Sensors and Materials, Vol. 37, No. 1 (2025) 441–451
MYU Tokyo

S & M 3919

Text-independent Hakka Speaker Recognition in Noisy Environments

Jie Peng,[1*] Chin-Ta Chen,[1] and Cheng-Fu Yang[2,3**]

[1]School of Electronic and Electrical Engineering, Zhaoqing University, Zhaoqing 526061, China
[2]Department of Chemical and Materials Engineering, National University of Kaohsiung, Kaohsiung 811, Taiwan
[3]Department of Aeronautical Engineering, Chaoyang University of Technology, Taichung 413, Taiwan

(Received November 11, 2024; accepted January 20, 2025)

Keywords: speaker identification, estimate-maximize algorithm, Gaussian mixture model, mel-frequency cepstral coefficient, Hakka

In this study, we introduce a robust text-independent speaker recognition system tailored specifically for Hakka speakers, operating in diverse noisy environments. Our system employs mel-frequency cepstral coefficients for effective feature extraction, coupled with a Gaussian mixture model and the expectation-maximization algorithm to enhance accuracy under challenging conditions. Through comprehensive experimentation, we achieved impressive recognition rates, underscoring the system's effectiveness for real-world applications. This innovation not only addresses the unique characteristics of Hakka speech but also demonstrates significant potential for deployment in various settings where background noise poses a challenge.

1. Introduction

Language serves as a fundamental medium for human communication, and text-independent speaker recognition technology has become essential for improving interactions between humans and machines. However, Hakka, unlike more widely spoken languages, suffers from a lack of extensive digital corpora, which poses challenges in training robust machine learning models.(1) Additionally, Hakka features significant dialectal variation, resulting in inconsistencies in pronunciation and vocabulary usage across different regions. The tonal nature of the language further complicates recognition tasks.(2) Recent research has made strides in Hakka speech recognition by leveraging the Whisper model, which incorporates several advanced techniques. This model utilizes semisupervised learning embeddings and multi view self-attention mechanisms to enhance feature representation. Furthermore, it combines a chain-based discriminative autoencoder with multistream convolutional neural networks and employs lattice-free maximum mutual information to optimize performance. A recurrent neural network language model is used for rescoring, which significantly boosts accuracy even in the face of

*Corresponding author: e-mail: pengjie@zqu.edu.cn
*Corresponding author: e-mail: cfyang@nuk.edu.tw
https://doi.org/10.18494/SAM5466

ISSN 0914-4935
https://myukk.org/

limited training data. Collectively, these innovations represent a promising direction for overcoming the unique challenges posed by Hakka in the realm of speaker recognition.[3–5]

While these advancements are promising, significant challenges persist in effectively recognizing Hakka speech, especially in noisy environments.[6] Background noise can severely degrade the quality of audio input, resulting in reduced accuracy for speaker recognition systems. This degradation is particularly problematic for Hakka, given its tonal nature and dialectal variations, which can make distinguishing between speakers even more difficult under adverse conditions. To tackle these challenges, it is essential to develop innovative methods that can robustly extract and process speech features, even amidst background noise. These methods should focus on enhancing feature extraction techniques and employing noise-reduction algorithms to improve the signal quality. Additionally, incorporating adaptive models that can learn and adjust to varying noise levels will be crucial for maintaining high recognition accuracy. By addressing these factors, we can pave the way for more effective Hakka speaker recognition systems that operate reliably in real-world scenarios.

An approach that utilizes mel-frequency cepstral coefficients (MFCCs)[7,8] in conjunction with a Gaussian mixture model (GMM)[9,10] and the expectation-maximization (EM) algorithm[11,12] has demonstrated success in improving accuracy under noisy conditions for Cantonese speaker recognition.[13] By leveraging MFCCs for robust feature extraction and GMM for the effective modeling of speech characteristics, we aim to enhance the performance of Hakka speaker recognition systems, particularly in challenging acoustic environments. The MFCC method provides a powerful means of capturing the essential features of speech, making it resilient to variations in noise. Meanwhile, the GMM allows for the modeling of complex speech patterns, which is crucial for accurately identifying speakers. By integrating these methodologies, we aim to address the specific challenges posed by Hakka's unique characteristics, thereby improving recognition accuracy in real-world settings where background noise is prevalent. In this work, we seek to contribute to the development of more reliable and effective speaker recognition systems for Hakka, ultimately facilitating better human–machine interactions.

2. Research Model

The design of our text-independent speaker recognition system for Hakka is akin to fingerprinting, as each individual's voice possesses distinct characteristics. We will segment audio recordings from multiple speakers into various frequency bands, carefully selecting the most suitable spectral features from these bands to create both training and test sets. By calculating the differences between these sets, we can effectively identify the corresponding speaker in the audio recordings. This approach allows us to capture the unique vocal traits of each speaker, enhancing the system's ability to differentiate between them even in challenging acoustic environments. The segmentation process ensures that we focus on the most relevant frequency ranges, improving the robustness and accuracy of our recognition model.

2.1 Framework and flowchart

The framework for text-independent speaker recognition in Hakka is structured into two primary stages, the model training stage and the testing stage. In the first stage, we collect audio segments from 50 Hakka speakers. A 30-second audio clip is designated for training, while a 5-second audio clip is used for testing. These audio segments are then organized into training and testing sets. During the model training stage, we extract speech features from the training set for all 50 speakers. Each speaker's unique features are used to train an individual model, which is subsequently stored in a model library. In the testing stage, we aim to recognize speech audio from the test set by extracting features in the same manner as in the training stage. These extracted features are then compared with those stored in the model library. Using GMM, we calculate the similarity between the test features and the models, resulting in difference values. Finally, on the basis of these values, we identify the corresponding Hakka speaker. Figure 1 illustrates the flowchart of this recognition process, providing a visual representation of the steps involved in both training and testing stages. This structured approach ensures a systematic method for achieving accurate speaker recognition within the Hakka language.

2.2 Model description

The audio recordings for this design were captured using the built-in recorder of a Redmi Note 5 smartphone and saved in waveform audio file format. Each of the 50 Hakka speakers contributed two audio segments, one approximately 30 s long for training purposes and the other around 5 s for testing. The recordings were conducted in a quiet indoor environment; however, owing to suboptimal sound insulation, there was some degree of external noise interference. To minimize variability, each speaker maintained a consistent position relative to the microphone, ensuring that the microphone direction remained unchanged throughout the recording sessions. Additionally, speakers were instructed to maintain a steady emotional state during the recordings to further reduce potential interference from extraneous factors. The content of the recorded

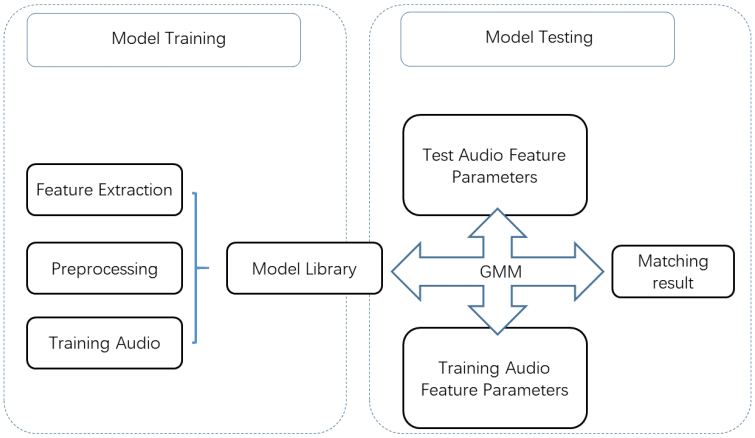

Fig. 1. (Color online) Realization flowchart for text-independent Hakka speaker recognition.

speech consisted either of self-composed phrases or randomly selected text provided by the speaker. This controlled approach was aimed at enhancing the quality and consistency of the audio data, facilitating more accurate training and testing of the speaker recognition system.

Voice signals encompass a variety of distinctive feature parameters, each conveying different physical and acoustic characteristics. The process of feature extraction is critical, as it is aimed at minimizing the effect of factors unrelated to speaker recognition while also reducing the volume of data needed for the subsequent recognition phase. The primary objective is to extract feature parameters that capture the speaker's unique information embedded within the voice signal. Depending on the intended application of the voice signal, different feature parameters are selected to optimize the accuracy of speaker recognition. For instance, parameters such as MFCCs may be used to effectively represent the spectral properties of speech, while pitch and tone features can help differentiate speakers on the basis of their vocal qualities. By carefully choosing and extracting these parameters, we enhance the system's ability to accurately identify speakers, even under varying acoustic conditions. This tailored approach to feature extraction is essential for developing a robust and effective speaker recognition system.

In audio signal processing, window functions play a crucial role in smoothing out the truncated parts of a signal, which helps reduce the loss of important speech information while enabling the extraction of signals with varying amplitudes from closely spaced frequency components. Directly truncating a signal can introduce edge effects, leading to signal distortion and spectral leakage. By applying window functions, these issues can be mitigated; they smooth the edges of the segmented signal, preserving more of the original information. There are several types of window functions, with common ones including the rectangular window, triangular window, Hamming window, and Gaussian window. In this design, the Hamming window is preferred for its effectiveness in addressing edge effects and minimizing signal loss. The use of the Hamming window allows for better handling of these edge effects, resulting in more accurate and reliable signal processing. The formula for the Hamming window is as follows.

$$W(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right). 0 \le n \le N-1 \tag{1}$$

Here, $W(n)$ is the windowing function, n is the index of sampling, and N is the length of the window. The primary mathematical tool used in speech signal processing is the Fourier transform. Since analyzing a signal in the time domain can be challenging, we often convert it into the frequency domain to reveal its characteristics more clearly. This process involves transforming a time-domain signal $X(n)$ into its frequency-domain representation $X(m)$. The result of applying the Fourier transform to a segment of the signal is referred to as the spectrum. In practice, the input to the fast Fourier transform (FFT) operation is typically a complex number. However, since speech signals are real-valued, the imaginary part is often set to zero during the transformation. This approach simplifies the analysis while still allowing us to extract valuable frequency-domain information, which is crucial for understanding the spectral properties of the speech signal. By employing the FFT, we can efficiently compute the spectrum, facilitating further processing and analysis of the speech data.

After completing the FFT treatment, we obtain an energy spectrum, which is represented as a complex matrix of speech features. Since the phase spectrum in the energy spectrum carries less relevant information for many applications, it is common practice to discard the phase spectrum and focus on the amplitude spectrum. This amplitude spectrum, often referred to as the short-time power spectrum, is computed by squaring the magnitude of the frequency-domain signal $X(m)$ obtained from the FFT. By retaining the amplitude spectrum, we emphasize the power of the frequency components, which provides crucial insights into the characteristics of the speech signal. This approach enhances the ability to analyze and process the speech features effectively, allowing for improved performance in tasks such as speaker recognition, speech synthesis, and other applications in audio signal processing.

$$m = 2595 \cdot \log_{10}\left(1 + \frac{f}{700}\right) \tag{2}$$

The relationship between mel frequency and linear frequency is illustrated in Fig. 2. In this representation, the mel scale is designed to reflect human auditory perception, where frequencies are spaced more closely at lower ranges and further apart at higher ranges. This scale is particularly useful in speech processing, as it aligns more closely with how we perceive pitch and sound.

2.2.1 GMM

The GMM is used to describe the distribution of speech feature signals in the feature space. During the training phase, the MFCC of the speech feature parameters is utilized to train the GMM, resulting in a model that captures the statistical properties of the speaker's voice. In a Gaussian speaker recognition system, each speaker's unique characteristics are represented through their corresponding GMM. When a speaker's training audio is processed, the MFCC features are extracted from the speech. These features serve as the foundation for establishing the speaker's GMM. When conducting speaker training, our primary objective is to identify the optimal parameters that effectively describe the observed speech data. To accomplish this, we employ the EM algorithm. The EM algorithm operates on a fundamental principle: it begins with an initial set of parameters and iteratively refines these parameters to improve the model. In

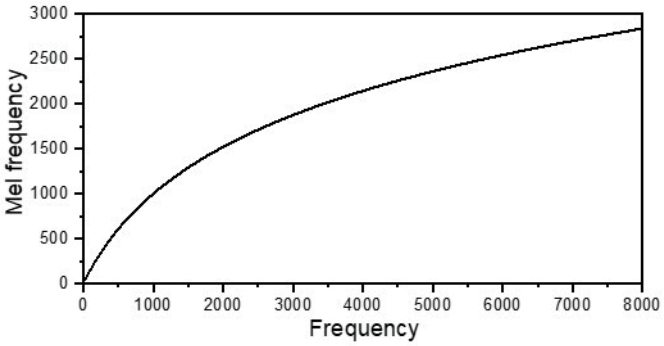

Fig. 2. Relationship between mel frequency and linear frequency.

each iteration, we estimate a new set of parameters on the basis of current estimates, replacing them for the next round of calculations. This cycle of estimation and replacement continues until the algorithm converges, yielding stable parameters that best represent the speech data. The process can be outlined as follows.

(a) Randomly initialize the λ initial values of the model parameters, denoted as $\overline{\lambda}$.

(b) For j from 1 to J, the algorithm iteration begins.

(1) Calculate the conditional probability expectation for the joint distribution as follows.

$$Q(\lambda) = E\left[L\left(\frac{\lambda}{X} \right) \right] = \sum_{i=1}^{M} \sum_{t-1}^{T} \log\left(p_i b_i \left(x_t / \lambda_i \right) \right) p\left(\frac{i}{x_i}, \lambda \right) \tag{3}$$

(2) Run the following equation iteratively. If $p(X|\lambda)$ converges, terminate the algorithm. Otherwise, proceed to the next step and continue the iterative calculations.

2.2.2 Classification

Suppose there are NNN target speakers, each represented by a GMM, denoted as $\{\lambda_1, \lambda_2, ..., \lambda_N\}$. The observed feature sequence of speech to be recognized is represented as $\{o_1, o_2, ..., o_T\}$. In the speaker recognition system, the goal is to identify the speaker from whom the speech originates. This involves calculating the posterior probability of the observed features OOO for each GMM and determining the speaker corresponding to the maximum posterior probability. To achieve this, for each speaker s in the speaker set, the classification system must compute the value of s that maximizes $p(\lambda_n|O)$. This is accomplished by applying Bayes' theorem, which states that

$$p\left(\lambda_n \mid O \right) = \frac{p(O \mid \lambda_n) p\left(\lambda_n \right)}{p\left(O \right)}, \tag{4}$$

where $p(\lambda_n|O)$ is the likelihood of the observed features given the GMM of speaker n, and $p(\lambda_n)$ is the prior probability of the speaker model. Thus, the classification system evaluates this expression for each speaker to find the one that maximizes the posterior probability. The recognition result is determined by the maximum posterior probability criterion, which identifies the speaker model that yields the highest posterior probability, as follows.

$$n^* = \underset{1 \le n \le N}{\operatorname{argmax}}\, p\left(\lambda_n \mid O \right) \tag{5}$$

3. Matlab and Field-programmable Gate Array (FPGA) Experimental Results and Discussion

The recognition rate of a speech recognition system is significantly affected by the number of Gaussian components in the GMM and the number of coefficients in the MFCC. MFCCs are used to extract features from the speech signal, capturing its essential characteristics, while

GMMs model these characteristics on the basis of the extracted features. It is well-established that increasing the number of MFCCs enhances the detail captured regarding the signal's spectral properties, and increasing the number of Gaussian components allows for the more accurate modeling of the speech data. As a result, higher values for both parameters typically lead to improved recognition rates. However, this also increases computational complexity, which can pose challenges related to processing time and resource requirements. Therefore, determining the optimal number of MFCCs and Gaussian components necessitates a convergence calculation approach.

3.1 Input voice analysis

In the experiments, we used a dataset of Hakka speakers recorded in both clean and noisy environments, with noise types including flowing water and thunderstorms. Figure 3 shows both clean and noise-contaminated voice signals. The waveform plots in Figs. 3(a), 3(c), and 3(e) illustrate the amplitude versus time for the voice signals, highlighting variations in speech amplitude over the time interval. Meanwhile, the spectrum plots in Figs. 3(b), 3(d), and 3(f) show the frequency content of the voice signals over time, with power per frequency in dB/Hz indicated on a color scale. The *y*-axis represents frequency in Hz, while the *x*-axis represents time in seconds. In practical environments, speech signals are frequently disrupted by various types of uncontrollable background noise, which can significantly impact the performance of recognition systems. Typically, background noise acts as an additive signal, resulting in collected signals that contain both the actual speech and the background noise. In this study, three types of noise were used: thunderstorm noise, flowing water noise, and a combination of both. These noise signals, with signal-to-noise ratios (SNRs) of 10 dB, were added to the recorded speech signals to create noisy speech signals.

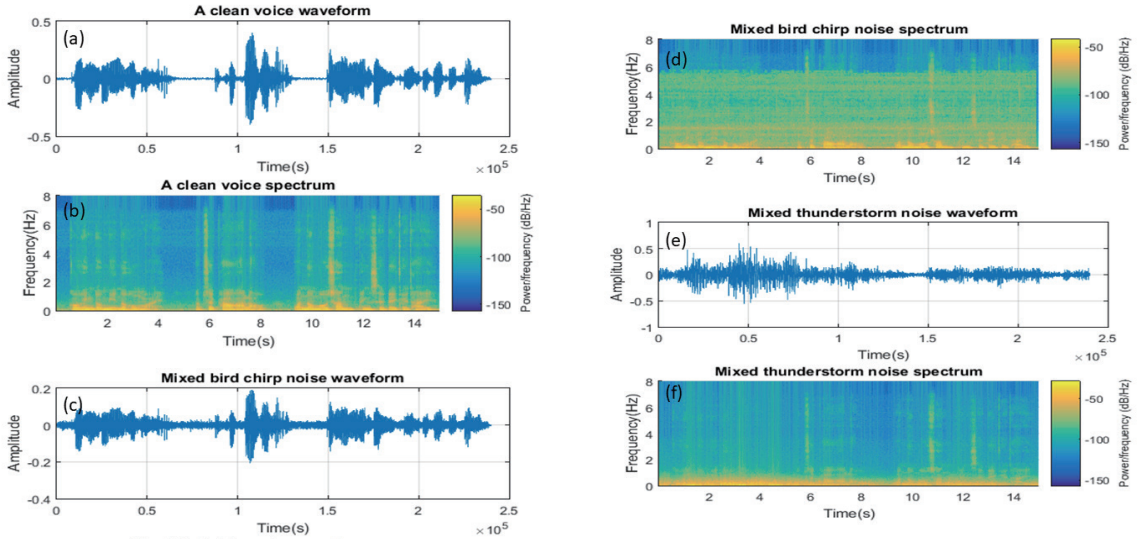

Fig. 3. (Color online) The following components are included: (a) clean voice waveform, (b) clean voice spectrum, (c) mixed bird chirp noise waveform, (d) mixed bird chirp noise spectrum, (e) mixed thunderstorm noise waveform, and (f) mixed thunderstorm noise spectrum.

3.2 Impact of MFCC number

Figure 4 illustrates the recognition rates for clean voice signals and those recorded in noisy environments. The *x*-axis represents the number of MFCCs used to capture features of the speech signal. The dimension of the MFCC determines the size of the feature vector extracted from the speech. Increasing the number of MFCCs generally captures more detail about the signal's spectral properties, thereby enhancing the accuracy of speech recognition systems. However, this also leads to increased computational complexity. As shown in Fig. 4, recognition rates improved as the number of MFCCs increased from 2 to 12. Specifically, the recognition rate for clean voice signals increased from 92% to nearly 100%, while rates for signals mixed with flowing water and thunderstorms rose from 69% to 82% and from 66% to 78%, respectively.

3.3 Impact of Gaussian component number

Figure 5 illustrates the recognition rates of clean voice signals based on different numbers of MFCCs and Gaussian components. The *x*-axis represents the number of MFCCs, with solid lines indicating the recognition rate and dotted lines representing the recognition time. The line colors

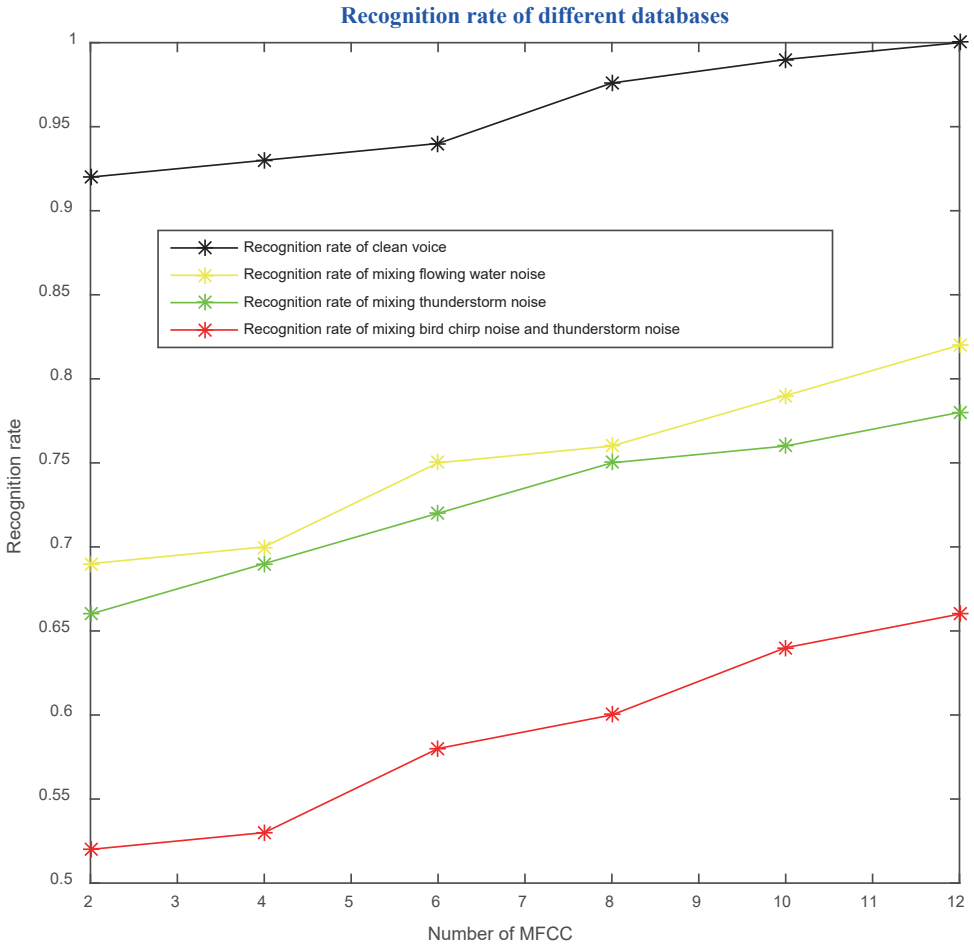

Fig. 4. (Color online) Recognition rates of clean voice signals and those in noisy environments.

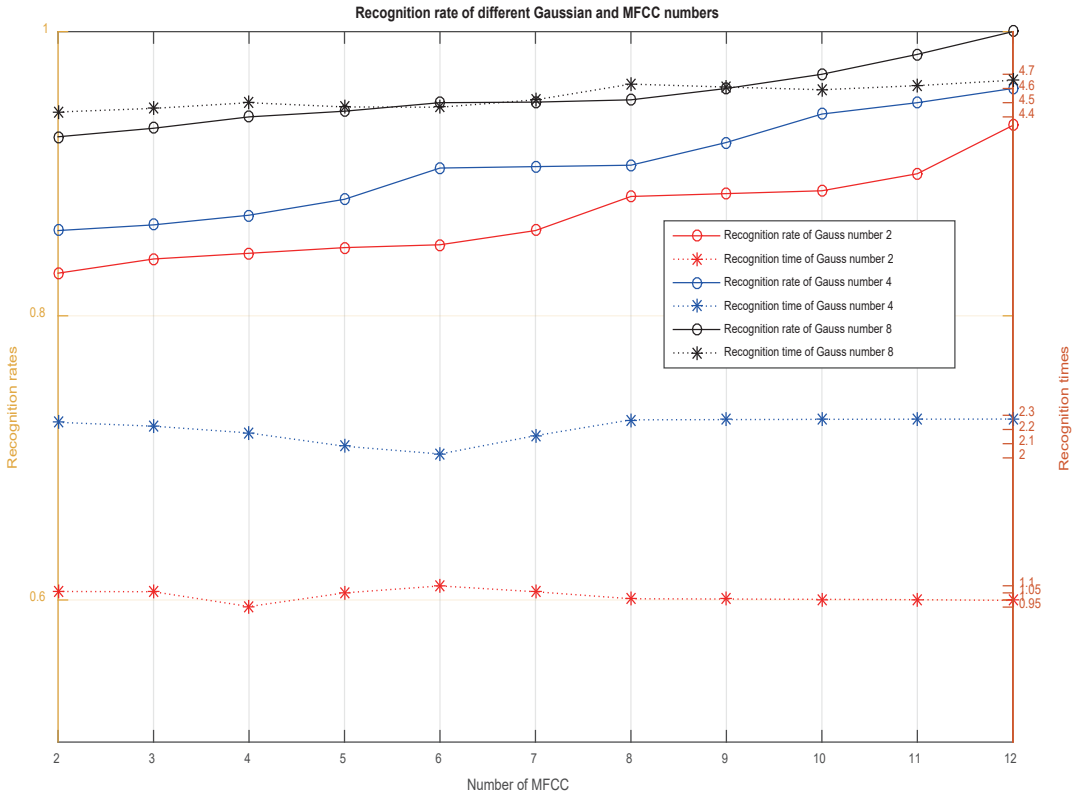

Fig. 5. (Color online) Recognition rates of different numbers of Gaussian components.

red, blue, and black correspond to 2, 4, and 8 Gaussian components, respectively. The figure demonstrates that the recognition rate increases with the numbers of both MFCCs and Gaussian components. However, while the recognition time is significantly affected by the number of Gaussian components, it is only slightly affected by the number of MFCCs.

3.4 FPGA simulation result

The implementation of feature extraction algorithms in speaker recognition systems on FPGA boards is highly significant for research and practical applications. FPGAs enable not only efficient processing and analysis but also provide crucial support for various fields. Their inherent capabilities for parallel computing and hardware acceleration facilitate the execution of complex feature extraction algorithms, allowing for real-time and low-latency processing. This characteristic is especially vital for applications that handle large volumes of data and have stringent time constraints, such as real-time monitoring systems and autonomous driving technologies. By utilizing FPGAs for feature extraction, high-performance processing can be achieved while also benefiting from lower power consumption. This makes FPGAs an attractive choice for energy-efficient solutions in high-demand environments. Moreover, the use of FPGAs allows for greater flexibility in adapting to evolving algorithms and system requirements. As speaker recognition technology advances, the ability to modify and optimize hardware

implementations can lead to improved accuracy and efficiency. This adaptability is crucial in fields such as security, where speaker verification is increasingly important. The feature extraction module is illustrated in Fig. 6, while Fig. 7 presents the speaker recognition test module.

Together, these components highlight the potential of FPGA-based systems to enhance speaker recognition performance across diverse applications. Among these parameters, "Feature_val" represents the effective signal of the MFCC, which is significant at high levels. The parameters "wPosition_X" and "wPosition_Y" indicate the respective row and column numbers of the feature function. These specifications are crucial for accurately mapping the features extracted from the speech signal, allowing for efficient processing and analysis in the recognition system. Among these parameters, the "Store" parameter represents the posterior probability of the GMM. This parameter is used to identify the speaker with the highest probability within the speaker model library. If a match is found, it indicates that the test speech has been successfully associated with the corresponding speaker model. This process is essential for accurate speaker recognition, as it helps ensure that the system effectively distinguishes between different speakers on the basis of their unique vocal characteristics.

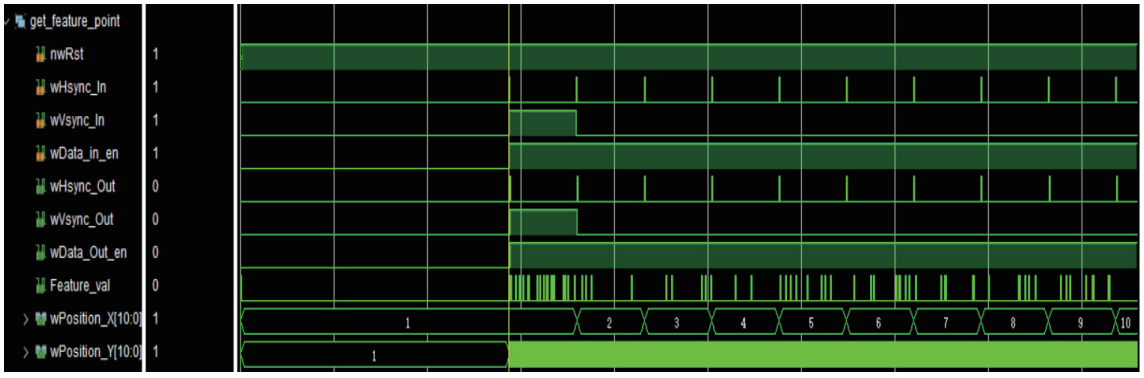

Fig. 6. (Color online) Feature extraction module.

Fig. 7. (Color online) Speaker recognition test module.

4. Conclusions

In this study, we introduced an effective speaker recognition system tailored for Hakka speakers in noisy environments. By utilizing MFCC for feature extraction and employing GMM with the EM algorithm for modeling, the system achieves high robustness and accuracy. Future work will be aimed at integrating this software into hardware circuits, facilitating real-time applications and further enhancing the system's performance in practical settings. This transition to hardware implementation is crucial for applications requiring immediate response times, such as in security and communication systems.

Acknowledgments

The study was supported by Summit-Tech Resource Corp. and by projects under Grant Nos. NSTC 113-2622-E-390-001 and NSTC 113-2221-E-390-011.

References

1 Y. Chen and R. Zhou: Brain Sci. **12** (2022) 1629.
2 M. H. Chiang, C. H. Lai, and H. S. Chiu: Proc. 35th Conf. Computational Linguistics and Speech Processing (ROCLING, 2023) 390–396.
3 A. R. Douglas and R. C. Rose: IEEE Trans. Audio Speech Process. **3** (1995) 72.
4 N. Chauhan, T. Isshiki, and D. Li: SN Comput. Sci. **4** (2023) 531.
5 C. Y. Chen, Y. H. Hsu, and C. C. Chang: Proc. 35th Conf. Computational Linguistics and Speech Processing (ROCLING, 2023) 367–370.
6 I. Bilik and J. Tabrikian: Proc. 13th IEEE Workshop on Statistical Signal Processing (2005) 399–404.
7 M. A. A. Albadr, S. Tiun, M. Ayob, M. Mohammed, and F. T. AL-Dhief: Cognit. Comput. **13** (2021) 1136.
8 K. S. Ahmad, A. S. Thosar, J. H. Nirmal, and V. S. Pande: 8th Int. Conf. Advances in Pattern Recognition (IEEE, 2015) 1–6.
9 D. A. Reynolds, T. F. Quatieri, and R. B. Dunn: Digital Signal Process. **10** (2000) 19.
10 S. Farsiani, H. Izadkhah, and S. Lotfi: Comput. Electr. Eng. **100** (2022) 107882.
11 M. Afify, Y. Gong, and J. P. Haton: Comput. Speech Lang. **10** (1996) 23.
12 A. S. Dhanjal and W. A. Singh: Multimed. Tools Appl. **83** (2024) 23367.
13 Y. Fan, C. T. Chen, and C. F. Yang: Sens. Mater. **34** (2022) 2809.