Audio-Visual Classification of Palatalization in British English

Gabriela Adams, Austin Brailey-Jones and Daniel Harper

Introduction to Artificial Intelligence
University of Georgia
Athens, GA 30602

Abstract

A feed forward neural network is tasked with classifying short utterances by phonetic variation. The training dataset was collected from the Audio British National Corpus representing speakers from across the British Isles, and it was annotated by specialists to produce a set of greater than 6000 gold standard tokens. The model is trained on two types of feature sets representing audio and visual exploration of palatalization in naturalistic speech. Due to the prescriptive limitations of current forced alignment techniques, neural networks offer an alternative methodology to classify acoustic signals. The models outlined in this work perform with at least 87% accuracy in testing to demonstrate the applicability of neural networks in this problem domain and the diverse sets of features that can be used to represent the data.

1. Introduction

Modern approaches to reduce the time commitment of phonetic segmentation have employed deep neural networks [9] and speech-to-text pipelines [5] to supply the orthographic transcriptions for the input into forced aligners. While this does drastically reduce the time investment of orthographic transcription, it does not solve the underlying issue that forced aligners can only represent what is listed in a provided dictionary. Common methods for audio classification include the use of Convolutional Neural Networks, but this methodology often transforms the audio waveform to its corresponding spectrogram to perform image classification. This process works for whole file classification, but it does not address whether neural networks can learn acoustic distinctions from the audio file which would be necessary for phonetic segmentation. The present work aims to explore the application of deep learning to acoustic data to test the success of a neural network on the classification of a waveform by phonetic variation.

Our presented models will attempt both audio and image classification to allow for the comparison of feature extraction methodologies for the classification of short utterances based on labels of phonetic variation. These models will share the same neural network architecture, but the feature inputs used for training the models will be changed for each classifier allowing the direct comparison of model performance based solely on feature extraction methods. The exploration of not only whether deep learning is suitable for phonetic classification, but also the comparison of different feature sets gives this work theoretical importance by providing benchmarks for future research.

2. Background

This section describes the background literature which informs the current project. Section 2.1 discusses the tasks of phonetic segmentation and acoustic classification using neural networks. Section 2.2 overviews the literature of visual classification of phonetic variation using neural networks.

2.1. Background - Phonetic Segmentation and Audio Classification

Research in the Linguistic fields concerned with Phonetics requires detailed annotation and segmentation of audio recordings to access the desired data. This is a time-consuming process which serves as a bottleneck for programs of phonetic study. As data continues to become more available and larger in scale, the bottleneck of audio segmentation only becomes more apparent. The field has appealed to automatic speech recognition to address the issue using tools such as forced alignment. Forced alignment systems process audio data along with an orthographic transcription of the audio file to return the word and phoneme boundaries of the input audio file. Linguists can drastically reduce the time required to manually annotate a sample set through use of these tools. Popular systems like the Montreal Forced Aligner [7], Dartmouth Linguistic Automation [11], and the Penn Forced Aligner [2] contain the same fundamental architectures relying on acoustic Gaussian Mixture Models (GMMS) and Hidden Markov Models (HMMs) of phone transitions to break the continuous acoustic signal into its constituent phonetic segments [4].

The chosen constituents parts, however, are not solely determined by the acoustic signal but in conjunction with the look-up table supplied to the system through the HMM dictionary. This is the forced nature of forced alignment. The possible phonetic labels for an input word are restricted to the HMM dictionary. If there is linguistic or acoustic variation in the audio input to the system that is not represented by the orthographic transcription or HMM dictionary, the output phonetic segmentation will not represent that variation [6]. This poses an issue for phonetic and sociolinguistic research as the true nature of the acoustic signal can be masked by misalignment. While still faster than manual annotation and segmentation, modern systems still require hand-checking and orthographic transcriptions. To explore this problem and further automate the task, neural networks have been developed to classify audio segments. However, these models tend to classify based on semantic or acoustic categories not concerned with specific phonetic variation [14]. While current work applies neural networks to speech problems [12], the question of model ability to determine phonetic variation from audio input remains an area of active research.

2.2. Background - Spectrographic Classification

The inclusion of a spectrographic image classification comes from both the common use of image transformations for audio classification, and from recent work exploring its application to phonetic classification. This paper discusses the disadvantages of facial recognition for speech analysis, instead opting for spectrogram analysis [13]. Using a spectrogram allows for avoiding differences between some accents and different facial patterns by different speakers. Spectrograms plot the amplitude and frequency of sounds over time through colors. By analyzing the colors corresponding to frequency and amplitude, you can detect specific phonetic landmarks such the high frequency frication seen in palatalized /t/. The *imread* function, implemented in this program, uses three RGB values to gather the color value at each pixel of a spectrogram image.

3. Experimental Study

This section presents the models as built, trained, and tested on the Audio BNC palatalization dataset. Section 3.1 overviews the general pipeline from data setup to classification which is shared by both models. Further, each model has a distinct feature extraction methodology which is individually outlined in sections 3.1.1 and 3.2.1. Section 3.2 presents the image classification model, and lastly, 3.3 presents the overall feed forward model architecture used for both classifiers.

3.1. Audio Classification

The audio classifier is built using the *pytorch* [10] and *pandas* [8] packages for python 3.7. All utilities for feature extraction and dataset preparation come from the *pytorch* utilities. An overview of the model pipeline is as follows. Audio files are loaded into a dataset alongside their corresponding labels. The dataset is split into a testing and training sets of 80% for training and 20% for testing. Once split, the audio files are pre-processed for the model to standardize the input. These uniform files are passed to a feature extractor which extracts Mel-Frequency Cepstral Coefficients using standard practice for ASR systems which will be outlined below. The features are input into a 3-layer feed forward network for classification. A training loop uses cross entropy loss for the evaluation criterion. The remaining 20% of the data is passed through the model using a testing loop to evaluate model performance.

3.1.1. Feature Extraction

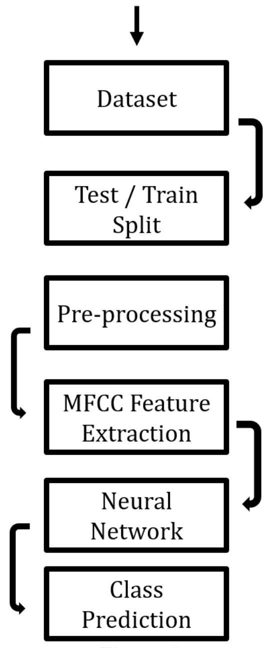


Figure 1

Each signal underwent a preprocessing step to standardize the duration of each file. The dataset contained tokens ranging from a minimum of 100ms to a maximum of approximately 7 seconds all with a sampling rate of 16000 Hz. An arbitrary cutoff of 700ms was chosen for each file. Files shorter than 700ms were padded with zeros and files longer than 700ms were cut off at the 700ms mark. While crude, this method ensured features arrays maintained a uniform size. Given that few tokens were longer than 700ms with the phone of interest appearing in the first word, training data loss in tokens greater than 700ms is minimal.

After standardization, the tokens are passed to an MFCC extraction method from the *pytorch torchaudio kaldi* utility. Pre-emphasis was performed on audio files to boost energy in high frequencies. The signal was then windowed with a frame length of 25ms and a frame shift of 10ms. The Hamming window was selected to lessen noise generated from the windowing process. A Discrete Fourier Transform was applied to each Hamming window to extract frequency information. The DFT windows next were passed through a Mel filter bank with 23 bins and the output logarithmically transformed. From the log-Mel filter bank windows, the cepstral representation of the signal is computed. Lastly, the first 13 coefficients are taken from the cepstral representation to produce the final MFCC features for each window. These 13 features are concatenated for each 25ms window to produce the feature array. The feature array is then flattened for input into the neural network. Figure 1 shows the model pipeline.

3.2. Image Classification

In order to make the audio and image classifiers as compatible as possible, they were built with essentially the same pipeline. The image classifier is built using the *pytorch* [9], *pandas* [8], *matplotlib* [3], and *scikit-image* [15] packages for python 3.7. All utilities for feature extraction and dataset preparation come from the *pytorch* utilities. An overview of the model pipeline is as follows. The directory of audio files are sent through a loop with the *pyplot* function *specgram*. This spectrogram is saved as a Portable Network Graphics file. Image files are then loaded into a dataset alongside their corresponding labels. The dataset is split into a testing and training sets of 80% for training and 20% for testing. Once split, the image files are pre-processed for the model to standardize the input. These uniform files are passed to the *imread* function, which returns a 3-dimensional array of pixel colors for the image before being flattened. These flattened features are input into a 3-layer feed forward network for classification. A training loop uses cross entropy loss for the evaluation criterion. The remaining 20% of the data is passed through the model using a testing loop to evaluate model performance.

3.2.1. Feature Extraction

Each signal underwent a preprocessing step to standardize the duration of each file. The dataset contained tokens ranging from a minimum of 100ms to a maximum of approximately 7 seconds all with a sampling rate of 16000 Hz. An arbitrary cutoff of 700ms was chosen for each file. Similar to the audio standardization, files longer than 700ms were trimmed to this length, while files shorter than 700ms had zeros (representing no noise) added to reach the desired cutoff length. For spectrograms, there was no effect caused by this as longer files simply appeared as if they were 700ms and shorter files had extra whitespace on the right side of spectrograms. This whitespace had no effect because image feature extraction is based on RGB values per pixel and white will not have an effect on these values.

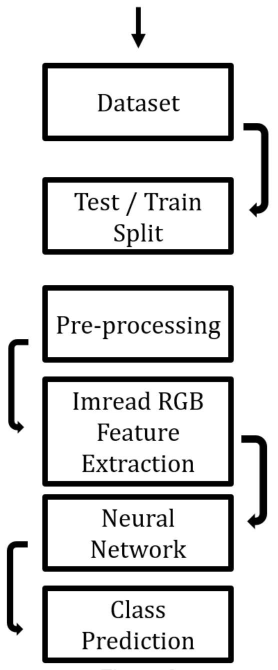


Figure 2

After standardization, the tokens are passed to an image reading function from the *scikit-image* package. This package takes an image as input and outputs a three dimensional array of size MxNx3. The M and N are the width and height of the spectrograph image and the 3 represents the three colors used in the color scale: red, green, and blue. The output of this function is an array from the *numpy* package holding the color values of red, blue, and green within each pixel of the image. This final feature array is flattened for simpler dimensions before being passed as input into the neural network. Figure 2 above shows the model pipeline for image feature extraction and classification.

3.3. Model Architecture

Figure 3 to the right outlines the model architecture. A feedforward network was created using *torch.nn* linear layers. Two hidden layers were used due to better performance compared to one hidden layer. Input dimensions are set by the size of the input vector for each feature set (audio or image); however, the hidden dimensions were kept consistent between each feature set to ensure maximal compatibility. Hidden layer 1 had input dimensions of 512. Hidden layer 2 had input dimensions of 100. A rectified linear unit layer was used as the activation function between each hidden layer. Lastly, the output had dimensions of 2 to represent the two classes palatalized or not palatalized. The learning rate for both models was set at 0.001. In training, *pytorch*'s cross entropy loss function was used as the evaluation criterion. Due to this, no SoftMax function was used after the output layer [16]. The formula for *pytorch's* cross entropy loss calculation is given below. It is worth noting that in this package, a softmax calculation is applied within the *nn.CrossEntropyLoss()* loss criterion where *x* is the input, ***y*** is the target, ***w*** is the weight, and N spans the minibatch dimension as well as $\mathbf{d}_1,\ldots,\mathbf{d}_\mathbf{k}$ for the K-dimensional case.

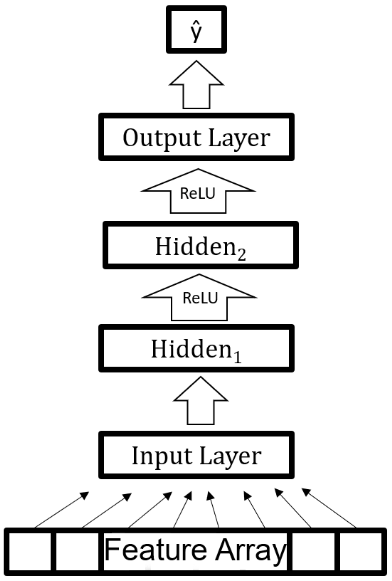


Figure 3

3.3.1. Loss Function

$$l(x,y) = \begin{cases} \sum_{n=1}^{N} \frac{1}{\sum_{n=1}^{N} w_{y_n} \cdot 1\{y_n \neq \text{ignore\_index}\}} l_{n,} & \text{if reduction} = \text{'mean'}; \\ \sum_{n=1}^{N} l_n & \text{if reducuction} = \text{'sum'}. \end{cases}$$

Figure 4

4. Results

With each model and feature set detailed above, the performance of each model will be compared in section 4. Firstly, section 4.1 presents a table of model accuracy and training loss over 50 epochs. 4.2 shows the training loss and accuracy curve over time to allow comparison of feature sets and their ease of fitting for each model.

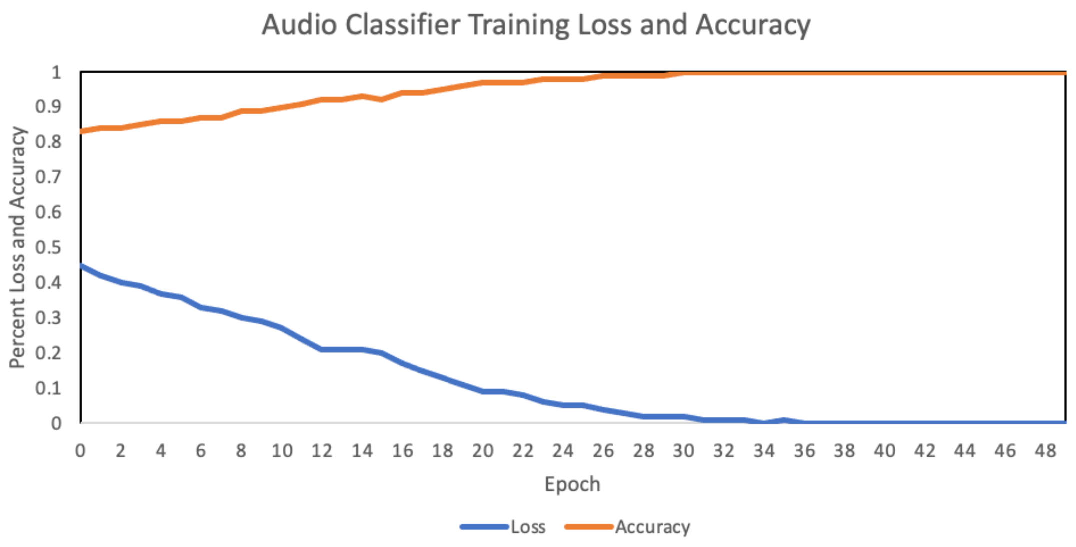
4.1 - Data

Classifier	**Training Min Loss / Max Loss**	**Training Min Accuracy / Max Accuracy**	**Average Training Loss**	**Average Training Accuracy**	**Test Accuracy**
Audio Classifier	0% / 45%	83% / 100%	12%	95%	88%
Image Classifier	22% / 52%	81% / 92%	34%	87%	86%

4.2 - Graphs

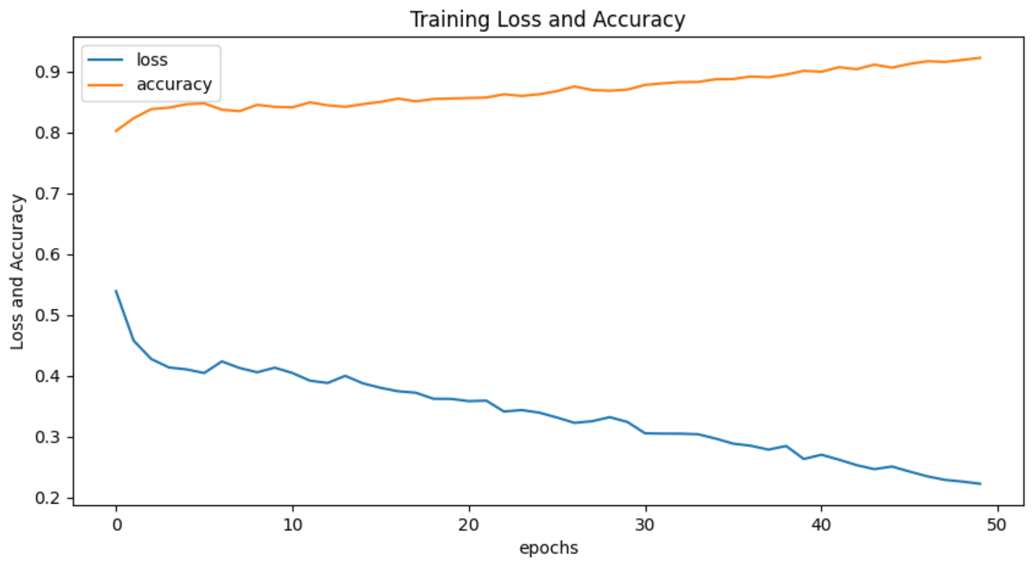
4.2.1 - Audio Features Training Graph

Figure 4.2.1 below shows the training of the network on the extracted MFCC acoustic features. By epoch 34, the training loss achieved 0 and the accuracy achieved 100%.



4.2.2 - Image Features Training Graph

Figure 4.2.2 below shows the training of the network on the extracted Imread RGB image features. By epoch fifty, the training loop achieved 22% loss and 92% accuracy.



5. Conclusion

This section presents the data analysis presented in 5.1, what the designed next steps are, 5.2 Future Works and what relevance this information holds.

5.1. Data Analysis

The feed forward neural network is tasked with classifying short utterances by phonetic variation. To reiterate, the training dataset is above 6,000 gold standard tokens that were collected from the Audio British National Corpus (British Isles), and hand-checked by specialists. These neural networks offered an alternative methodology to classify and segment acoustic signals (a.k.a. The exploration of palatalization in naturalistic speech.). The models outlined in this work perform with greater than 86% accuracy in testing.

Analyzing the table under section 4.1 Data, the conclusion is clear that still audio classifiers outperform image classifiers when it comes to training on feature extraction methodologies for the classification of short utterances based on labels of phonetic variation. This is what is determined from our trials. In every metric measured, the audio classifier outperformed. Keep in mind, the "Test Accuracy" metric was close for the image classifier by being only one percent lower. "Average Training Accuracy", audio classifier outperformed with 8 percent higher at a solid 95% and for the "Average Training Loss" image classifier has the higher scoring at 34% , which was approximately 1.4 times the amount of the audio classifier at 12%. This work has demonstrated the applicability of neural networks in this problem domain and the diverse sets of features that can be used to represent the data. Differences also noted of these sensory features is the amount of time it took for the inputs to train in its respective sense. The audio classifier took approximately 20 to 30 minutes in comparison to the image classifier that trained for 2 and 15 minutes to 2 hours and 30 minutes.

5.2. Future Works

Further, the aim is to utilize this project to explore the feasibility of combining both the audio and visual feature sets for the richest input possible to a model for any possible boost in performance. The assumption is that this will be done with using the *tf.concat* function, as the method implies, concating both tensor models in order to achieve a higher performance score. Also noting that since the tensors on their own took either approximately 2 hours and 30 minutes or 30 minutes to train, it is safe to assume the combined tensor will take more than the 2 hours and 30 minutes to train. The idea behind this is that, the “Test Accuracy” metric showed both the audio and visual classifiers held a decent score of 88% and 87% respectively, but could have been each scoring high in classifying different acoustic distinctions. Since these two models share the same neural network architecture, and the feature inputs used for training the models are changed for each classifier, based solely on feature extraction methods, the probability of a boost in performance is extremely likely.

5.3. Relevance

What can be used with the data derived and the newly created tensors, is from a business aspect, companies could be able to incorporate these classifiers into their software. Ideas of softwares that can analyze recorded interviews, meetings, and for their clients ease of communication. Creating closed captions on videos that are not readily provided help with accessibilities and creates a welcoming environment that allows for more benefits with less drawbacks. An example of an application that would do well is WhatsApp, or any other phone service, that allows for users to send voice messages and videos. If there is an option to opt for the transcript on videos / voice notes, the availability of people would improve and correspondence would increase due to the new feature offered. The second most beneficial use is in the context of live online interviews. Now that we have switched to a heavier reliance on technology, having the ability to use the combination of audio and video classifiers will substantially impact and improve the culture of computer mediated discourse. This revelation is what drove us to investigate if deep learning is suitable for phonetic classification, but also the comparison of different feature sets. All in all aiming to provide benchmarks for future research.

6. References

1. BNC Consortium. (2007). British national corpus. Oxford Text Archive Core Collection.
2. Gorman, K., Howell, J., & Wagner, M. (2011). Prosodylab-aligner: A tool for forced alignment of laboratory speech. Canadian Acoustics, 39, 192.

3. Hunter, J.D. (2007) Matplotlib: A 2D Graphics Environment, Computing in Science & Engineering, 9(3), 90-95. https://matplotlib.org
4. Jurafsky, D., & Martin, J. H. (2021, Unpublished). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. (Ed. 3).
5. Kim, J. Y., Liu, C., Calvo, R. A., McCabe, K., Taylor, S. C., Schuller, B. W., & Wu, K. (2019). A comparison of online automatic speech recognition systems and the nonverbal responses to unintelligible speech. arXiv preprint arXiv:1904.12403.
6. Lipani, L. (2021) Subphonemic Variation in English Stops: Studies using automated methods and large-scale data (Doctoral dissertation, University of Georgia).
7. McAuliffe, M., Socolof, M., Mihuc, S., Wagner, M., & Sonderegger, M. (2017, August). Montreal Forced Aligner: Trainable Text-Speech Alignment Using Kaldi. In Interspeech (Vol. 2017, pp. 498-502).
8. McKinney, Wes. (2015) "Pandas, python data analysis library." URL http://pandas. pydata. org .
9. Moritz, N., Hori, T., & Le, J. (2020, May). Streaming automatic speech recognition with the transformer model. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 6074-6078). IEEE.
10. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems, 32.
11. Reddy, S., & Stanford, J. N. (2015). Toward completely automated vowel extraction: Introducing DARLA. Linguistics Vanguard, 1(1), 15-28.
12. Rejaibi, E., Komaty, A., Meriaudeau, F., Agrebi, S., & Othmani, A. (2022). MFCC-based recurrent neural network for automatic clinical depression recognition and assessment from speech. Biomedical Signal Processing and Control, 71, 103107.
13. Shah, V. H., & Chandra, M. (2020). Speech Recognition Using Spectrogram-Based Visual Features. Algorithms for Intelligent Systems, 695–704.
14. Soares, B. S., Luz, J. S., de Macêdo, V. F., e Silva, R. R. V., de Araújo, F. H. D., & Magalhães, D. M. V. (2022). MFCC-based descriptor for bee queen presence detection. Expert Systems with Applications, 117104.
15. Van der Walt, S., et al. (2014). scikit-image: image processing in Python. PeerJ, 2, 453. https://scikit-image.org
16. Wadekar, S. (2021, January 3). Why Softmax not used when cross-entropy-loss is used as loss function during neural network... Medium. Retrieved May 1, 2022, from https://shaktiwadekar.medium.com/why-softmax-not-used-when-cross-entropy-loss-is-used-as-loss-function-during-neural-network-d77abd708715

7. Team Member Contributions

Member	**Audio Classifier**	**Image Classifier**	**Audio Classifier Write-up**	**Image Classifier Write-up**	**Conclusion Write-up / Editing**	**Future Works Write-up**

Gabriela Adams					X	X
Austin Brailey-Jones	X		X			
Daniel Harper		X		X		