Challenge 4 Report

Juan Reyes
University of California, Santa Barbara

Marco Chavez
University of California, Santa Barbara

***Abstract*—Speech recognition is a widely popular and useful nowadays. It is the basis for voice assistants on mobile phones. This report discusses a particular implementation of speech recognition, to distinguish between spoken digits from 0 to 9. This was achieved using an MFCC (mel frequency cepstral coefficient) and using DTW (dynamic time warping), along with a training set. The results show that it has an accuracy of 80.5%.**

I. INTRODUCTION

As technology is becoming more automated voice assistants have gained popularity in achieving this automation. Being able to control and use modern devices by speaking is convenient and sometimes necessary. This can be done in many different ways and involves the understandings of signals and fundamentals of sounds. The method described here is known as MFCC, after the MFCC of a signal is computed it can be compared using DTW with a training set. MFCC has been widely used for speech recognition as it extracts important information from an audio signal that distinguish vocal features. In this particular implementation, it is to distinguish spoken digits from different speakers.

II. METHODS

To implement the MFCC the first step is to remove noise from signal. First we eliminate all frequencies outside of the human speech frequency range (0-3400 Hz) by applying a low pass filter. To eliminate noise within human speech range a pre-emphasis filter is applied, which maintains high frequencies. This is achieved using Equation (1) where alpha is 0.97 and s[] is the signal already processed by a low pass filter [3].

$$y[n] = s[n] - \alpha s[n-1] \qquad (1)\text{Ref.}[3]$$

$$M(f) = 1125 \ln(1 + f/700) \qquad (1)$$

$$M^{-1}(m) = 700(\exp(m/1125) - 1) \qquad (2)$$

The next step is to frame the input signal into windows that hop a certain number of samples to create another window. We used window sizes of 3ms and hop sizes of 2ms in order to extract small details within the audio signal.

After the framing, a hamming window is generated that is the same size as the window sizes of each frame. The hamming window is used to create a precise representation of the signal components in the frequency domain. Afterwards the fft of these new windowed frames are calculated to find the frequency content.

A filterbank is made using the Mel scale, this scale helps to replicate the interpretation of sound the way a human ear would. The filterbank can be made with any number of filters and Figure 2 shows a graphical display of 10 filters for a range of frequencies from 0Hz to 8000Hz. To explain Figure 2, the mel frequencies indicate where each point of the triangle filter should be. As the frequencies get higher it will space the triangles more as humans do not perceive the difference of a small change in higher frequencies compared to a small change in lower frequencies. Equations (2) and (3) show how to convert from frequencies to mel frequencies, (2) shows the conversion from frequencies to mel frequencies and (3) shows the reverse conversion.

This filter is applied to each windowed frame and then a log scale is used to replicate how humans perceive sound, which is not on a linear scale. The last step is calculate the DCT (discrete cosine transform) of these frames to decorrelate each filter as they overlap [1].

Training was performed using a provided test set of spoken digits 0 to 9, each 50 times and with 3 different speakers. To create a more diverse training set, an average of 6 of each digit from each speaker was calculated and stored. This stored information

would then used to compare to a new input in order to classify it.

The comparison method used is known as Dynamic time warping , DTW. As shown in Equation (4) DTW algorithm is implemented to calculate least distance between features of a digit uttered and reference digit set of possible digits. The unknown digit is guessed to be the reference digit with the least distance. The main idea behind DTW is stretching or shrinking along the time axis of one signal to match another signal. To find the least distance a two dimensional array ,dist[m n], is used, where the column represents the unknown signal and the row represents a reference signal. Let dist(i,j) hold the minimum distance between point a_i from unknown signal and b_j from known signal. dist(i, j) is calculated using the following equation:

$$D(i,j) = \min[D(i-1,j-1), D(i-1,j), D(i,j-1)] + d(i,j)$$

(4) Ref. [2]

Once the distance matrix has been populated, starting from position dist(m,n) a sum is calculated to get minimum cost distance. First the value at dist(m,n) is added to the sum. Then nearest three elements to dist(m,n) are compared and the least cost element is chosen as next step and is added to the sum. The process repeats until element dist(1,1) is reached.

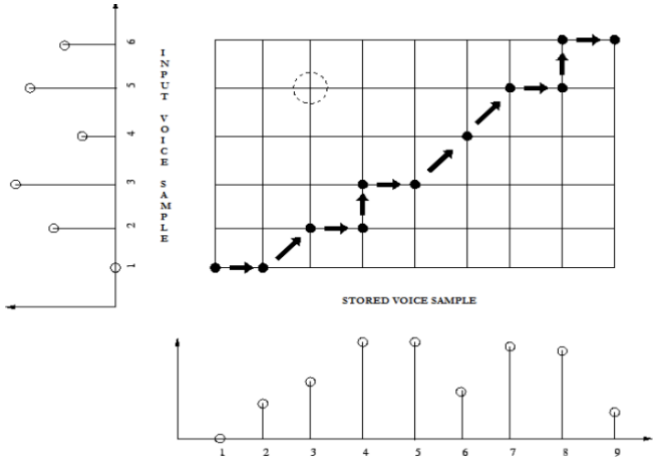

Figure 1: Dynamic Time Warping of two voice samples [2]

III. RESULTS

Applying the entire algorithm to old and new inputs yields and accuracy of about 80.5%. Different number of coefficients and mel filters were tried and the best results were with values of 6 and 18, respectively. This also helped to keep the data size of the training set smaller than the restriction.

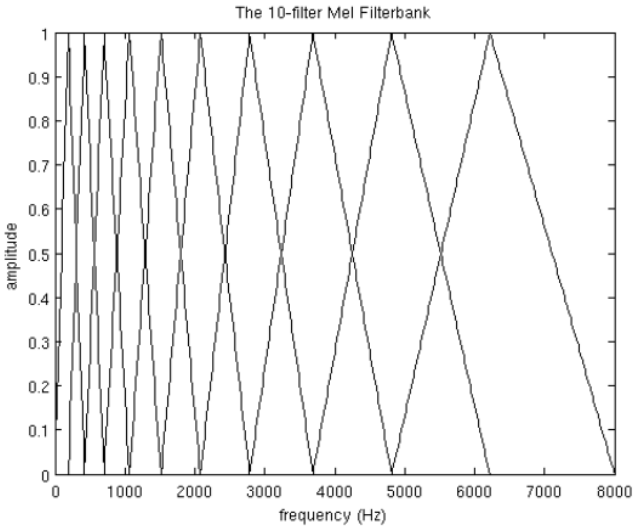

Figure 2: Mel filterbank, x-axis (Hz), y-axis(amplitude)

IV. DISCUSSION

The algorithm seems to work with desirable results and running time. The running time is determined by the training set size. As we kept ours to 30 reference MFCCs it was rather small and efficient. The running time could be improved with more vectorized code in MATLAB. Although the training set was limited to a small number of speakers, the accuracy could be decreased if the audio is filled with background sounds or if there are different speakers. Such as those with heavy accents or mispronunciations.

V. CONCLUSION

In conclusion the results for the algorithm are relatively reliable. As well as the running time. This can be useful in smaller applications but for more robustness a bigger training set would be required, along with refinement of the code. This can easily be expanded to include more digits or even words.

REFERENCES

[1] Practicalcryptography.com. (2013). Mel Frequency Cepstral Coefficient (MFCC) tutorial. [online] Available at: http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs/#computing-the-mel-filterbank [Accessed 25 May 2018].

[2] *Speech Recognition using MFCC and DTW* by Bhadragiri Jagan Mohan and Ramesh Babu at School of Electrical Engineering VIT University Vellore, India [online] Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6838564 [Accessed 25 May 2018].

[3] *Spoken Word Recognition Using MFCC and Learning Vector Quantization,* by Esmeralda C. Djamal* , Neneng Nurhamidah and Ridwan Ilyas Jurusan Informatika at Universitas Jenderal Achmad Yani Jl. Terusan Jenderal Sudirman, Cimahi [online] Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8239119 [Accessed 25 May 2018].

Juan Reyes

- Contributed to MFCC implementation
- Contributed to Mel filter bank implementation
- Debugged code

Marco Chavez

- Contributed with MFCC implementation
- Implemented DTW
- Implemented training algorithm
- Contributed to Mel filter bank implementation
- Implemented recognition algorithm
- Debugged code