Speaker Identification

A Text-Dependent Speaker Identification System

Hani Mohammed, R. Kumar, V.Y. Karthik, S. Bharath

December 15, 2018

Abstract

Speaker Identification is rapidly gaining popularity with the rise of Voice Assistants and the like. Personalization of such products often require Speaker Identification. There are numerous issues that need to be addressed before Speaker Identification can be implemented on a large scale. At the highest level, all speaker recognition systems contain two main modules:*feature extraction and feature matching.* One popular feature extraction technique, MFCC, is often susceptible to noise. MFCC is used in many speech identification applications and has a range of other applications. VQ, a classical feature mapping technique, works by dividing a large set of vectors into groups having approximately the same number of points closest to them. In this project, a user authentication system has been realised with a second-layer of security by speaker identification, implemented by training a single recording of a passphrase upon registration and testing a real-time recording against the trained model to authenticate user access using Matlab.

1 Introduction

The sound of each individual's voice is entirely unique not only because of the actual shape and size of an individual's vocal cords but also due to the size and shape of the rest of that person's body, especially the vocal tract, and the manner in which the speech sounds are habitually formed and articulated. This forms a strong foundation for the use of voice as a biometric. Although voice may not be the best form of biometrics in terms of security, it has a lot of other advantages like the need of zero physical contact, ease of use and it's inherent property. Hence, to simulate a real-world application, we have implemented speaker identification only as a second layer of security with the first one being a textual password using Matlab.

Speaker Identification can be broadly categorised into two types: modules: *Text-Dependent and Text-Independent.* In this project, we have focused on the Text-Dependent application,i.e., identifying the speaker as well as what is being said.

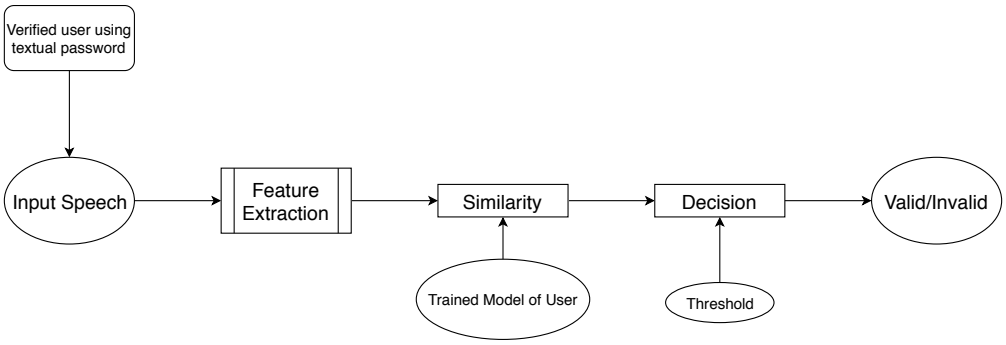

Figure 1: Basic Structure of User Authentication using Speaker Identification as a second layer of security.

2 Procedures

The two main modules of this project are the extraction of Mel-frequency cepstral coefficients (MFCCs) and pattern matching with Vector Quantisation (VQ). The other modules such as speech pre-processing enhance the performance of the main modules.

2.1 Audio Pre-Processing

The first step that is done on the audio file is denoising. This is done with the `ddencmp` and the `wdencmp` functions from the Wavelet Toolbox.

Now that we have a denoised signal, we significantly trim the beginning and end of the signal by taking one frame at a time using the `remove_silence` function. A minimum threshold is set for the maximum amplitude of the signal in that particular frame. All of the frames from the beginning to the one where the threshold is exceeded are excluded. The end of the signal is trimmed in the same manner.

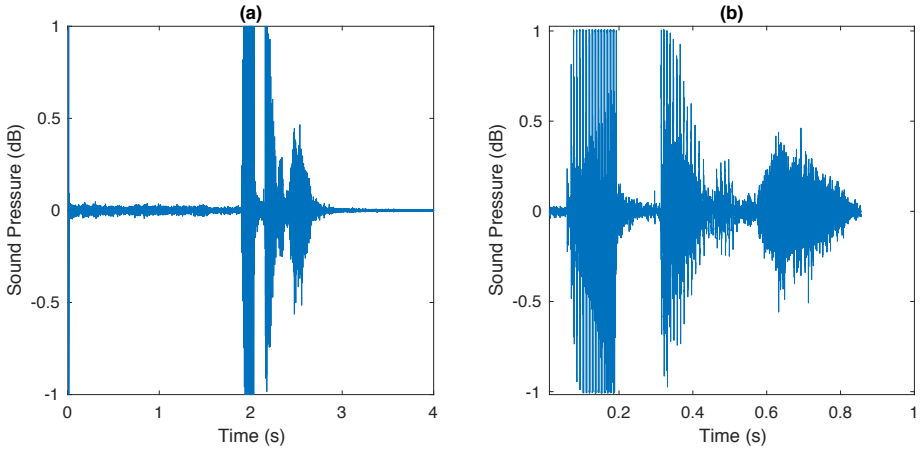

Figure 2: Audio Pre-Processing: (a) Input Audio Signal and (b) Denoised, Trimmed Audio Signal.

2.2 Mel-frequency cepstrum coefficients processor

A block diagram of the structure of an MFCC processor is given in Figure 3. The speech input is recorded at a sampling rate of 22500 Hz. This sampling frequency was chosen to minimize the effects of aliasing in the analog-to-digital conversion. These sampled signals can capture all frequencies up to 5 kHz, which cover most energy of sounds that are generated by humans. The main purpose of the MFCC processor is to mimic the behavior of the human ears.

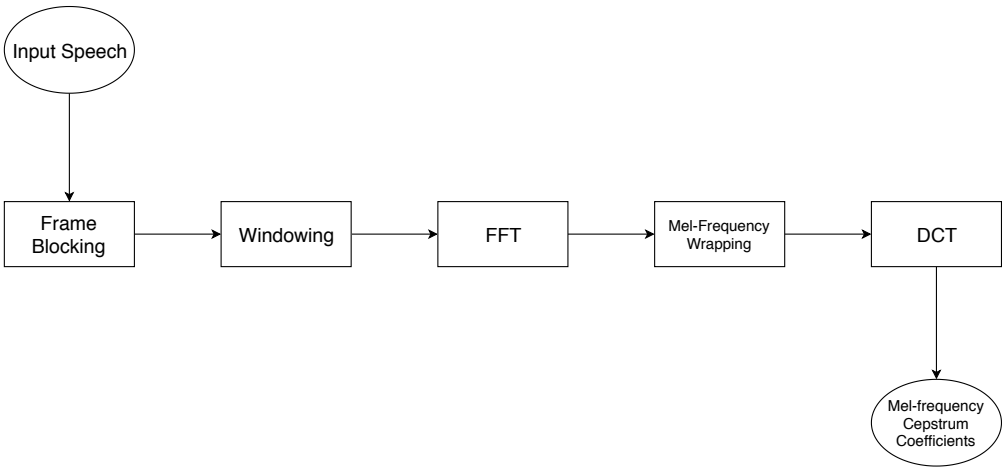

Figure 3: Structure of the MFCC Processor.

2.2.1 Frame Blocking

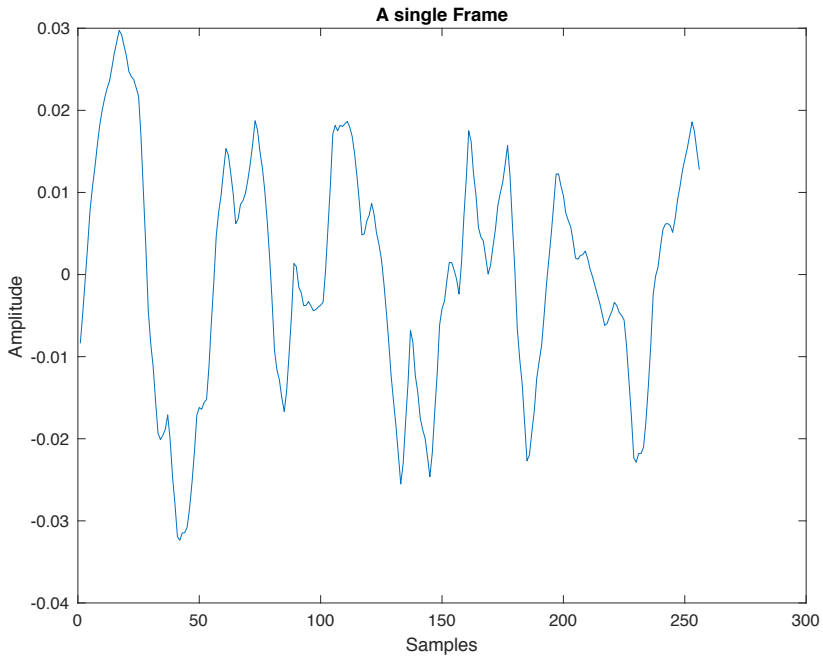

Figure 4: A single frame after Frame Blocking of 256 samples is applied

2.2.2 Frame Windowing

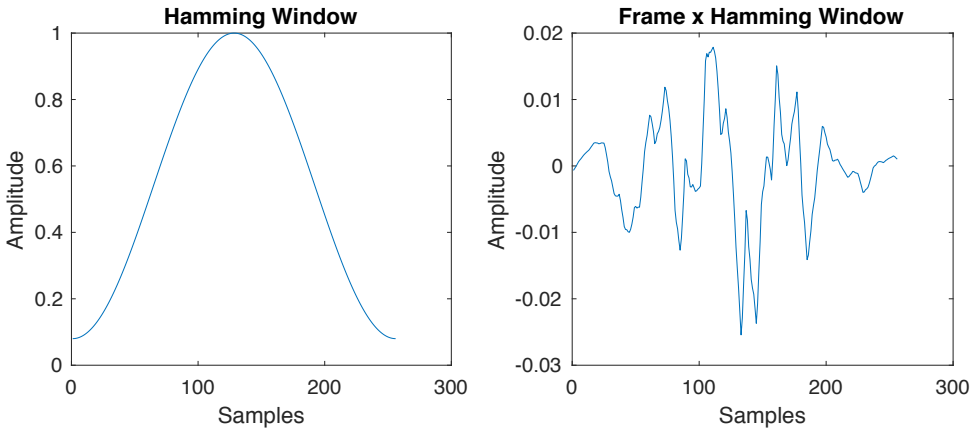

Figure 5: The hamming window for a frame and the frame after application

2.2.3 FFT

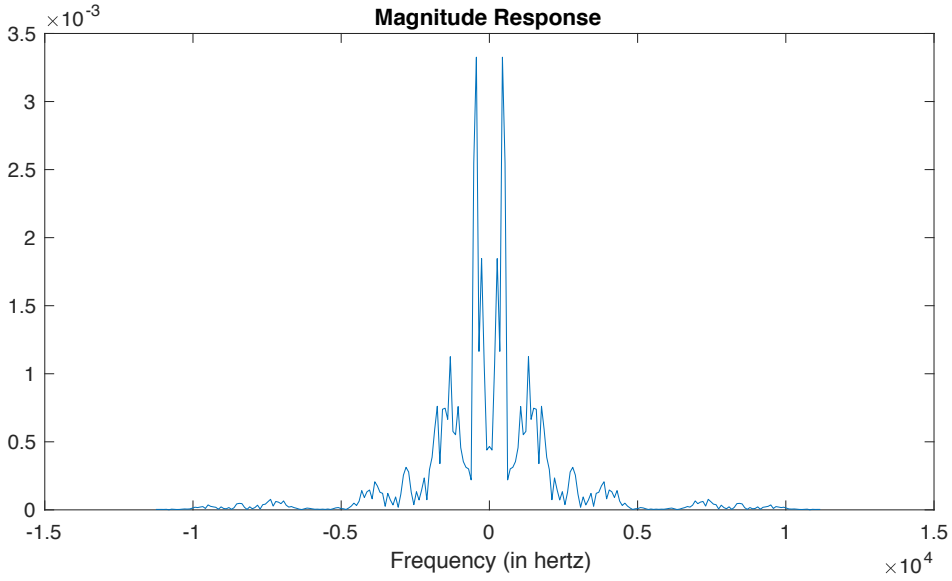

Figure 6: Frame after applying Fast Fourier Transform.

2.2.4 Mel-Frequency Wrapping

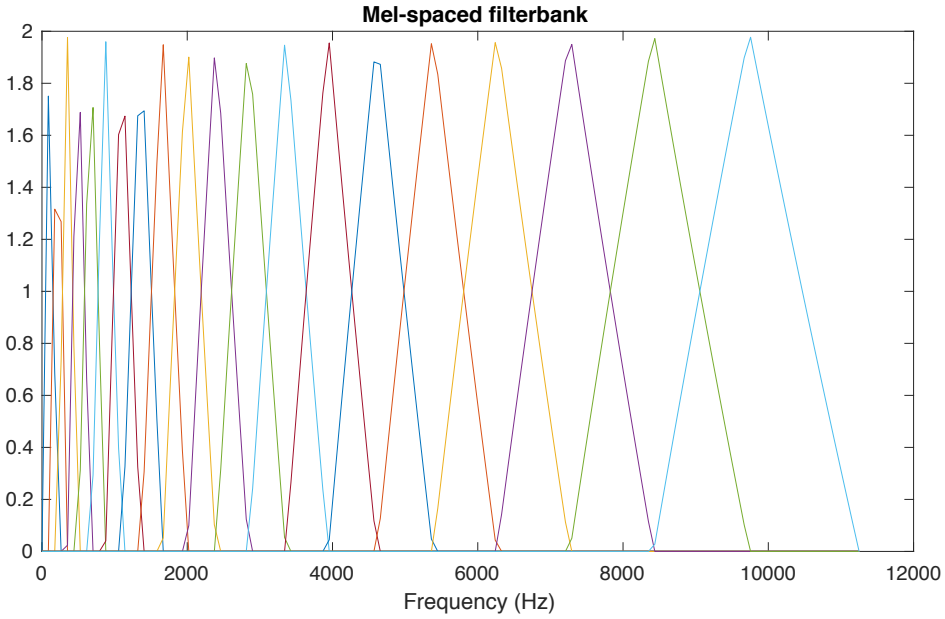

Figure 7: The twenty mel-spaced filterbanks generated for each frame.

2.2.5 DCT

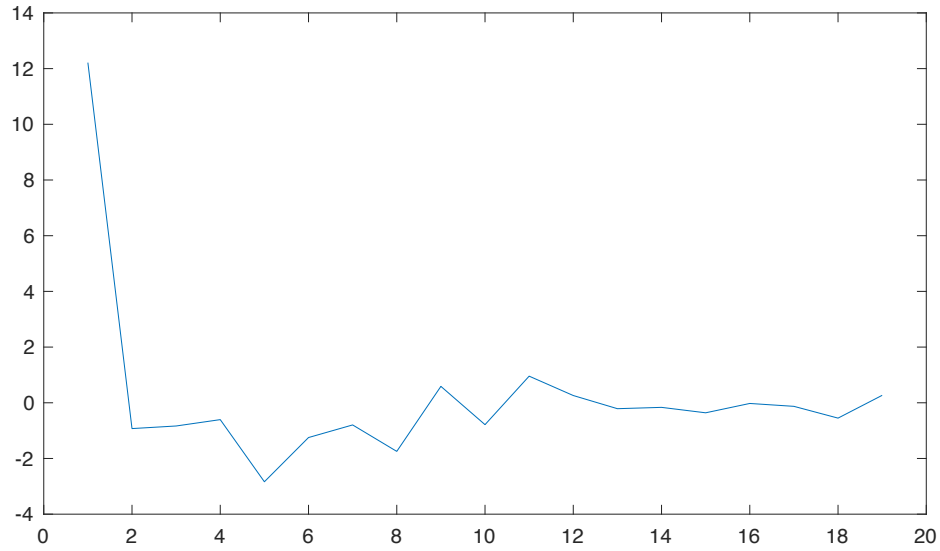

Figure 8: The Values of the 19 mfc coefficients (the first one is ignored as it contributes very little to the speaker information).

Finally, the values of the 19 mfc coefficients, all superposed are shown below.

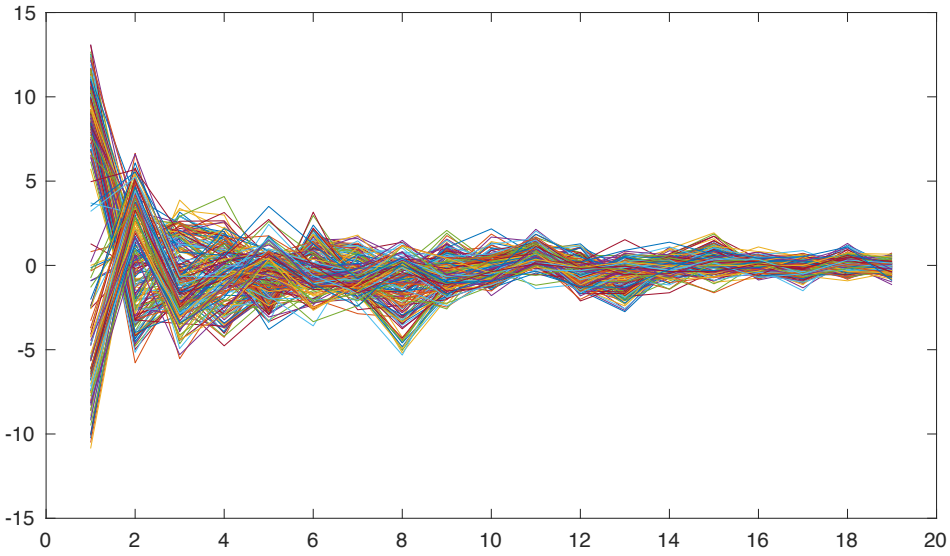

Figure 9: The clustering of an mfcc coefficient

3 Vector Quantisation

The extracted MFCCs are clustered using this technique. VQ has been used due to it's ease of implementation and high accuracy. In this process vectors from a large vector space are mapped to a finite number of regions in that space. Each region is called a cluster and can be represented by its center called a codeword. The collection of all codewords is called a codebook. Here, we have defined 16 clusters for the features to be mapped to.

When the speaker is to be identified, his audio signal is recorded, pre-processed and then the mfcc coefficients are extracted. Next, the total distortion is computed with repect to the codebook generated in the registration period. If this distortion falls below a minimum threshold (4.65 in our project), the speaker is authorised and given access. Otherwise, access is rejected.

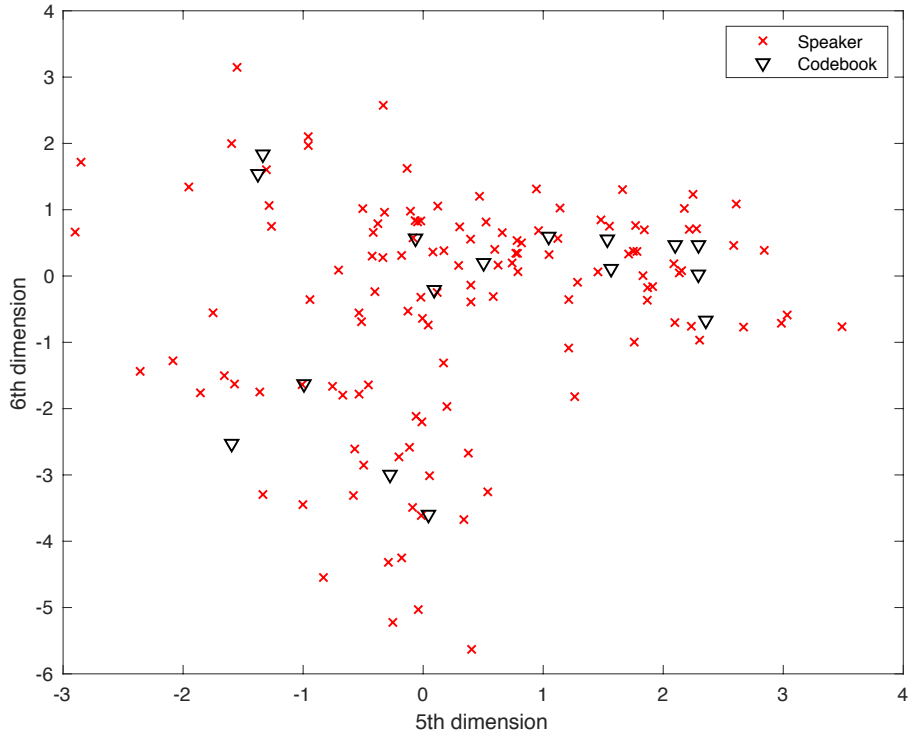

Figure 10: The clustering of two dimensions of the mfcc coefficients into 16 clusters

4 Analysis of the Results

4.1 Dataset

The dataset used has three speakers, Jackson, Nicolas and Theo, each of whom pronounces the English digits from zero to nine, fifty times each, thereby giving us a total of 1500 audio files. The files are in the Waveform audio file format (.wav) and sampled at 8 kHz. The recordings are already trimmed for minimal silence in the beginning and the end.

4.2 Procedure

A single pronunciation of each digit of Jackson is trained and tested against all other pronunciations of every digit including the same one for each of the three speakers. The results were then tabulated and analysed.

4.3 Results

Jackson x Jackson

Out\ Rec	0	1	2	3	4	5	6	7	8	9	Total
TP	20	16	14	11	13	33	21	35	12	7	**182**
TN	450	430	450	450	450	450	448	443	450	450	**4471**
FP	0	20	0	0	0	0	2	7	0	0	**29**
FN	30	34	36	39	37	17	29	15	38	43	**318**

Figure 11

Jackson x Nicholas

Out\ Rec	0	1	2	3	4	5	6	7	8	9	Total
TP	0	0	0	0	0	0	0	0	0	0	**0**
TN	500	500	500	500	500	500	500	500	498	500	**4998**
FP	0	0	0	0	0	0	0	0	2	0	**2**
FN	0	0	0	0	0	0	0	0	0	0	**0**

Figure 12

Jackson x Theo

Out\ Rec	0	1	2	3	4	5	6	7	8	9	Total
TP	0	0	0	0	0	0	0	0	0	0	**0**
TN	500	499	500	500	500	468	497	474	482	500	**4920**
FP	0	1	0	0	0	32	3	26	18	0	**80**
FN	0	0	0	0	0	0	0	0	0	0	**0**

Figure 13

	True Positive	True Negative
Predicted Positive	182	111
Predicted Negative	318	14389

Figure 14: Confusion Matrix

4.4 Inferences

From the confusion matrix, we obtain a 97.14% accuracy and a near perfect specificity of 99.23%. Since we are dealing with security, the false positives are the most important aspect which should ideally be zero. This is moderately achieved by our system although there is still scope to improve this.

From experimentation, it has been found that passwords work best when they are a word or two in length, approximately 750 ms in length, and have characteristic phonetic sounds. More words in the password average out the phonemes making it non-characteristic. Near-close front Unrounded Vowel i and Voiced labiodental fricative v are most prone to be mistaken. The microphone distance limit is found to be 60 cm in a low noise environment.

On further experimentation, the VQ process is found to be almost ten times faster than the Dynamic Time Warping technique with 0.043925s to train a model and 0.0062s to test one. The DTW also does not have any significant improvements when compared to Vector Quantisation. Time is of importance as we cannot hold the user waiting to be authenticated for a long time. The system passes this constraint effectively too.

5 Obstacles

The first obstacle that we faced was the false positive detection of a blank, noisy input whenever it was passed. This was a major issue as the whole system could collapse if this was not rectified. To tackle this problem we wrote our own `remove_silence` module and to trim the audio signal from both ends until a significant sample is attained.

Next, we faced the issue of text independency. This was tried to be resolved by using DTW technique on the MFCC coefficients which did not provide any progress but rather

decreased it fatally with a tenfold increase in the computational time. Next, We tried integrating both the VQ and DTW techniques which also failed. Finally, we decided to go with a stricter threshold for validation and noise removal using the functions `ddencmp` and `wdencmp` from the Wavelet Toolbox which gave us a significant improvement.

Finally, the GUI was a little difficult to be integrated especially since we had real-time audio recording and registration. It was quickly resolved with upon learning more about the `GUIDE` module in Matlab.

6 Scope

This project has only uncovered the potential of speaker identification and much of it remains unexplored. For e.g, we could train multiple audio signals instead of just one and combine them which could tremendously improve the accuracy. Also, machine learning could be added so that the trained model can be updated upon every successful attempt so as to accommodate even the tiniest of changes. The number of tries can also be increased with an algorithm to further stricten the threshold upon every failed attempt. A multi-word phrase could be effectively windowed into single words, thereby eliminating noise and silence between words as well as introducing word ordering (which is currently lacking in our system,i.e, "My name is Ranjan" and "Ranjan is my name" are both considered the same) as well as increased security, nullifying the two-word length for an optimal password.

7 GUI

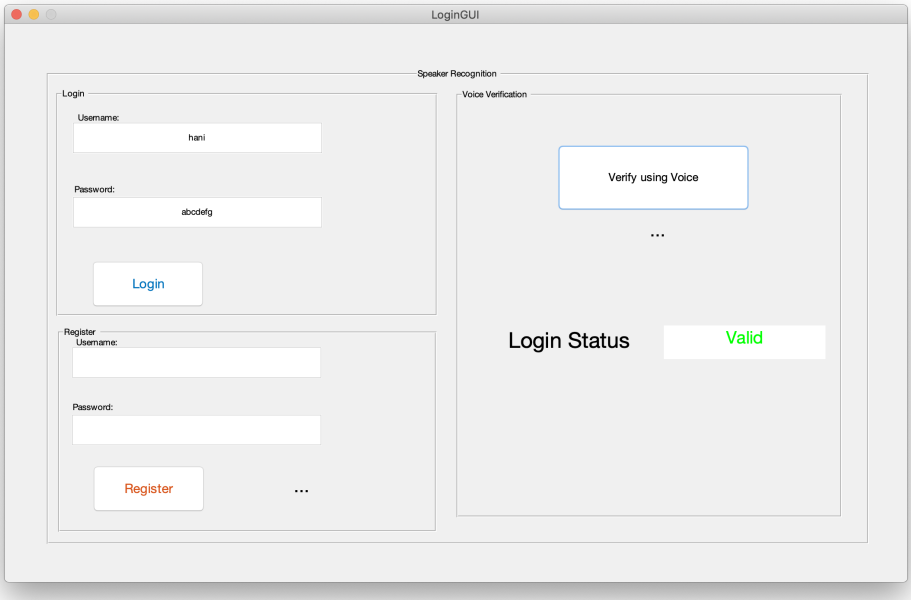

Figure 15: A successful attempt at logging in.

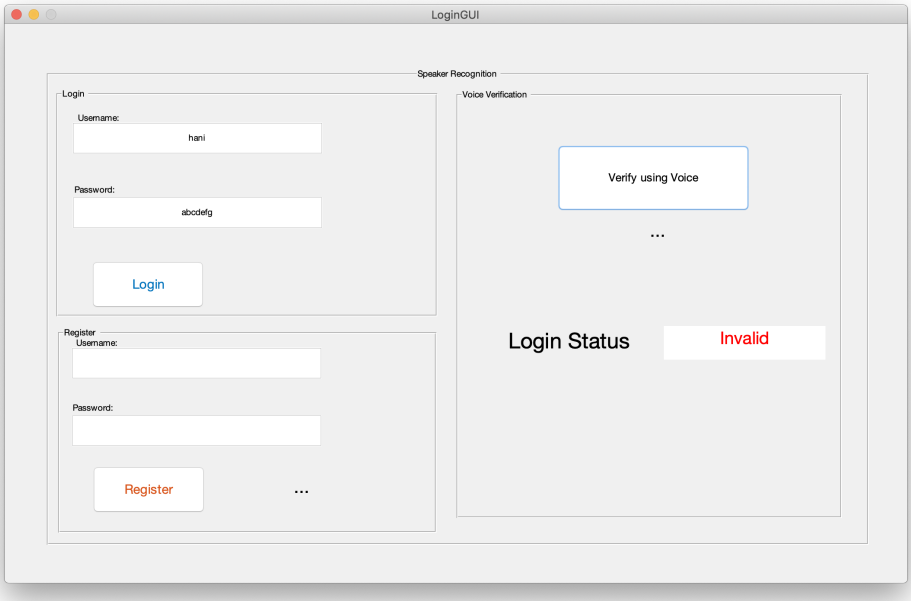

Figure 16: An unsuccessful attempt.

References

[1] *An Automatic Speaker Recognition System,* from `http://www.ifp.uiuc.edu/~minhdo/teaching/speaker_recognition`

[2] Atahan Tolunay, *Text-Dependent Speaker Verification Implemented in Matlab Using MFCC and DTW* (2010)

[3] Jakobovski, *Free Spoken Digit Dataset*, available at `https://github.com/Jakobovski/free-spoken-digit-dataset`.

[4] *Mel-frequency cepstrum*, available at `https://en.wikipedia.org/wiki/Mel-frequency_cepstrum`.

[5] *Matlab code*, available at `https://github.com/thehanimo/speaker-identification`.