# Impact of Noise for Accent Recognition

Anurag Pendyala

May 3, 2023

## Contents



## List of Tables

# List of Figures

# 1 Abstract

Communication is an integral part of human life. Humans have developed various languages, accents, and dialects during evolution. Recognizing accents can help tune applications at a personal level. This project aims to create a classification model that can classify accents even when the recording is noisy. Noisy data with background noises correspond to real-life in-the-wild scenarios. Building a robust noise-resistant accent recognition model can prove to be essential in making many personalized applications possible. Training a model on both clean and noisy data can make the model perform well for real-life scenarios.

# 2 Introduction

Communication is the key to the survival of humans for a long time. After humans evolved to start using voice to communicate, various languages emerged. However, it did not stop there. For a single language, humans have developed accents and dialects based on the regions they are living in. These accents also behave like different languages. Although the language is the same, say English, people speaking one accent can, sometimes, not understand other accents.

Accents are essential in our daily communication in the modern world, whether it be in business or social situations. Strong accent recognition models that can withstand noisy conditions are required due to the rising demand for personalized applications based on accents. For example, a person might understand basic knowledge but however, has a thick Irish accent. They might find it challenging to communicate in the Indian subcontinent where the accent is completely different, despite the language still being English. An accent recognition device can help personalize content for them.

However, to implement these applications in real life, accent recognition is a difficult task. One of the reasons is that most of the data on which current models are trained is "studio-quality" data. However, most of the applications can not use state-of-the-art hardware to achieve this high-quality data. They often have some accompanying noise from the background or static noise from the poor microphones. Most of the current models fail on classifying accents on this noisy data. Hence, a robust, noise-resistant model is needed for many applications to be successful.

## 2.1 Goals

The data considered, from AccentDB, is again studio quality. However, additional noise can be augmented to it to make it viable for the project. The final goal is to build and train a model that can understand there is noise in the audio and eventually learn to filter out the noise and classify the accent. The following are some milestone steps that were followed for the project:

- Take the original data and split it into train and test (80%-20%) split.
- Add Cafe and Babble noise as background to the original test and train data respectively.

- Train 3 models on only original data, only babble noisy data, and on both non-noisy as well as babble training data.

- Test these models on various test sets to evaluate their performance and choose the best one.

# 3 Related Work

There has been extensive research to detect the language spoken from an audio signal. [2] has worked on exactly that. The results were used to run Automatic Speech Recognizers to get the transcript of what was spoken. The problem that is being posed is similar to that. However, it is just the idea that is taken. Models from [2] can be considered as well if the [1]'s models fail.

The authors of [1] have worked extensively. However, they have not experimented with noisy data. Having noise can correspond well with real-life scenarios. AugLy [3] is a tool that can be used to inculcate noise into datasets. Most of the current state-of-the-art architectures working on accent/dialect recognition have mostly worked on studio-quality-level clean audio files.

## 3.1 Related Work on the dataset area

The authors of [1], around which the basic model and data are based, have shown an accuracy close to 98%. Hence, seems like the data collected by the authors is very clean and easy to understand. However, noisy data was not tested.

## 3.2 Related Work on the Pattern Recognition Approach

Most of the language, speech, dialect, and accent recognition is done by using features like Mel Frequency Cepstrum Coefficients [4]. MFCCs capture harmonics and resonances caused in the throat while speakings. These changes every time a syllable is pronounced. Hence, these have information that can uniquely identify the language or dialect, or accent. [5] has worked on dialect recognition for the Arabic language. They use a similar approach to frequencies as feature extraction methods. AugLy ([3]) lets users add noise to various available settings like airport, babble, cafe, etc. The amount can be varied by adjusting the SNR (Sound to Noise Ratio).

## 3.3 Related Work for Noisy Accent Recognition

There isn't any current work that addressed the issue of accent recognition on noisy data. However, I have drawn inspiration from my earlier work named "Impact of Noise for Speech Emotion Recognition.". The task involved speech emotion recognition on audio files on various models that were trained and tested on combinations for various feature extractors, architectures, and combinations of datasets [6].

# 4 Data

## 4.1 Summary

The dataset has multiple sections. The ones that I will be looking into are for the following 4 classes: American, Australian, British, and Indian. The number of data points for each class is summarized in Table 1.

| Class | Total Audio Files | Train Files(80%) | Test Files(20%) |
|---|---|---|---|
| American | 1484 | 1187 | 297 |
| Australian | 1484 | 1187 | 297 |
| British | 1484 | 1187 | 297 |
| Indian | 1484 | 1187 | 297 |
| **TOTAL** | 5896 | 4748 | 1188 |

Table 1: The overall data summary from AccentDB [1].

Each audio file is a single channel .wav file which is approximately 3.5 $\pm$ 0.35 seconds long. The sampling rate is 22050 Hz. In each audio file, the speaker talks about a particular sentence. There are 8 speakers for American with 742 audio files. However, there are only 2 speakers for other accents. Hence, to maintain the balance, I have only chosen 2 speakers with an American accent.

There would be 3 models in total. One which is trained only on clean data. One which is trained only on Noisy data, say Cafe at SNR = 10dB and SNR = -10dB. Finally, the last model is trained on both of these sets of data. To maintain the balance, each class has an equal number of audio files associated with them. The summary of this is given in Table 2. Finally, for every model, it will be tested on 2384 audio files which are noisy versions of the original test set.

| Model | Total Train Files |
|---|---|
| Clean | 4748 |
| Noisy | 9496 |
| Clean+Noisy | 14244 |

Table 2: Summary of files that will be used for training. The Noisy data will have 2 sections with SNR=10dB and SNR=-10dB.

Another kind of noise, which is white static noise was also considered later on to test the models. They were approximately 2.4k in total. The data flow can be seen in Figure 1.

## 4.2 Preprocessing

As described in the previous section, Mel Frequency Cepstrum Coefficients will be used as features for each and every audio file. MFCCs are the most commonly used features for speech recognition tasks. One of the main reasons for this is that MFCCs capture the vocal response of individuals. In different accents, certain phonemes and sounds are

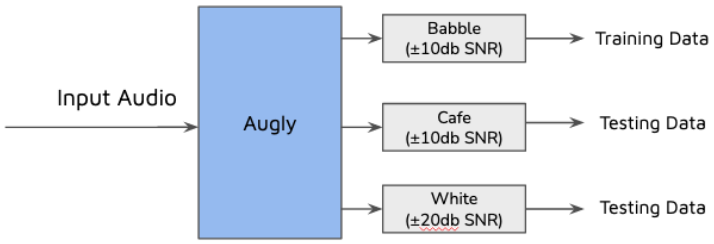


Figure 1: Augmentation Data Flow.

pronounced differently than others. MFCCs capture these responses in 13 coefficients. Hence, they can be considered for accent recognition as well.

For a given window, there are 13 coefficients. However, since the size of each audio file is uneven, it is important to trim down every audio file to a particular size so that the number of features for each file is the same. The minimum audio length of the dataset is 1.8s. Hence, all the audio files were trimmed down to 1.8s. Instead of taking the first 1.8s or the last 1.8s, the middle part was chosen since there were unnecessary pauses at the beginning and end of the audio files. Once these 1.8s are extracted, a series of LTI functions like Fourier Transforms, Inverse FTs, Logarithms, etc. are applied to obtain 13 coefficients in total. The sample heatmap for one of the audio files can be seen in Figure 2. Each feature map for an audio file has 78 distinct windows since the window size is 0.1s with a 25% overlap. Therefore, every audio file now has a feature image of dimension $78 \times 13$.

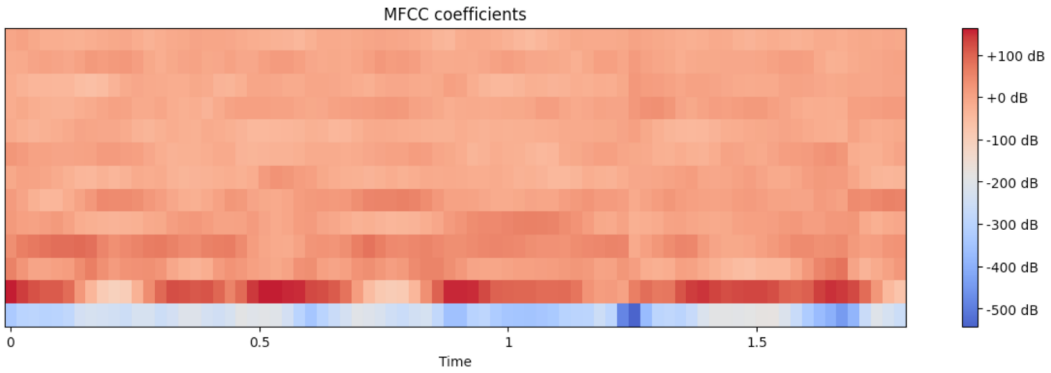


Figure 2: Sample MFCC HeatMap for an audio file

All the augmented and raw data can be downloaded from link.

# 5 Implementations

## 5.1 Model details

Every audio file after preprocessing has a $78 \times 13$ feature vector. A simple convolutional neural network can be used on these feature images followed by a fully connected layer that can classify the accent of the audio file. AccentDB [1] has suggested a ConvNet

which I will be using for the project. The authors claim a near 98% accuracy with that model on clean data. The architecture can be seen in Figure 4. The whole model pipeline can be seen in Figure 3.

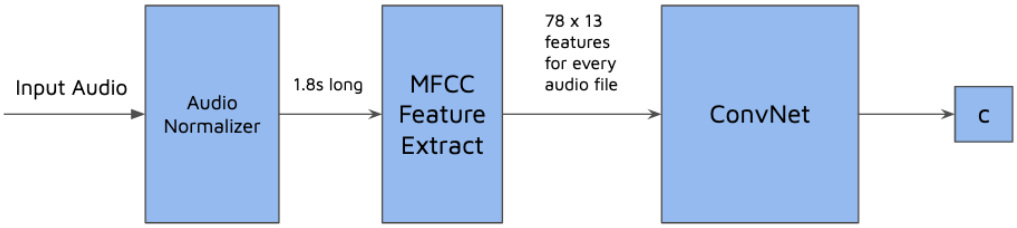


Figure 3: Pipeline for the project. Audio files are first normalized to have equal lengths and then a $78 \times 13$ MFCC features are extracted from each of them. The ConvNet is trained on these features to output a class.

| Layer (type) | Output Shape | Param # |
|---|---|---|
| Conv1d-1 | [-1, 32, 76] | 1,280 |
| MaxPool1d-2 | [-1, 32, 38] | 0 |
| Conv1d-3 | [-1, 32, 36] | 3,104 |
| MaxPool1d-4 | [-1, 32, 18] | 0 |
| Conv1d-5 | [-1, 64, 16] | 6,208 |
| MaxPool1d-6 | [-1, 64, 8] | 0 |
| Linear-7 | [-1, 128] | 65,664 |
| Dropout-8 | [-1, 128] | 0 |
| Linear-9 | [-1, 32] | 4,128 |
| Dropout-10 | [-1, 32] | 0 |
| Linear-11 | [-1, 4] | 132 |

Figure 4: Architecutre of ConvNet

The batch size chosen is 8, with a Cross Entropy Loss function. The optimizer used is an Adam Optimizer. All the other hyperparameters match that of AccentDB since ConvNet is essentially based on that. There are approximately 80k parameters that are trained. The model started giving desirable results right after 10-15 epochs.

## 5.2 Experiments

In total, there were 3 models trained. Each corresponds to a specific set of training data. They are summarized in Table 3.

| **Model Name** | **Traning Sets** |
|---|---|
| no_noise | clean data |
| only_babble | babble |
| clean_and_noise | clean+babble |

Table 3: Training sets for various models trained.

The training losses and accuracies accuracy progress can be seen in Figure **??**. You can see that the models have trained really quickly and accuracy reached close to 1.

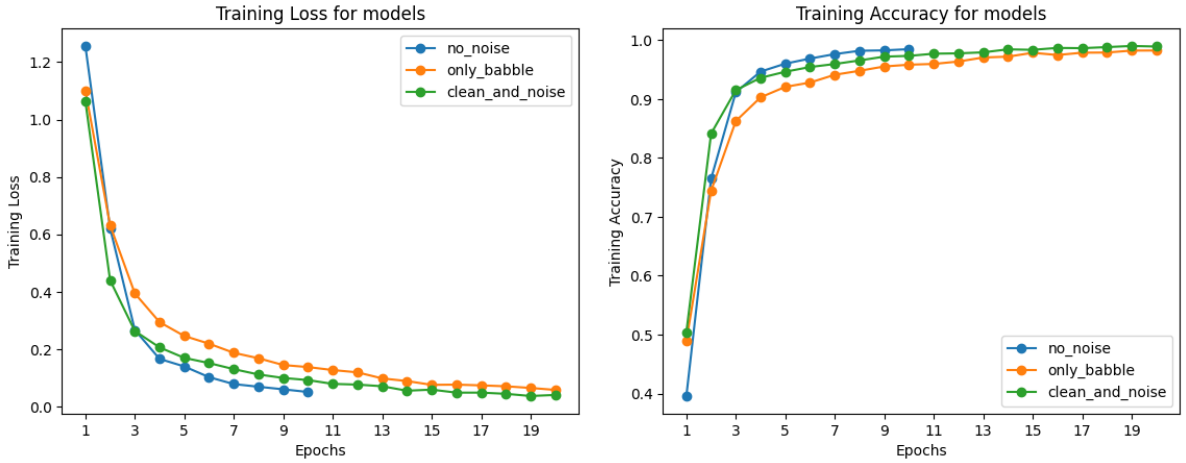


Figure 5: Starting left, the plots show the progress of loss and training accuracies with respect to accuracy.

The testing sets considered are clean testing set, which was already provided by AccentDB [1]. All the models were tested on various combinations (one at a time, two at a time, etc.) of these testing sets.

## 5.3 Results and Inferences

### 5.3.1 Results

As described earlier, all the models were tested on various training sets. Numerically all the results are summarized in Table 4.

| Model Name | Test Set | American | Australian | British | Indian | Overall | Std | Model Wise Std |
|---|---|---|---|---|---|---|---|---|
| no_noise (Baseline) | Clean | 99.66 | 99.32 | 99.66 | 99.32 | 99.49 | 0.20 | 40.95 |
| | Cafe Test | 15.99 | 100.00 | 0.00 | 0.00 | 29.00 | 47.93 | |
| | White Noise | 83.18 | 29.34 | 0.03 | 0.00 | 28.14 | 39.21 | |
| only_babble | Clean | 71.04 | 98.66 | 29.29 | 70.70 | 67.42 | 28.60 | 14.28 |
| | Cafe Test | 97.32 | 97.15 | 98.00 | 98.15 | 97.66 | 0.49 | |
| | Clean + Cafe | 89.89 | 99.32 | 76.43 | 90.23 | 88.97 | 9.43 | |
| | White Noise | 95.97 | 98.32 | 98.15 | 98.32 | 97.69 | 1.15 | |
| clean_and_noise (Proposed) | Cafe Test | 99.66 | 99.66 | 100.00 | 100.00 | 99.83 | 0.20 | 1.37 |
| | Clean + Cafe | 99.77 | 99.77 | 100.00 | 99.66 | 99.80 | 0.14 | |
| | White Noise | 96.74 | 96.66 | 97.45 | 97.45 | 97.08 | 0.43 | |
| | Clean + Cafe + White | 97.11 | 97.52 | 98.66 | 98.66 | 97.99 | 0.79 | |

Table 4: Results summary for each model for different combinations of testing sets.

### 5.3.2 Inferences

- The no_noise model when tested on clean test data yields close to 99%. This is in line with AccentDB's network [1]. It is safe to assume that the network implemented was almost the same.
- As expected, when the no_noise model was tested on noisy data, the accuracy was very poor. It was just 28%. The model couldn't even recognize a few classes.

- The only_babble model performed relatively better. However, when it was tested on clean data, the accuracy achieved is only 68%. However, it performed close to 98% when tested on noisy data. One possible reason could be that the model learned what the noise looks like and just filtered it out. When it did not find noise in clean data, it failed.

- The proposed model clean_and_noisy is trained on both clean and noisy data. This model performed the best for all the possible test combinations with an average accuracy close to 98% with a variation of only 1.4%.

Apart from the inferences presented above, there were some surprises. Accuracies close to 98% were not expected at all. The models were tested on white noise as well which wasn't part of initial plans. The reason for choosing this noise is to mimic bad microphones with terrible audio reception. The SNRs chosen were 20dB and -20dB. The model still worked fine to classify the accent.

In the initial proposal, I suggested the usage of text recognizers if this model fails. However, upon further thought this was not a viable method since the words spoken are same, but the way they are spoken is different. Hence, the textual features would've been the same for all the classes.

# 6 Conclusion

The idea of augmentation is a viable tool to expand and extrapolate the already clean datasets. Adding noises to these clean datasets can help in mimicking many real-life scenarios. Especially for audio files, most of the applications that are planned to be implemented in the wild do not use clean data for testing. Hence, training the models on noisy data can make them robust. To create noisy data, augmentation tools can be used. Again, models that are trained only on clean data cannot perform well in the wild since it has never seen noisy data. Hence, training models with noisy data will in general perform better.

# 7 Future Scope

The model was tested only on SNR up to ±20dB. However, the noise can further be increased and checked if the model works for higher noise. If the model fails to perform well for higher noise levels, then the model can be retrained with more noisy data. Eventually, the model should hopefully perform better. The model can be tested on various noise levels and then checked where the model would break. This can estimate the tolerance of the model. This can even estimate the amount of augmentation needed. The same methodology can be extended to more accents and languages if possible.

# References

[1] A. Ahamad, A. Anand, and P. Bhargava, "Accentdb: A database of non-native english accents to assist neural speech recognition," in *Proceedings of The 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 5351–5358. [Online]. Available: https://www.aclweb.org/anthology/2020.lrec-1.659 1, 4, 5, 6, 8

[2] C. Bartz *et al.*, *Language Identification Using Deep Convolutional Recurrent Neural Networks*. DOI, 2017. 4

[3] Z. Papakipos and J. Bitton, "Augly: Data augmentations for robustness," 2022. 4

[4] U. Shrawankar and V. M. Thakare, "Techniques for feature extraction in speech recognition system: A comparative study," *" ArXiv. Org*, vol. 6, May 2013. 4

[5] Y. Fares *et al.*, *"Arabic Dialect Identification with Deep Learning and Hybrid Frequency Based Features.* " Proceedings of the Fourth Arabic Natural Language Processing Workshop, Association for Computational Linguistics, 2019. 4

[6] A. Pendyala, "Speech emotion recognition," https://github.com/done-n-dusted/SpeechEmotionRecognition, 2021. 4

...