CS 698: Identifying Demographics from Speech
Project Report

Kei Fukutani

Instructor: David Guy Brizan

University of San Francisco
Summer, 2020

1. Project Overview

The goal of this project was to identify demographics, such as native languages and regions where the speakers were raised, using the voice of the speaker as the only input. The reason why I chose these demographics is because I thought this analysis would help understand the differences of various English languages people speak. Mel Frequency Cepstral Coefficients and Mel spectrograms were extracted from input audios. Convolutional Neural Networks took those features as input and predicted the speaker's native language and region where they were raised.

2. Dataset

2.1 Data Preparation

I used the Fisher English Corpus [1] in this project. This corpus contains a collection of 11699 conversational telephone speech (CTS) files that were created at the Linguistic Data Consortium (LDC). It contains transcript data for each conversation which lasts up to 10 minutes. There is a complete set of tables describing the speakers, the properties of the telephone calls, and the set of topics that were used to initiate the conversations.

First, I cut the 11699 conversation wav files into utterance wav files using the timestamp in the transcript file. The number of utterance wav files is 2,227,788. Then, to train neural networks as a model to predict demographics, I picked up the labels "NATIVE_LANG" and "WHERE_RAISED" from the description of the speakers. "NATIVE_LANG" indicates the speaker's native language and "WERE_RAISED" indicates the region where the speaker was raised.

Of the 2,227,788 utterances, Figure 1 shows the number of labels of top 20 native languages. Figure 2 shows the number of labels of top 20 regions for people whose native language is English. Native languages are highly imbalanced. So, I decided to have the model predict native languages, and if the native language is English, then it predicts regions. For example, the label is "Eng_NY" if the speaker's native language is "English" and the region raised is "NY".

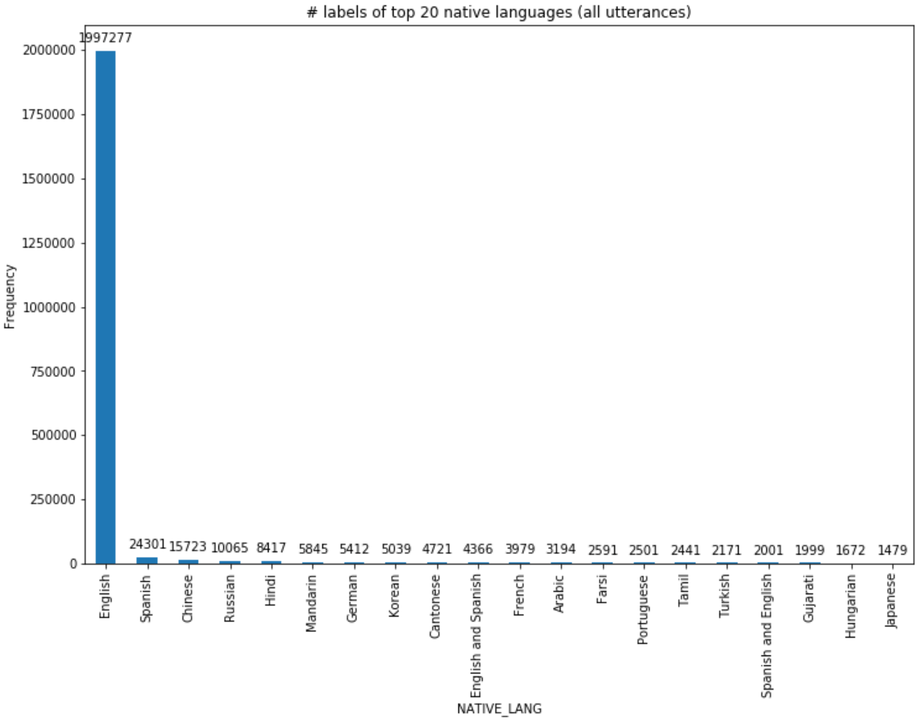


Figure 1: Distribution of labels of top 20 “NATIVE_LANG” across the entire dataset.

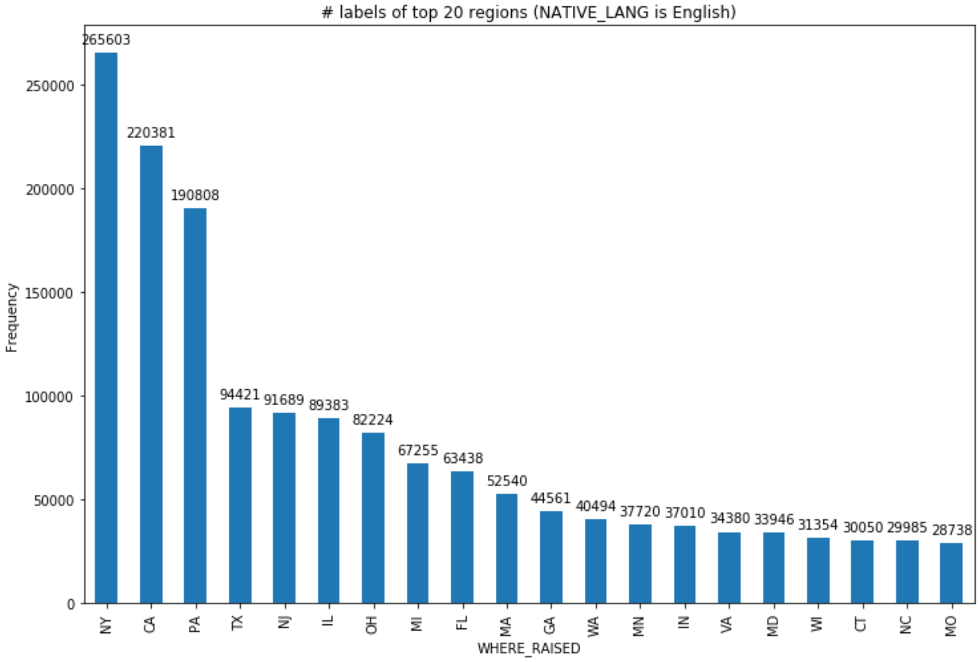
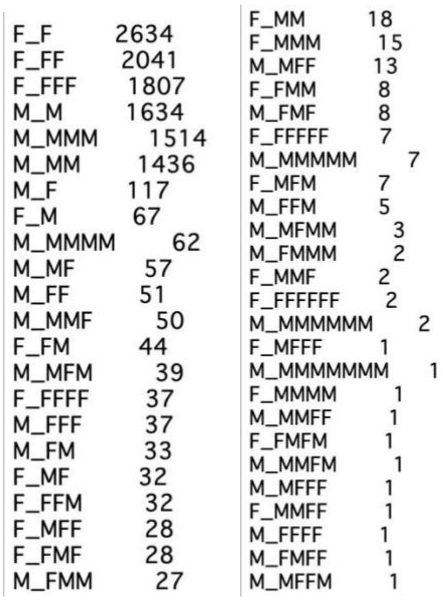


Figure 2: Distribution of labels of top 20 “WERE_RAISED” across the dataset in which the “NATIVE_LANG” is “English”.

2.2 Data Cleaning

There are 11971 participants (different PINs) in the dataset, and the number of people who reported their gender is 11917. Of the 11917 participants, I checked the patterns of [gender reported by speaker]_[[gender reported by annotator] …]. For example, "F_FFF" indicates that the gender reported by the speaker is female, and she recorded 3 conversations and for each conversation, her gender was reported by annotator as female. Figure 3 shows the distribution of these patterns.



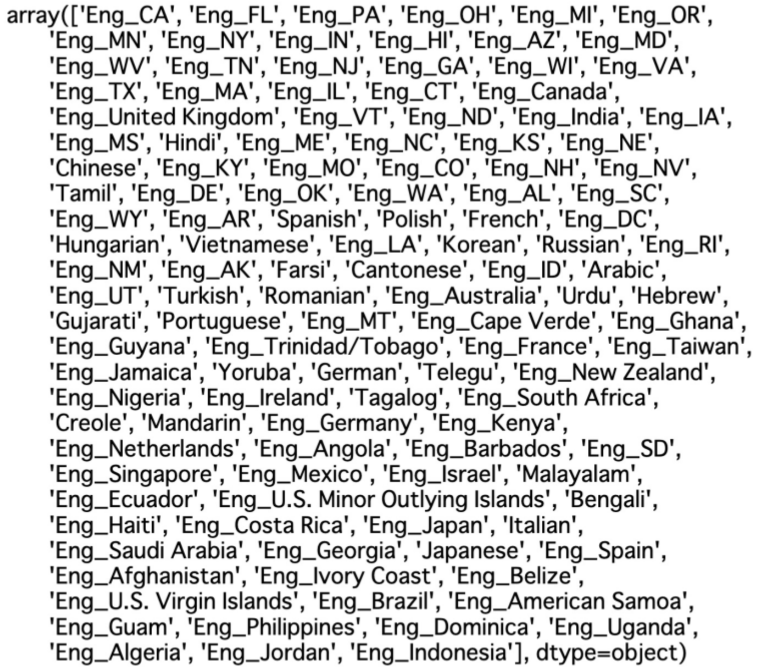
F_F	2634
F_FF	2041
F_FFF	1807
M_M	1634
M_MMM	1514
M_MM	1436
M_F	117
F_M	67
M_MMMM	62
M_MF	57
M_FF	51
M_MMF	50
F_FM	44
M_MFM	39
F_FFFF	37
M_FFF	37
M_FM	33
F_MF	32
F_FFM	32
F_MFF	28
F_FMF	28
M_FMM	27
F_MM	18
F_MMM	15
M_MFF	13
F_FMM	8
M_FMF	8
F_FFFFF	7
M_MMMMM	7
F_MFM	7
M_FFM	5
M_MFMM	3
M_FMMM	2
F_MMF	2
F_FFFFFF	2
M_MMMMMM	2
F_MFFF	1
M_MMMMMMM	1
F_MMMM	1
M_MMFF	1
F_FMFM	1
M_MMFM	1
M_MFFF	1
F_MMFF	1
M_FFFF	1
M_FMFF	1
M_MFFM	1

Figure 3: Distribution of gender patterns.

According to the data description file: (/home2/magics/corpora/fisher-english/transcripts/doc/part1/doc_pindata_tbl.txt), "it can (and does) happen that the person who answers the phone and records a call is NOT the person who was recruited and assigned a given PIN." Another readme file: (/home2/magics/corpora/fisher-english/transcripts/doc/part1/fe_03_readme.txt) says, "… , but it's probably safe to assume that most of the demographic data is applicable, given the matching gender."

If a person who reported their information is completely different from a person who recorded the audio data, it is a problem. Therefore, I used only gender-matched data.

In addition, I removed the data that has NaN in either "NATIVE_LANG" or "WHERE_RAISED" and selected the top 30 non-English native languages and 94 regions within native language "English". As a result, the number of classes is 124 in the whole dataset. At this point, the number of utterances is 1,978,945. Figure 4 shows those labels. Note that I used the labels preprocessed in this way for this project, but there are some concerns about the labels. I discuss these issues in section 6.



```
array(['Eng_CA', 'Eng_FL', 'Eng_PA', 'Eng_OH', 'Eng_MI', 'Eng_OR',
       'Eng_MN', 'Eng_NY', 'Eng_IN', 'Eng_HI', 'Eng_AZ', 'Eng_MD',
       'Eng_WV', 'Eng_TN', 'Eng_NJ', 'Eng_GA', 'Eng_WI', 'Eng_VA',
       'Eng_TX', 'Eng_MA', 'Eng_IL', 'Eng_CT', 'Eng_Canada',
       'Eng_United Kingdom', 'Eng_VT', 'Eng_ND', 'Eng_India', 'Eng_IA',
       'Eng_MS', 'Hindi', 'Eng_ME', 'Eng_NC', 'Eng_KS', 'Eng_NE',
       'Chinese', 'Eng_KY', 'Eng_MO', 'Eng_CO', 'Eng_NH', 'Eng_NV',
       'Tamil', 'Eng_DE', 'Eng_OK', 'Eng_WA', 'Eng_AL', 'Eng_SC',
       'Eng_WY', 'Eng_AR', 'Spanish', 'Polish', 'French', 'Eng_DC',
       'Hungarian', 'Vietnamese', 'Eng_LA', 'Korean', 'Russian', 'Eng_RI',
       'Eng_NM', 'Eng_AK', 'Farsi', 'Cantonese', 'Eng_ID', 'Arabic',
       'Eng_UT', 'Turkish', 'Romanian', 'Eng_Australia', 'Urdu', 'Hebrew',
       'Gujarati', 'Portuguese', 'Eng_MT', 'Eng_Cape Verde', 'Eng_Ghana',
       'Eng_Guyana', 'Eng_Trinidad/Tobago', 'Eng_France', 'Eng_Taiwan',
       'Eng_Jamaica', 'Yoruba', 'German', 'Telegu', 'Eng_New Zealand',
       'Eng_Nigeria', 'Eng_Ireland', 'Tagalog', 'Eng_South Africa',
       'Creole', 'Mandarin', 'Eng_Germany', 'Eng_Kenya',
       'Eng_Netherlands', 'Eng_Angola', 'Eng_Barbados', 'Eng_SD',
       'Eng_Singapore', 'Eng_Mexico', 'Eng_Israel', 'Malayalam',
       'Eng_Ecuador', 'Eng_U.S. Minor Outlying Islands', 'Bengali',
       'Eng_Haiti', 'Eng_Costa Rica', 'Eng_Japan', 'Italian',
       'Eng_Saudi Arabia', 'Eng_Georgia', 'Japanese', 'Eng_Spain',
       'Eng_Afghanistan', 'Eng_Ivory Coast', 'Eng_Belize',
       'Eng_U.S. Virgin Islands', 'Eng_Brazil', 'Eng_American Samoa',
       'Eng_Guam', 'Eng_Philippines', 'Eng_Dominica', 'Eng_Uganda',
       'Eng_Algeria', 'Eng_Jordan', 'Eng_Indonesia'], dtype=object)
```

Figure 4: 124 classes in the entire dataset.

3. Feature Engineering and Model Architectures

3.1 Mel Frequency Cepstral Coefficients (MFCC)

MFCC is widely used as a feature of sound data. MFCCs are derived as follows [2]. I used a sampling rate of 8,000 Hz, which is a native sampling rate of each file, length of the FFT window of 2048 (256 ms), and hop length of 512 (64 ms). The number of MFCCs (n_mfcc) is 20, and the number of Mel bands (n_mels) is 128. For each time frame of 256 ms, the following steps are executed:

1. Take the Fourier transform of (a windowed excerpt of) a signal.
2. Map the powers of the spectrum obtained above onto the mel scale, using triangular overlapping windows.
3. Take the logs of the powers at each of the mel frequencies.
4. Take the discrete cosine transform of the list of mel log powers, as if it were a signal.

The MFCCs are the amplitudes of the resulting spectrum. An example of MFCCs for a 5 seconds utterance is shown in Figure 5.

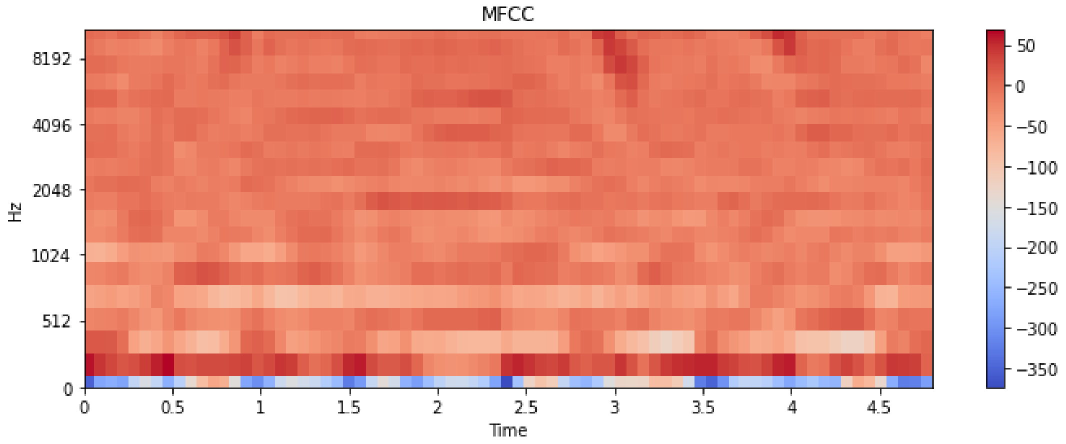


Figure 5: An example of MFCCs for a 5 sec utterance.

3.2 Mel Spectrogram

Mel Spectrogram is also used as an input and produces better results for some neural networks [3][4]. Log-mel spectrum is obtained after the step 3 above. In other words, the Log-mel spectrum is a vector before being applied by discrete cosine transform. An example of Log-mel spectrogram for a 5 seconds utterance is shown in Figure 6, which is the same utterance as that of Figure 5.

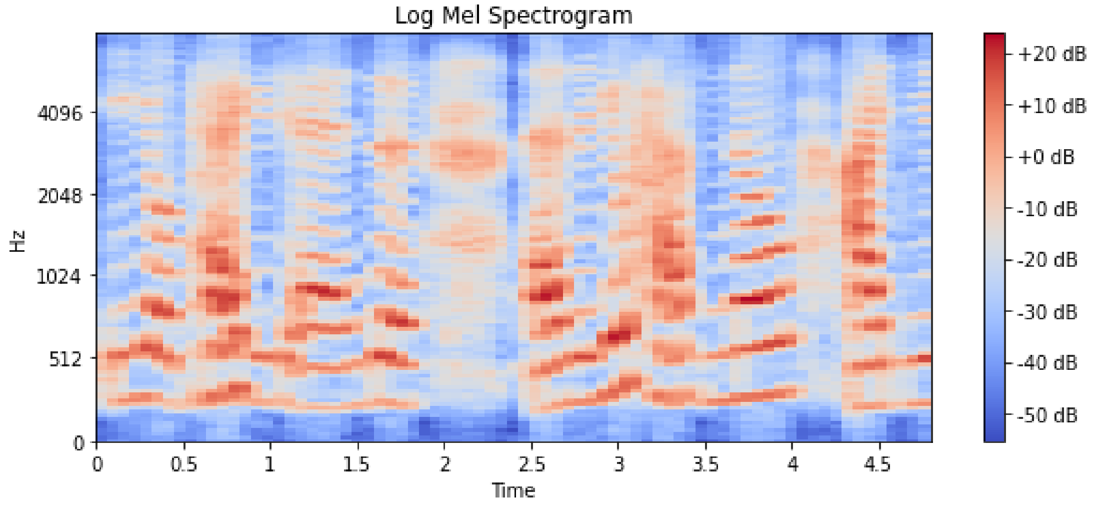


Figure 6: An example of Log-mel spectrogram for a 5 sec utterance.

Those two features of sound described above are of time and frequency domain. They contain information about transitions of Formants and could have something else that humans cannot see but computers can. Note that I did not apply any noise reduction technique.

3.3 Model Architectures

Simple Convolutional Neural Networks (CNNs) were trained in this project. I built 5 different architectures. Figure 7 shows these architectures. Paddings were set for all convolutional layers such that output has the same height/width dimension as the input. All activation functions for convolutional layers and fully connected layers were Rectified Linear Unit (ReLU). I did not use strides in this project.

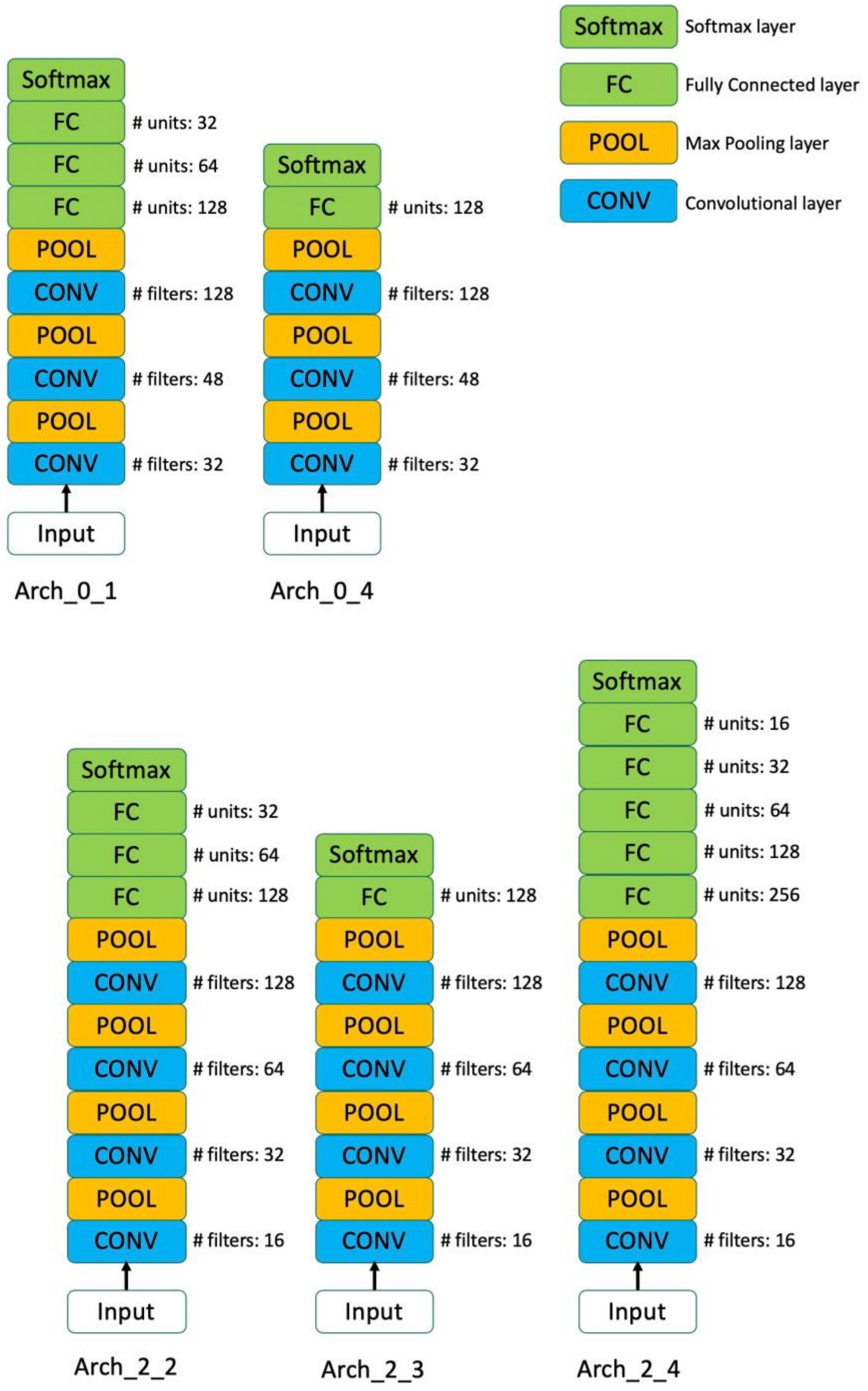


Figure 7: Five different architectures of CNNs.

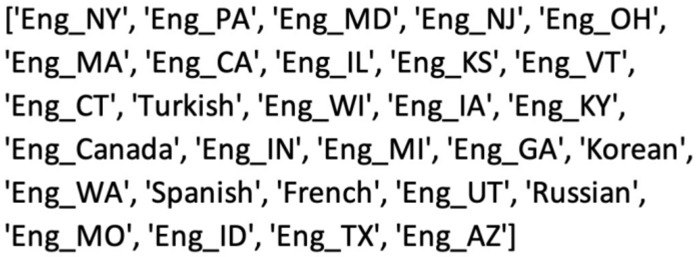
4. Experiments

4.1 Data

Of the 1,978,945 utterances described in Section 2.2, I picked up 10,000 and cut off 5 seconds waveform data out of each utterance. The utterances that were shorter than 5 seconds were padded with zeros so as to be taken as input for CNN models. The two-dimension feature of MFCC is of size 20 x 75, and 128 x 75 for Log-mel spectrogram.

4.2 Train, Validation, and Test Data

Those 10,000 MFCCs or Log-mel spectrograms were randomly split into training set, validation set, and test set, which contain 8,100, 900, and 1,000 utterances respectively. In the 8100 utterances in the training set, there are 29 classes shown in Figure 8.



```
['Eng_NY', 'Eng_PA', 'Eng_MD', 'Eng_NJ', 'Eng_OH',
'Eng_MA', 'Eng_CA', 'Eng_IL', 'Eng_KS', 'Eng_VT',
'Eng_CT', 'Turkish', 'Eng_WI', 'Eng_IA', 'Eng_KY',
'Eng_Canada', 'Eng_IN', 'Eng_MI', 'Eng_GA', 'Korean',
'Eng_WA', 'Spanish', 'French', 'Eng_UT', 'Russian',
'Eng_MO', 'Eng_ID', 'Eng_TX', 'Eng_AZ']
```

Figure 8: 29 classes our models predict.

4.3 Methodology

All models were trained for 200 epochs with early stopping using patience 50. Adam was used as an optimizer and the learning rate was 0.0005 for every experiment.

First, I chose one architecture Arc_2_2 and trained it with 5 different filter sizes 3x3, 5x5, 7x7, 9x9, and 11x11 in the hope that larger filters can learn something in the longer time span, which smaller filters cannot learn. Considering that the length of a phoneme is approximately 150 ms and the hop length is 64 ms, filter size 3x3 seems a little too small to see the transition of the phonemes. Second, I selected one filter size 9x9 and trained with 5 different architectures, Arc_0_1, Arc_0_4, Arc_2_2, Arc_2_3, and Arc_2_4. The MFCCs and Log-mel spectrograms were used as input for every experiment. Lastly, L2 regularization and dropout layers were added to some models.

In order to evaluate the model performance, accuracy was used since the distribution of the labels was not too imbalanced. Confusion matrices were also calculated for each experiment.

5. Results

Although the test accuracy fluctuated every time training was executed, filter size 7x7 and 9x9 seem a little better than others for MFCC inputs, and filter size 5x5 and 7x7 for Log-mel spectrogram. For architecture, it is hard to say which architecture is better than others. Arch_2_2 might be a little better than others. Also, it appears that the Log-mel spectrogram is better than MFCC when used as input.

Architecture	Feature	Filter size	Test Accuracy
Arch_2_2	MFCC	3x3	0.7760
Arch_2_2	MFCC	5x5	0.7850
Arch_2_2	MFCC	7x7	0.8140
Arch_2_2	MFCC	9x9	0.8260
Arch_2_2	MFCC	11x11	0.7610
Arch_0_1	MFCC	9x9	0.7990
Arch_0_4	MFCC	9x9	0.7150
Arch_2_2	MFCC	9x9	0.8260
Arch_2_3	MFCC	9x9	0.8130
Arch_2_4	MFCC	9x9	0.7990
Arch_2_2	Log-mel spectrogram	3x3	0.8380
Arch_2_2	Log-mel spectrogram	5x5	0.8440
Arch_2_2	Log-mel spectrogram	7x7	0.8430
Arch_2_2	Log-mel spectrogram	9x9	0.8290
Arch_2_2	Log-mel spectrogram	11x11	0.8070
Arch_0_1	Log-mel spectrogram	9x9	0.8250
Arch_0_4	Log-mel spectrogram	9x9	0.8380
Arch_2_2	Log-mel spectrogram	9x9	0.8290
Arch_2_3	Log-mel spectrogram	9x9	0.8110
Arch_2_4	Log-mel spectrogram	9x9	0.8290
Arch_0_1	Log-mel spectrogram	5x5	0.8070
Arch_0_4	Log-mel spectrogram	5x5	0.8120
Arch_2_2	Log-mel spectrogram	5x5	0.8440
Arch_2_3	Log-mel spectrogram	5x5	0.8120
Arch_2_4	Log-mel spectrogram	5x5	0.8400

Lastly, L2 regularization and dropout layers were added to Arch_2_2 to reduce overfitting. Figure 9 shows this architecture. The fraction of the input units to drop was 0.2 for all dropout layers. Lambda 0.001 was used for all convolutional layers and fully connected layers after some experiments. Regularization obviously improved the performance of the model. Figure 10 shows the training and validation accuracy and loss for the last model, which achieved a test accuracy of 89.8 %. Figure 11 shows the confusion matrix of the model.

Architecture	Feature	Lambda	Filter size	Test Accuracy
Arch_2_2_b	Log-mel spectrogram	None (Dropout only)	5x5	0.8690
Arch_2_2_b	Log-mel spectrogram	0.001	5x5	0.8980

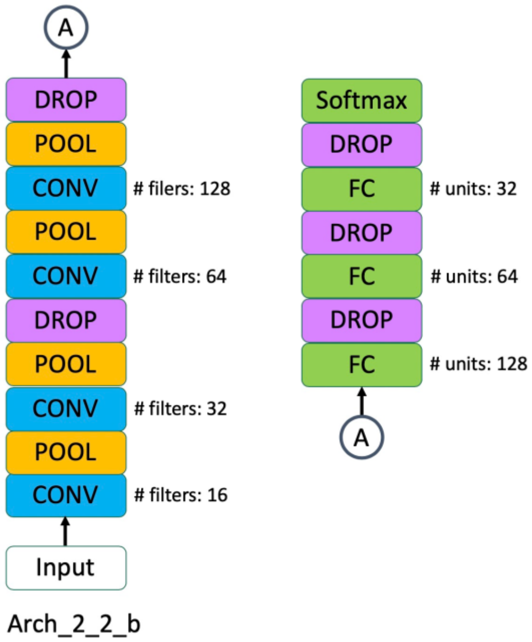


Figure 9: Architecture Arch_2_2_b.

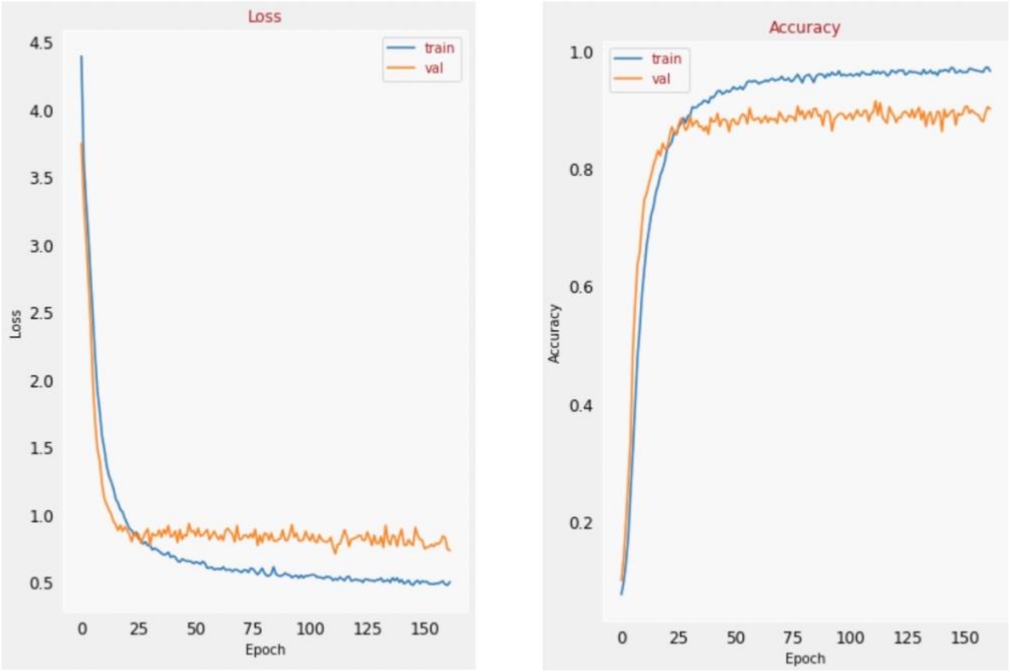
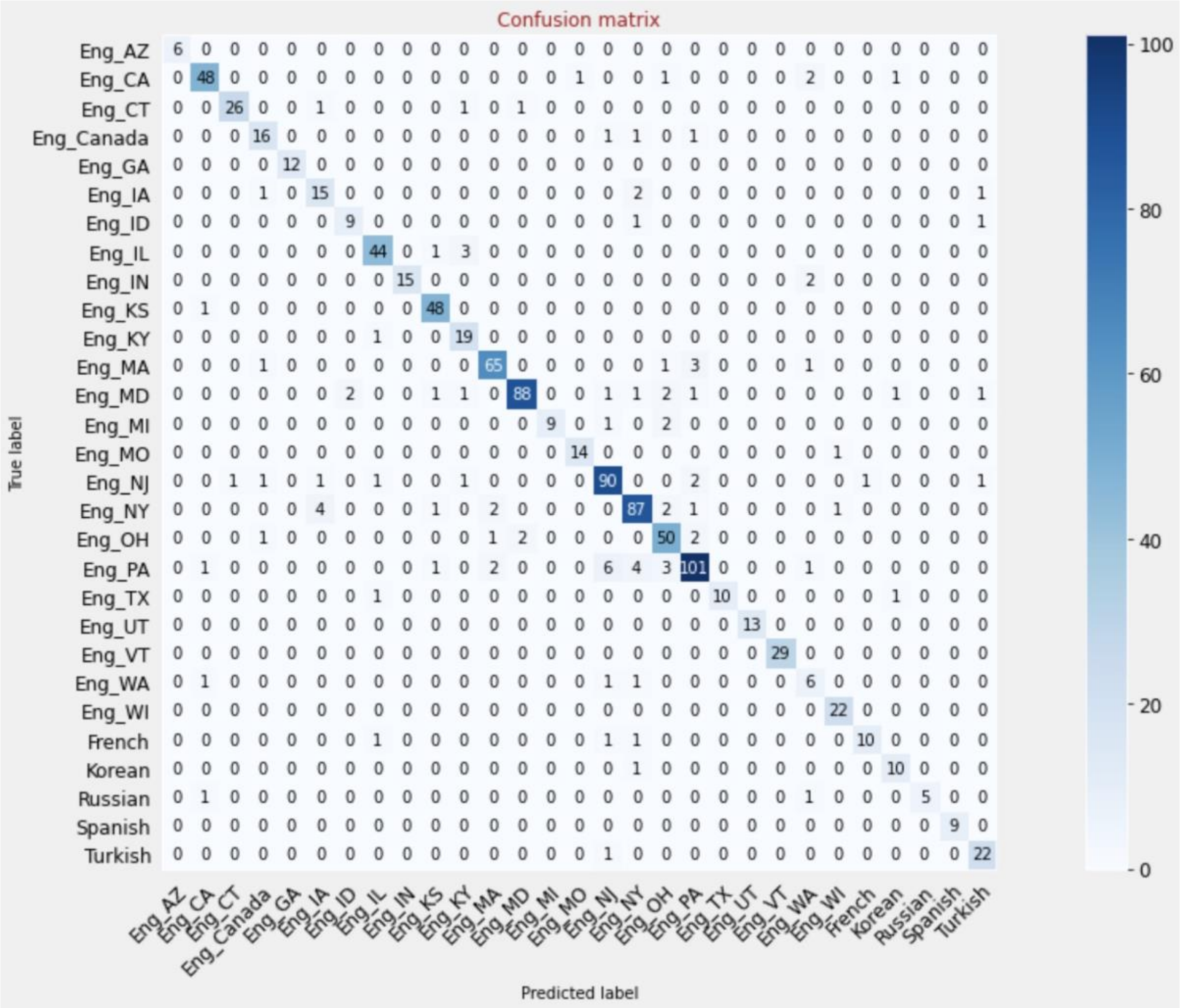


Figure 10: Training and Validation accuracy and Loss for the last model.



	Eng_AZ	Eng_CA	Eng_CT	Eng_Canada	Eng_GA	Eng_IA	Eng_ID	Eng_IL	Eng_IN	Eng_KS	Eng_KY	Eng_MA	Eng_MD	Eng_MI	Eng_MO	Eng_NJ	Eng_NY	Eng_OH	Eng_PA	Eng_TX	Eng_UT	Eng_VT	Eng_WA	Eng_WI	French	Korean	Russian	Spanish	Turkish
Eng_AZ	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Eng_CA	0	48	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	2	0	0	1	0	0	0
Eng_CT	0	0	26	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Eng_Canada	0	0	0	16	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0
Eng_GA	0	0	0	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Eng_IA	0	0	0	1	0	15	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	1
Eng_ID	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1
Eng_IL	0	0	0	0	0	0	0	44	0	1	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Eng_IN	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0
Eng_KS	0	1	0	0	0	0	0	0	0	48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Eng_KY	0	0	0	0	0	0	0	1	0	0	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Eng_MA	0	0	0	1	0	0	0	0	0	0	0	65	0	0	0	0	0	1	3	0	0	0	1	0	0	0	0	0	0
Eng_MD	0	0	0	0	0	0	2	0	0	1	1	0	88	0	0	1	1	2	1	0	0	0	0	0	0	1	0	0	1
Eng_MI	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	1	0	2	0	0	0	0	0	0	0	0	0	0	0
Eng_MO	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Eng_NJ	0	0	1	1	0	1	0	1	0	0	1	0	0	0	0	90	0	0	2	0	0	0	0	0	1	0	0	0	1
Eng_NY	0	0	0	0	0	4	0	0	0	1	0	2	0	0	0	0	87	2	1	0	0	0	0	1	0	0	0	0	0
Eng_OH	0	0	0	1	0	0	0	0	0	0	0	1	2	0	0	0	0	50	2	0	0	0	0	0	0	0	0	0	0
Eng_PA	0	1	0	0	0	0	0	0	0	1	0	2	0	0	0	6	4	3	101	0	0	0	1	0	0	0	0	0	0
Eng_TX	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	1	0	0	0
Eng_UT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0
Eng_VT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	29	0	0	0	0	0	0	0
Eng_WA	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	6	0	0	0	0	0	0
Eng_WI	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	22	0	0	0	0	0
French	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	10	0	0	0	0
Korean	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	10	0	0	0
Russian	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	5	0	0
Spanish	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0
Turkish	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	22

Figure 11: Confusion matrix of the last model.

6. Future Work

There is a lot of room for improvement:

- In the 124 classes, there are some labels that are not appropriate for this project, such as “Eng_Brazil”, “Eng_Japan”, and so on. Those labels indicate that the speaker’s native language is English and that their region of origin is a country whose native language is not English. Whether the speech data reflect those demographics or not could be further explored. Also, if the number of utterances for a label is too small, then models will catch the specific person’s way of speaking, not the pattern of dialect. It is necessary to analyze how many speakers there are in each label.

- Different architectures, not only CNN, but also LSTM, Attention mechanism, or a combination of those could produce better results. Also, it is important to analyze why some performances are good or not good. It seems that there is a technique to identify which part of an image contributes to classification decisions that CNNs make. Needless to say, domain knowledge of accent classification should give some hints for further exploring neural network models.

- Only less than 1 % of the data available were used in this project. Adding more data with carefully selected classes should improve the performances. However, it will be harder to store, clean up, transform large amounts of data, and train models with it. It took about 7 hours to convert 50,000 utterances to Log-mel spectrograms. The size of 1,978,945 utterances is more than 100 GB.

In conclusion, it took me a lot of time to preprocess and analyze the data and implement a pipeline to train the models, but I learned a lot from this project.

7. References

[1] https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/lrec2004-fisher-corpus.pdf

[2] https://en.wikipedia.org/wiki/Mel-frequency_cepstrum

[3] Hendrik Purwins, Bo Li, Tuomas Virtanen, Jan Schlüter, Shuo-yiin Chang, Tara Sainath (2019). “Deep Learning for Audio Signal Processing”, https://arxiv.org/pdf/1905.00078.pdf

[4] Kannan Venkataramanan, Haresh Rengaraj Rajamohan (2019). “Emotion Recognition from Speech”, https://arxiv.org/pdf/1912.10458.pdf