The Role of Processing Cough Audio to Detect Covid-19

Stephen Zhu
Department of Electrical Engineering
Stanford University srzhu3@stanford.edu

1 Abstract

This project aims to examine the role of signal processing in aiding audio classification, specifically detecting Covid-19 in cough audio files. During the pandemic, it took some time before the first accessible covid tests were released, posing a global health issue. If we were to collect data early on and use machine learning in the future, we would potentially be able to deploy these accessible tests at a much faster rate, and at significantly less cost.

This project compares the accuracies of different classification techniques, namely KNN, SVM, and ML, combined with different forms of signal processing on the COUGHVID dataset [1]. The types of processing used are no processing, DFT, and MFCC. As shown in this project, adding even one layer of signal processing can boost classification accuracy.

2 Methods

2.1 Dataset

The COUGHVID dataset is a crowdsourced dataset comprised of around 35,000 audio files containing coughs, json files containing basic information about each of the audio files, and a metadata CSV file containing information on all the files. Within the metadata file, each audio file is given either no label, a healthy label, a symptomatic label, or a covid (positive) label. The audio files in question are either mp3, ogg, or webm files with varying sampling rates, lengths, and coughs.

2.2 Data Preprocessing

Since the raw data was so varying, it was necessary to preprocess them in order to prepare them for signal processing and classification. Luckily, the COUGHVID dataset authors provided some starter code that included many preprocessing functions. First, all of the ogg and webm files were converted to mp3 files using an application called FFMPEG. Then, the metadata CSV file

was parsed, and all of the files that did not have a label were removed, while all the ones that did have a label were separated into a directory labeled as "healthy" and a directory labeled as "covid." For simplicity, the files labeled as symptomatic in the metadata file were also grouped together with the covid files. The healthy directory contained around 15,000 files, while the covid directory contained around 5,000 files.

Lastly, each file in each directory was resampled at a set frequency (22,050 Hz) before a provided function was used to extract each cough within each audio file. These coughs were then normalized, lowpass filtered, downsampled, and then cropped or extended to be exactly 22,050 samples long (1 second). The first 1000 coughs of each label (healthy and covid) were then recorded into txt files for storage.

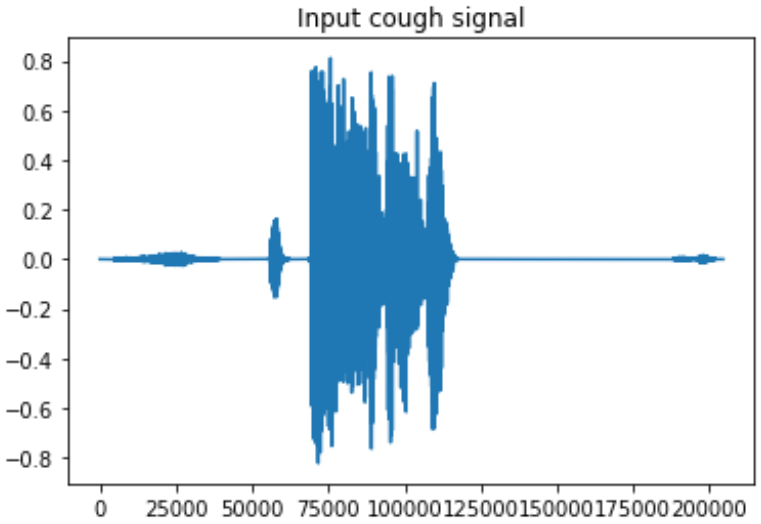

Figure 1. An example of an unprocessed, covid labeled audio file.

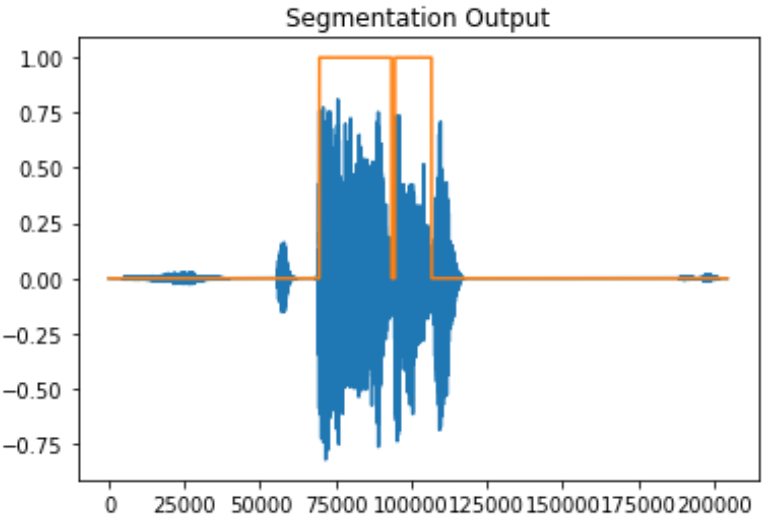

Figure 2. Plot showing ranges of detected coughs.

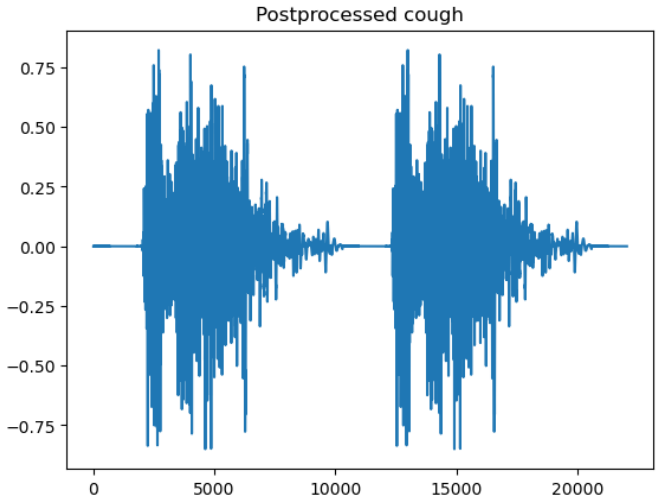

Figure 3. Postprocessed example cough.

2.3 Signal Processing Methods

To examine the effects of signal processing in covid detection, 3 variations of signal processing were used: no signal processing, DFT, and MFCCs. The naïve approach to covid detection from the cough audio would be to classify based off of the raw audio file; this approach also serves as a baseline to compare the other tests with. The DFT method was chosen since it would convert the time domain audio into the frequency domain, which contains information such as the fundamental frequency and harmonics. Lastly, MFCCs were used as a well-known signal processing method used for extracting features from audio [2,3].

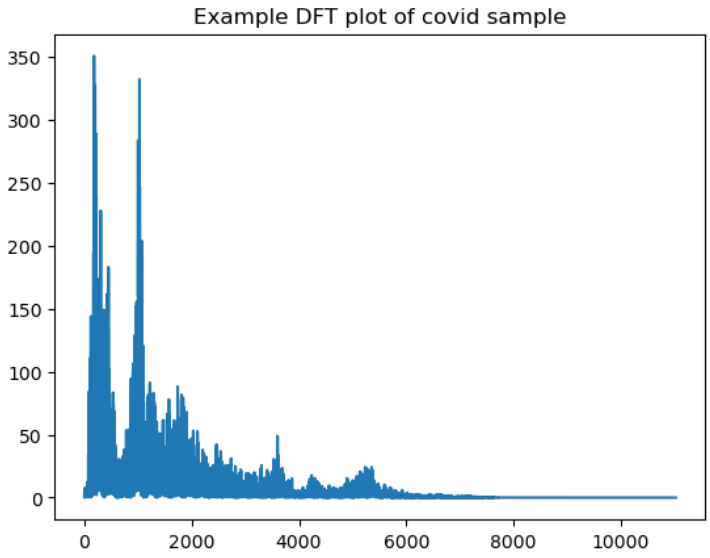

Figure 4. DFT of a covid (positive) cough.

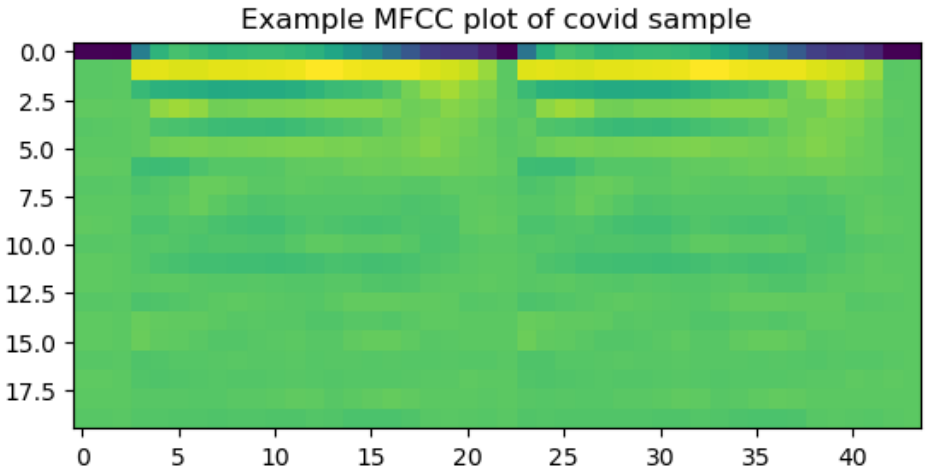

Figure 5. Image depicting the first ~40 MFCCs of a covid (positive) cough.

2.4 Classification Methods

3 types of classification methods were chosen to classify the raw or signal processed coughs: KNN, SVM, and ML. Similar to the concept of using the raw cough audio, the KNN method was chosen as a naïve implementation for classification. For each signal processing method, 1-10 nearest neighbors were used, and the highest accuracy was reported. One step above this, the SVM method was chosen to see if the classification could be split using a hyperplane, either linear or defined from the precomputed kernels in the sklearn Python library. The kernels used were: linear, poly, rbf, and sigmoid. For each of the signal processing methods, the regularization parameter C was varied between 0.01 and 10, and each of the kernels was used.

Lastly, a simple ML model was constructed to see if a non-precomputed function could be trained to fit the data. The model in question was implemented through Pytorch, and it has a simple architecture: Input -> Linear -> ReLU -> Linear -> ReLU -> Linear -> Softmax -> Output. To keep the different models for each signal processing method as comparable as possible, the loss function was kept as MSE loss, the optimizer was fixed as Adam, and each model was trained for 20 epochs with batch size 4. The (constant) learning rate had to be adjusted for each model though, as it would either not converge or converge too slowly, depending on the signal processing method used. Thus, the raw cough audio model used a learning rate of 1e-3, the DFT model used a learning rate of 1e-5, and the MFCC model used a learning rate of 1e-4.

From the 2000 preprocessed coughs (1000 healthy, 1000 covid (positive)), the data was split into a 80-20% ratio for training and validation respectively.

3 Results

Figures 6-8 show example accuracy graphs generated using the MFCC signal processing method and each of the classification methods.

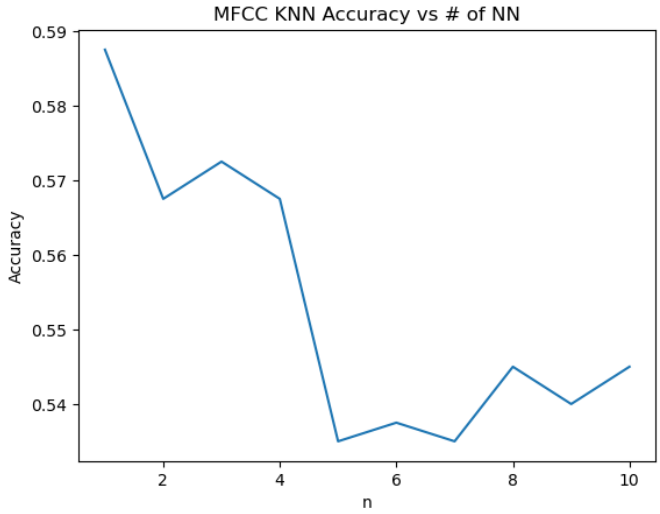

Figure 6. MFCC KNN accuracy vs # of nearest neighbors plot.

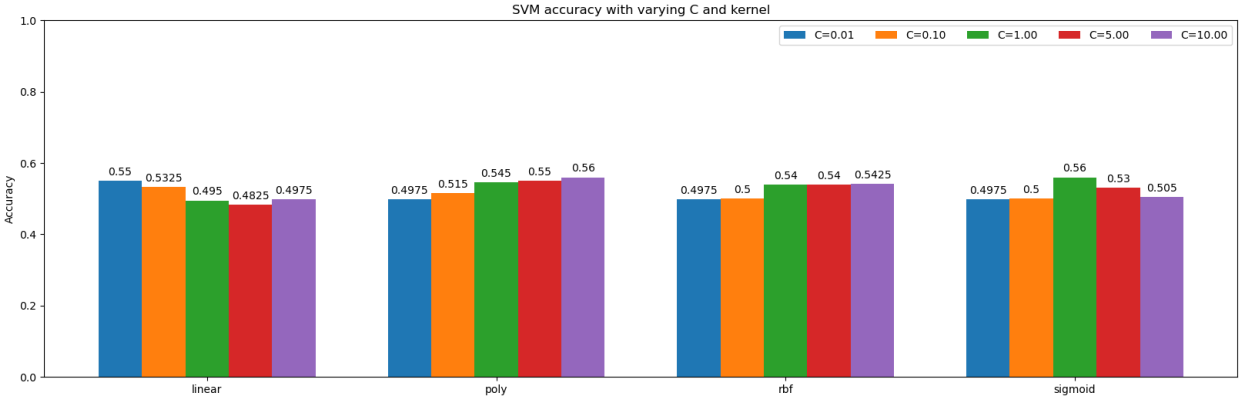

Figure 7. MFCC SVM accuracy vs C and kernel chart.

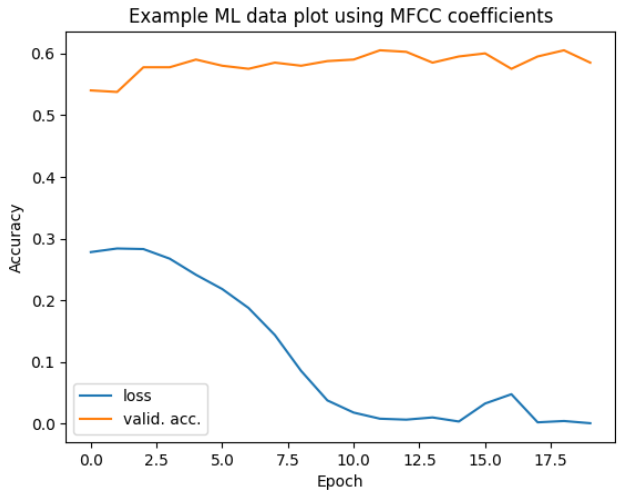

Figure 8. MFCC ML accuracy vs epoch graph.

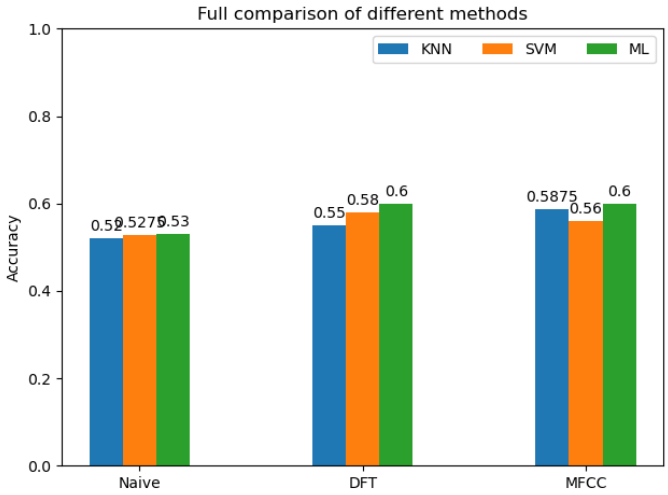

Figure 9. Highest accuracy comparison between different signal processing and classification methods.

It is clear to see from Figure 9, the highest accuracy graph for each method, that from the signal processing methods, using the raw audio file performed the worst, while DFT and MFCC both had higher accuracies. From the classification methods, KNN mostly performed the worst, SVM usually performed slightly better, and ML performed the best consistently. The highest accuracy obtained by KNN was 58.75% using MFCCs, by SVM was 58% using the DFT, and by ML was 0.6 using both DFT and MFCC.

4 Discussion, Conclusions, and Future Work

As expected, using the naïve method for both signal processing and classification performed the worst. For signal processing, this is expected because the DFT and MFCCs contain more frequency based information, which is likely more distinguishable due to the lack of structural complexity that comes from the time domain signal. For classification, since the data is so high dimensional in all of the signal processing cases, it is hard for KNN to work unless exactly the right features are extracted to separate each label into clear clusters in n-space. SVM is also limited by its kernels, and if the data points don't fall under its precomputed kernels, then it is hard for the algorithm to fit to the data. A manually defined kernel could be used, but without knowledge of how the data is shaped in n-space, it would be hard to construct a good kernel. Thus, the reason why ML works well for this application is because of the model's ability to be trained to learn which features are actually important (which KNN can't do), and its ability to fit an arbitrary function to the datapoints (which SVM can't do).

To expand on this in the future, the rest of the dataset can be used, different signal processing methods can be combined, and the ML models can be trained for longer or their hyperparameters tuned. Wavelets, or other signal processing methods [3], along with other classification methods [2] can also be used to obtain higher accuracies.

5 References

[1] Orlandic, L., Teijeiro, T. & Atienza, D. The COUGHVID crowdsourcing dataset, a corpus for the study of large-scale cough analysis algorithms. *Sci Data* **8**, 156 (2021). https://doi.org/10.1038/s41597-021-00937-4

[2] Verde, L., De Pietro, G. & Sannino, G. Artificial Intelligence Techniques for the Non-invasive Detection of COVID-19 Through the Analysis of Voice Signals. *Arab J Sci Eng* **48**, 11143–11153 (2023). https://doi.org/10.1007/s13369-021-06041-4

[3] Ulukaya, S., Sarıca, A.A., Erdem, O. *et al.* MSCCov19Net: multi-branch deep learning model for COVID-19 detection from cough sounds. *Med Biol Eng Comput* **61**, 1619–1629 (2023). https://doi.org/10.1007/s11517-023-02803-4