CS230

Predicting Voice Pathology

Nishant Badal
Department of Computer Science
Stanford University
nibadal@stanford.edu

Louie Kam
Department of Statistics
Stanford University
lkam8096@stanford.edu

1 Introduction

Disorders of the voice are referred to as dysphonia and can be categorized into two groups - hyperkinetic and hypokinetic. In hyperkinetic disorders, phonation becomes forceful and exaggerated with sphincter action similar to gagging. Excessive muscular action and a tightened larynx lead to a grating sound. In hypokinetic disorders phonation instead becomes subdued and the larynx is largely sluggish leading to huskiness or a breathless sound. Reflux laryngitis is another voice disorder that is caused by back flow of stomach acid to the throat. Up to 9% of Americans may have voice disorders, however only around 1% seek treatment. Thus, it is important to investigate ways to quickly screen for vocal fold pathologies.

In this project, we implement different models that take in Mel-Frequency Coefficients (MFCCs) or spectrograms processed from the audio signal data. MFCCs typically represent phonemes and are frequently used in voice recognition. These types of data show how the frequencies of sound vary over time while reducing the large amount of noise present in audio data. The input of our algorithm will be MFCC and spectrogram data. We use deep learning models to predict the voice pathology.

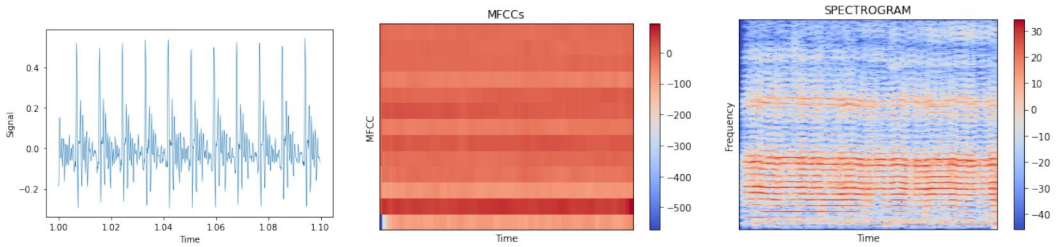
2 Dataset and Features

We use data from the VOICED database [1], which contains 208 voice samples from patients with different voice pathologies. For each patient, the vocalization of the vowel "a" was recorded for approximately five seconds with a constant voice intensity. Recordings were sampled at 8000 Hz with 32-bit resolution. The vocalizations were performed in a carefully controlled environment, accounting for variables such as background noise, humidity, and measurement device. The pathology of each voice was described and verified by medical experts as healthy, hyperkinetic dysphonia, hyperkinetic dysphonia, or reflux laryngitis. There are 72 voices with hyperkinetic dysphonia, 57 voices associated that are healthy, 41 voices with hypokinetic dysphonia, and 38 voices with reflux laryngitis.

We processed the audio signal of each subject into Mel-frequency cepstral coefficients (MFCCs). MFCCs are features extracted from audio signals commonly used in audio and voice tasks. The coefficients are calculated by separating the signal into windows of a designated size and stride, both in terms of the number of samples. The log of the discrete Fourier transform is applied to each window to compute the logarithmic power spectrum. MFCCs are obtained by conducting mel-scaled filter bank analysis and applying discrete cosine transform (DCT) for each window [8]. We rely on the LIBROSA package in Python to compute the MFCCs and spectrogram for each recording. Signals were padded from the left to keep the dimensions consistent between samples. Padding zeros from the left preserves the integrity of the signal, which is 0 at the beginning of all recordings. A example is shown in Figure 1.

We aim to predict voice pathology of patients given their audio data and other features. The dataset is split in 60/40 train/dev set. Nishant uses the dev set as a test set due to small sample size, and Louie

CS230: Deep Learning, Spring 2022, Stanford University, CA.



(a) Signal (truncated) (b) MFCCs (c) Spectrogram

Figure 1: Different representations of the audio data for voice ID 001

further splits the dev set into 50/50 validation/test set (so 60/20/20 overall split). The data consists of 124 train and 84 dev observations. It is a valid concern that a validation set of 42 observations is not enough to assess the validation loss robustly. Several data augmentation strategies were also utilized whenever possible. The samples were time-shifted 5 times to allow the model to better generalize by adjusting the left zero-padding of the signal during the initial stages of the recording (the recording starts before the voice, and the voice is cut off abruptly at the end of the recording). The initial half-second of every of every 4-second recording was silence. The shifting resulted in this moving to different locations in a uniform distribution. Certain models, such as VGGish [6] and YAMNET [7], slice the mel spectrogram into 0.96s duration with 50% overlap to increase the number of observations.

3 Methods

We train models to predict the voice pathology of patients using MFCCs provided from the recordings. As there are four possible diagnoses (healthy, hyperkinetic dysphonia, hyperkinetic dysphonia, or reflux laryngitis), this is a multi-classification problem. Thus, we use the categorical cross-entropy loss function as follows (for one sample):

$$\mathcal{L}(y, \hat{y}) = -\sum_{c=1}^{C} y_c log \hat{y_c}. \tag{1}$$

Note that C is the number of classes, y_c is the output label for class c, and $\hat{y}$ is the probability assigned to class c by the model. This loss function tries to maximize the probability of the ground truth label in each training example.

Due to the small number of samples in the VOICED database, we try a variety of models for transfer learning: VGGish, ResNet50, and YAMNET. These models were used as feature extractors. We freeze the pretrained weights in these networks to create embeddings for the input audio. VGGish is a model released by the Google Sound Understanding team that outputs 1x128 dimensional embeddings and has labels from more than 600 event classes [3] [6]. The model was trained using the AudioSet data set which contains 2 million human-labeled 10-second YouTube soundtracks. ResNet50 is a residual learning model that achieved a 3.57% error rate on the ImageNet database which consists of hundreds of thousands of images linked to words or phrases [2]. YAMNET is similar to VGGish in design, although it employs the MobileNet architecture, which is known for its efficiency [4]. We not only append dense layers to these pretrained networks but also try architectures related to few-shot learning: a ResNet 50-based Siamese network and a ResNet50-based prototypical network. We also implemented a RNN model that takes in a similar style of input used in VGGish and YAMNET. An Adam optimizer was used in all models along with Xavier initialization. The models were implemented using Keras and PyTorch.

4 Experiments

We selected F1 score as our primary metric to evaluate the performance of our models. In addition to this, accuracy was recorded for all models. All models had a softmax activation output. A 10^{-3}

learning rate. A learning rate of 10^{-2} often failed to converge as the gradients were large could not reduce loss effectively while 10^{-4} and lower learning rates took many additional iterations to converge.

4.1 Model A: ResNet50-based fully-connected network

ResNet50 expects an AxBx3 input size (3 channels). The 96x64 MFCCs were stacked thrice and fed to a ResNet50 base model with weights frozen. Next an average pooling layer was applied. Average pooling was selected over max pooling because the window size is small and information is not sparse in the embeddings. After this were two fully-connected layers with 128 units and 40% dropout followed by batch normalization and the output layer. The model was trained 10 epochs. Model achieved a weighted F1 score across all classes of 0.30.

4.2 Model B: VGGish-based fully-connected network

VGGish expects an input size of AxBx1. The MFFcs were fed into the VGGish model with the its weights frozen. Then 3 fully-connected layers of 128 units were applied. The model was trained 100 epochs.

Class	F1 Score	Precision	Recall
Hypokinetic	0.43	0.37	0.50
Hyperkinetic	0.33	0.40	0.28
Reflux Laryngitis	0.07	0.37	0.19
Healthy	0.43	0.51	0.37

(a) Performance by class

Metric	Score
Weighted Avg. F1 Score	0.36
Test Accuracy	0.34
Train Accuracy	1.0

(b) Overall performance

4.3 Model C: ResNet 50-based Siamese network

Two identical networks consisting of a ResNet50 base model followed by an average pool layer and two 32 unit fully-connected layers were constructed. The two networks shared the same weights. The euclidean distance between the output embeddings of the networks was calculated and fed to an output unit with a sigmoid activation function. Binary cross-entropy was used as the loss function. The model was trained for 20 epochs.

Class	F1 Score	Precision	Recall
Different	0.67	0.50	1.00
Same	0.00	0.00	0.00

(a) Performance by class

Metric	Score
Weighted Avg. F1 Score	0.33
Test Accuracy	0.50
Train Accuracy	0.83

(b) Overall performance

Figure 3: Model C performance indicators

4.4 Model D: ResNet50-based prototypical network

A prototypical network was implemented using ResNet50. The architecture of the network was created as Snell et. al. have described [5]. The network was 4-way 5-shot classification with a softmax output. The final layer of ResNet50 was flattened and used to calculate the euclidean distance between classes. The weights were unfrozen. The model was trained for 20 epochs.

4.5 Model E: LSTM Model

We implemented a basic LSTM model with a LSTM layer (512) followed by two Dense layers (128 and 4). The input shape is $(76, 13)$ representing the 13 MFCCs over 76 windows of the audio signal.

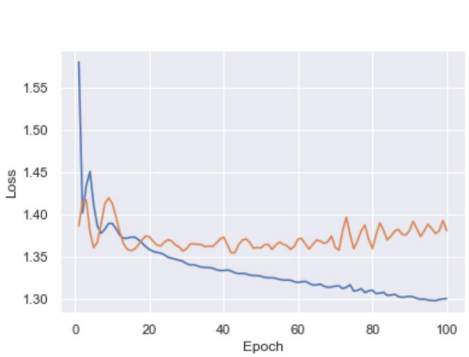
Class	F1 Score	Precision	Recall
Hypokinetic	0.53	0.33	0.53
Hyperkinetic	0.59	0.40	0.57
Reflux Laryngitis	0.39	0.27	0.33
Healthy	0.53	0.51	0.47

(a) Performance by class

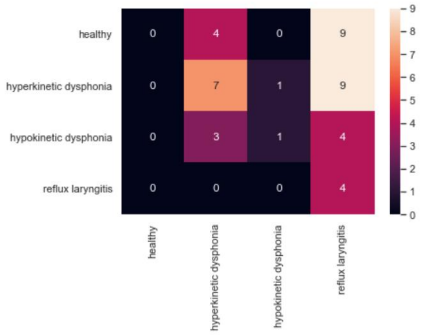
Metric	Score
Weighted Avg. F1 Score	0.52
Test Accuracy	0.51
Train Accuracy	0.66

(b) Overall performance

Classes were weighted according to their frequency. No dropout was applied, and the model was trained for 100 epochs. The test classification accuracy was 0.286, and the weighted-F1 score was 0.229. See Figure 5 for more details.



(a) Training and validation loss



(b) Confusion matrix evaluated on the test data

Figure 5: Results of the best LSTM model

4.6 Model F: YAMNET-based fully-connected network

We replace the dense output layer of 521 nodes, the classes in the AudioSet-YouTube corpus the YAMNET model was pretrained on, to a hidden dense layer with 256 nodes and an output layer with 4 nodes for our classification problem. We use the validation set to roughly scope out some of the hyperparameters, such as learning rate (10^{-2}, 10^{-3}, 10^{-4}), dropout (0, 0.2), and epoch (10, 20, 50). The best-performing model (according to the validation loss) had a 10^{-2} learning rate, no dropout, and 10 epochs of training. The test accuracy across the mini-samples (the various 0.96s observations) is 0.355. This model also differs in that it trains on mini-samples and the results can be aggregated by taking the mean of the softmax probabilities across all mini-samples in the recording. This allows us to visualize segments of the audio to determine which areas are most indicative of the voice pathology, as shown in Figure 6.

Unfortunately, this model performs degenerately, predicting hypokinetic dysphonia in the training and validation set before predicting healthy in the test set.

5 Discussion

Initially tested convolutional neural network architectures had large gaps between train and test set performance. This is likely in large part caused by a small sample size of 208 4 second voice clips in the VOICED database. Transfer learning using ResNet50 and VGGIsh increased the performance by 5-10%. Several data augmentation strategies were also applied to mitigate this described in the data section above. These increased F1 scores by around 5% in both models A and B. Larger networks were also considered for models A and B. Model A with four fully-connected layers with 128 units with and without dropout were both trialed. The larger networks did not increase performance or shorten the train and test set gap.

Few-shot learning learning approaches seemed appropriate given the small data size. Model C attempted to use a Siamese neural network to distinguish between two classes. The F1 score by class

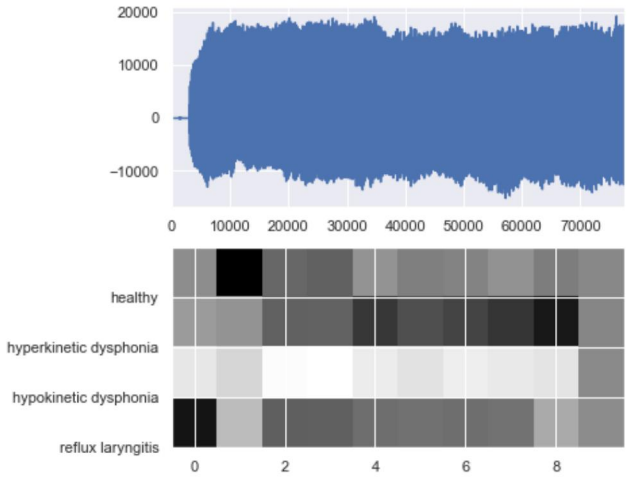


Figure 6: Evaluation of YAMNET-based model on a test recording (signal above). The dark squares indicate higher softmax probabilities

for this model is useful and showed the model is not predicting any pairs as the same. This is likely because of the small amount of data despite data augmentation. A few-shot learning approach using a prototypical network was also attempted. A euclidean distance was calculated between classes because because Snell et. al. found this to greatly improve performance[5]. 4-way comparisons performed better than 3-way and 2-way comparisons. This may be because 4-way comparisons are more difficult and force the model to generalize better and make more tuned decisions. The prototypical network yielded an F1 score of 52% which is the highest of all models. It performed slightly worse on reflux laryngitis classification. This could be influenced by reflux laryngitis having the smallest number of samples of all classes.

The design of the YAMNET-based network was promising with its additional ability to classify small windows in the recording, giving experts an opportunity to narrow down voice characteristics associated with the pathologies. However, this requires a good model, and the YAMNET-based network in this paper fails to perform better than a baseline. We also tried unfreezing the layers, but this did not prove helpful. This suggests that the embeddings provided by the network lie too far from those that are truly beneficial to voice pathology prediction, and subsequent layers of YAMNET may need to be set to trainable. There is also a possibility that the implementation was done incorrectly.

The LSTM model is interesting because it can take in variable inputs, which gives flexibility in real-word applications compared to the numerous CNN networks that rely on “image" inputs. There is some skepticism over the optimal processing of the MFCCs [8] that might allow the model to improve. Stacking LSTM layers might also help performance, although we believed that this would not improve performance training from scratch.

6 Conclusion/Future Work

We explored several different network architectures to classify 4 vocal fold pathologies. Our prototypical network classifier yielded the best results, although it still struggles with classifying reflux laryngitis and did not achieve an F1 score greater than 0.52. Although prototypical networks have been shown in the literature to to have high classification rates even with as little as 10 examples on visual data [5], it could be that audio requires more examples. It may be useful to compare the performance of this network to a group of experts diagnosing these requirements. It is also possible that the quality of the audio data with a sample rate of 8000 is not high enough. Another strategy that could be explored is adding data from other datasets which have similar diagnoses. Finally, the interpretation of the parameters that are used to constructs MFCCs and spectrograms as hyperparameters might also yield task-specific optimizations.

7 Contributions

Nishant Badal

- Pre-processed data to extract MFCCs by porting Google Sound Understanding labs pre-processing audio script
- Augmented audio data using time-shifts
- VGGish model was not available in Keras. Found GitHub repository containing PyTorch implementation. Learned PyTorch to implement prototypical network
- Created, optimized, and analyzed ResNet50-based model using Keras
- Created, optimized, and analyzed VGGish-based model using Keras
- Created, optimized, and analyzed Siamese-based model using Keras
- Created, optimized, and analyzed Prototypical-based model using PyTorch
- Assisted with writing final report

Louie Kam

- Produced visualizations of MFCCs, spectrograms, and voice samples.
- Trained and optimized LSTM model over space of hyperparameters.
- Imported pretrained YAMNET model throught GitHub. Trained and (attempted to) optimized model over space of hyperparameters with pretrained weights frozen and unfrozen.
- Contributed to the final report (dataset, model explanation, conclusion, formatting).

References

[1] Ugo Cesari, Giuseppe De Pietro, Elio Marciano, Ciro Niri, Giovanna Sannino, and Laura Verde. A new database of healthy and pathological voices. *Computers & Electrical Engineering*, 68:310–321, 2018.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[3] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, and Kevin W. Wilson. CNN architectures for large-scale audio classification. *CoRR*, abs/1609.09430, 2016.

[4] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017.

[5] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning, 2017.

[6] TensorFlow. Vggish, 2021.

[7] TensorFlow. Yamnet, 2021.

[8] Saska Tirronen, Sudarsana Reddy Kadiri, and Paavo Alku. The effect of the mfcc frame length in automatic voice pathology detection. *Journal of Voice*, 2022.

8 Appendix

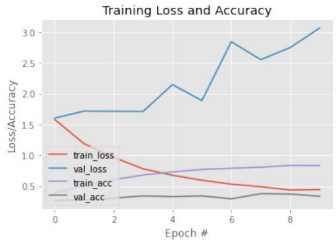


Figure 7: Model C Loss and Accuracy

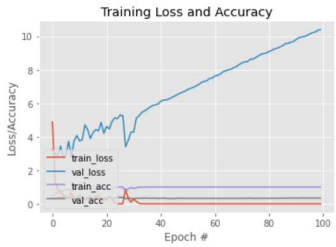


Figure 8: Model B Loss and Accuracy

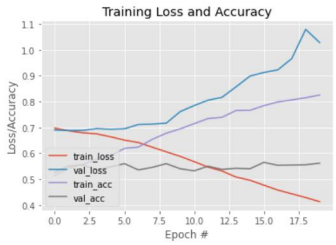


Figure 9: Model C Loss and Accuracy