SpiroSound

Jake Garrison and Farshid Salemi Parizi
Department of Electrical Engineering
University of Washington

Abstract

Spirometry is a clinical sensing technique for identifying or monitoring pulmonary exacerbations in patients. The expense of the equipment and requirement for adequate training interferes with widespread adoption of spirometry, especially in developing regions where it is most needed. We present several methods for performing spirometry on a mobile phone using the internal microphone and evaluate the reliability of each method.

1 Introduction

We first present the motivation, related work and dataset, then go into detail on the two categories of methods used for mobile phone spirometry, classical machine learning and deep learning. We conclude with a comparison of the results followed by ideas for future work.

1.1 Motivation

Lung disease contributes to roughly 10% of deaths in the world and approximately $50 billion in US healthcare costs annually [3, 7]. Spirometry [2] is the most widely employed objective measure of lung function and is central to the diagnosis and management of chronic lung diseases, such as asthma, chronic obstructive pulmonary disease (COPD), and cystic fibrosis. Forced expiratory volume in one second (FEV1) is the volume exhaled in the first second which has great importance on diagnosing lung diseases or COPD. However, challenges currently facing spirometry include hardware cost, patient compliance and usability [5,6]. In this project we present SpiroSound, a smartphone-based approach that calculates FEV1 using the phone's built-in microphone.

1.2 Related Work

Our work draws motivation from prior research exploring solutions that utilize sensing and computing capabilities of smartphones as well as technologies that leverage audible sensing for improvement in healthcare. Larson in SpiroSmart [1], which is prior research from our lab, relies on complicated algorithms that require a remote server for prediction, while our simple implementation runs locally and efficiently on the phone. In SpiroSound we use a subset of the dataset used in SpiroSmart, and suffer from similar accuracy issues and limitations.

2 Dataset

2.1 Data Collection

Our dataset consists of two trials for each of the 500 patients where a trial contains a waveform recorded from the mobile phone, groundtruth data from a clinical spirometer, and other patient flags such as gender, smoker, ethnicity. The patient data comes from a mix of university students and various local VA hospital patients. The mobile data is collected on various android phones held approximately 15 cm from the patient's face. The patient takes a deep breath, then forcibly exhales until their lung is depleted. The breathing technique is exactly as described in the Spirometry Procedures Manual [4]. The groundtruth spirometry data is collected in the traditional

clinical manner and rather than extracting the entire flow volume curve, we extract the FEV1 which, according to pulmonologists, is the most useful result of a spirometry test. A healthy FEV1 increases with height and weight, but generally is between 3 and 4.

2.2 Preprocessing

The raw waveforms from the mobile phone are preprocessed such that they are uniform in duration (10 seconds), resampled to 8kHz, and trimmed to start and end so that only the exhale audio data is captured. In order to best capture the sound of the airflow, the waveform is low pass filtered to eliminate sound above 400 Hz. While this may remove useful audio, it helps by removing speech and wheezing noises that are specific to a trial or patient. More advanced filtering could be used to better extract the airflow sound. Figure 1 below shows an example waveform and spectrogram after preprocessing.

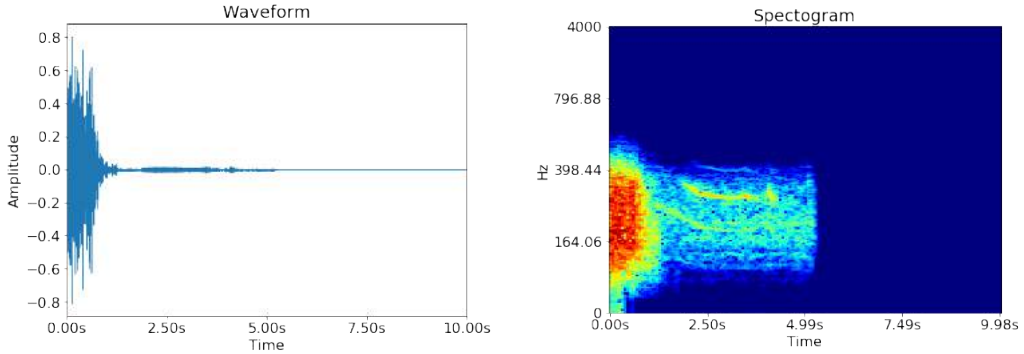


Figure 1: Filtered input waveform and spectrogram respectively

2.3 Data Split

Given our data comes from the general population, it is unsurprisingly biased towards healthy people as they are the majority. As a result, the machine learning technique is at risk for essentially guessing the mean FEV1 rather than generalizing to the full variance of the data. To remedy this, we truncated our dataset to be more uniform across our FEV1 range. The results of truncation are shown in Figure 2. This step greatly improved our model's ability to generalize to all patients.

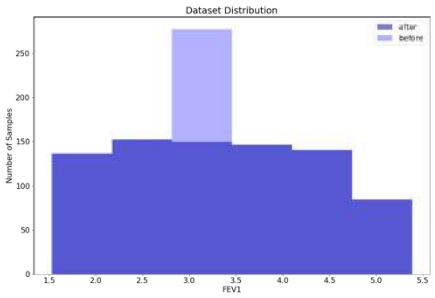


Figure 2: Histogram showing data

The refined dataset is split into a test and train set such that 80 percent of the data is used for training. Of the 80 percent training data, 20 percent is set aside as our validation set.

3 Machine Learning

We tested various methods of feature extraction and classical machine learning techniques in order to regress to the true FEV1 from the audio input, optimizing for mean squared error (MSE). Each of the following methods used the scikit-learn library for implementation.

3.1 Feature Extraction

We use the absolute value of our recorded audio file, shown in Figure 3, as our palette for feature extraction. Since most of the information is in the initial one second, we divided the first second into eight equally distant portions and calculated the area underneath the waveform in each portion and subsequently treated each of the areas as a feature. This significantly down samples the input

data. We also included the decay feature which is the time when the sound decays to room noise level. Other extracted features included max amplitude and the area underneath the envelope. We coupled these features with patient info such as gender, ethnicity and if the patient was a smoker.

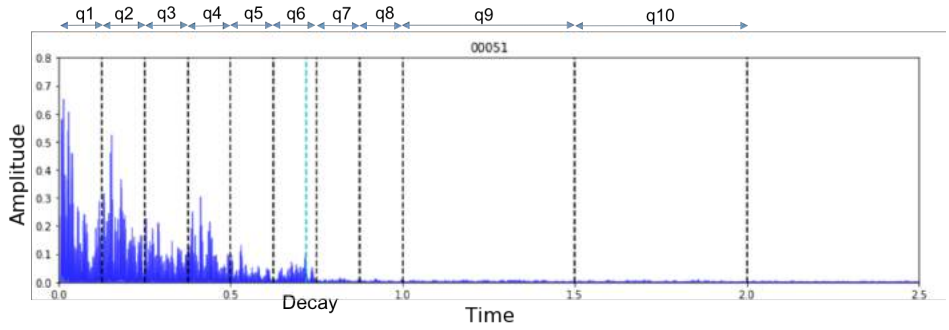


Figure 3: Feature extraction from waveform envelope

3.2 Models

We evaluated five different machine learning approaches to regress to FEV1. These approaches were random forest, gradient boosting, SVM, ridge regression and KNN. We used grid search and three-fold cross validation to tune different hyper parameters in order to optimize for MSR.

3.3 Results

The results of the five approaches are summarized in Table 1. The random forest model had the least mean square error, while the ridge regression trained the fastest. For reference, a MSR of 0.7 corresponds to an average percent error of around 30%.

Model	Mean Squared Error	Train Time (seconds)
Random Forest	0.631	19.88
Gradient Boost	0.676	1.04
SVM Linear	0.761	454.6
Ridge Regression	0.798	0.212
KNN	0.954	7.67

Table 1: Regression results sorted by error

3.3 Feature Importance

In this section, we explore the importance of our input features for the different machine learning approaches. As shown in Figure 4, each model used the features differently, for example a feature might have a significant importance on one model and not have any effect on the other model. For example, in the random forest model gender is the most impactful parameter, whereas in SVM, asthma has the greatest significance.

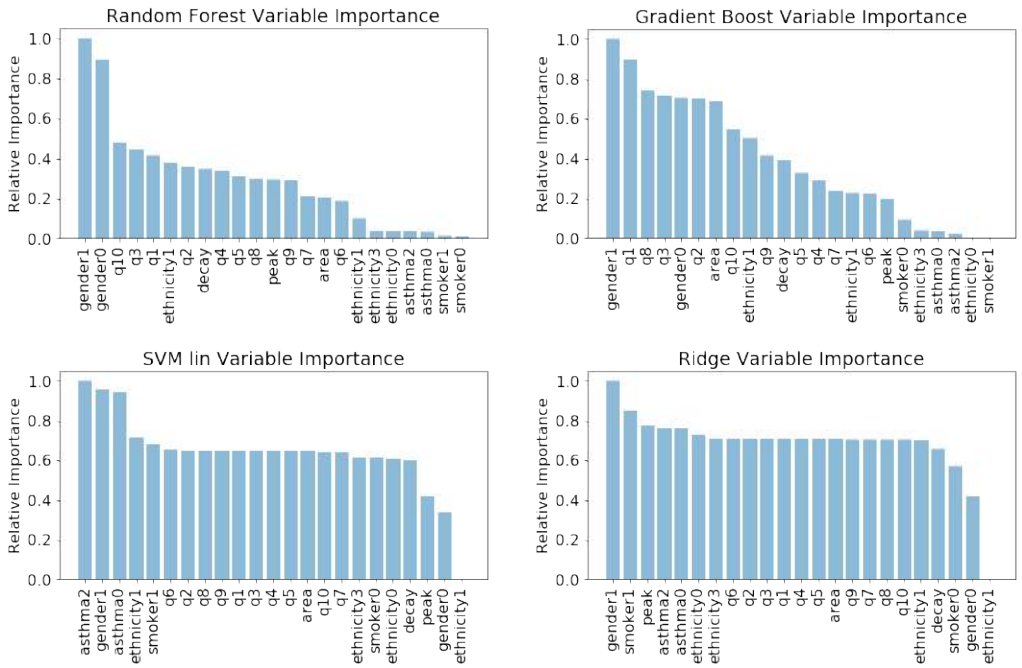


Figure 4: Feature importance for top machine learning methods

4 Deep Learning

We experimented with deep learning via Tensorflow with the hopes of achieving smaller error rates and obtaining new insights on feature extraction. Convolutional neural networks (ConvNet) were used so the trained filter layers could be analyzed for ideas on feature extraction.

4.1 Input

Unlike the approach used in the machine learning, our deep learning input was simply the preprocessed waveform. No patient information or extracted features were used. The input audio was in the form of a spectrogram image with dimensions 256 x 157 where the axis was frequency by time resolution. Refer back to Figure 1 for an input spectrogram example.

4.2 Models

Two different ConvNet architectures were explored, the first had three layers and the second had five. The three layered net was more compact, trains faster and seems to generalize better on unseen test data. Conversely, the five layer net was much deeper, contained many more hidden nodes and had a symmetrical max pooling size. The ConvNet architectures are shown in greater detail in Appendix A.

Both architectures were trained with 50 epochs and a batch size of 32 (constrained by the available GPU memory of our Titan X). The ConvNets were setup to classify the correct FEV1 class from 16 classes spanning from 1.25 to 5.5 in increments of 0.25. The output softmax layer was used to regress from the confidence associated with each bin to a single FEV1 value from which MSR can be evaluated. The biggest limitation to this approach is while training, the nets were optimizing for accuracy (identifying the correct FEV1 bin), but the metric we actually want to optimize for was MSE obtained through regression.

4.3 Training

Despite the obvious differences in the two ConvNet models, training results were somewhat model independent. Figure 5a reveals that the five layer model indeed achieved a higher training accuracy, but this did not translate to a better validation accuracy. The five layer model overfits to the train data and in 50 epochs nearly reaches a point of diminished learning from the data. Since

the validation accuracy is continually climbing, it could be worth training for more epochs to when the validation accuracy finally settles, especially for the three layer net since it only reaches 50% accuracy on the training data.

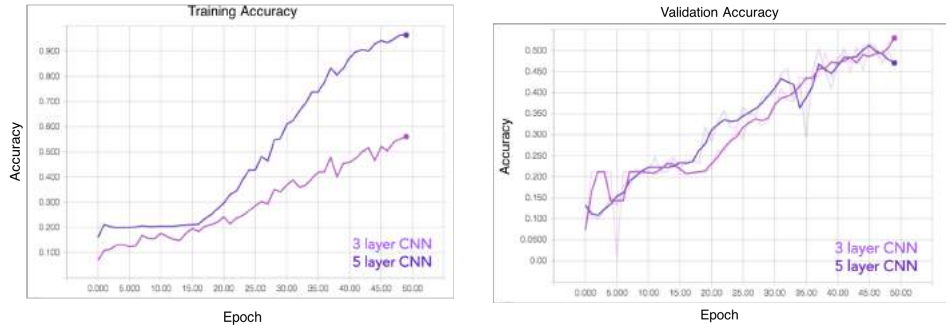


Figure 5: Training accuracy and validation accuracy for both networks

4.4 Results

The results for the two classifiers are shown in Table 2, along with the softmax conversion to mean square error. When compared to Table 1, the deep learning methods fail to do as well as classical models; however, they are not directly comparable since the deep learning only used the raw audio as input. Furthermore, it is interesting to see that classification accuracy is not a good proxy for MSR in regression since models with higher accuracy do not necessarily have lower MSR. Overall it appears the three layer model is more effective in terms of MSR, less prone to overfitting, and faster to train. More analysis may reveal specific features in the spectrogram that aid the classification task. Such analysis would be useful for improving feature extraction in section 3.1.

Model	Classifier Accuracy	Mean Squared Error	Train Time (seconds)
3 Layer	0.47	1.08	85
5 Layer	0.52	1.11	146

Table 2: ConvNet model results

5 Conclusion

Our results indicate we can regress from audio to FEV1 with a MSR of around 0.63 which corresponds to around 20% error on average. While this is not effective enough to be routinely used in a clinical setting, it is far better than guessing. Our model is clearly better at predicting FEV1 for healthy patients. While the intention of the deep learning was to discover new insights on feature extraction and ideally achieve lower error, the results ended up being less favorable.

5.1 Model Comparison

The Bland-Altman plots in Figure 6 indicate possible bias in the model. Ideally the scatter is clustered along the 0 horizontal line meaning there is no bias toward any particular FEV1 value. Our results suggest a clear bias towards the mean FEV1 (~3.5), especially in the Ridge regression model which has a clear linear trend. The random forest and three layer models appear to be the most generalized. Overall, random forest is our best model based on the low MSR and the Bland-Altman plot.

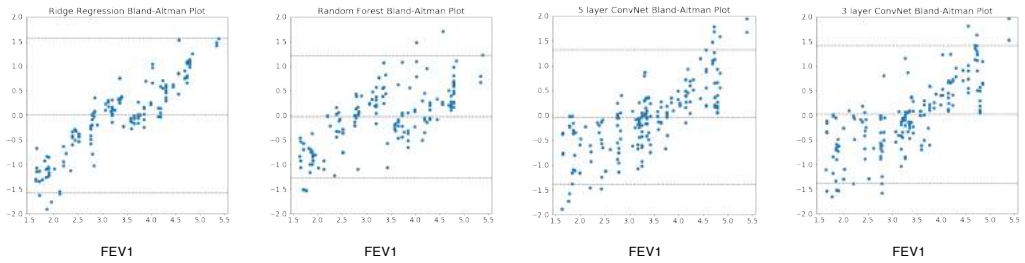


Figure 6: Bland Altman plots for bias analysis

6 Future Work

This project serves as a checkpoint in a larger research problem. Since this problem has not been well explored in prior related work, our methods are very much experimental. Our findings indicate there is room for improvement and exploration in both feature engineering and deep learning architecture design. Coming up with better ways of extracting useful information from the sound will be key to the success of this technique. The ConvNet can be improved by altering it to optimize for MSE in regression, rather than classification accuracy. Additionally, different inputs such as the sound envelope or differently scaled spectrograms could prove effective, or other neural network architectures such as an RNN may learn better temporal features from the data.

7 References

[1] Eric C Larson, Mayank Goel, Gaetano Boriello, Sonya Heltshe, Margaret Rosenfeld, and Shwetak N Patel. 2012. SpiroSmart : Using a Microphone to Measure Lung Function on a Mobile Phone. UbiComp'12

[2] Townsend, M.C. Spirometry in the occupational health setting. Journal of occupational and environmental medicine / American College of Occupational and Environmental Medicine 53, 5 (2011).

[3] The clinical and economic burden of chronic obstructive pulmonary disease in the USA https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3694800/

[4] Spirometry Procedures Manual https://www.cdc.gov/nchs/data/nhanes/nhanes_07_08/spirometry.pdf

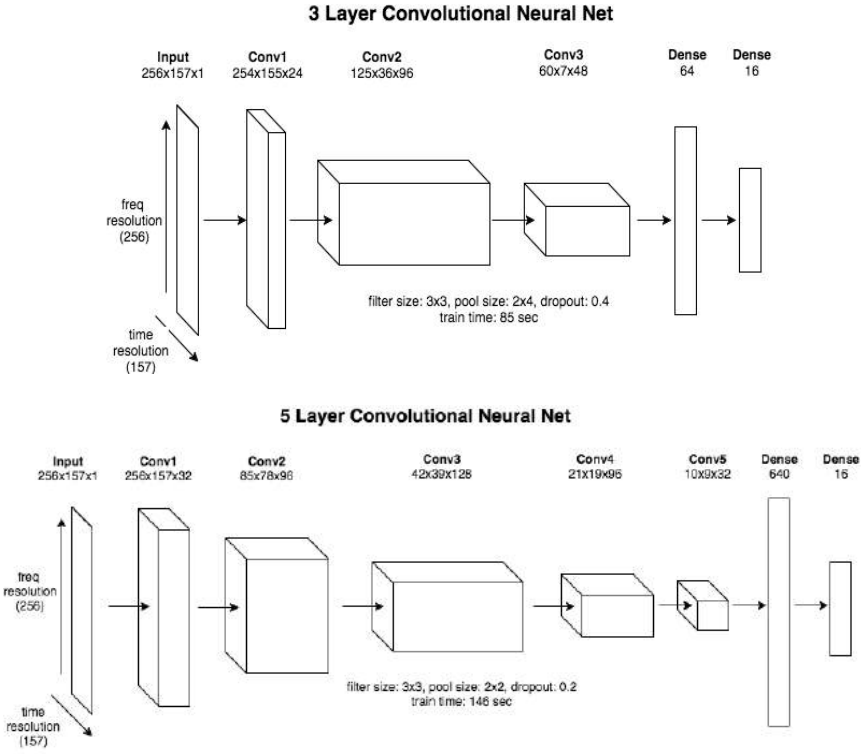
[5] Finkelstein J, Cabrera MR, H.G. internet-based home asthma telemonitoring: can patients handle the technology. Chest 117, 1 (2000)

[6] Grzincich, G., Gagliardini, R., and Bossi, A. Evaluation of a home telemonitoring service for adult patients with cystic fibrosis: a pilot study. J. of Telemedicine, (2010).

[7] Top causes of death http://www.who.int/mediacentre/factsheets/fs310/en/

8 Appendix

A Neural network architectures



ConvNet architecture diagrams for 3 and 5 layers