Accent Classification

Sandeep Chilukuri, Brian Conn, Blake Hawes

Group 25

Introduction

Introduction: In our world, where so many different languages and accents mix, it's important to understand each other. But accents can make this hard. Most voice recognition systems are trained on "standard" accents and struggle with different or foreign ones. This can lead to misunderstandings and can make people feel less confident or less part of the community when they're trying to fit into a new place.

Goal: We want to solve this problem by creating a new Machine Learning model that can recognize and tell apart all kinds of accents in spoken language. This model will help voice recognition systems understand everyone better, no matter how they speak. This will make communication clearer and help build stronger connections between people. Also, our model will help people who want to reduce their accents, making it easier for them to blend in with the native speakers of the language, which can help them find better jobs and feel more at home in their new environment.

Data

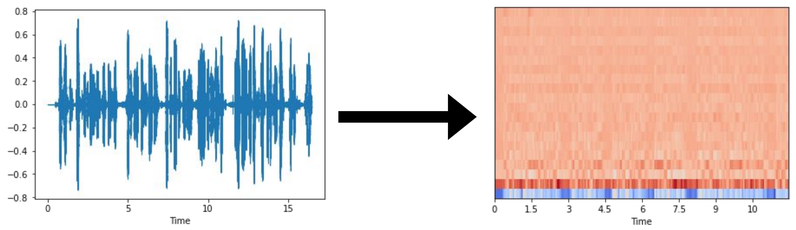
The dataset utilized in our project is the Speech Accent Archive and is on Kaggle. It contains audio samples from speakers with 214 different native languages. Each audio sample is spoken in English with the speaker reading the same paragraph. To ensure an adequate amount of samples we chose to use the 5 labels with the most data. These were English, Spanish, Arabic, French and Mandarin.

Language	Sample Count
English (Eng.)	579
Spanish (Spa.)	162
Arabic (Ar.)	102
French (Fr.)	65
Mandarin (Mdr.)	63

This data was collected by Dr. Steven H. Weinberger over the course of 20 years. These audio samples came from students, fellow researchers, and volunteer submissions. Most of the data was collected by Dr. Weinberger's students who were primarily affiliated with U.S. universities. This collection process has resulted in a notable class imbalance, with the number of English samples surpassing the combined total of all other languages. This imbalance poses challenges for data analysis and may impact the performance and generalizability of machine learning models trained on this dataset.

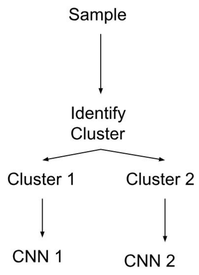
Preprocessing

The recordings were naturally all of varying lengths. To guarantee a consistent input shape for our model, recordings were truncated down to the length of the shortest recording. After splitting the recordings into training, validation, and testing sets, SMOTE resampling was applied to the training set to help remedy class imbalance between languages. Augmented samples were added to the training set by adding white noise to each training example. Lastly, the MFC (Mel-Frequency Cepstral) transformation was applied to each example as is commonly done in audio classification. Treating this transformation of the data loosely as an "image" of the sound, we can now use a CNN-based architecture to classify the data.

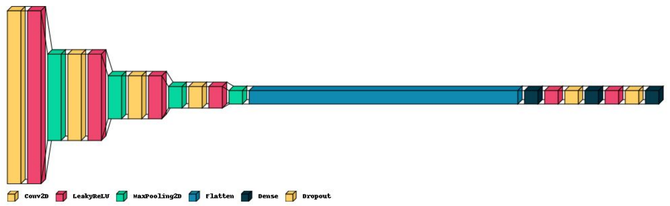


Methods

We decided to test two methods. First, a simple approach where we just use our CNN-based model to predict on the data. Then, a second approach where we first cluster the data into two clusters using K Means, then fine-tune the original model on each cluster, creating two new CNNs specialized to classify on a particular cluster. We hope that training the model on data that is more similar to each other will help the model pick up on the intricacies of the data. A diagram of this second approach is shown to the right.



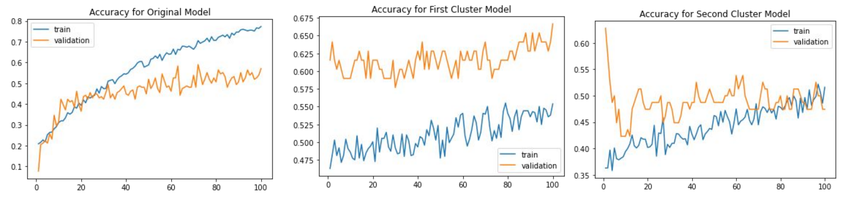
For our model architecture, we have used four convolutional layers with Leaky Relu as the activation function followed by the Max Pooling layers at each layer. Then we flatten the CNN layers and feed it forward using two fully connected dense layers, each with a Leaky Relu activation function and 0.5 dropout. Finally, we use a softmax function to output our predictions.



Results

After clustering, cluster 1 contained 486 samples while cluster 2 contained 485. Below are the testing results of our models. As shown in this table, the cluster models each performed better than the original model. Weighted together, the cluster models have an accuracy of 67%, a whole 15% increase from the original model. All other metrics show similar improvements.

Test Metric	Original Model	Cluster 1 Model	Cluster 2 Model	Weighted Average of Cluster Models
Loss (CCE)	1.3154	0.9285	0.9604	0.9444
Accuracy	0.52	0.67	0.68	0.67
Precision	0.55	0.69	0.68	0.69
Recall	0.52	0.67	0.68	0.67
F1 Score	0.53	0.68	0.67	0.68



Confusion Matrix for Original Model

T\P	Ar.	Eng.	Fr.	Mdr.	Spa.
Ar.	5	6	3	1	5
Eng.	4	80	6	8	18
Fr.	1	6	3	0	3
Mdr.	0	7	2	3	1
Spa.	4	10	5	3	11

Confusion Matrix for Cluster Models

T\P	Ar.	Eng.	Fr.	Mdr.	Spa.
Ar.	11	3	1	2	3
Eng.	1	91	5	9	10
Fr.	1	2	7	1	2
Mdr.	1	1	1	9	2
Spa.	4	11	0	3	14

Conclusion

In this study, we explored the task of predicting a speaker's native language from speech recordings. Our approach involved training a Convolutional Neural Network (CNN) on the entire dataset and experimenting with a clustered fine-tuning technique using K Means clustering. We found that these cluster-specific models performed significantly better than the original model trained on the entire dataset. This result suggests that there may be underlying patterns or characteristics within the data that are better captured by models trained on more homogeneous subsets.

Future research could explore how this clustered fine-tuning method performs on other tasks, as well as explore the effect of increasing the number of clusters used.