# BUET Bus Entrance System using Voice Recognition

Wasifa Mashiyath
*Department of Electrical and Electronic Engineering*
*Bangladesh University of Engineering and Technology*
Dhaka, Bangladesh
1806003@eee.buet.ac.bd

Khadiza Sultana
*Department of Electrical and Electronic Engineering*
*Bangladesh University of Engineering and Technology*
Dhaka, Bangladesh
1806013@eee.buet.ac.bd

Raihan Mahmud Chowdhury
*Department of Electrical and Electronic Engineering*
*Bangladesh University of Engineering and Technology*
Dhaka, Bangladesh
1806014@eee.buet.ac.bd

M. M. Yeamin
*Department of Electrical and Electronic Engineering*
*Bangladesh University of Engineering and Technology*
Dhaka, Bangladesh
1806016@eee.buet.ac.bd

***Abstract*— This project represents a text dependent voice recognition based digital system for BUET Bus Service over the existing conventional system to help the students. To design, the software part of our proposed system, MFCC is used for extracting feature of the given voice. In this algorithm, the cepstral co-efficient are calculated at Mel frequency scale. vector quantization and Euclidean distance methods are used for feature matching through MATLAB. Here, the pattern of a student's voice will be compared with the patterns stored in the student database in training session of the system. The quality and testing of this speaker recognition system is completed and analyzed, and we have found that when we use strong noise cancelling devices for voice input, the accuracy of the system is very satisfactory.**

***Keywords— MFCC, FFT, DCT, Vector quantization, Euclidean distance.***

## I. INTRODUCTION

BUET bus service plays a very important role in the transportation facilities of the attached students of BUET. But the students have to face many troubles at every step of collecting tickets. The sufferings are not ended here, rather they have to carry tickets all the time and there is a chance of losing them. To ease the suffering of the students, we have tried to convert the system to a digitalized process from start to end. Before entering the bus or testing phase, students will have to say their name and id to a microphone as the system is speaker-dependent and speech-dependent. The sampled form of the audio data will be converted into a set of feature arrays as their voice patterns. These patterns of voices which are unique from person to person will be extracted using the MFCC algorithm and compared with the previously stored voice patterns of the students collected in the same way in the training phase in the student database following the vector quantization method. Then the software will be able to select the best matching person. When a successful input will occur, that means when our proposed system successfully recognizes the students, they will get access to enter the bus and the bill will be counted on their account.

## II. LITERATURE REVIEW

Text-dependent speaker recognition systems have been designed following many algorithms over the past years. Among different methods, MFCC along with vector quantization methods are used in this project because of the simplified yet moderately high-accuracy algorithm [1][2]. To increase the precision,10 training data sets are used, and a better output is found. As a classifier, KNN is another one with the comprehensible method [3], but with the increase in data set in columns, the Euclidean distance method gives better output. Besides, there are various methods of voice recognition perfectly using more complex algorithms like Hidden Markov Models (HMM) [4]. The architecture of neural networks helps in this field [5]. DNN and end-to-end automatic speech recognition methods all are used nowadays for complex scenarios [6]. With the comparison of all the methods, MFCC with vector quantization is easy to work with high accuracy and maintenance.

## III. METHODOLOGY

### *A. Voiced/ unvoiced/ silence detection and silence removal:*

Voiced parts of a speech are almost periodic and have more correlation among successive samples. These contain more useful information and higher energy than the silence and unvoiced portion. Moreover, silence/unvoiced portion of speech is affected more by noise than voiced portion. So, removal of this redundant information through proper segmentation does not only ensure the reduction of number of computation but also increases the accuracy of speech processing. At first, the data was sampled at 44100 samples/sec. Frame size was considered as Fs/100. Using first (44100/5) =8820 samples, the parameters $\mu$ and $\sigma$ were calculated because the first 8820 samples of the input speech contain background noise/ white noise and so its distribution is normal distribution. Then, for any sample x if $(x-\mu)/\sigma >$ the adjusted threshold, then we consider that it belongs to the distribution of background noise and hence it can be eliminated from the speech part. Thus, the number of voiced samples and unvoiced samples of each frame was counted. Based on this, we divided the frames as voiced frames and unvoiced frames. Our silence removed signal was formed by including the voiced frames only.

### *B. Framing*

We split signal up into (overlapping) frames: one per row and keep the samples of each frame along their respective row. We chose hamming windowing. By default, the number of frames will be rounded down to the nearest integer and the last few samples of x() will be ignored unless its length is lw more than a multiple of inc where w is the frame size and inc is frame increase in samples. If the 'z' or 'r' options are given, the number of frames will instead be rounded up and zero padding is done or last few samples were reflected for final frame.

Depending on the number of samples, total number of frames can be varied. At first, frame size was 100ms, but after hamming windowing with length 20ms window for improving efficiency, frame size changed into 20ms.

### *C. Calculating Discrete Fourier Transform (DFT):*

After performing framing and windowing, it was required to calculate DFT for each point of each frame in order to get a frequency spectrum of the pre-processed speech and also to understand better what frequencies it is composed of, so that we can do further processing of it. For fast computation, we used the build in function fft of MATLAB to find DFT. This transform is defined by

$$Y(k) = \sum_{j=1}^{n} X(j)\, W_n^{(j-1)(k-1)}$$

where,

$$W_n = e^{(-2\pi i)/n}$$

and X(j) was the time domain speech after framing and windowing. Here 1≤k≤K, where K=the size of hamming window = n= number of samples = 882.

Before performing the DFT, if the frame duration was not mentioned, then it was required to convert the speech length to 2's power, because 2's power point DFT helps to make the computation and understanding easier. It was done by rounding the number of samples to its nearest smallest 2's power value.

For human speech, most of the information lies within 4kHz.

### *D. Calculating Power Spectrum*

Analyzing the speech in frequency domain is easier than in time domain because we can measure the power contained in each frame just by taking the square of the absolute value of DFT of each point and then measuring their average. This is compared to the human cochlea, which vibrates at different spots depending on the frequency values, and based on its location of vibration, nerves send the signal to inform the amount of any frequency component present.

The average power of each frame is taken following the equation

$$P(k)= \frac{1}{N}\sum_{i=1}^{N} P_i\,(k) \text{ where } P_i(k)=|Y(k)|^2.$$

Here Y is the DFT calculated in the previous step. Again, 1≤k≤K, where K=the size of hamming window = n= number of samples = 882.

### *E. Mel Filter Bank*

Human ears perceive frequency logarithmically, whereas machines perceive the sound linearly. Human ears have higher resolution at low frequencies, but machines treat all ranges of frequencies in a similar way. So, modeling the human hearing property at the feature extraction stage will improve the performance of the model.The formula for mapping the actual frequency to the frequency that human beings will perceive is given below:

$$\text{mel}(f)=1127*\ln\left(1+\frac{f}{700}\right)$$

At first, lowest and highest frequencies were converted to MEL units and then divided into 32 filter banks having equally spaced points. After that, these points were converted back to Hertz following the inverse mel equation and rounded to the nearest frequency bins so that spectral leakage did not happen. It is not possible to provide infinite resolution in frequency domain. After that, triangular filter banks are formed by:

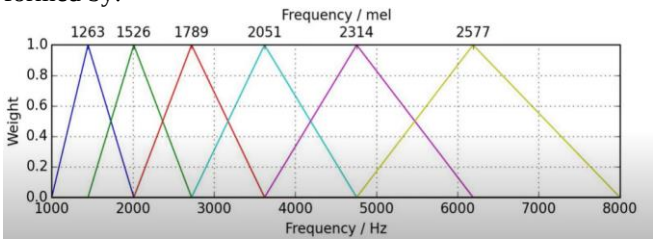


Figure1: Triangular Filter Banks

To estimate the amount of energy present in various frequency ranges, we take groups of periodogram bins and add them together. The Mel filter bank does this; the first filter is extremely narrow and indicates how much energy is close to 0 Hertz. As the frequencies increase, the filters get wider to have the same perceptual difference in terms of frequency distances.

### *F. Discrete Cosine Transform(DCT)*

DCT expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. DCT is inverse Fourier transform with reduced computation because it deals with real numbers only. Our filter banks were all overlapping. The filter bank energies were quite correlated with each other. The DCT decorrelated the energies in different Mel bands. But only 15 of the 32 DCT coefficients are kept. This is because the higher DCT coefficients represent fast changes in the filter bank energies, and these fast changes degrade the performance of speech processing, so we get a slight improvement by dropping them.

We prepared the function of DCT following the equation:

$$y(k,l) = w(k)\sum_{m=1}^{M} u(m,l)\cos\frac{\pi(2m-1)(k-1)}{2M}, \quad k = 1,\ldots,M$$

where

$$w(k) = \begin{cases} \frac{1}{\sqrt{M}}, & k = 1 \\ \sqrt{\frac{2}{M}}, & 2 \le k \le M \end{cases}$$

### *G. Delta and delta-delta co-efficients:*

Although the MFCC feature vector only captures the power spectral envelope of a single frame, speech would also contain information about dynamics, or more specifically, about the trajectories of the MFCC coefficients over time. It turns out that adding the MFCC trajectories to the original feature vector after computing them significantly improves the performance of speech processing. We had 15 MFCC coefficients, we would also get 15 delta coefficients and 15 delta-delta coefficients. A feature vector of length 45 would result from the combination. As we performed hamming windowing, we skipped these two steps. But our project code includes it and so this part of code can also be executed by changing the window type.

To calculate the coefficients, the following formula is used:

$$d_t = \frac{\sum_{n=1}^{N} n(c_{t+n} - c_{t-n})}{2\sum_{n=1}^{N} n^2}$$

where $d_t$ is a delta coefficient, from frame t computed in terms of the static coefficients $c_{t+N}$ to $c_{t-N}$. A typical value for N is 2.

### *H. Vector Quantization*

The technique of mapping vectors from a huge vector space to a limited number of regions in that space is known as vector quantization. Each area is referred to as a cluster and can be depicted by its center called a centroid [2,4]. The VQ encoder encodes a given set of k-dimensional data vectors with a much smaller subset. The subset is called a codebook and its elements Ci are called codewords, code vectors, reproducing vectors, prototypes, or design samples. Only the index i is transmitted to the decoder. The decoder has the same codebook as the encoder, and decoding is operated by table look-up procedure. The commonly used vector quantizers are based on nearest neighbor called Voronoi or nearest neighbor vector quantizer. Both the classical K-means algorithm and the LBG algorithm belong to the class of nearest neighbor quantizers.

A key component of pattern matching is the measurement of dissimilarity between two feature vectors. The measurement of dissimilarity satisfies three metric properties such as Positive definiteness property, Symmetry property and Triangular inequality property. Each metric has three main characteristics such as computational complexity, analytical tractability and feature evaluation reliability. The metrics used in speech processing are derived from the Minkowski metric. The Minkowski metric can be expressed as

$$D_p(X,Y) = \sqrt[p]{\sum_{i=1}^{k} \left| x^i - y^i \right|^p},$$

Where $X = \{x^1, x^2, ..., x^k\}$ and $Y = \{y^1, y^2, ..., y^k\}$ are vectors and p is the order of the metric.

The performance of the vector quantizer can be evaluated by a distortion measure D which is a non-negative cost D (X j, X j) associated with quantizing any input vector X j with a reproduction vector X j. Usually, the Euclidean distortion measure is used. The performance of a quantizer is always qualified by an average distortion Dy = E [D (X j Xj)] between the input vectors and the final reproduction vectors, where E represents the expectation operator. Normally, the performance of the quantizer will be good if the average distortion is small.

### *I. Euclidean Distance Measurement*

The voice of an unknown speaker is represented by a series of feature vectors (x1, x2,....xi) during the speaker recognition phase, and it is then compared to codebooks from the database. Based on reducing the Euclidean distance, it is possible to determine the speaker's identity by calculating the distortion distance between two vector sets[6]. The Euclidean distance is calculated using the following formula:

The Euclidean distance between two points P = ($p_1$, $p_2 \cdots p_n$) and Q = ($q_1$, $q_2 \cdots q_n$),

$$= \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

The speaker with the lowest distortion distance is chosen to be identified as the unknown person.

## IV. RESULTS AND DUSCUSSION

To evaluate our model, we have calculated the Precision, Sensitivity, Specificity, Accuracy and F-measure along with mixing various levels of white noise with 240 Name audio samples and 240 ID audio samples to test for robustness. We have also generated the confusion matrix to test the performance of the classification algorithm.

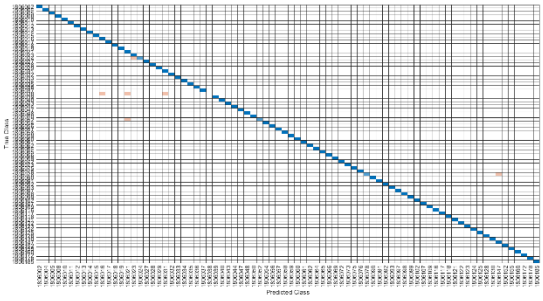

Figure2: Confusion Matrix for Name Audio Samples

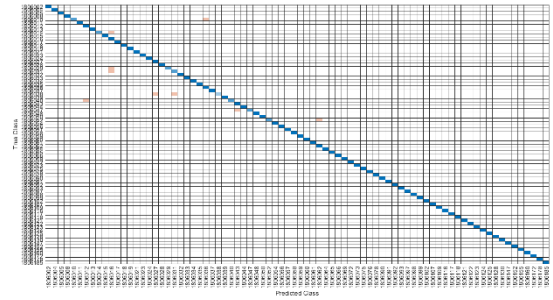

Figure3: Confusion Matrix for ID Audio Samples

| SNR(dB) | Precision | Sensitivity | Specificity | Accuracy | F-measure |
|---|---|---|---|---|---|
| 10 | 0.079167 | 0.079167 | 0.98834 | 0.079167 | 0.079167 |
| 15 | 0.10833 | 0.10833 | 0.98871 | 0.10833 | 0.10833 |
| 20 | 0.2 | 0.2 | 0.98987 | 0.2 | 0.2 |
| 25 | 0.35833 | 0.35833 | 0.99188 | 0.35833 | 0.35833 |
| 30 | 0.55417 | 0.55417 | 0.99436 | 0.55417 | 0.55417 |
| 35 | 0.65417 | 0.65417 | 0.99562 | 0.65417 | 0.65417 |
| 40 | 0.7 | 0.7 | 0.9962 | 0.7 | 0.7 |
| 45 | 0.77083 | 0.77083 | 0.9971 | 0.77083 | 0.77083 |
| 50 | 0.8375 | 0.8375 | 0.99794 | 0.8375 | 0.8375 |
| 55 | 0.925 | 0.925 | 0.99905 | 0.925 | 0.925 |
| 60 | 0.96667 | 0.96667 | 0.99958 | 0.96667 | 0.96667 |
| 65 | 0.97083 | 0.97083 | 0.99963 | 0.97083 | 0.97083 |
| 70 | 0.98333 | 0.98333 | 0.99979 | 0.98333 | 0.98333 |
| 75 | 0.97917 | 0.97917 | 0.99974 | 0.97917 | 0.97917 |
| 80 | 0.975 | 0.975 | 0.99968 | 0.975 | 0.975 |

Figure4: Performance Table based on the Name Audio Samples

| SNR(dB) | Precision | Sensitivity | Specificity | Accuracy | F-measure |
|---|---|---|---|---|---|
| 10 | 0.079167 | 0.079167 | 0.98834 | 0.079167 | 0.079167 |
| 15 | 0.0875 | 0.0875 | 0.98845 | 0.0875 | 0.0875 |
| 20 | 0.14583 | 0.14583 | 0.98919 | 0.14583 | 0.14583 |
| 25 | 0.275 | 0.275 | 0.99082 | 0.275 | 0.275 |
| 30 | 0.4875 | 0.4875 | 0.99351 | 0.4875 | 0.4875 |
| 35 | 0.64167 | 0.64167 | 0.99546 | 0.64167 | 0.64167 |
| 40 | 0.75417 | 0.75417 | 0.99689 | 0.75417 | 0.75417 |
| 45 | 0.8 | 0.8 | 0.99747 | 0.8 | 0.8 |
| 50 | 0.8875 | 0.8875 | 0.99858 | 0.8875 | 0.8875 |
| 55 | 0.925 | 0.925 | 0.99905 | 0.925 | 0.925 |
| 60 | 0.92917 | 0.92917 | 0.9991 | 0.92917 | 0.92917 |
| 65 | 0.95 | 0.95 | 0.99937 | 0.95 | 0.95 |
| 70 | 0.97083 | 0.97083 | 0.99963 | 0.97083 | 0.97083 |
| 75 | 0.95833 | 0.95833 | 0.99947 | 0.95833 | 0.95833 |
| 80 | 0.97083 | 0.97083 | 0.99963 | 0.97083 | 0.97083 |

Figure5: Performance Table based on the ID Audio Samples

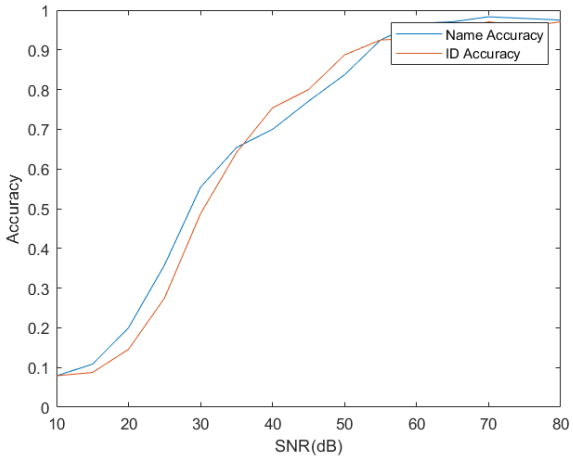


Figure6: Accuracy vs SNR (dB)

Here, we can see a significant increase in accuracy when the noise is lower. The accuracy and F1 scores are above 80% when the SNR is above 45dB. Hence, using a better noise reduction algorithm is needed to expect better performance from the designed system in noisy environment where SNR is less than 45dB.

## V. CONCLUSION

The goal of our project was to build up a voice-based bus entry system. For this purpose, we took 5 name samples and 5 ID samples collected from 92 students. We trained these data with our code. We used MFCC for feature extraction and KNN for feature matching. Then we took input voice to match with the samples.

Our system can accurately identify users 94% times. However, because our project is built for the roads, in high noise scenario, the percentage of success may fall. So, we need very good hardware setup to get maximum success from our project. Our budget was estimated to be 3,36,000-taka one time and 25,000 taka monthly for maximum output. This may vary considering new hardware setup. Our project was voice dependent. If it could be made voice independent, it could be more efficient. Also, better output will come if the samples were to be noise-free.

## ACKNOWLEDGMENT

We would like to thank everyone who helped us in developing our project. We are humbly grateful to Professor Dr. Celia Shahnaz mam and Shafin Bin Hamid sir for giving us a chance to present our project and for giving us constant guideline, valuable insights. We are also thankful to our classmates who provided the necessary training and testing data which helped us frame and test our project to the better end.

## REFERENCES

[1] Shaneh, M., & Taheri, A. (2009). Voice command recognition system based on MFCC and VQ algorithms. *World Academy of Science, Engineering and Technology*, *57*, 534-538.

[2] Patel, K., & Prasad, R. K. (2013). Speech recognition and verification using MFCC & VQ. *Int. J. Emerg. Sci. Eng.(IJESE)*, *1*(7), 137-140.

[3] Faek, F. K., & Al-Talabani, A. K. (2013). Speaker recognition from noisy spoken sentences. *International Journal of Computer Applications*, *70*(20).

[4] L. Rabiner, "A tutorial on Hidden Marcov Model and selected applications in Speech Recognition", Proceedings of the IEEE, Vol. 77, No. 2, 1989, pp.257-286

[5] Patricia Melin, Jerica Urias, Daniel Solano, Miguel Soto, Miguel Lopez, and Oscar Castillo, "Voice Recognition with Neural Networks, Type-2 Fuzzy Logic and Genetic Algorithms"

[6] Jiang, W., Wen, F., & Liu, P. (2018). Robust beamforming for speech recognition using DNN-based time-frequency masks estimation. *IEEE Access*, *6*, 52385-52392.

[7] Kiran, U. (2021, June 13). MFCC Technique for Speech Recognition. AnalyticsVidhya. . https://www.analyticsvidhya.com/blog/2021/06/mfcc-technique-for-speech-recognition/

[8] A.Thakur (2013) Speech Recognition Using Euclidean Distance (https://www.semanticscholar.org/paper/Speech-Recognition-Using-EuclideanDistanceThakurSahayam/0b25724f5160b054ea3c0ce2bcc28b284e2b40ec)

[9] Balwant A. Sonkamble (2012) Speech Recognition Using Vector Quantization through Modified K-meansLBG Algorithm (https://core.ac.uk/download/pdf/234644507.pdf )