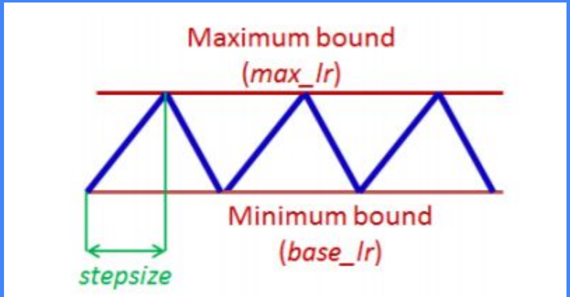
Preliminary Results

Results of multi-classifier on full dataset:

Train_loss	Valid_loss	valid_acc
1.416949	1.447258	0.503581

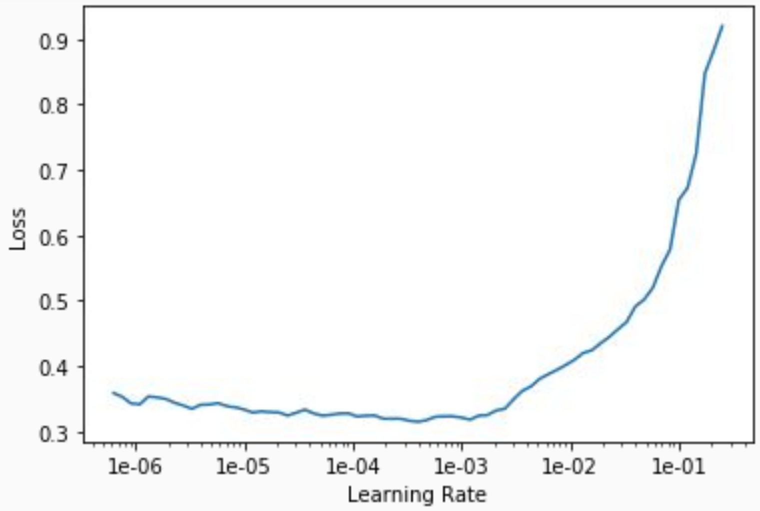
The 10, semi-accurate labels have been difficult to classify together.

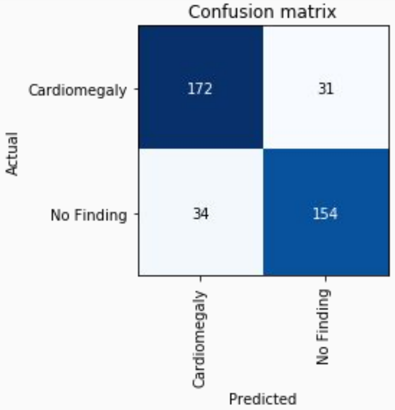
One Cycle policy



- Learning rate is often seen as the most important hyper-parameter, but there is often a need to experimentally find the best values and schedule for the global learning rates.
- **Cyclical Learning Rates** eliminates the need of monotonically decreasing the learning rate, this method lets the learning rate cyclically vary between reasonable boundary values.
- Provides a simple way to estimate "reasonable bounds" -- linearly increasing the learning rate of the network for a few epochs.
- The benefits of the cyclic learning rate (CLR) methods. Is that a short run of only a few epochs where the learning rate linearly increases is sufficient to estimate boundary learning rates for the CLR policies. Then a policy where the learning rate cyclically varies between these bounds is sufficient to obtain near optimal classification results

Binary classification with One-Cycle Policy





Train_loss	Valid_loss	valid_acc
0.500656	0.361455	0.891432

The learning rate plot obtained from the LRfinder method using one cycle policy gives us the optimal lr to train the model on.

Next Steps / Questions

- Bounding Boxes seem to be an area of exploration that could provide promise.
- Additionally, a dataset with improved label accuracy could provide significant improvement in results.
- Alterations to our preprocessing procedures to allow for more flexible manipulations (i.e., changes to normalization process, etc.).
- Deployment of other models, such as AdaBoost & SVM.
- CLAHE image pre-processing