Classification and Clustering of Ultrasound Tongue Images in Vowel Production—Project Report

Sam Fisher & Yen-Chen Lu

Background

Ultrasonography has become a widely employed tool in diverse research domains, encompassing articulatory phonetics, laboratory phonology, speech pathology, and second language acquisition, just to name a few. Its popularity lies in providing a non-invasive, safe, and real-time means of visualizing the intricate movements of the tongue. Traditionally, the analysis of ultrasound data was characterized by slow and laborious processes, involving contour tracing to quantify image data into x, y coordinates. However, recent studies such as Kochetov et al. (2019) and Faytak et al. (2020) are starting to explore raw-image-based methods for their efficiency. Our project is centered around the classification and clustering of ultrasound tongue images during vowel production, aiming to leverage these advanced methods. Specifically, we employ a Convolutional Autoencoder for vowel classification and explore Principal Component Analysis (PCA) and Uniform Manifold Approximation and Projection (UMAP) for visualizing vowel clusters.

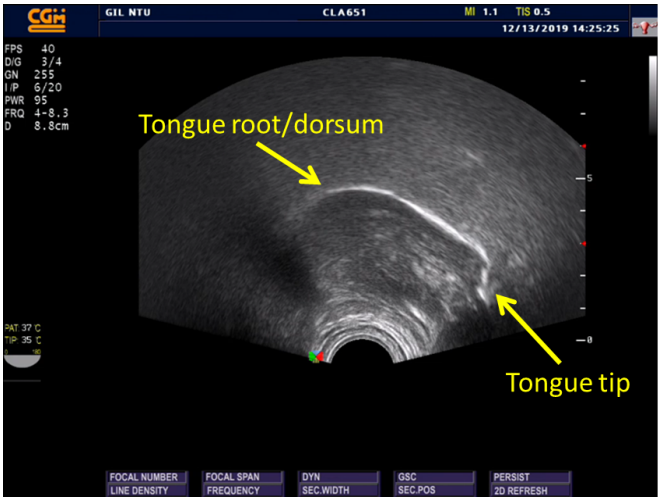


Figure 1. Demonstration of an ultrasound midsagittal lingual image.

Data Collection

We conducted our study with two native Mandarin speakers, one female and one male, from whom we acquired midsagittal lingual ultrasound images for six

vowels (/a i u e o ə/), each pronounced independently 100 times. The resulting dataset comprises 1200 trials (6 vowels * 100 repetitions * 2 subjects), with images extracted from the midpoint of each trial for subsequent analysis. The raw ultrasound images of the two speakers can be accessed in our repository, specifically from the folders sam_frames/ (containing raw images of the male speaker) and yenchen_frames/ (containing raw images of the female speaker).

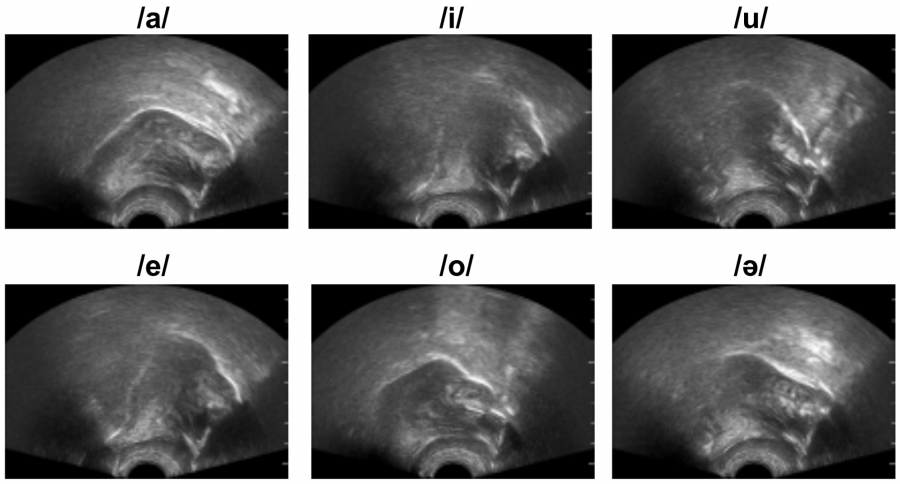


Figure 2. Representative images of the six vowels from one of the speakers

Data Pre-processing

We performed a series of pre-processing steps to transform the data into a format compatible with further analyses. This involved converting the images to grayscale, cropping them to exclude areas that do not contain meaningful information (e.g., image border), downscaling them to a lower 96 (height) x 140 (width) resolution, and, depending on the specific requirements of classification and clustering algorithms, flattening them to a 1 x 13440 vector of pixels and normalizing the pixel values to fall within the range of [0, 1]. These image preprocessing procedures are integral for subsequent analyses.

Results: Image Classification

We employed an autoencoder as a classifier, benefiting from its ability to learn a compressed representation that proves valuable for various downstream tasks, including the subsequent classification task.

The dataset, consisting of 1200 images, underwent an 80-20 split into training and test sets. Ensuring equal representation, images from both speakers were distributed evenly between the training and test sets.

As part of the preprocessing pipeline, the numpy array representing these images was transformed into a 3D numpy array (tensor). Consequently, the dimensions for the training set images became (960, 96, 140), and for the test set images, (240, 96, 140). To facilitate model input, each 96x140 image in both the train and test sets was converted into a matrix of dimensions 96x140x1, resulting in the shapes (960, 96, 140, 1) and (240, 96, 140, 1) for the train and test data, respectively. Furthermore, labels underwent one-hot encoding, for instance, transforming an original label of 0 into [1. 0. 0. 0. 0. 0.].

For the reconstruction task, the autoencoder was trained on the training dataset using an 80-20 train-validation split, with the objective of minimizing the discrepancy between the input and the reconstructed output. The training batch size was set to 64, and the model underwent 200 epochs. Upon completing the training of the autoencoder, we obtained the full model, comprising both the encoder and decoder pairs. The batch size for the full model remained 64, with a total of 100 epochs.

Visualization of the training and validation accuracy (Figure 3 and 4), along with the training and validation loss, demonstrated convergence over the course of 100 epochs. The model exhibited stability, and no overfitting was observed during the training process. Test results affirmed the model's accuracy, revealing a test loss of 0.095 and a test accuracy of 0.992. Impressively, only two incorrect predictions out of 240 were recorded, both associated with mid vowels (/ə/ and /o/). Figure 5 shows the instances where the model made incorrect predictions in vowel categorization.

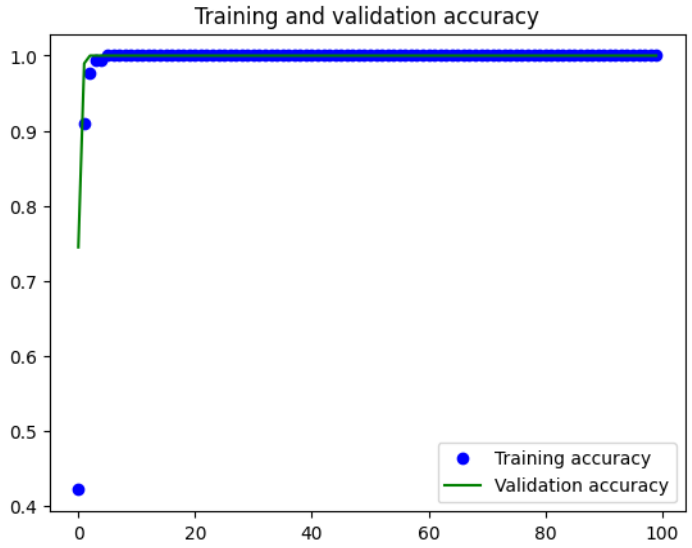


Figure 3. Training and validation accuracy plot

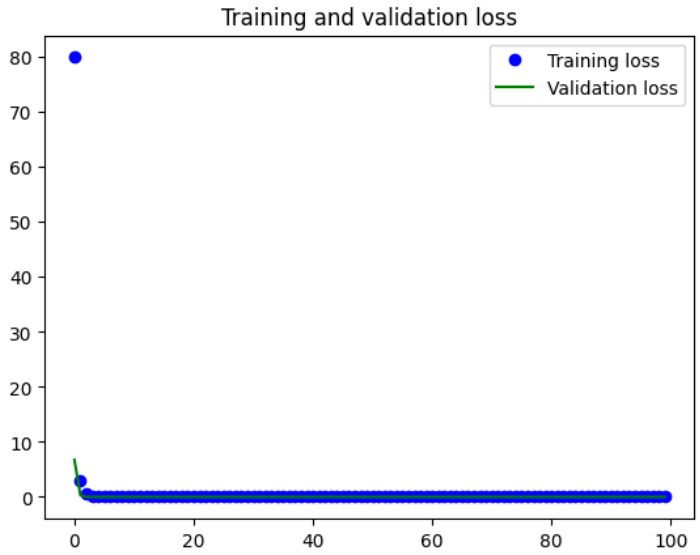


Figure 4. Training and validation loss plot

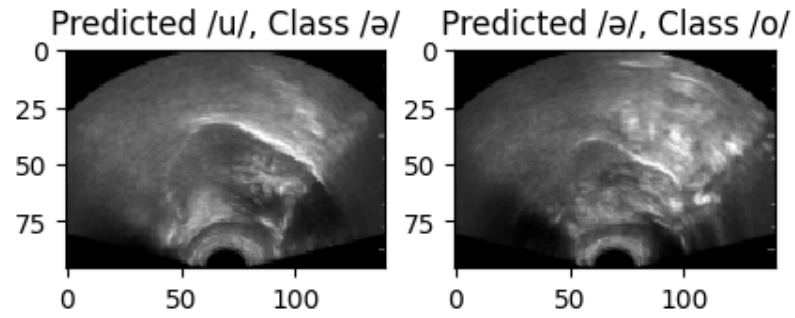


Figure 5. Two incorrect vowel predictions

Results: Clustering

We started our PCA analysis by overviewing how much of our dataset's variability is explained as we progressively include more principal components. The number of PCs needed to explain a certain percentage of the total variance (such as 80%) serves as a rough estimate of the intrinsic dimensionality of the data. Figure 6 shows that the male speaker requires a less amount of PCs compared to the female speaker for a given amount of explained variance, suggesting more consistent articulatory gestures in the male speaker's data. Generally however, this kind of ultrasound data seems to require roughly 30 PCs in order to account for 80% of the total variance, though there is a sharp plateau going above roughly 5 PCs.

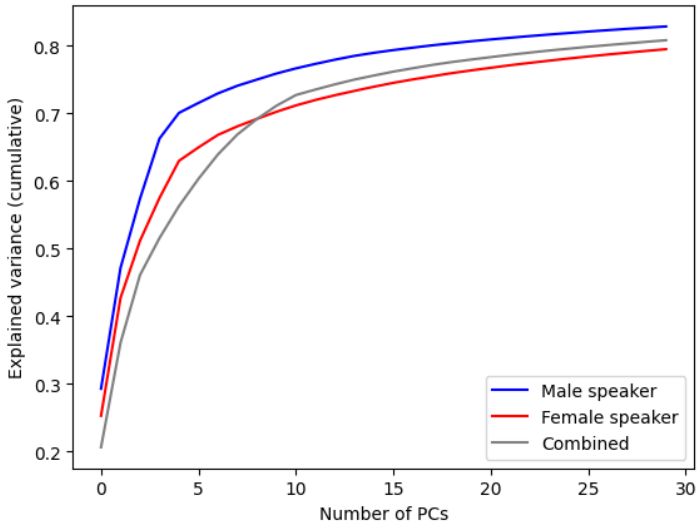
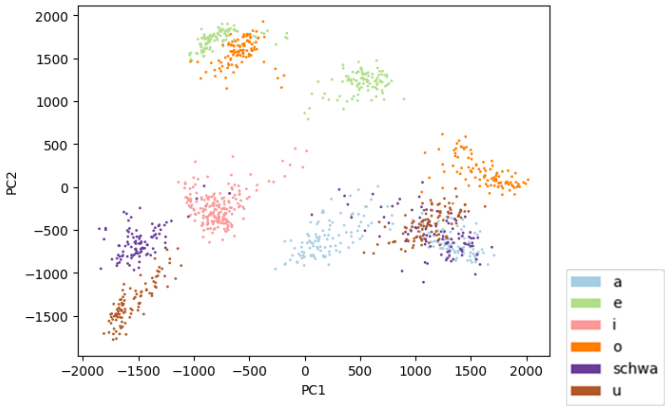


Figure 6. Cumulative explained variance of PCA analysis

We then plotted the first 3 PCs of the combined data (both speakers pooled together) as 2D scatterplots, as shown in Figure 7. PCA seems to do a decent job of teasing apart the majority of vowels, however for some vowels such as /u/ (brown) and /ə/ (purple) there appears to be significant overlap. We expect the overlap to be diminished as more PCs are included, however visualization will become difficult. Additionally, the same vowels from both speakers don't seem to be located near each other, indicating considerable inter-speaker difference in tongue posture during production of the same vowels.



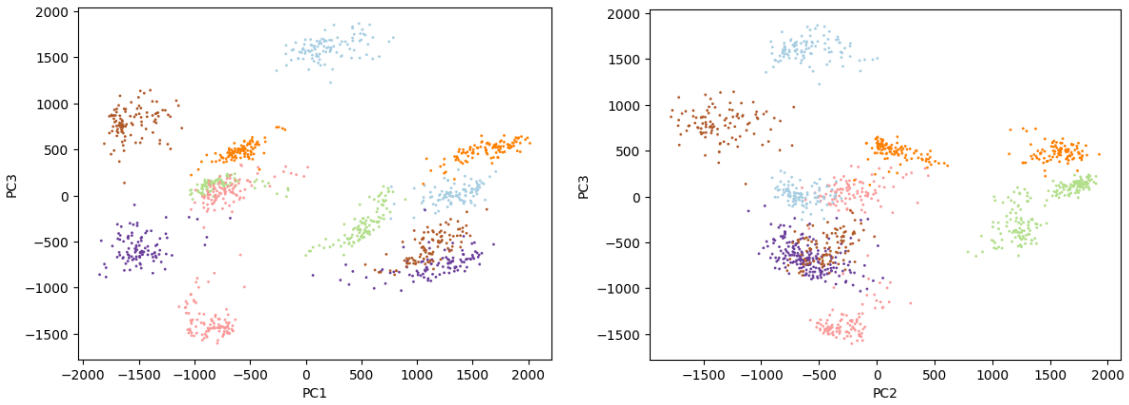


Figure 7. 2D scatterplots of the first 3 PCs of the combined data; each point represents an image; “schwa” represents the vowel /ə/

By using matplotlib’s `Axes3D` module and setting `%matplotlib widget`, we can create an interactive 3D plot of the first 3 PCs, allowing us to more easily get a sense of how the data points are clustered in space. Due to limitations in this document, here we only show a static 3D figure set with `%matplotlib inline`.

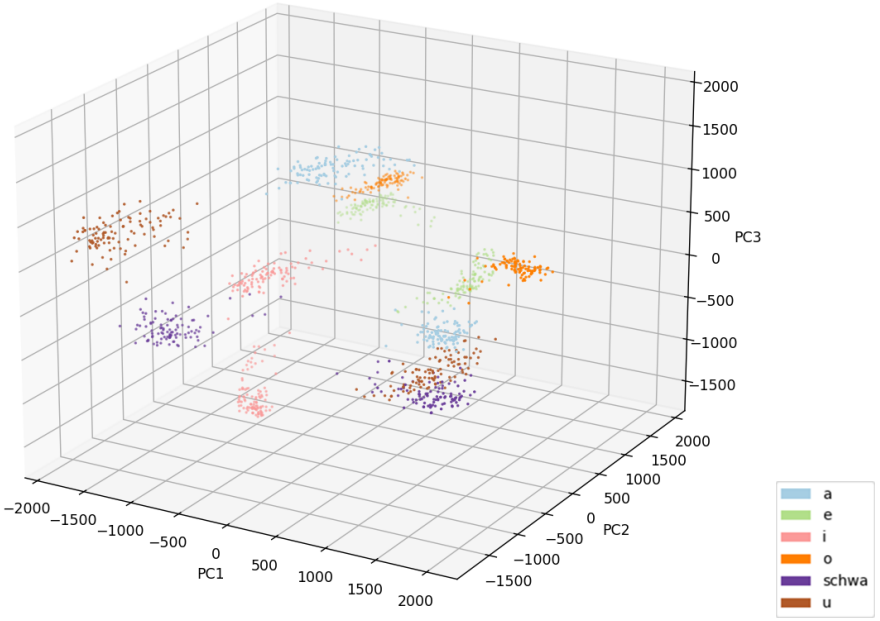


Figure 8. 3D scatterplot of the first 3 PCs

To better distinguish between different vowel categories, we analyzed our data again using a very recent and popular non-linear dimensionality reduction algorithm: UMAP (McInnes et al., 2018). UMAP estimates the latent structure of the data by calculating its local manifold properties and projecting the estimated manifold to a lower dimension. Non-linear methods can provide more faithful data visualizations in most practical applications. Compared to other well-known non-linear dimensionality reduction methods like t-SNE, UMAP offers significantly improved calculation performance. We use a variant of UMAP called densMAP which preserves local density information of the data (Narayan et al., 2021). Applying

densMAP to our combined dataset shows significantly better inter-cluster separation between vowel categories compared to PCA (Figure 9).

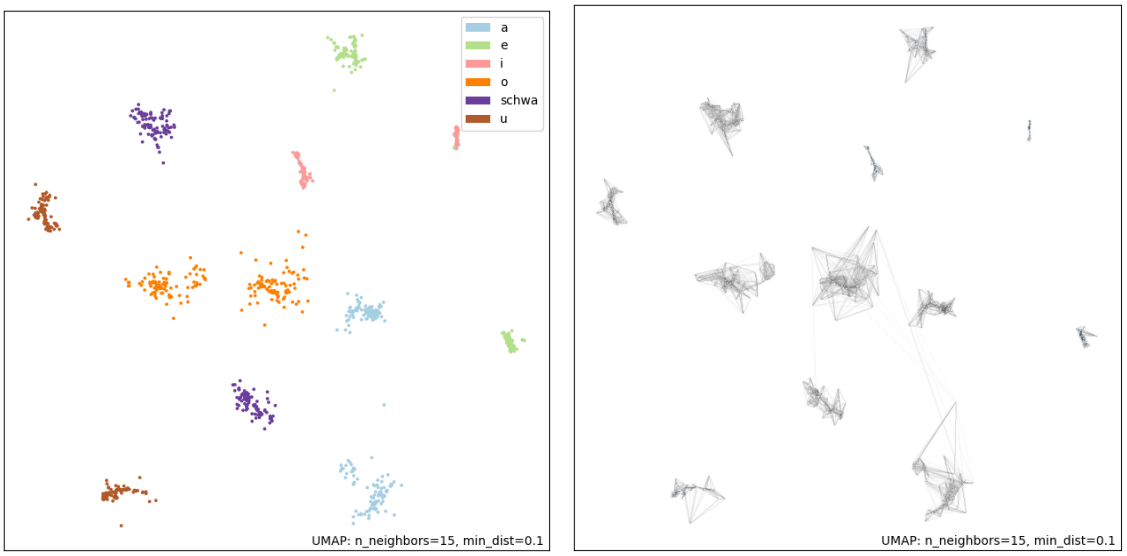


Figure 9. 2D densMAP of the combined data; figure to the right shows connectivity among data points

Despite this improvement, from the connectivity plot we can clearly see that most clusters are isolated, meaning the algorithm fails to find a connection between data points in different clusters. This makes interpreting the relative distance and locations of clusters very unreliable. We can improve the result by increasing the `n_neighbors` parameter. The highest possible value is the number of observations minus one, so we set `n_neighbors=1199` to maximize the benefit (Figure 10).

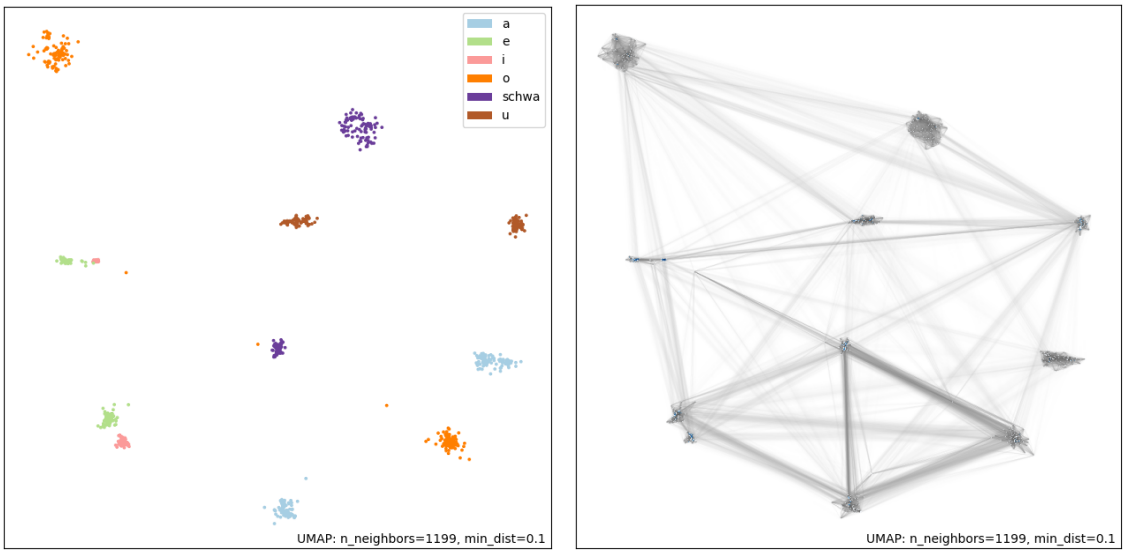


Figure 10. 2D densMAP with the `n_neighbors` parameter set to 1199 to maximize connectivity detection

The improvement in connectivity detection is substantial. Additionally, the cluster of each vowel appears to have shrunken, perhaps due to better global

optimization. We can see each vowel forms a distinct cluster, with virtually no overlap. There are, however, some straggler data points that lie closer to a different vowel category, providing a plausible explanation to some of the mis-classifications in the CNN autoencoder classification result. Similarly, some vowel clusters are very close to each other, especially /e/ and /i/, hinting at their similarity in terms of midsagittal lingual gesture. Yet just as the PCA result, the same vowels from different speakers are still placed separately, suggesting that the speaker-dependent lingual gestural difference is not caused by a specific algorithm.

Finally, we plotted our data in a 3D densMAP (Figure 11). Again, the result compared to PCA shows impressive cluster separation since a 2D densMAP already has virtually no overlap.

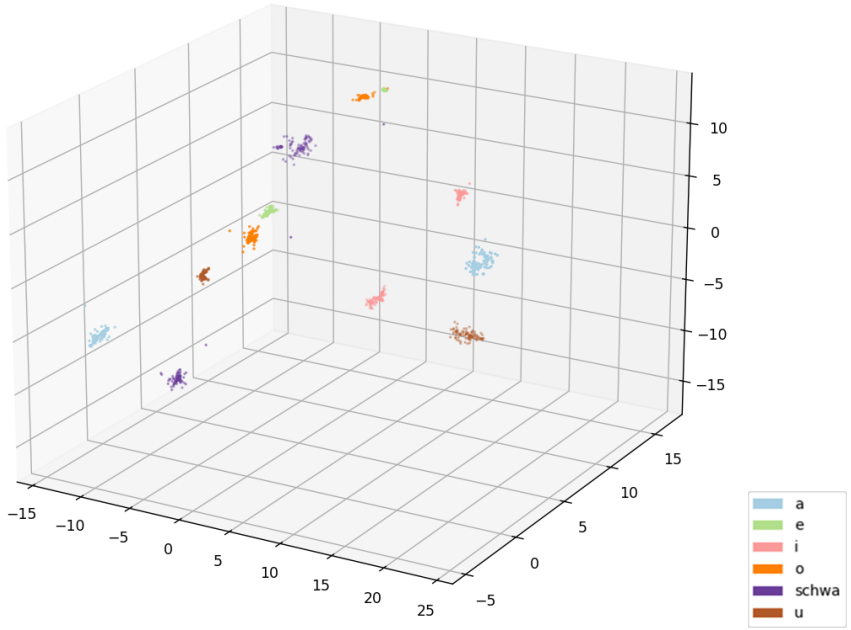


Figure 11. 3D densMAP of the combined data with `n_neighbors` set to 1199

Discussion

The remarkably high classification accuracy may stem from the simplicity of our dataset, which involved only two speakers and vowels produced in isolation, thus overlooking the coarticulation effects from neighboring sounds. Future studies should incorporate vowels from diverse sequences and test the model on datasets from other speakers to assess its robustness. If inclusion of more speakers and higher variety of data significantly reduces classification accuracy, it may also be promising to use densMAP as a preprocessing step before classification. Overall, our project lays the groundwork for leveraging raw-image-based methods in ultrasound analysis, with the potential for broader applications in various linguistic contexts.

Division of Work

Data Collection: Sam Fisher & Yen-Chen Lu
Data Pre-processing: Sam Fisher & Yen-Chen Lu
Classification: Yen-Chen Lu
Clustering: Sam Fisher

References

Faytak, M., Liu, S., & Sundara, M. (2020). Nasal coda neutralization in Shanghai Mandarin: Articulatory and perceptual evidence. *Laboratory Phonology, 11*(1).

Kochetov, A., Faytak, M., & Nara, K. (2019). Manner differences in the Punjabi dental-retroflex contrast: an ultrasound study of time-series data. In *Proceedings of the 19th International Congress of Phonetic Sciences* (pp. 2002-2006).

McInnes, L., Healy, J., & Melville, J. (2018). Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

Narayan, A., Berger, B., & Cho, H. (2021). Assessing single-cell transcriptomic variability through density-preserving data visualization. *Nature biotechnology, 39*(6), 765-774.