UNIVERSITY OF MINNESOTA
Driven to Discover℠

Pronunciation Tool for Learning Tonal Languages

Prepared for: EE 4951 Senior Design

Brendan Bagley, Kirsten Olson, Derrick Loke, Kiana Vang, Jake Wang
Advisor: Professor Gerald Sobelman
University of Minnesota, Twin Cities
Department of Electrical and Computer Engineering
Minneapolis, MN

December 20, 2020

Table of Contents

Executive Summary

The Pronunciation Tool for Learning Tonal Languages senior design project is a continuation of the Spring 2020 semester's software. This software gives visual and textual feedback on tone pronunciation in tonal languages such as Mandarin Chinese. In tonal languages, inflection means just as much to a word's meaning as its consonants and vowels. Producing tones is one of the hardest components of learning tonal languages for native English and other non-tonal language speakers since words with the same spelling but different tones can sound extremely similar.

The software works by first selecting a language, tone, and word in a visually appealing, simple graphical user interface. If desired, the user can listen to an audio sample of the selection. Next, the program takes in the user's pronunciation and converts it into a spectrogram by performing dozens of Fast Fourier Transforms (FFTs) and plotting the frequency over time. This is normalized to account for each person's unique pitch. Then this sample is compared to a database of native speakers' pronunciations, and the two samples are plotted on top of each other. The mean-squared error (MSE) of the two samples is calculated to give the user a numerical score and textual feedback on what specifically needs to be improved.

Furthermore, a type of artificial intelligence called a convolutional neural network (CNN) is created from the entire database of native speakers. This CNN is used to detect the tone the user produced. This provides feedback on if the pronunciation would be understood correctly by native speakers. It does not, however, give any hints on how to improve; that is why the team included the spectrogram plot and textual feedback.

While the team hasn't conducted user testing yet, in-house testing demonstrates that the software provides satisfactory feedback. The spectrogram plot is potentially the most useful aspect of the project; it shows more information than the mean-squared error's summarized score, and it gives more indication on

how to improve than the CNN. Additionally, humans are typically much better at pattern detection than machines, so giving the user the ability to visually compare their pronunciation against a native speaker's is a great way to provide feedback.

This project's feedback is what differentiates it from other similar products. After testing several smartphone apps for learning languages, the team found that Duolingo was the only app that recorded user audio and scored it. However, this score was merely a qualitative audio score of a phrase, and it didn't give any recommendations on how to improve or what went wrong.

Moving forward, there are many improvements that the team recommends. However, only two improvements are considered high priority. First, most of the native speaker databases are incomplete and must be expanded in order to use the CNN effectively. Mandarin is the only language with an adequate sample size; the Mandarin database has every character in all four tones spoken by three male and three female speakers. In contrast, the Hmong database only has one speaker, and the Vietnamese database doesn't even have every word. This means that the CNNs for languages besides Mandarin are ineffectual to the point of being useless. Second, the current program has several software requirements, including Python 3, various Python libraries, and Visual Studio. The end user should instead receive a simple one-click program with minimal requirements; to accomplish this, the Python program should be bundled into an executable that is easy to download and install.

Problem Definition

Background and Relevant Topics

The ability to speak more than one language possesses many benefits. It especially opens a gateway for one to engage in different perspectives, exchange cultural awareness, and increase their understanding of the world. However, not all languages are created equal. Some languages have multiple writing systems, while others only have one. For example, Japanese has three writing systems (Katakana,

Hiragana, and Kanji) whereas English only has one (the Alphabet). Some languages are only written, while others are a mix of written and oral. The languages that the team is particularly interested in for the senior design project are tonal languages, such as Chinese Mandarin, Vietnamese, Hmong and Thai. While a given word can be spelled the same, the tone tied to that specific word can give it an entirely new and different meaning. This presents many challenges to the person learning and teaching said language. The project expands on previous semester's team's work and aims to complete a pronunciation tool for learning tonal languages. Different techniques were used to achieve the final development of the tool, such as digital signal processing, statistical analysis, machine learning, and software engineering.

Keywords/Glossary

Convolutional Neural Network (CNN): a specialized type of neural network model used to analyze visual imagery.

Fast Fourier Transforms (FFT): a discrete digital signal processing technique that converts a signal from its original domain to the frequency domain, and vice versa.

Graphical User Interface (GUI): A software application interface that allows users to easily interact with features that are supported by the application (buttons, search bar, text bar, etc.).

Mean-Squared Error (MSE): a statistical method that computes the average squared difference between the estimated values and the actual value.

Pitch Contour: A simplified, scattered plot representation of a perceived pitch/sound over time. Also known as tone contour.

Spectrogram: A 2D visual representation of signal frequencies (y-axis) as it varies with time (x-axis); it is essentially a picture of a sound.

Tonal Language: A type of language where the meaning of a word is affected and conveyed by a specific tone. The number of tones in a tonal language varies from language to language.

Customer Needs Requirement and Discussion

Feature	Requirement
Compatibility	Tool works on Windows 7 and newer, macOS 10.10 and newer, Ubuntu 14.04.06 LTS and newer. GUI is consistent between operating systems.
Design	GUI should be graphically pleasing. GUI should be easy to use (buttons to navigate each page, select words/tones/languages). The user should have the option to display the written form of the word in the target language as well as the English description of the tone. GUI must be simple enough for all users to use without training
Functionality	Must be simple to select/change language, word, and tone The time between recording finishing and a rating appearing shall be no more than five seconds
Learning Mode	The user will have three learning modes to choose from, within a single language: 1. The user selects one specific word and tone 2. The user selects one tone and the tool generates words only with that specific tone 3. The tool generates random words and tones for the user to practice
Languages	The user will be able to practice words and tones from Mandarin Chinese, Vietnamese, and Thai. The tool should allow for the easy addition of other tonal languages such as Hmong.
Play Audio	The selected word and tone is played back to the user from an audio database of native speakers.
Record Audio	The user is able to record their own pronunciation.

	Pitch contours are displayed for both the native speaker's and user's pronunciation.
Scoring System	A score between 0 and 100 is displayed to the user based on accuracy of user pronunciation.
Feedback	Textual and/or visual feedback is provided to explain how the user can refine their pronunciation.
Repeatability	After the user records their pronunciation and receives feedback, the user is able to repeatedly attempt the pronunciation to improve on the previous score.

Product Design Specification

The current software design allows the user to:

- Choose a language
- Select a word to practice or let software randomize selection via "Shuffle" mode
- Play the native pronunciation
- Search for a word using text bar feature
- Record user pronunciation
- Classify the tone produced
- Visualize the pitch contours of reference and word produced
- Improve pronunciation given a rating score from 0 to 100 and textual feedback

Concept Design

The software tool is a continuation of a similar tool created by a Senior Design Project team in spring 2020, prepared by Abdirahman Abdirahman, Ali Adam, Hassan Ali, Elijah Nguyen, and Chee Tey and mentored by Professor Gerald Sobelman. The spring 2020 team found a method to use Python to convert an audio sample to the frequency domain and create a Mel spectrogram. The spectrograms of a Mandarin and Vietnamese audio database could then be fed through a CNN to

categorize tones. Additionally, this preliminary product had the capability to play and record audio files, run them through the appropriate CNN, and print out the predicted tone. The team found that this method has the capability of being significantly more accurate than linear or quadratic regression, which would potentially fail when different languages and tones were implemented; for example, Vietnamese has two falling-rising tones, which would confound a linear regression algorithm. The GUI of the spring 2020 program is shown in Figure 1. Using this proof of concept, the fall 2020 team is expanding on both functionality and usability.

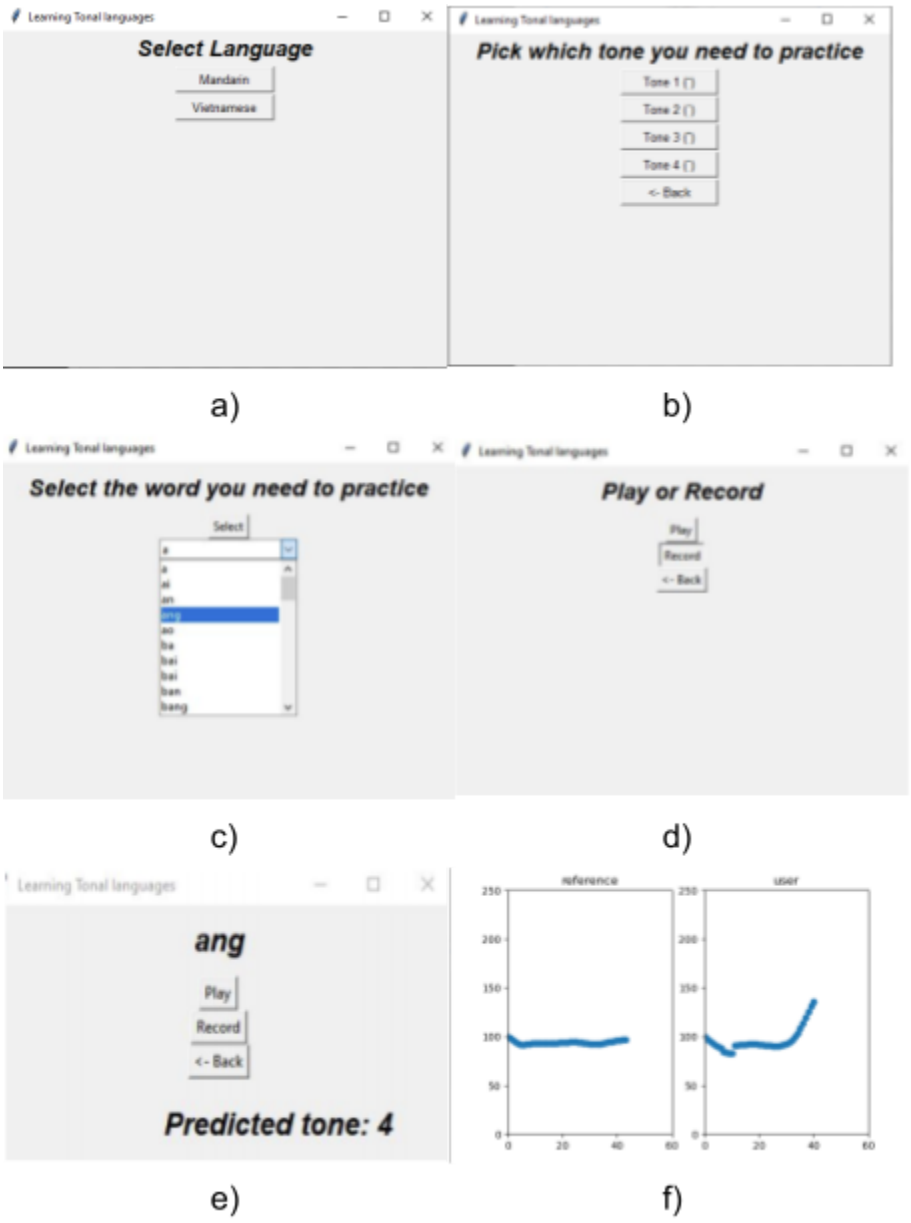

Figure 1. The five pages of the spring 2020 GUI, in order: language selection page, tone selection page, word selection dropdown, play page, and record page (a through e). The reference and user spectrograms plot is a pop-up window (f).

After identifying a method to categorize tones, the team researched external sources to discover similar products on the market. Of approximately a dozen language learning apps the team tested on the Apple App Store, the only one to let the user pronounce a word is Duolingo; however, as shown in Figure 2, this app

merely holistically rated the user's pronunciation but provided no guidance on how to improve. The team decided that this app's layout—specifically having the phrase, play button, and feedback on one screen—is an improvement over the preliminary project's GUI.

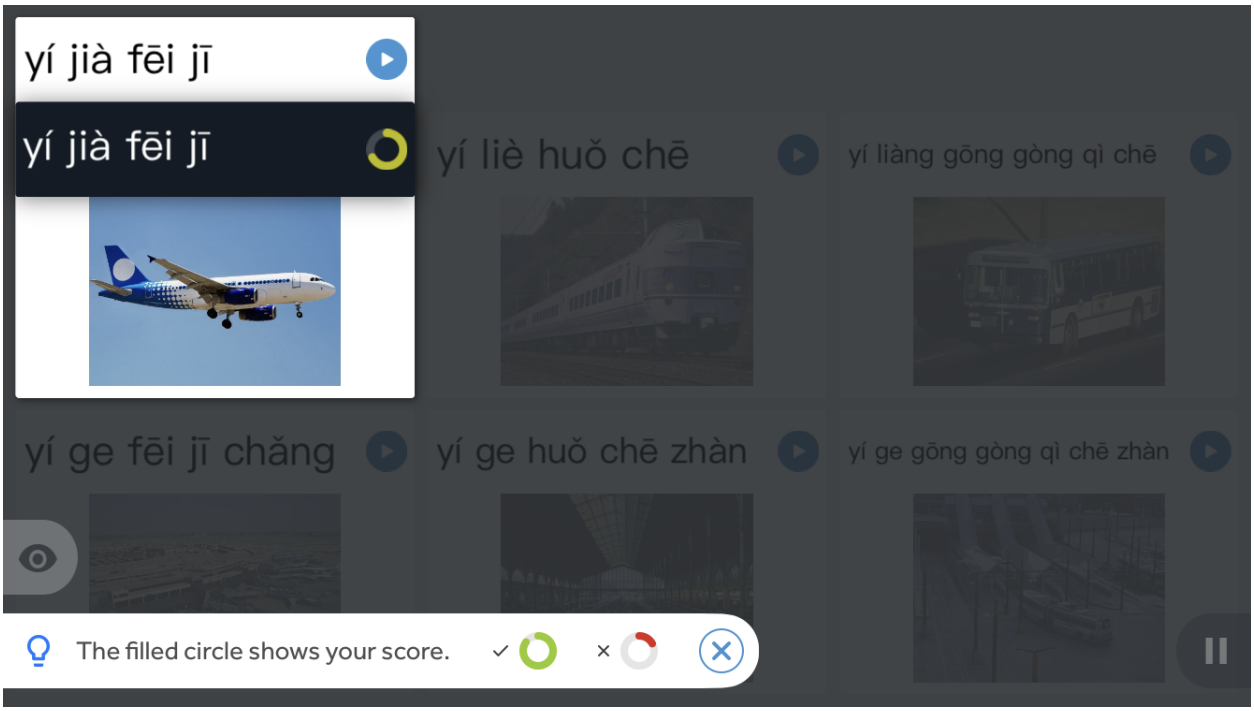

Figure 2. Rating on user pronunciation of a Mandarin Chinese phrase in the iOS language learning app Duolingo.

Using the lessons learned about a GUI, the team sketched a new potential layout for the program, depicted in Figure 3. Specifically, the new GUI should be fully contained within one screen, remove the pop-up spectrogram graphs, and be more aesthetically pleasing than the previous GUI.

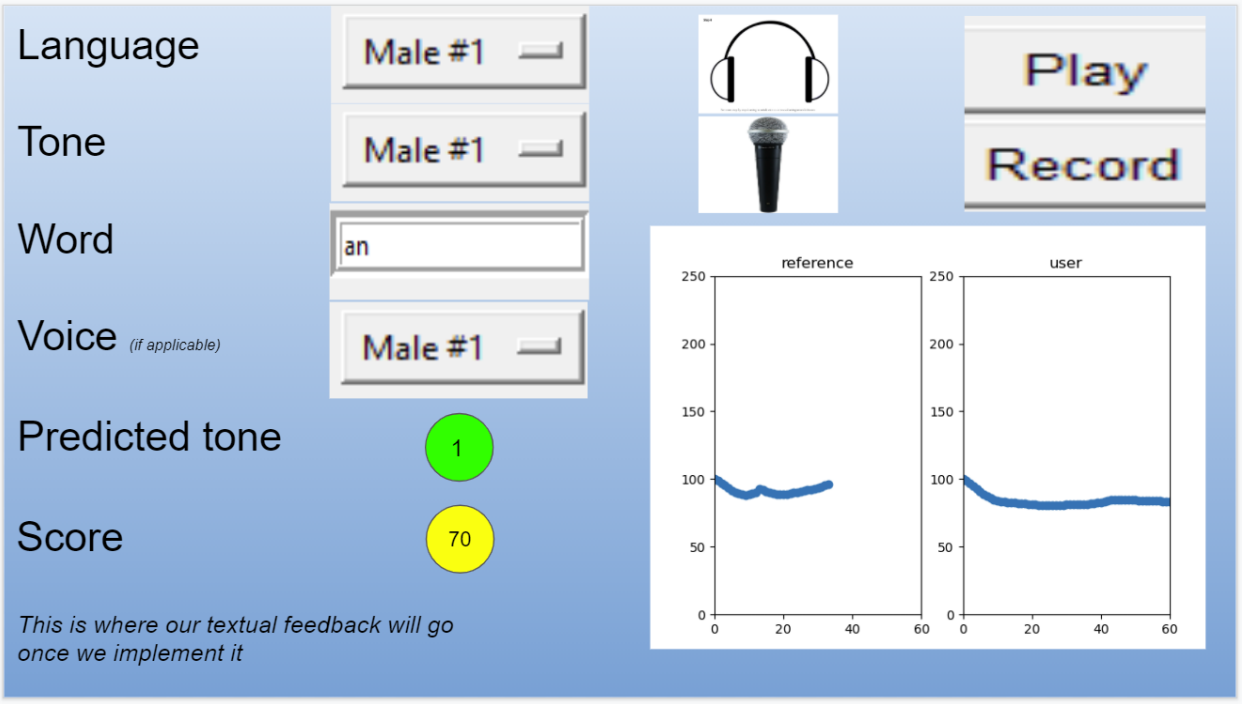

Figure 3. The sketch of the team's generated idea for the main program's GUI. The predicted tone bubble changes from green to red if the tone is correct, and the score bubble shifts on a spectrum from green to yellow to red depending on the numerical score.

Design Description

The tool is designed to help the user correctly pronounce words from tonal languages. All features of the program are displayed in a single window where the user can select their language, tone, and word they would like to pronounce. They have the option to hear the pronunciation of the word which is played from a dataset of words recorded by native speakers. The user can record their own pronunciation and is presented with feedback in the form of a visual representation of their tone, a tone prediction, a score between 0 and 100, and textual feedback. The tone is predicted using a CNN, and the score and feedback are based on the

difference in the user and reference pitch contours. A functional block diagram describing the flow of the program is shown in Figure 4.

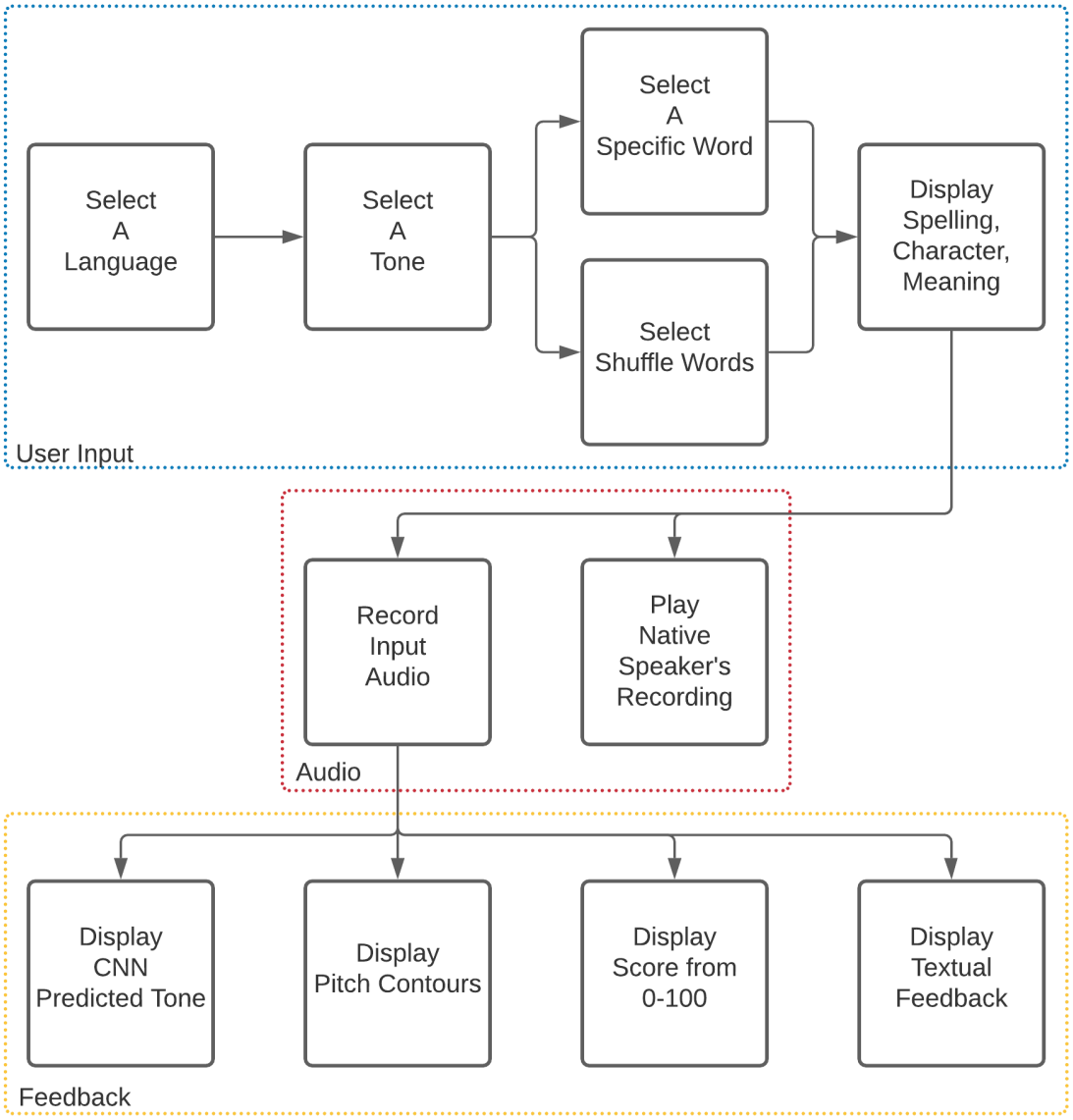

Figure 4. Functional block diagram of how the program functions. After a user inputs the language, tone, and word to practice, the program can either play a sample of a native speaker or record the user's pronunciation. After recording the user, the program analyzes the audio file by interpreting the tone using the CNN, graphically displaying the pitch contours, calculating the MSE and converting it into a numerical score, and providing textual feedback accordingly.

Prototype Construction Details

GUI: All of the program's features and functionality are displayed in one window to make the program easier to use. The user can change their language, tone, and word without having to navigate through several screens, and the pitch contours and feedback are all displayed in one area. Words can be selected by scrolling through a drop-down list, typing in a search bar, or selecting the shuffle button which chooses words at random. For each word, a romanized spelling is displayed along with an English translation.

Sample normalization: When determining the tone of a recording, only the pitch contour matters; the duration and starting pitch should not have a significant impact on the tone. For example, the tool must recognize that someone with a naturally higher voice or someone that speaks more slowly than the reference audios is not necessarily pronouncing the tone incorrectly despite their pitch contour being different. In order to address this, reference and user pitch contours were normalized in two ways.

First, when generating the user and reference pitch contours used in the feedback system, all samples in both are scaled such that their starting pitches are the same. This compensates for the differences in natural speaking pitch between people. The pitch contours are scaled multiplicatively rather than additively in order for the relative proportional distances between samples in the same pitch contour to stay the same. This makes it easier for the user to visually compare their pronunciation with the reference pronunciation. It also makes it easier to implement the scoring and feedback system.

Next, the pitch contours are normalized along the x-axis such that they are the exact same length when placed on a graph. This compensates for the difference in speaking speed between people. For training the CNN, all audio files are normalized to a predetermined number of samples. When a user audio file is run against the CNN, it is also normalized to that same number of samples. For the feedback

system, the user pitch contour is always manipulated to be the same length as the reference pitch contour before analysis is done. In cases where stretching is required, interpolation is used.

Convolutional Neural Network: A CNN is used in order to classify a user's recording into a tone family within the language that the user is practicing. A CNN is a class of deep learning algorithms that is widely used for processing images.

In order to train the CNN, all of the reference audio files for each language are used to generate spectrograms. These spectrograms are very similar to images, which is why a CNN was chosen as the machine learning algorithm for this tool. A 70/30 train and test data split is used to train the CNN over 16 epochs. A separate model is generated for each language.

Once a CNN model has been made, it is used to classify user recordings. When a user records an attempt, the recording is normalized to a predetermined number of samples, it is used to generate a spectrogram, and the CNN predicts a tone using that spectrogram.

Scoring/Feedback system: After the user attempts their own pronunciation of a word, the program generates a scatter plot containing graphs of both the user and reference pitch contours (normalized in time and starting pitch). It then calculates the mean squared error (MSE) between the data points for the two pitch contours using Equation (1). The MSE value is then translated into a score from 0 to 100 based on the accuracy of their pronunciation.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2 \qquad (1)$$

In order to translate the MSE value into a score, several pronunciations of various tonal words were made and their respective MSE values were recorded. The

pronunciations that were judged as “good” were given a high score and the ones judged as “bad” were given a low score. In Excel, a table of the MSE values and their corresponding scores was created, and the resulting trendline function from the plotted graph was used as the scoring function in the program. Because some tones are more difficult to pronounce, such as tone 3 compared to tone 1 in Mandarin, the team decided to use separate scoring functions for each of the tones. The tones that are easier to pronounce have scoring functions that grade more harshly.

The feedback system is also based on the difference between the user and reference pitch contours. In order to determine which part of the tone the user needs help pronouncing, the difference in pitch contours is calculated for both the beginning and the end of the tone. Based on which difference is greater, feedback is presented to the user to help them more accurately match the reference pronunciation. For example, if the end of a user’s tone is much lower than the end of the reference’s, textual feedback will suggest that the user make the change in their tone more dramatic. Additionally, more intuitive hints and tips are presented to the user that connect the tones to sounds they are more familiar with. As an example, the Mandarin tone 2 starts low and ends high, and the textual feedback will tell the user that this tone should sound like they are asking a question which is a sound all English speakers are familiar with. This type of feedback is customized for each tone.

Design Evaluation

Since the project is software-based and fairly qualitative, the team assessed the design and functionality themselves based on the customer requirements document. The finished interface is shown in Figure 5. The team determined the visual design satisfies the design requirement of the customer requirements

because the entire program is displayed in a single screen, it is easy to change any setting, and the buttons, fonts, and graph all resize when the window is.

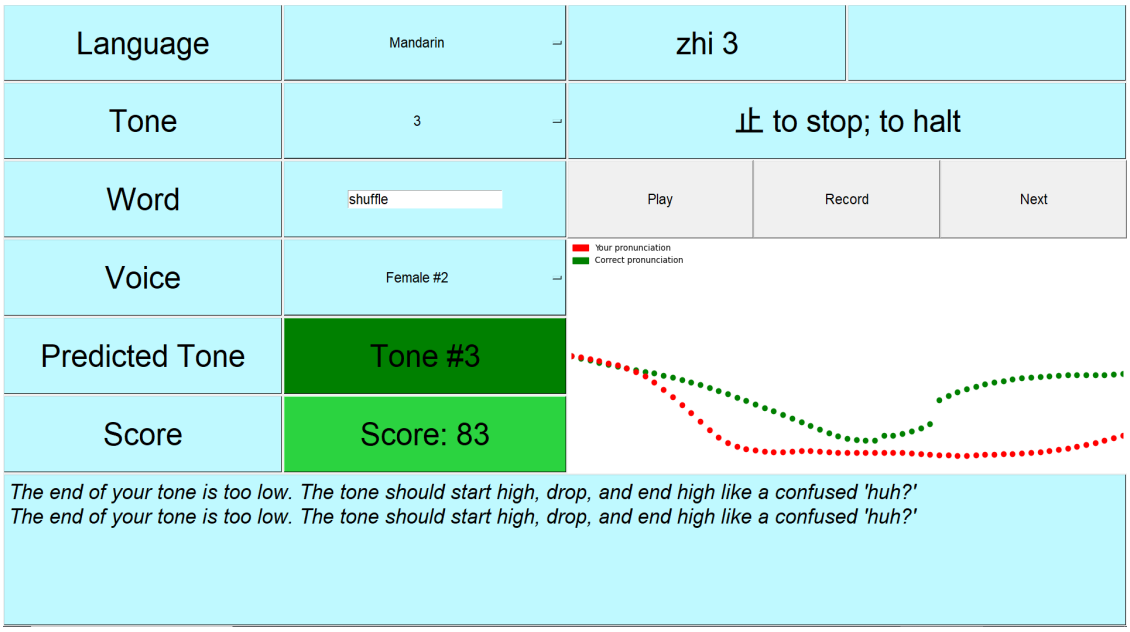

Figure 5. The program's display after making the appropriate selections and recording the user twice. Since the predicted tone (using CNN) is the same as the desired tone, the box is green; since the numerical score is relatively high, the box is light green. The GUI displays appropriate textual feedback below indicating how to improve the last attempts.

To ensure robustness and accuracy of the team's CNN implementation and results, the team performed data augmentation on their original audio libraries. For a given library, four more augmented audio files are generated: 2 whole steps down, 4 whole steps down, 2 whole steps up, 4 whole steps up. As a result, the audio libraries are five times larger than their original allowing for better training and testing of CNN models. After in-house testing, the team concludes that the CNN augmented with more data is significantly more accurate than with the original dataset. The CNN predicted tone, numerical feedback, textual feedback, and spectrogram plot are determined to be satisfactory for the feedback requirements.

Conclusions and Recommendations

Strengths and Weaknesses

Strengths:

1) Current GUI is simpler and easier to use. All features are supported in one window
2) A score rating between 0 and 100 is displayed along with textual feedback and a graphical representation of the reference and user's pitch contours, all of which allows for specific user pronunciation correction and improvement
3) "Shuffle" feature allows randomized selection of words
4) In-text search bar allows for easy search and selection of word
5) (Specific only to Chinese) An English definition is provided for each word along with its written form/character in the mother language.

Weaknesses:

1) GUI: The selected voice box and the box of definition of Chinese words are still displayed even though they are not in use.
2) Software: It is not easy for users to install all needed libraries for the tool, and the team has not had any success converting the project into a single-click executable file.

Future Work

Moving forward, recommendations for improvement and future work include multiple parts. First, the GUI's features can be further improved by administering user-testing and obtaining user feedback. The team did not perform any user-testing due to the complexity of the internal review process. Before any user-testing can be done, the team must prepare a formal report of the project and receive approval from an internal review board. Due to time limitations, the

team did not proceed with this. However, if given more time, the team believes that user feedback will allow for better development of the GUI resulting in an overall better experience for users.

An added feature that the team recommends adding is quiz mode. This feature tests users' pronunciation given any chosen amount of random words with random tones. In other words, the user will choose the number of words they want to practice and the GUI will randomly select that amount from the audio libraries and test them. This quiz mode feature expands usability making the GUI more purposeful and intuitive.

The team recommends soliciting help from native speakers to expand the current audio libraries. This will allow for better training and testing of CNN models. Thus, resulting in providing a more accurate score, visual and textual feedback to users.

Lastly, the team recommends expanding this project from a desktop application to a mobile application allowing users to carry this tool with them wherever they go. This further allows for better and easier accessibility and usability since mobile devices are commonly used on a daily basis.

Delivery and Cost

Overall, this project was completed on schedule once the team decided to eliminate the mobile application requirement as it would require a completely unique program compared to the PC/Mac version.

There are no costs and manufacturing or fabrication issues since the project is completely software designed. While there was a $400 budget set aside for this project, it was not used since all team members used their own devices (PC or Mac, built-in speakers and microphones or headsets) and no other costs were associated with the project.

The team considers the project resolved because it has successfully achieved working functionality for four main components of the feedback system: 1) produce and show a numerical score to the user based on their pronunciation, 2) display a visual graph of both the user and reference pitch contours, 3) improve the CNN model for a more accurate tone prediction, and 4) provide textual feedback to the user on how to improve their pronunciation.

References

Native speaker audio database sources

- Mandarin Chinese - MSU Tone Perfect
- Vietnamese - DLIFLC
- Thai - Tuttle Publishing via UC Berkeley
- Hmong - Original audio database created by Kiana Vang for this project