Accent Detection Within the Amateur Singing Voice

Camille Noufi, Sarah Ciresi, Vidya Rangasayee
Stanford University
Stanford, CA
[cnoufi],[sciresi],[rvidya]@stanford.edu

Abstract

In this paper, we investigate the feasibility of detecting innate speech characteristics, namely characteristics of accent, still present during solo singing by using singer-provided country and language as a proxy. We investigate variants of convolutional neural networks to classify the associated country and language of both native and non-native English speakers during their karaoke-style singing performance of English standard "Amazing Grace." The most successful architecture provides an 7.83% improvement in overall accuracy compared our baseline statistical model, demonstrating the networks' ability to detect subtle accent-based speech characteristics. We see an overwhelming prediction of variants of English accent, suggesting style-influenced modification of pronunciation and intonation when singing a well-known English song.

1 Introduction

The past several years have seen increasing performance in speaker recognition and accent detection from spontaneous speech when utilizing deep neural networks. These neural models have moved toward using minimal input preprocessing in order to allow complex feature learning to occur within the layers of the networks. Convolutional architectures have been moderately successful in identifying articulatory and intonation features, and more so in identifying features representing tone in speech and music [1-3,5] (see Section 2.2 for more detail).

In this paper, we implement two variants of deep convolutional neural network architectures (CNNs) to classify ten different accents. Both architectures attempt to detect accent (via proxy "country-language") by learning both long and short term speech characteristics from solo-singing audio undergoing minimal feature pre-processing.[1] We design our models around a large and novel dataset of people from ten different countries singing "Amazing Grace." We analyze the per-class performance of each model to understand if speech characteristics indicative of accent are still present during singing and, if so, how they are modified or confounded.

The paper is organized as follows. Section 2 details our proposed feature and model architecture, and Section 3 provides experimental setup and procedures. In Section 4 we analyze the classification results of each model and discuss the learned speech features and how they influenced classification. Section 5 concludes our work.

[1] Class labels are hereafter referred to synonymously as "country" or "accent."

Preprint. Work in progress.

Table 1: Ten Most-Represented Accent Classes within DAMP-Amazing Grace

Accent Label	Language (Country)	Accent Label	Language (Country)
de-DE	German (Germany)	fr-FR	French (France)
en-AU	English (Australian)	id-ID	Indonesian (Indonesia)
en-CA	English(Canadian)	nb-NO	Norwegian (Norway)
en-GB	English (Great Britain)	pt-BR	Portuguese (Brazil)
en-US	English (United States)	sv-SE	Swedish (Sweden)

2 Approach

2.1 Dataset Selection

We use a subset of the the Stanford DAMP Database, a proprietary database owned by Smule, developer of the Sing! Karaoke smartphone app.[2] Because of its recent release and proprietary nature, only two pieces of existing research have utilized this dataset. Additionally, it is the largest dataset to date of uniform solo singing recordings. We use a portion of the database consisting of approximately 17,000 raw audio recordings of both trained and untrained singers from around the world singing "Amazing Grace" on the app. The karaoke accompaniment is played at the same key and tempo to all users. The recordings are captured via the singers' smartphone microphone. The files are monophonic, single-channel, and do not contain any preprocessing or compression beyond what the users' headphones and smartphone applies. Along with each recording, the dataset contains metadata information provided by the user. We use the provided "country-language" label for our classification. In this paper, we focus on learning and classifying the ten most represented countries in the dataset. We do this to ensure ample training data per class, as many of the countries in the complete dataset only have one or two recordings. Table 1 details our dataset.

2.2 Input Feature Selection

Recent work in accent classification has shown that learning from both long-term and short-term speech features leads to significant improvement in performance [4]. These long- and short-term features are those that describe articulation (the way the vocal tract articulators like the jaw, tongue and lips shape to form certain sounds), prosody and intonation (the time-domain patterns of articulation). Both articulation and intonation are related to the property of sound known as timbre. Timbre is often described as the "color," "quality," or "tone" of a sound [6]. It is quantitatively analyzed via proxy measurements of the spectral envelope shape, and change in timbre is often measure via the proxy of spectral content variation over time [7]. Therefore, we assume that timbre is a property reliant on both time and frequency, and select magnitude spectrograms as input to our model. Taking note from much of the literature [1-7], we discard phase information. Specifically, we calculate the mel-scale magnitude spectrogram of each audio recording. Although we take steps to prevent the model learning pitch-dependent weights by using the same melody across all recordings, we want to further dissociate accent detection from sung pitch by utilizing a pitch invariant input. Mel spectrograms use a perception-based frequency scaling, and often lead to better timbre- and temporally-based classification over other kinds of spectrograms such as the short-time-Fourier-transform or constant-Q transform [1-3,5]. The conversion from frequency to mels is defined in Equation 1,

$$m = 1127 \ln(1 + f/700) \tag{1}$$

where m is the mel coefficient corresponding to the original frequency f in Hz.

2.3 Model Architecture

We design two variants of convolutional neural networks: a deep vanilla network and a ResNeXt block network [8], depicted in Figure 1. Our classification task is novel and therefore requires us to additionally train a comparative statistical baseline model. All model architectures are hereafter described in detail.

[2] https://ccrma.stanford.edu/damp/

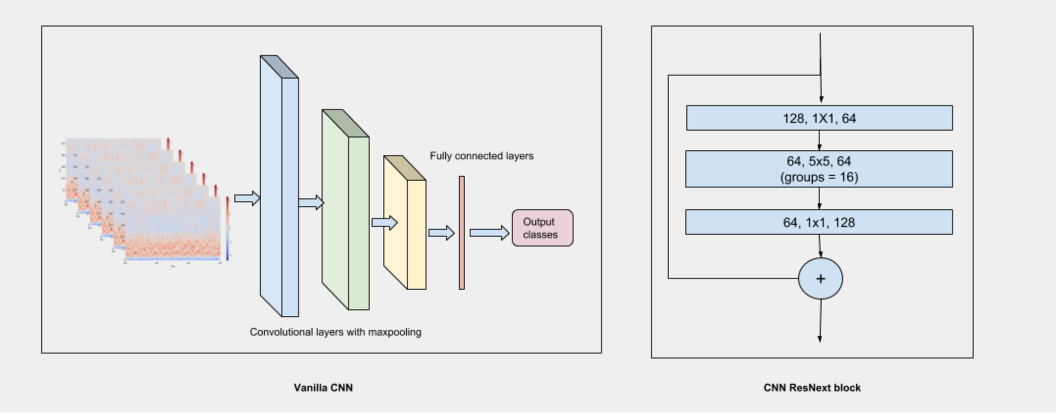


Figure 1: ResNeXt Convolutional Neural Network Architecture

2.3.1 Vanilla CNN

We design a vanilla CNN with three convolutional blocks followed by two fully-connected feed-forward layers. Each convolutional block contains a 2D convolutional layer with a ReLU nonlinear activation followed by max pooling. The first 2D layer uses 1 input channel, a kernel size of (3, 3), stride of (1, 1) and padding (1, 1) and 64 output channels. Subsequent 2D layers use identical kernel size, stride, and padding, but contain 64 channels for both input and output. The max pooling layer has kernel size (2, 2) and stride of (2, 2). A dropout layer follows the three convolutional blocks to prevent overfitting of the model [9]. The final output layer consists of 10 nodes corresponding to the 10 possible classes.

2.3.2 CNN with ResNeXt Block

We design our architecture similar to [1]. The first layer is a vanilla convolutional layer with 32 output channels, kernel size of (10,10) and strides (1, 1), followed by a max pooling layer with kernel size (2, 2) and strides (2, 2). A ResNeXt convolutional block follows this max pooling layer. The first convolutional layer inside the block has 64 output channels with (1, 1) kernel. The middle layer is a grouped convolution layer that slices the input on the channel axis according to the cardinality parameters. The convolutional layer is applied with 16 channels of kernel size of (5, 5) and strides (1, 1). A final convolutional layer inside the ResNeXt block takes the concatenated layer (64 channels) and applies a 128 channel convolutional layer with kernel size (1, 1). The skipped connection from the input to the ResNeXt block is added to the output of the last layer inside the block. Another max pooling layer having kernel size (2, 2) and strides (2, 2) is applied after the ResNeXt block. This is followed by the final fully connected layer with 10 nodes corresponding to the 10 classes. Dropout is applied at this last fully connected dense layer to prevent overfitting. The ReLU non-linear activation function is used in all convolutional and fully connected layers.

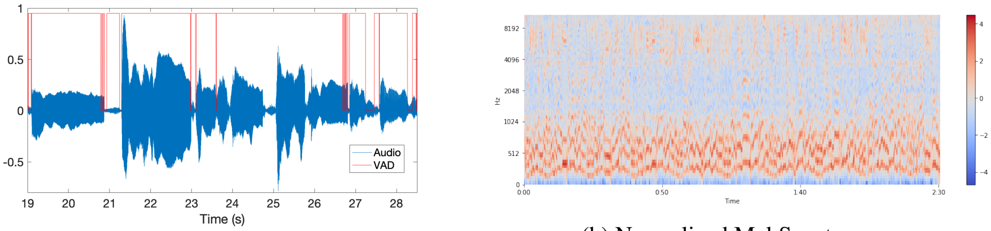
2.4 Baseline

We use a k-nearest neighbors (KNN) clustering-based classification model in order to gauge the accuracy and efficiency of our proposed convolutional models. For this baseline, we use scikit-learn's KNeighborsClassifier multi-class classification algorithm with $k = 10$ and uniform weighting. Mel-spectrograms sliced along the time-domain into vectors are used as input samples to reduce dimensionality and make each sample time-invariant.

3 Experiments

3.1 Preprocessing and Feature Extraction

We refine the DAMP Amazing Grace dataset to include only monaural recordings sampled at 22,050 Hz. Additionally, we want to perform analysis on solo singing only, so we discard all audio files that



(a) Voice Activity Detection (0.95 = voiced)

(b) Normalized Mel Spectrogram

Figure 2: Feature Extraction of Singer Recording

include any background accompaniment or noise having greater than 4 percent of the root-mean-square energy of the sung portions of the recording. This brings our dataset to 10,937 recordings. After refining our dataset to the top ten most-represented countries, we divide the dataset into training, validation and test sets, split at 80/10/10 respectively. This dataset is still very imbalanced, with approximately 75% of the data labeled as US-English. We balanced our training set by undersampling our two majority classes (US-English and UK-English) to contain the same number of recordings as the remaining eight classes. This brings our training set down to 1015 audio recordings, with an average of 100 recordings per class. The validation and test sets are kept as representative (and therefore imbalanced) partitions.

Voice activity detection is performed on all recordings.[3] Frames estimated to have speech energy with 0.95 confidence or more are labeled as voiced and concatenated together, discarding silences in the audio file. This is done to ensure minimal amounts of noise or silence as inputs into our models. Mel-scale magnitude spectrograms are are computed on the voiced audio, using a frame size of 2048 samples (with zero-padding) and hop size of 512 samples. The magnitudes are squared to obtain the power spectrum, logarithmic compression is applied, and the spectrograms are normalized to zero mean and unit variance[3]. Figure 2 displays the voice activity detection and normalized mel spectrogram of a singer's recording in the training set. We slice each mel-spectrogram into 4.75 seconds chunks (204 frames/chunk, 2 musical measures/chunk) with 50% overlap. Each chunk consists of 80 mel-scaled frequency bins. These spectrogram slices are used as the two-dimensional inputs into our neural models, with the input dimensions being M = 80, N = 204.[4] The "truth" accent associated with each recording is used as the "truth" accent for each spectrogram slice.

3.2 Model Training and Evaluation

The dropout rate, learning rate, and batch size are tuned specifically to each model based on validation performance. Table 2 lists the selected training hyperparameters. For all models, batch normalization is applied before each activation function in the convolutional blocks. L2 weight regularizations with a weight 1e-6 are applied on all learnable weights. We use cross entropy loss with ADAM optimization during training. Overall classification accuracy, F1, precision and recall are selected as evaluation metrics and confusion matrices are computed for analysis.

The models are trained using the training set over 40 epochs with early stopping. Initially the models suffered overfitting when training over all 40 epochs. A validation set of 20% and a patience value of 7 is used to determine early stop. At the end of every epoch, the validation loss is calculated and if the loss is better than an earlier epoch, the model is updated. The patience threshold is applied to any epoch that results in a worse loss than the preceding epoch. When the patience threshold is breached, training is terminated and the best model learnt thus far is returned. This is applied to reduce overfitting and results in slightly improved accuracy.

[3] http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html

[4] Throughout this paper, M and m stand for number of frequency bins, and N and n stand for number of time frames. M and N are the dimensions of the input spectrograms, while m and n are the convolutional filter dimensions.

Table 2: Classification Performance. (p = Dropout Rate, lr = Learning Rate, b = Batch Size)

Architecture	# Params	Accuracy	F1	Precision	Recall	p	lr	b
CNN-3x3	11371792	11.03%	15.93%	57.63%	11.03%	0.5	0.001	128
CNN-ResNeXt	11302160	15.64%	22.41%	56.31%	15.64%	0.3	0.001	128
KNN	10	7.81%	11.84%	60.89%	7.81%			

4 Discussion

4.1 Classification Analysis

Table 2 summarizes metric-based model performance on our test set. Overall classification accuracy scores are at the upper limit of the results found in [1] using the same feature input. The overall accuracy is a few percentage points above random (10%) on our test set, consistent with performance results on our validation sets. This indicates the system is learning, but not in enough detail to distinguish accurately between the ten classes. Precision scores were similar for the CNN-3x3 and CNN-ResNeXt, at 57.63 and 56.31, respectively, suggesting that both models identified almost as many false positives as true positives amongst all classes. Both models performed worse in terms of recall, indicating a low probability of correctly identifying all true positives amongst each class. We note that the CNN-ResNeXt model did achieve about 4.6% higher recall than the CNN-3x3 model, contributing to a higher weighted F1 score as well, at 22.41% for the CNN-ResNeXt compared to 15.93% for the CNN-3x3.

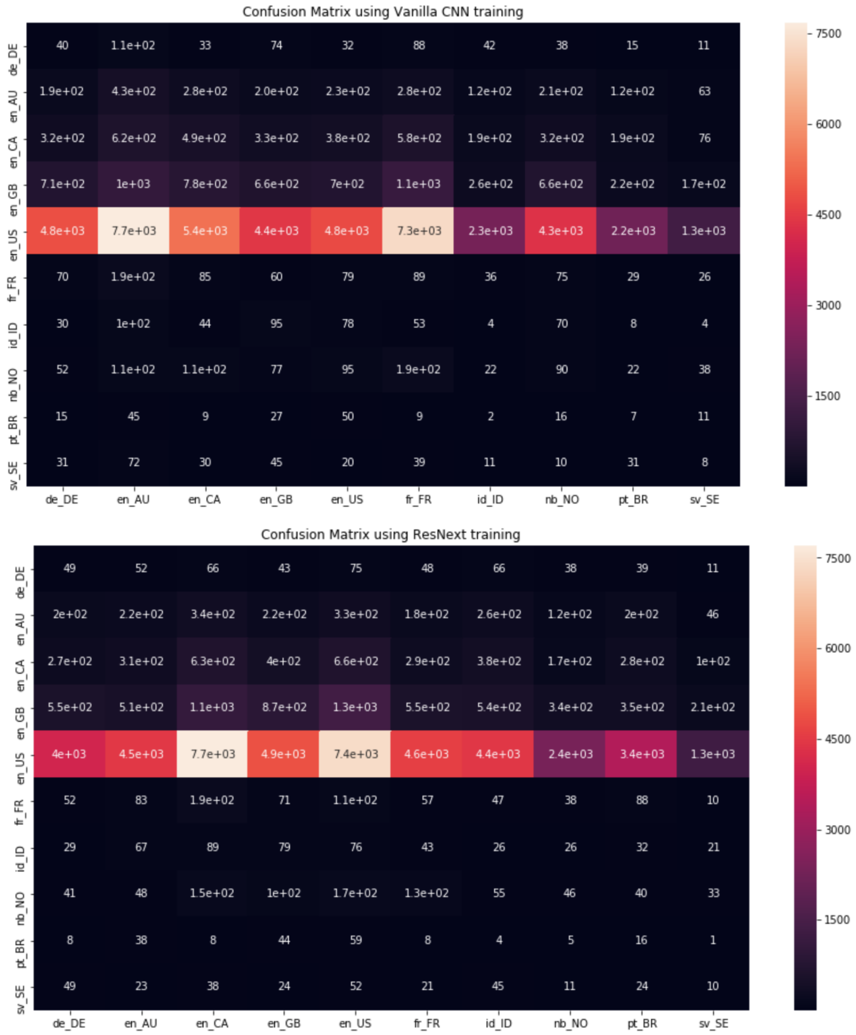
The resulting confusion matrices of test set classification shown in Figure 3 provide a more useful window into how the network has learned the input singing features. We see that our test set is imbalanced, with the European and North American countries providing a majority of the samples. However, amongst the imbalanced samples, we see some common misclassification trends.

Not surprisingly, we see that the network has difficulty distinguishing between very similar labels such as Canadian English and American English. The ResNeXt model is able to classify American English and Canadian English the most accurately. Further, misclassifications as English across all classes are primarily misclassified as Canadian English or American English. These self- and cross-classifications are correct 34.1% of the time, over twice the overall model accuracy. The Scandanavian and French singers are most likely to be misclassified as English singers. Both models misclassify Norwegian singers as English (with slightly different country variants), followed by French. Surprisingly, Swedish singers are not classified similarly. Both models seem to have more difficulty associating Swedish singing with any particular class. The models perform similarly for German and Indonesian, the single non-European/non-American accent in the dataset.

The overall low accuracy of both models, alongside the baseline, are indicators of two systematic issues. The first being that the country-language labels might not be the best proxy for accent. The labels are user-provided and do not guarantee that they will embody their innate day-to-day speech accent in their singing performances. Further, singing a well-known, traditional English song that has been stylized in many genres could confound and modify pronunciation and intonation. The mel-spectrograms are indeed pitch-invariant, but other aspects of musicality, including skill-level and transient stylistic choices that may be completely independent of country, accent or language identity, might outweigh speech-like pronunciation. The over-classification of English dialect even using a balanced training set may suggest that an English-style pronunciation is used to fit the target language of the song.

4.2 Future Work

Future work to separate confounding speech styling would help to further identify accent-specific pronunciation in singing. One route could be to apply phoneme-level segmentation, extract a collection of specific phones from each file, and attempt to discriminate between pronunciation of a single phone's pronunciation. We began investigation in implementing a stacked CNN-RNN in order to capture time-varying nuance, but were not able to make training efficient enough to aquire results thus far. Modifying this network to train efficiently would provide us insight into the usefulness of time-influenced intonation in detecting accent.



Confusion Matrix using Vanilla CNN training

	de_DE	en_AU	en_CA	en_GB	en_US	fr_FR	id_ID	nb_NO	pt_BR	sv_SE
de_DE	40	1.1e+02	33	74	32	88	42	38	15	11
en_AU	1.9e+02	4.3e+02	2.8e+02	2.0e+02	2.3e+02	2.8e+02	1.2e+02	2.1e+02	1.2e+02	63
en_CA	3.2e+02	6.2e+02	4.9e+02	3.3e+02	3.8e+02	5.8e+02	1.9e+02	3.2e+02	1.9e+02	76
en_GB	7.1e+02	1e+03	7.8e+02	6.6e+02	7e+02	1.1e+03	2.6e+02	6.6e+02	2.2e+02	1.7e+02
en_US	4.8e+03	7.7e+03	5.4e+03	4.4e+03	4.8e+03	7.3e+03	2.3e+03	4.3e+03	2.2e+03	1.3e+03
fr_FR	70	1.9e+02	85	60	79	89	36	75	29	26
id_ID	30	1e+02	44	95	78	53	4	70	8	4
nb_NO	52	1.1e+02	1.1e+02	77	95	1.9e+02	22	90	22	38
pt_BR	15	45	9	27	50	9	2	16	7	11
sv_SE	31	72	30	45	20	39	11	10	31	8

Confusion Matrix using ResNext training

	de_DE	en_AU	en_CA	en_GB	en_US	fr_FR	id_ID	nb_NO	pt_BR	sv_SE
de_DE	49	52	66	43	75	48	66	38	39	11
en_AU	2e+02	2.2e+02	3.4e+02	2.2e+02	3.3e+02	1.8e+02	2.6e+02	1.2e+02	2e+02	46
en_CA	2.7e+02	3.1e+02	6.3e+02	4e+02	6.6e+02	2.9e+02	3.8e+02	1.7e+02	2.8e+02	1e+02
en_GB	5.5e+02	5.1e+02	1.1e+03	8.7e+02	1.3e+03	5.5e+02	5.4e+02	3.4e+02	3.5e+02	2.1e+02
en_US	4e+03	4.5e+03	7.7e+03	4.9e+03	7.4e+03	4.6e+03	4.4e+03	2.4e+03	3.4e+03	1.3e+03
fr_FR	52	83	1.9e+02	71	1.1e+02	57	47	38	88	10
id_ID	29	67	89	79	76	43	26	26	32	21
nb_NO	41	48	1.5e+02	1e+02	1.7e+02	1.3e+02	55	46	40	33
pt_BR	8	38	8	44	59	8	4	5	16	1
sv_SE	49	23	38	24	52	21	45	11	24	10

Figure 3: Confusion Matrices of CNN-3x3 (top) and CNN-ResNeXt (bottom) Classification Results

We are also in the process of expanding the supplied metadata to include age, gender, and user-perceived expression and skill levels. Multi-tag classification in conjunction with a deeper look into the feature maps of the convolutional neural networks would provide much needed insight into how these different tags influence particular manifestations of speech characteristics during singing.

5 Conclusion

In this paper, we consider two variants of deep convolutional networks for the novel task of detecting characteristics of accent within the amateur singing voice. We train a simple 3x3 vanilla CNN as well as a more complex ResNeXt CNN to classify ten different accents (using country and language as a proxy) amongst both native and non-native English speakers, based on performances of the English song "Amazing Grace". The results show that the CNN-ResNeXt model is able to outperform both our baseline KNN clustering model as well as our vanilla CNN-3x3 model, obtaining higher accuracy, recall, and F1 score than both simpler models. Nevertheless, these metrics are still generally low, with a high source of error stemming from over-prediction of variants of English accent. This may be due to the fact that the performances were of a well-known, traditional English song, which could confound and modify pronunciation and intonation in singing much more so than speech. Analysis of the confusion matrices also suggests that there may be other aspects of musical speech, such as stylistic choices, that outweigh characteristics of accent, making the task of accent classification more difficult. Additionally, further analysis to discern whether country-language labels are indeed an accurate proxy for accent is also needed moving forward.

References

[1] Wang, C & Tzanetakis, G (2018) Singing Style Investigation by Residual Siamese Convolutional Neural Networks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).*

[2] Pons, J & Serra, X (2017) Designing Efficient Architectures For Modeling Temporal Features With Convolutional Neural Networks. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).*

[3] Pons, J, Slizovskaia, O, Gong, R, Gomez, E & Serra, X. (2017) Timbre Analysis of Music Audio Signals with Convolutional Neural Networks. *arXiv Preprint, arXiv:1703.06697v2, 2017*

[4] Jiao, Y, Tu, M, Berisha, V, & Liss, J (2016) Accent Identification by Combining Deep Neural Networks and Recurrent Neural Networks Trained on Long and Short Term Features. In *17th Annual Conference of the International Speech Communication Association (INTERSPEECH 2016)*

[5] K. Choi, G. Fazekas, K. Cho, & M. Sandler (2017) A Tutorial on Deep Learning for Music Information Retrieval. *arXiv Preprint, arXiv:1709.04396v2, 2017*

[6] Wessel, D (1979), Timbre space as a musical control structure. *Computer Music Journal*, pp. 45–52.

[7] G. Peeters, B. L. Giordano, P. Susini, N. Misdariis, & S. McAdams, (2011) The timbre toolbox: Extracting audio descriptors from musical signals. In *The Journal of the Acoustical Society of America* vol. 130, no. 5, pp. 2902–2916.

[8] S. Xie, R. B. Girshick, P. D. Zhuowen & K. He, (2016) Aggregated Residual Transformations for Deep Neural Networks. *arXiv Preprint, arXiv:1611.05431*

[9] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, & R. Salakhutdinov, (2014) Dropout: A Simple Way to Prevent Neural Networks from Overfitting. In *Journal of Machine Learning Research* vol. 15, pp. 1929-1958.