APS360 Final Report

12/9/2020

Team 44
Catherine Glossop, Julia Chae, Mingshi Chi, Rocco Ruan

Word Count:

Specifications

1. Word Count limit: 2500 words (1% penalty per word)
2. There are about 44 marks tied specifically to text sections. As such, each section should be about 60 words long per mark it's worth.
    a. "Evaluating on new data" is worth 10 marks but probably not going to be 550 words - so can probably bleed a bit longer on other sections

Introduction (2 marks, 120 words)

The goal of this project is to develop a CNN speech accent classifier which intakes an accented English phrase as input and correctly identifies the origin of the speaker's accent. This goal is motivated by direct experiences with voice-controlled devices not understanding heavily accented English, which is especially prominent in immigrant families and friends. An accent classifier contributes to solving this problem by enabling the subsequent use of accent-specific text-to-speech algorithms to improve the voice-controlled device experience for accented speakers.

Machine learning is appropriate for this task as accent classification is difficult to code an explicit algorithm for, but possible for humans to do. The diversity in the types of audio inputs from gender, audio length, accent, quality and volume makes it hard for any classical algorithms to be able to distinguish different classes. With machine learning, this classification function can be learned and improved to the desired level of performance with sufficient

Illustration (2 marks, no words)

- Can be of model, or of general idea of our project
- If model:
    - Possible to reuse and improve with details
    - If a new architecture is used, can draw a new one (CAT)
- If general idea of project:
    - we have one in our presentation, in slide 6 - can reuse or slightly edit

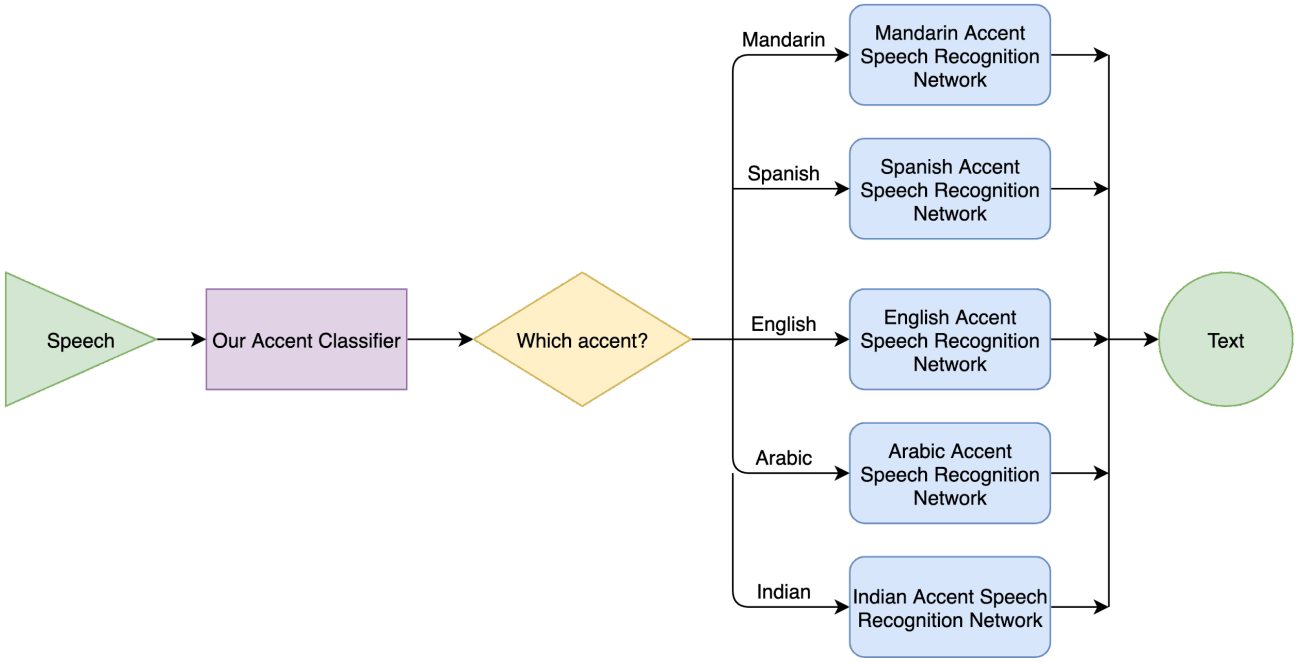


Figure X. An illustration of the use-case of an accent classifier in a speech recognition system.

Background and Related Work (2 marks, 120 words)

- **Rocco's section is still relevant here**

We heavily referenced the work of Mak and Mok [1], who created an accent classifier with similar motivations to ours. Their paper referenced the use of mel-frequency cepstral coefficients (MFCCs), which are representations of the amplitudes of a sound's component frequencies over time commonly used to preprocess data for speech recognition [2] and accent classification. This paper also gave us ideas on expanding our dataset, which are elaborated upon in Data Processing.

Mak and Mok also cite several papers that implement models such as SVMs [3][4] for accent classification, giving us an idea of the accuracy that we can expect. Depending on the number of accents being classified, validation accuracy ranges between 50% to 90% in papers we have reviewed. However, most papers used their "test" data to adjust hyperparameters, which may have inflated their reported accuracies.

More detail into our background research can be found here.

Data Processing (2 marks, 120 words)

We selected classes of data from George Mason University's Speech Accent Archive, with the following criteria:

1. How common the accent was in Canada and the United States
2. How much data was available

To this end, we selected English, Mandarin, Spanish, and Arabic as classes. We also put together native speakers of many Indian languages into a single class, based on other research that has done the same[1]. The number of files per class can be observed in Appendix A. Each class from the Archive consisted of a 20-30 second .mp3s of speakers repeating a specific sentence, which we loaded into a resampled, normalized audio-time series and clipped into 10-second sections (this number was decided experimentally).

To extract useful featured, we then converted each section to a logarithmic mel spectrogram, which illustrate auditory power over time for various frequencies, approximating the human perception of sound while preserving more frequency bands than the more common MFCC method[2].These images were then consumed by our model.

We attempted to generate additional data from our existing data by pitch-shifting, applying Gaussian noise, and using autoencoders. We were successful in generating this data, but these approaches increased training time without significant improvement in performance. We conducted many experiments with this additional data, but in the end our best-performing models were trained without it.

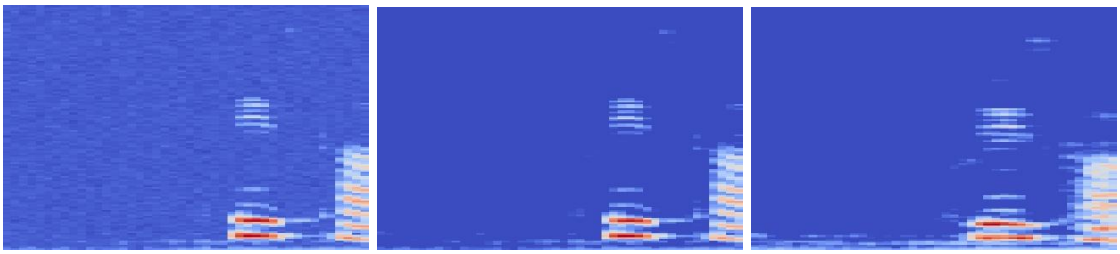


Figure X, Y, Z. A noisy MFCC (left) and a pitch-lowered MFCC (right) generated from a raw MFCC (middle). Notice the “fuzziness” of the left image and the “smushed-ness” of the right image (location on the y-axis indicates frequency, and colour indicates power).

Architecture (4 points, 240 words) (Julia)

The final architecture for our project is called CRNN, which is a combination of CNN, RNN and fully connected layers. The CRNN begins with convolutional layers followed by RNN and finally the RNN output gets reduced to an appropriate number of classes through funny connected layers.

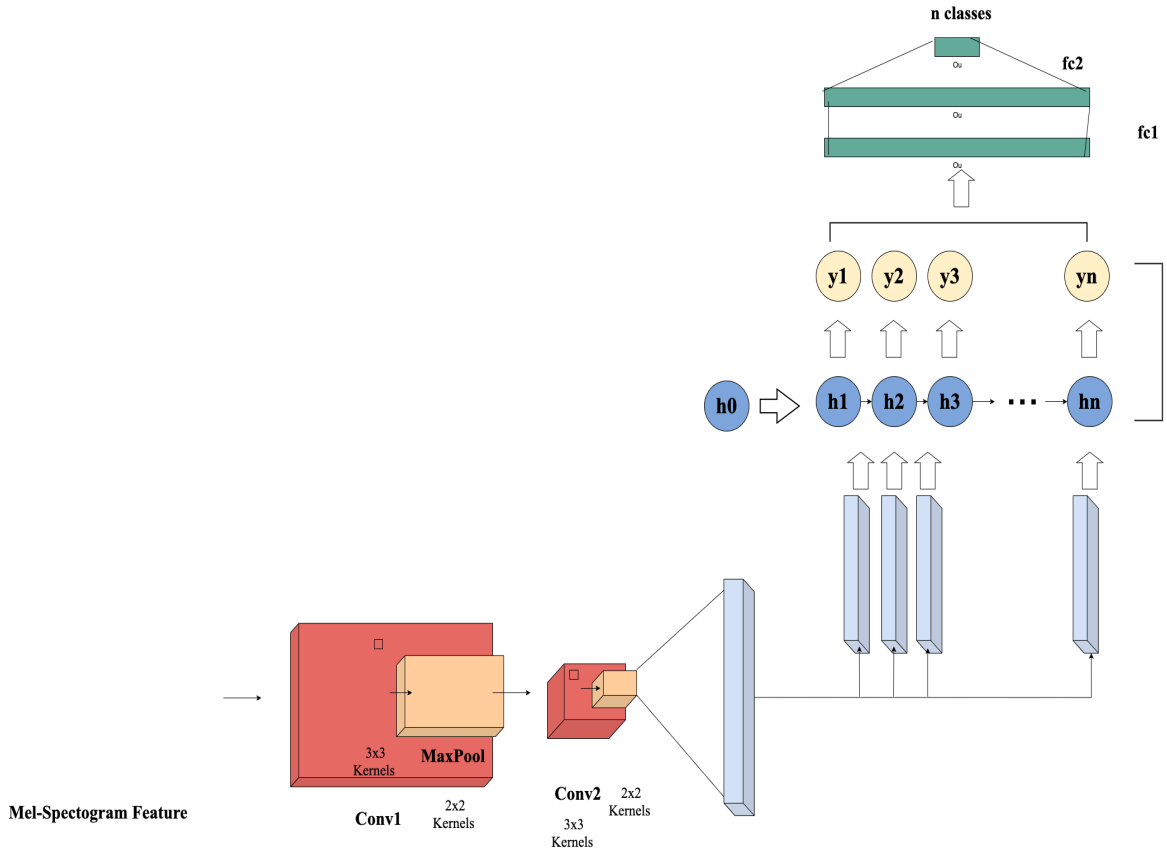
The CNN layers are composed of two convolutional layers with kernel size of 3, both followed by maxpool of kernel 2 and stride 2. The first convolutional layer takes in 3 channels as input and outputs 32 channels. Second layer takes in the 32 as input and outputs 64. At the end, the input feature of size 3x128x192 produces features of size 64x30x46.

1

2

The feature from the CNN then gets reshaped into a feature of 27x46x1920 for the RNN. The features are fed into an RNN with hidden size = 50, number of layers = 2, and input_size = 1970. Afterwards, the maximum of the RNN output is extracted to yield a feature of 50 for the fully connected layers.

The linear layers reduce the feature from 50 to 20, and then from 20 to 10 and finally from 10 to 2. Afterwards, a feature (scores) of size 2 are returned for the loss calculation and optimization. The loss function used is Cross Entropy and the optimizer is Adam optimizer.



The CRNN was selected as the past architecture after experimenting with 2D CNN, 1D CNN (time series) and RNN-only architectures and comparing the accuracy performance of those models. In addition, different types of RNN such as LSTM and GRU as well as additional layers such as batch-norm and drop-out were experimented with, however were omitted as they did not improve the performance of the model.

Baseline Model (4 points, 240 words) (Mingshi)

We compared our models for binary classification on English-Mandarin, English-Arabic, English- Spanish, English-India, and English-Foreign to an SVM. The SVM architecture was

imported from the sklearn library. The SVM functions used were linear, radial basis, polynomial, and sigmoid. All 4 functions performed relatively similarly with around 3% difference between the best performing and worst performing function in all experiments. Results of the comparison are shown in Figure _

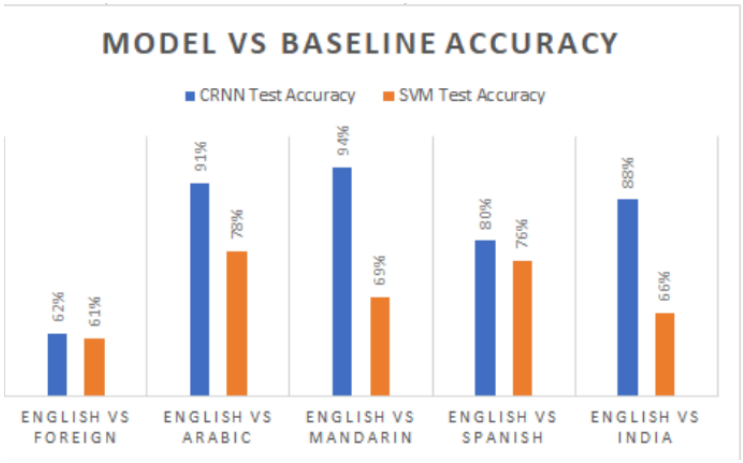


Figure _: SVM baseline model test accuracy compared to CRNN test accuracy

We also compared the performance of a tertiary classification task between English, Mandarin, and India classes with an ANN model using transfer learning. ANN baseline model with transfer learning gave a 54% validation accuracy compared to the highest validation from our models on tertiary tasks being 80%.

- Results on SVM and/or ANN
- Plot of results and comparison of performance

Quantitative Results (4 points, 240 words) (Rocco)

On a tertiary classification task between English, Mandarin, and Indian classes, we were able to achieve a validation accuracy around 75%. However, our test accuracy on holdout data was inexplicably 40% lower than the validation accuracy - perhaps implying that our models had manually overfit via extensive hyperparameter tuning. Our research showed that multi-class accent classification was quite difficult even for more advanced researchers, so we settled to demonstrate the efficacy of our architecture on binary classification instead.

Our training curves on the various binary classification tasks are shown below:

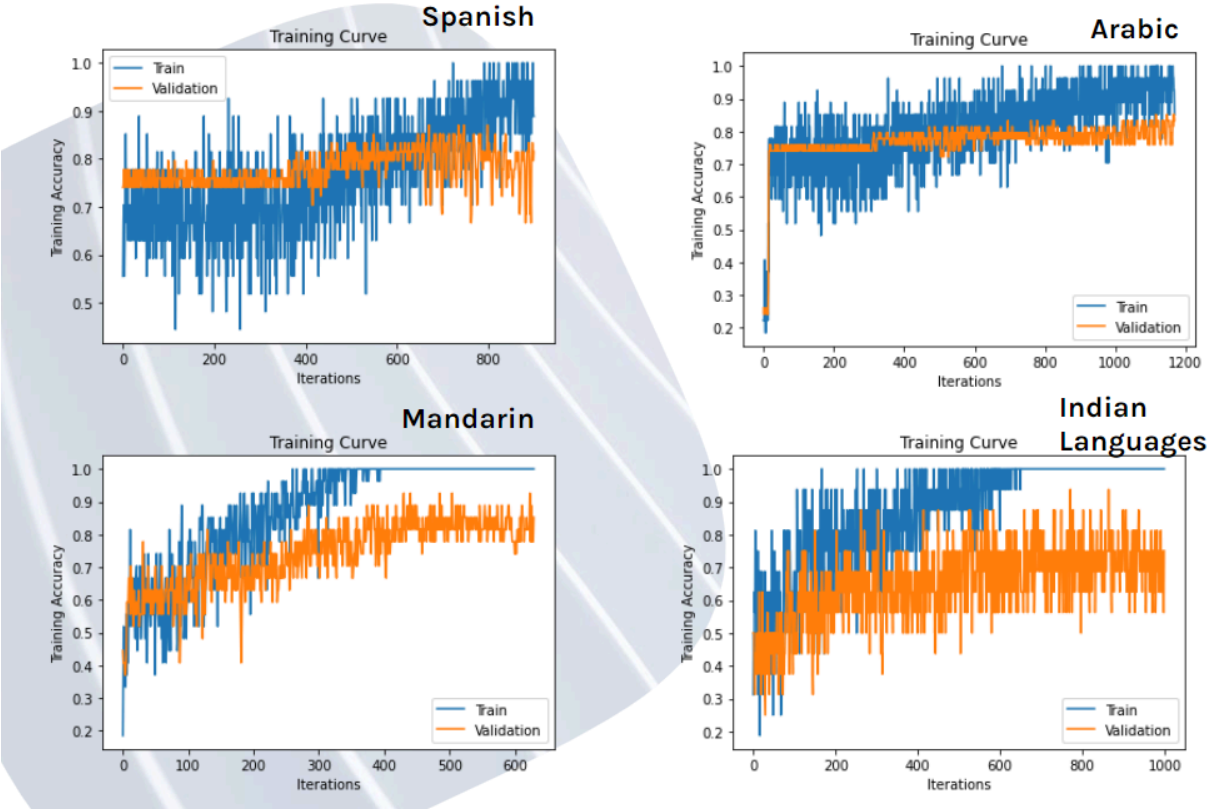


Figure X. Training curves for various classification tasks. Training accuracy reached 100% in all tasks, and validation accuracy peaked around 80-90% depending on the task.

Seeing satisfactory validation results, we then tested on holdout data, and compared to our baseline model's performance on the same test data. Results are summarized below; as mentioned prior in the Baseline section, we assessed our test results as satisfactory based on comparison with an SVM's performance.

Classification Task	CRNN Test Accuracy	
English vs Foreign	**62% (+1%)**	Correct/ Incorrect
English vs Arabic	**91% (+13%)**	**20/2**
English vs Mandarin	**94% (+25%)**	**14/1**
English vs Spanish	**80% (+14%)**	**24/5**
English vs India	**88% (+22%)**	**17/2**

Figure X. Results.

Qualitative Results (4 marks, 240 words) (Cat)

To qualitatively assess our model, we looked at the images that were passed in as input and the resulting classification. We noticed that there was little bias in our networks as the number of false positives for the both classes was not skewed towards either class. Additionally, we noticed that while the model performed well on holdout data, its performance dropped on collected data. Below are spectrograms of an english and mandarin sample that were correctly classified and two that were not in our collected dataset.

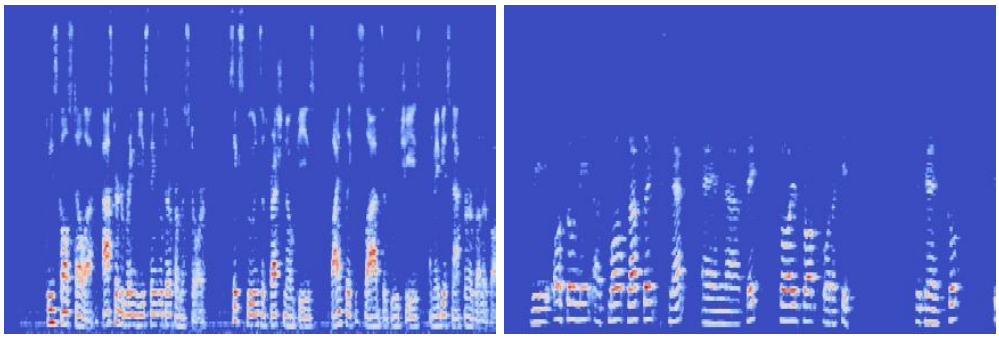


Figure X. Correctly classified English and Mandarin accents (left and right respectively)

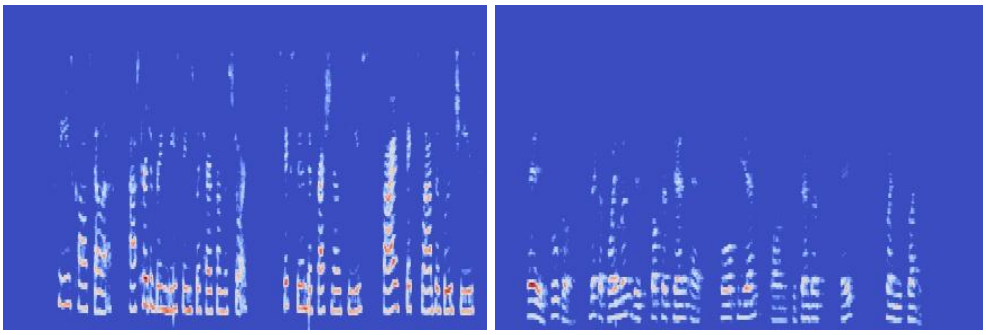


Figure X. Incorrectly classified English and Mandarin accents (left and right respectively)

The recordings we collected were lower quality and had less information for higher frequencies compared to the archive holdout test set and most were recorded on phone microphones and some had to be converted from m4a to mp3 meaning lossy conversions could have contributed to a loss in quality. This strongly suggests that it is reasonable that our model would perform better on higher quality data, similar to what it was trained on.

In the process of testing on our holdout data, to be discussed further in the evaluation section, we found that one specific sample, named english139, was consistently classified as a forgein in binary classification against english. After further investigation, the speaker in the clip is from Rhode Island and has idiosyncrasies often attributed to forgein accents. This results in a possible reason for this specific case of misclassification.

Evaluating on new data (400 words? if lower don't sweat it) - demonstration (Rocco and Mingshi)

Part of our evaluation on new data was conducted on holdout test data from the Speech Accent Archive, which was elaborated upon in Quantitative Results.

In order to test our model on some more diverse data, we also collected audio recordings of team members, friends, and family members of varying accents to run against our binary classifiers. Our audio samples included:

- 9 recordings from 4 native Mandarin speakers (7 + 2 Crystal)
- 1 recording from 1 native Hindi speaker
- 11 recordings from 7 English native speakers

We took various measures to ensure our demonstration dataset was comprehensive. They are listed below.

- 9 recordings were also audio samples of sentences other than the one present in the Speech Accent Archive, included to test the generalizability of the model.
- 2 of the recordings were taken in a noisy cafe to test robustness to noise.
- We made sure to include samples from both male and female speakers, for coverage.
- Speakers varied in terms of the strength and quality of their accents - for example, one speaker was classified as natively Mandarin, but had spoken English for 10 years, and another one was also classified as natively Mandarin but learned Cantonese before Mandarin.

We then wrote a test script to produce a prediction from each of these recordings using either our English-Mandarin or English-Indian binary classifiers. This was done using the following steps:

1. Loading, clipping, and resampling the audio from each recording, then generating the mel-spectrograms for each clip
2. Producing outputs for each clip, stored as floats for each of the two classes in the binary classification task being performed
3. Adding together the outputs for each clip, and taking the larger of the two outputs as the prediction for the model.

Our results were as follows:

English-India Classification - 1/1 recordings correct

English-Mandarin Classification:

- X / 20 correct (need to redo)
    - Y / 9 non-standard sentences correct
    - Z / 11 standard sentences correct
    - A / 2 noisy recordings correct

- B / n_male male speaker recordings correct
- C / n_female female speaker recordings correct

Discussion (480 words) (Cat and Julia)

Our results showed that test accuracy was higher for Mandarin and Arabic binary classification compared to the Indian dialects and Spanish classifiers. We recognized that the samples used for training the Indian Dialect-English classifier consolidated samples with native languages from across India. This meant that there was a lot of variation across the accents that were represented within the class similar to spanish which is also spoken across a large geographical area. This results in a variety of dialects that would make the dataset not as uniform compared to the Mandarin and Arabic samples that span across a much smaller geographical area. This is also represented in the PCA graphs we generated of the features in our dataset.

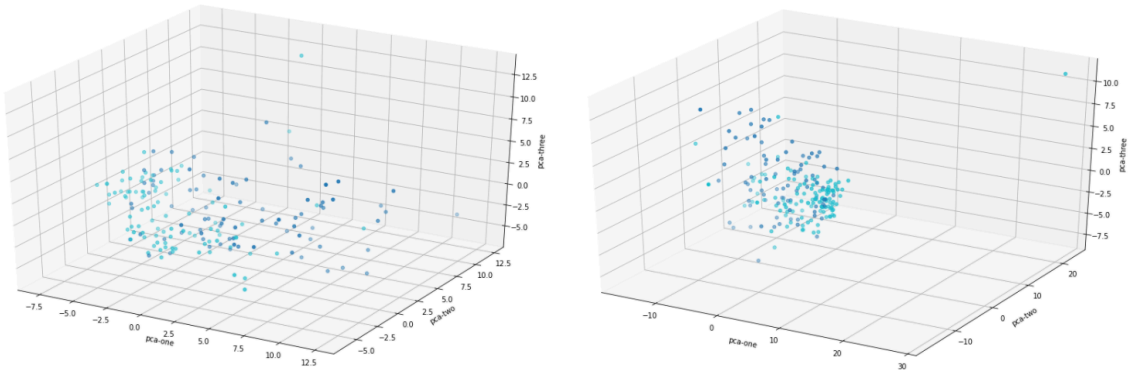


Figure X. Depicts the plot of the features of the Indian dialects vs. english dataset on the left and mandarin vs. english dataset on the right

Furthermore, accent classification proves to be a difficult problem in general due to the non-binary nature of accents. A person in the dataset was classified based on their country of origin and their native language but their degree of comfort in english is not taken into account. Therefore, within a given class there could be people with accents that would be considered the english standard and people with much heavier accents. Given our small dataset and the fact that processing audio data is considerably more difficult due to additional processing that is required for usable data, accent classification becomes a challenging problem.

Overall, our model performed quite well when compared to existing works in the problem area. For binary classification, the accuracies reported in many works seemed to range from 70% to 90% depending on the dataset, processing and the complexity of the model. On *Automatic Arabic Dialect Classification Using Deep Learning Models* paper published at *International Conference on Arabic Computational Linguistics*[3] in 2018, the maximum accuracy reported for classification between two dialect pairs was 83.8% using BLSTM architecture. In a paper published at *Artificial Intelligence and Interactive Digital Entertainment*[4]*,* the testing accuracy

[3] https://www.sciencedirect.com/science/article/pii/S1877050918321938
[4] https://www.aaai.org/ocs/index.php/AIIDE/AIIDE17/paper/download/15861/15222

between American and British dialect from Speech Accent Archive was 77% using AlexNet. In comparison to other existing projects, our model's average performance was 88% for binary-accent classification seemed to be a successful result.

From putting together this project, our group learned a lot in all aspects of machine learning. We learned about multiple ways to process audio data, and that the data processing decisions (i.e. length of audio clips, number of frequency bands, etc.) can drastically change the performance of the model. We also learned that combining architectures can boost performance, hence the CRNN architecture performing better than CNN and RNN models of similar caliber. Finally, we learned that some problems are more challenging than others and that the availability of data play a big role in performance.

- https://www.sciencedirect.com/science/article/pii/S1877050918321938 → binary arabic accent classification, which is around 70-80%
- https://www.aaai.org/ocs/index.php/AIIDE/AIIDE17/paper/download/15861/15222 → also around high 70s, 80s
- https://osuva.uwasa.fi/bitstream/handle/10024/9662/osuva_8534.pdf?sequence=1&isAllowed=y 96%

Ethical Considerations (2 marks, about 120 words)

Our research revealed that aside from accents, variations in speaking style produced by gender also significantly impact the ability of speech recognition algorithms [11].The Speech Accent Archive we used did not include people of any non-binary genders, but included exactly 588 male and 588 female speakers in the classes we selected, as shown.

As such, it is unlikely that our randomly-generated datasets produced significant bias, except in the case of the Mandarin dataset. It is possible that our Mandarin classifier was more likely to classify women as Mandarin due to the gender proportions in that class of data.

We also note that accent-classifying models have the capability to be used in ethically dubious ways, such as to discriminate against specific ethnic groups. However, we maintain that the existence of explicitly accent-classifying models actually fights discrimination more than it enables it. Without them, machine learning models can still become discriminatory by accident; accent classifiers combat this by facilitating quantitative screening of racial and ethnic bias in other developing or deployed models; we hold that unintentional bias is more prevalent today than explicit discrimination.

Appendix A - Class Proportions

Native Language	# of male speakers	# of female speakers	Total Speakers
English	**337**	**309**	646
Mandarin	**54**	**97**	151
Spanish	**117**	**111**	228
Indian Languages	**80**	**71**	151
Hindi	18	14	32
Bengali	8	12	20
Oriya	2	0	2
Punjabi	10	2	12
Tamil	3	10	13
Urdu	17	10	27
Gujarati	7	10	17
Kannada	6	3	9
Konkani	2	1	3
Marathi	5	6	11
Tibetan	2	3	5
Total	588	588	1327

Figure A1 - A summary of the number of audio files we had per class from the Speech Accent Archive.

Catherine	Mingshi	Rocco	Julia
25% 1. Data Retrieval and Filtering 2. Data Processing Script (70%) 3. Data Augmentation Script (30%) 4. Development of CRNN Hyperparameters - CNN portion 5. Training and Hyperparameter tweaking of Spanish English Binary Classifier 6. Training and Hyperparameter tweaking of Arabic English Binary Classifier 7. Training and Hyperparameter tweaking of Indian Languages English Binary Classifier 8. Presentation: - slides for introduction - initial presentation outline -video and audio recording 9. Research on	25% 1. Visualization for pca and some audio clips 2 **video editing** 3 collecting test data 80% 4 SVM 5 Transfer Learning 6 resizing image script 7 made w/o spanish and trained and did experiments 8 experimentation on architecture CNN (only for no spanish) 9 script for testing our in the wild stuff (had a bug lol) 10 dimension reduction code Failures: 1 a lot of visualization (ran into a ram error) 2 testing on reduced dimensions using different functions and performance on those 3 4 5	25% 1. RNN architecture 2. CRNN architecture (with help) 3. Pitch-shifting (70%) 4. Gaussian noise (70%) 5. Quality-of-life improvements in data augmentation 6. Debugging training script for binary classification (validation error bug) 7. Trained India model (helped by failing) (30%) 8. Trained multi-class RNN 9. Multi-folder data loading (for augmentation) 10. Tried augmented melspec images on CRNN tertiary classification 11. Fully connected layer experimentation (CRNN) 12. Added dimension calculator to CNN/CRNN	25% 1. 1D CNN time series experimentation 2. LSTM and GRU experimentation 3. Running binary classification experiment (Mandarin) 4. Initial CNN architecture set up 5. Writing clip-based evaluation 6. Architecture experimentation (CNN, RNN, CRNN) 7. Hyperparameter tuning of CRNN models (learning rate, batch size, etc.) 8. Tested with dropout 9. Slides for architecture + uh some other parts lmao 10. Recorded audio and video for presentation 1. Mozilla dataset training 2. Implement new audio clipping method 3. Clip-based evaluation (english vs foreign)

Different CRNN and CNN architectures 10. Research on data augmentation and processing techniques 11. Development of initial CNN architecture Failures: 1. Autoencoder 2. Model implemented augmented data 3. CNN Multi class (4 classes) testing 4. CNN Multi class (3 classes testing) 5. Data augmentation to separate out the words		13. Adam optimizer experiments (vs SGD) 14. Presentation stuff: a. Collected shit b. Scripted stuff c. Made slides Stuff I didn't finish 1. Confusion matrix (switched to binary) 2. Porting librosa to Windows (complete and utter failure)	