Received August 31, 2021, accepted September 26, 2021, date of publication October 4, 2021, date of current version October 15, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3117602

Self-Defined Text-Dependent Wake-Up-Words Speaker Recognition System

TSUNG-HAN TSAI, (Member, IEEE), PING-CHENG HAO, AND CHIAO-LI WANG
Department of Electrical Engineering, National Central University, Taoyuan 32001, Taiwan

Corresponding author: Tsung-Han Tsai (han@ee.ncu.edu.tw)

This work was supported by the Ministry of Science and Technology, Taiwan, under Grant MOST 108-2221-E-008-078-MY3.

ABSTRACT In recent years, wake-up-words (WUW) technology is highly developed in some speaker recognition system. It is the progress of verifying a person's claimed identity from their voice characteristics, and can be efficiently deployed in some consumer applications. In this paper, we proposed a self-defined text-dependent wake-up-words (WUW) speaker recognition system and its implementation. The whole system is divided into two phases: training phase and testing phase. In the training phase, a wake-up word by language is recorded, and the voice segment is cut out by using Voice Activity Detection (VAD). Then we use the Mel-Frequency Cepstral Coefficients (MFCC) as the pre-processing to extract the speech features. After obtaining the speech features, we use Gaussian Mixture Model (GMM) and Hidden Markov Model (HMM) simultaneously for training. In the testing phase, we build GMM and HMM continuously and use the Levenshtein Distance (LD) to calculate the differences of the state sequences between the dataset and the unknown speech input. If the unknown speech input passes the threshold, then it means a wake-up event is derived. The experimental results show that the average accuracy is 93.31 %, 82.42% and 3.38 % in 10dB, 5dB and 0dB of Signal Noise Ratio (SNR) respectively. The CPU and memory usage of entire system is around 757 MIPS and 40MB respectively.

INDEX TERMS Speaker recognition, customized wake-up word, mel-frequency cepstral coefficients, Gaussian mixture model, hidden Markov model, real-time operation.

I. INTRODUCTION

With the rapid development of human-computer interaction and the Internet of Things (IoT) technology, Natural Language Processing (NLP) becomes more and more popular. One of application is intelligent voice assistant. It helps users to get important information on the household appliances easily by using voice. Speech recognition problem can be described as seeking the most suitable word sequence based on a segment of voice. It is constructed by the model to convert and find a word sequence, such as the translation to a sequence of Hidden Markov model [1]–[3].

Key Word Spotting (KWS) and WUW are two main techniques with some similar basis. KWS is not user specific. It detects specific keywords within other words, sounds, and noises often without individually modeling the non-keywords [4]. Some research used HMM to build the specific keyword and other non-keywords to determine whether it is the correct wake-up words or not [5]–[7]. The problem of the most KWS methods is that they need many datasets to train the model to achieve high accuracy, such as Apple and Google. As a result, it is difficult to obtain such a large amount of speech data generally. The accuracy will decrease when the training data is not enough. To solve this problem, dynamic time warping (DTW) is a method of template matching proposed in [8]. However, it is not robust since it only used DTW for KWS. Some researches show a useful method with Human Factor Cepstral Coefficients ENS (HFCC-ENS) and DTW to improve the result [9]. It represents each speech frame by combining segmental DTW and GMM [10]. They used TIMIT dataset to train and test in their work. A result shows that too many Gaussian components will cause the model to be very sensitive to variations such as little noise in the training data since overfitting. Based on the results, the number of 50 for GMM components is the best choice in their work.

The WUW is related to KWS. The difference between them is that the goal of WUW system is to detect the right word. In [11], the authors explain that system will always detect the voice and wake up if the word is right. This means

The associate editor coordinating the review of this manuscript and approving it for publication was Yu-Cheng Fan.

WUW allows to activate these systems with speech commands. There are some researches about WUW paradigm in different aspects, such as the WUW with noise environments, the speed of utterance, the location of the target speaker, and so on. A research used the open source in CMU Sphinx to set up speech recognition WUW system and modified it as in [12].

Most of devices design the Wake-Up-Word (WUW) to activate the service in practical issue. As usual, any WUW between 3 and 6 syllables of WUW is very suitable in daily life. If there are too few syllables, it is easy to cause awakening when other words with the same syllable are spoken in daily conversations. On the other hand, when there are too many syllables (more than 6), it will be less intuitive and inconvenient to use. Also, the voiceprint comparison and sequence comparison will be also less accurate. Nowadays, most of the WUW is fixed and cannot be changed on mobile devices. If the device is going to wake up, users must say the words set up already by the developers. It is inconvenient for consumers.

In this paper, a self-defined WUW recognition system is proposed. As the important self-defined feature, it means the speaker can customize their own WUW by their wish. Any WUW between 3 and 6 syllables is allowable for awaken and become their own wake-up word. Our system performs well with high accuracy and low false accept rate. To widely apply our technique into most applications, we implement it an embedded system and operate smoothly in real-time. Another feature in our system is that it does not need to connect to Internet for using, so users' voice data will keep privacy and safety.

This paper is organized as follows: In Section 2, we introduce the related works of WUW. In Section 3, the proposed system is introduced. Section 4 presents the experimental results of software algorithm and embedded system implementation. Finally, Sections 5 gives the conclusions.

II. RELATED WORKS

Speaker recognition technology has been widely used. An important technique is to recognize the speaker by comparing the corresponding WUW stored in system. There are two kinds of speaker recognition: text independent and text dependent. A research shows the difference between them [13]. Any kind of text can be spoken during testing and training phase in text independent technique. On the other hand, the spoken text should be same during each phase in text dependent.

Text independent speaker recognition is introduced in [14]. They combined GMM and support vector machine (SVM) approaches to improve speaker identification system. To extract the voice feature, the authors take every 20 ms to make a frame where frame step is 10 ms. It used 20 dimensions of MFCC to make feature matrix. Then they used MFCC feature matrix to train a GMM with 50 components by Expectation-Maximization Algorithm. The Gaussians components were first shown to represent characteristic spectral shapes from the phonetic sounds which comprise a person's voice [15], [16]. Identification performance of the GMM is insensitive to the method of model initialization. This means model initialization can be random generated and maintains high identification performance with increasing the number of the speakers.

The text dependent speaker recognition technique is more suitable in a pre-defined system and will be introduced in following works. In [17], a method for text-dependent speaker recognition with Cepstral Compensating Vector and HMM has been proposed. The system is based on learning speaker-specific compensators for each speaker. The compensator is essentially a speaker to speaker transformation. This speaker-specific compensator captures the characteristics of the speaker to recognize. The experimental result shows high accuracy in this work. In [18], they used Vector Quantization (VQ) which designed a source codebook to represent a particular speaker saying a specific utterance. The speaker is accepted if the verification utterance's quantization distortion is less than speaker-specific threshold. It means that VQ can also be a method for text dependent speaker recognition. In [19], the author proposed a method for text-dependent speaker recognition in Vietnamese. The system is modeled for each speaker using GMM and the phonemes in the keywords are represented by HMM [20]. The prior and posterior probabilities for keywords and speakers have been combined together to identify speakers. The results show that using the posterior probability models has improved recognition results for sort keywords, giving a high correct recognition rate. Also, a GMM based speaker recognition system and a speaker verification system are introduced in [21], [22]. They presented a framework for speaker verification by preserving the speaker privacy and showed that GMM is a simple and effective approach for speaker recognition.

Speech feature extraction plays an important role in speech processing. It transforms the speech signal to a set of feature vectors. The speech spectrum has been shown to be very effective for speaker recognition. This is because spectrum reflects a person's vocal structure, and shows the main physiological factor which distinguishes one person's voice from others. Many feature extraction methods such as Mel-Frequency Cepstral Coefficients (MFCC) and Linear Predictive Codes (LPC) has been discussed in general speaker/speech recognition works [23]. Adapting LPC in speaker recognition can be found in [24]–[26]. A research combined MFCC method with the Hilbert spectrum has been proposed [27]. This structure is tested not only for clean speech, but also for speech corrupted by low-frequency noise and environmental noise. It shows a better result than the one only using MFCC. Furthermore, other improvement structures method is Modified Mel-Frequency Cepstral Coefficients (MMFCC) and Gammatone frequency cepstral coefficient (GFCC) in [28]. Overall, there are many methods and modified algorithms for extracting speech features. However. the extracting method depends on specification in different systems. Some systematic verifications had been discussed. In [29], it described about the speaker verification work. And a novel research demonstrates the classifiers and databases of text dependent speaker recognition [30].

To find out the exact and correct pronunciation, the work requires an assessment system to measure the distance in the pronunciation of English words [31]. The assessment system requires a method to measure the distance parameters that will be used in the assessment system. Parameters to be measured are Phonetic, Syllable, and Phonetic Length.

III. PROPOSED SYSTEM

The flowchart of the proposed system is provided in Figure 1. Since our motivation is to construct a complete and playable system, several individual techniques for voice processing is developed. In training part, we use Voice Activity Detection (VAD) to cut the voice first. Then we use MFCC to extract the features. By the feature data from MFCC, GMM is used to make the speaker recognition model, and HMM is focused on training the time sequence model at the same time. By the training work, we have the GMM and HMM into model pool for comparison. In our system, whenever we want to add a new wake-up words (no matter what it is), we must reenter the training phase. After recording the wake-up words for three times, the speech data are modeled and placed in the database. In the testing part, the system will check the similarity of the GMM models first. If the similarity score is higher than the threshold, then the system will compare the HMM model continuously. In comparison of the HMM models, Levenshtein distance algorithm is used to calculate the differences of the state sequences between the sequences data in the model pool and the unknown voice input. After the two different scores passing their individual thresholds sequentially, the system will be awakened.

A. PRE-PROCESSING ON VOICE ACTIVITY DETECTION

In the pre-processing stage, the system will capture the voice segment by VAD with 16K sample rate and 16 bits in every data point. Considering the system in [32], we cut every 20ms for one frame where the overlap is 10ms. Next, the VAD module is used to calculate every frequency bands energy. The frequency bands are 80Hz∼250Hz, 250Hz∼500Hz, 500Hz∼1kHz, 1kHz∼2kHz, 2kHz∼3kHz and 3kHz∼4kHz respectively. These bands are inputted to GMM to make the judgement whether the audio segment is silence or not. After the judgement, it will stack the segment if it is not a silence.

B. FEATURE EXTRACTION

MFCC is a mainstream method to extract features from sound data where Mel filter bank is the simulation of human ear [33]. The proposed system uses MFCC to extract features. Figure 2 shows the detailed steps. First, we use the pre-emphasis to emphasis the high frequency and eliminate the effects of the vocal cord and lips during speaking. Second, we use the sliding window and multiply it with Hamming

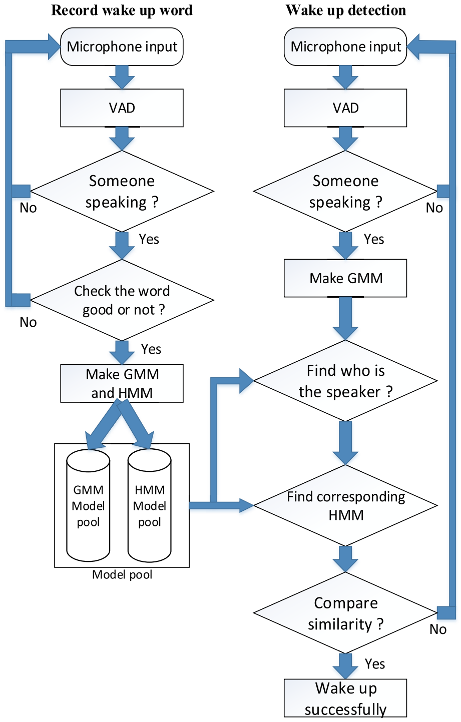

FIGURE 1. System overflow. After recording wake-up word, the system will make a speaker model pool and then enter the detection mode.

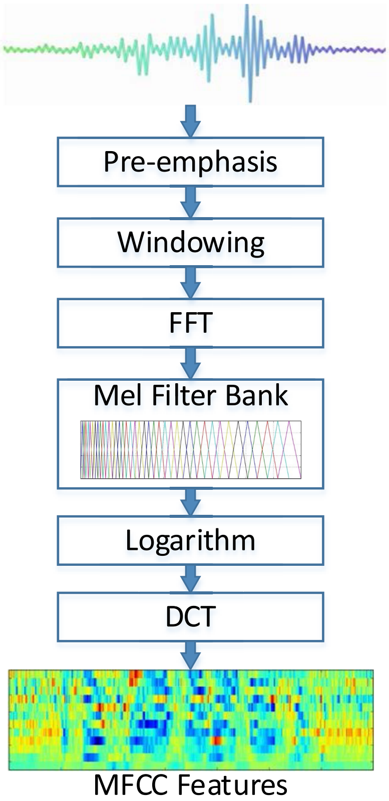

FIGURE 2. Each step of extracting Mel-frequency Cepstral coefficients.

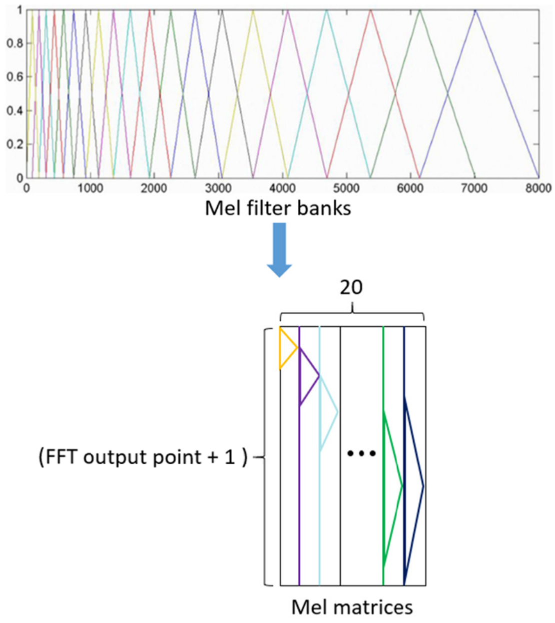

FIGURE 3. Manufacturing 20 sets of 257-dimentional Mel matrices for the following multiplication.

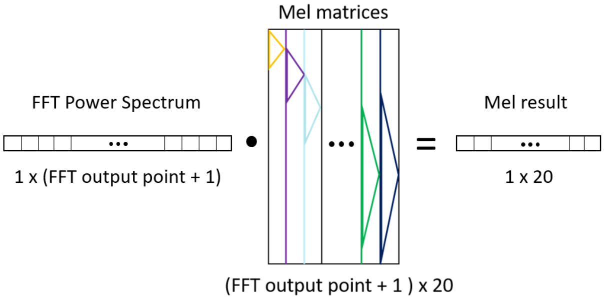

FIGURE 4. Matrix multiplying. The data in a frame processed by FFT will pass the Mel-filter banks and get 1 × 20 feature data.

window. Third, Fast Fourier Transform (FFT) will transform the signal from time domain to frequency domain. Fourth, the energy is calculated after FFT and then multiplied with 20 Mel filter banks. Figure 3 shows the Mel filter bank where 20 sets of 257-dimentional Mel matrices is constructed for the following multiplication. Figure 4 demonstrates the matrix multiplying. After FFT, the data in a frame will pass the Mel-filter banks and get 1 × 20 feature data. Finally, we take Logarithm and Discrete Cosine Transform (DCT) to convert to Cepstral, where we choose top 20 result as MFCC output. The Mel filter bank formula is shown as following [34]:

$$H_m(\mathrm{k}) = \begin{cases} 0, & \mathrm{k} < f(\mathrm{m}-1) \\ \dfrac{k - f(m-1)}{f(m) - f(m-1)}, & f(\mathrm{m}-1) \le \mathrm{k} < f(\mathrm{m}) \\ 1, & \mathrm{k} = f(m) \\ \dfrac{f(m-1) - k}{f(m+1) - f(m)}, & f(\mathrm{m}) < \mathrm{k} \le f(\mathrm{m}+1) \\ 0, & \mathrm{k} > f(\mathrm{m}+1) \end{cases} \tag{1}$$

where M is number of fillers, f () is the list of $M + 2$ Mel spaced frequencies calculated from:

$$m = 2595 * log_{10}(1 + f/700)$$

f is frequency. The number of FFT we choose is 512 point because there are 320 samples per frame. After taking 20 dimensions from MFCC result, we calculate the first order differential as another 20 dimensions feature and stack together to make a 40-dimension feature matrix. It displays more timing variation characteristics in the feature data.

C. SPEAKER MODEL

GMM is a probability model to describe irregular distribution, and is usually used in speaker recognition [35]. The parameter "Component" means how many Single Gaussian Model (SGM) in GMM. The researches give the details of derivation [36], [37]. When input data x is multiple dimension, SGM distribution is as:

$$P(x \mid \theta) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{(x-\mu)^T \Sigma^{-1} (x-\mu)}{2} \right) \tag{2}$$

where u is expectation vector, Σ is covariance matrix and D is dimension of input x.

The formula of GMM is below:

$$\lambda(x|\theta) = \sum_{k=1}^{K} \alpha_k P(x \mid \theta_k) \tag{3}$$

where λ is GMM, x is input, k is component, α_k is probability of each SGM. To train GMM for the speaker, we use expectation-maximization algorithm [38]. This algorithm is very popular for training GMM in unsupervised manner. In expectation-maximization algorithm, the expectation step is by use of the probability to find max likelihood, and the maximize step is by use of max likelihood to update parameters. The formula of E-step and M-step is as below:

E-Step:

$$\gamma_{jk} = \frac{\alpha_k \emptyset(x_j \mid \theta_k)}{\sum_{k=1}^{K} \alpha_k \emptyset(x_j \mid \theta_k)} \tag{4}$$

where

$$j = 1, 2, \ldots, N; \quad k = 1, 2, \ldots, K$$

M-Step:

$$\mu_k = \frac{\sum_{j=1}^{N} (\gamma_{jk} x_j)}{\sum_{j=1}^{N} \gamma_{jk}}, \quad k = 1, 2, \ldots, K \tag{5}$$

$$\Sigma_k = \frac{\sum_{j=1}^{N} \gamma_{jk} (x_j - \mu_k)(x_j - \mu_k)^T}{\sum_{j=1}^{N} \gamma_{jk}} \tag{6}$$

where

$$k = 1, 2, \ldots, K$$

$$\alpha_k = \frac{\sum_{j=1}^{N} \gamma_{jk}}{N}, \quad k = 1, 2, \ldots, K \tag{7}$$

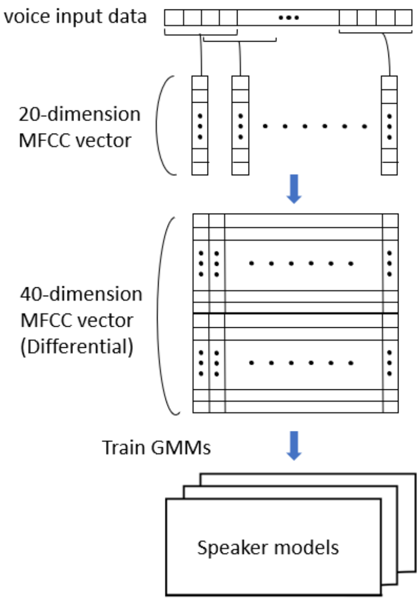

FIGURE 5. The development of the speaker model flowchart.

We repeat E and M step until the model converges so that the phase of making speaker model will be finished. Figure 5 shows the development of the speaker model. In Fig. 5, we differentiate it once and stack it with the original one by the feature data.

D. WAKE-UP-WORD (WUW) MODEL

We use Baum-Welch algorithm to train a HMM in Fig. 6 and use Viterbi algorithm to find the state sequence of the WUW model. The reason we choose HMM to make the word model is that HMM is a statistical model. The characteristic is that hidden states can only be calculated from observation. It can describe a time-dependent series perfectly to solve the problems which needs to consider timing information [39], [40]. This method is very useful in speech processing [41]. Simultaneously, we train a 16 component GMM and save these models and state sequence into model pool. In our simulation, we observe that when the 16 sets of Gaussian models is used in GMM, the accuracy will be the highest and not too sensitive. An HMM can be expressed as:

$$\theta = (A, B, \pi) \tag{8}$$

where A is state transition probability matrix, B is observation probability matrix and π is initial state probability vector. To train the HMM by Baum-Welch algorithm, we set the random parameters of A, B and π first. Then we use the forward procedure as:

$$\alpha_i(t) = P(Y_1 = y_1, \ldots, Y_t = y_t, X_t = i \mid \theta) \tag{9}$$

The probability of seeing the observations $y1$ to yt and state i at time t. This is found recursively as:

$$\alpha_i(1) = \pi_i b_i(y_1) \tag{10}$$

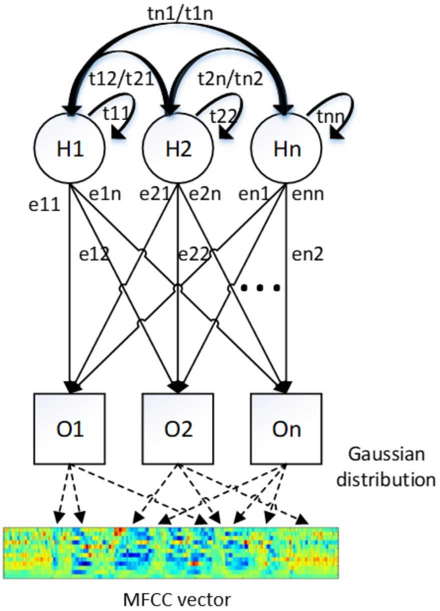

FIGURE 6. WUW Gaussian distribution HMM. This step calculates the probability of each syllable and make a priority model.

$$\alpha_i(t+1) = b_i(y_t + 1) \sum_{j=1}^{N} \alpha_j(t)\, a_{ji} \tag{11}$$

Then we use backward procedure as:

$$\beta_i(t) = P(Y_{t+1} = y_{t+1}, \ldots, Y_T = y_T, X_t = i \mid \theta) \tag{12}$$

That is the probability of the ending partial sequence y_{t+1} to y_T given starting state at time t. We calculate this by:

$$\beta_i(T) = 1 \tag{13}$$

$$\beta_i(t) = \sum_{j=1}^{N} \beta_j(t+1)\, a_{ij} b_j(y_{t+1}) \tag{14}$$

Now we can calculate the probability variables as:

$$\gamma_i(t) = \frac{P(X_t = i, Y \mid \theta)}{P(Y \mid \theta)} = \frac{\alpha_i(t)\,\beta_i(t)}{\sum_{j=1}^{N} \alpha_i(t)\,\beta_i(t)} \tag{15}$$

$$\begin{aligned}\delta_{ij}(t) &= \frac{P(X_t = i, X_{t+1} = j, Y \mid \theta)}{P(Y \mid \theta)} \\ &= \frac{\alpha_i(t)\, a_{ij} \beta_j(t+1)\, b_j(y_{t+1})}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i(t)\, a_{ij} \beta_j(t+1)\, b_j(y_{t+1})}\end{aligned} \tag{16}$$

The parameters of the HMM can now be updated as:

$$\pi_i^* = \gamma_i(1) \tag{17}$$

which is the expected frequency spent in state i at time 1.

$$a_{ij}^* = \frac{\sum_{t=1}^{T-1} \delta_{ij}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)} \tag{18}$$

$$b_i^*(v_k) = \frac{\sum_{t=1}^{T} 1_{y_t = v_k} \gamma_i(t)}{\sum_{t=1}^{T} \gamma_i(t)},$$

$$1_{y_t = v_k} = \begin{cases} 1; & if\ y_t = v_k \\ 0; & otherwise \end{cases} \tag{19}$$

Here, a_{ij}^* is the expected number of transitions from state i to state j and $b_i^*(v_k)$ is the expected number of times. The output observations have been equal to v_k while in state i over the expected total number of times in state i. These steps are now repeated iteratively until converge or a certain number of times.

Next, we can find the state sequence after completing training HMM. To do this, the Viterbi Algorithm is used:

$$p_l(i,x) = e_l(i)\max_k\left(p_k(j,x-1)\,p_{kl}\right) \tag{20}$$

where x is input and i is the most possible state of input x. Figure 7 shows the schematic diagram of the state sequence by Viterbi algorithm. The proposed system will save HMM and state sequences with speaker GMM models into the model pool for testing in the next step.

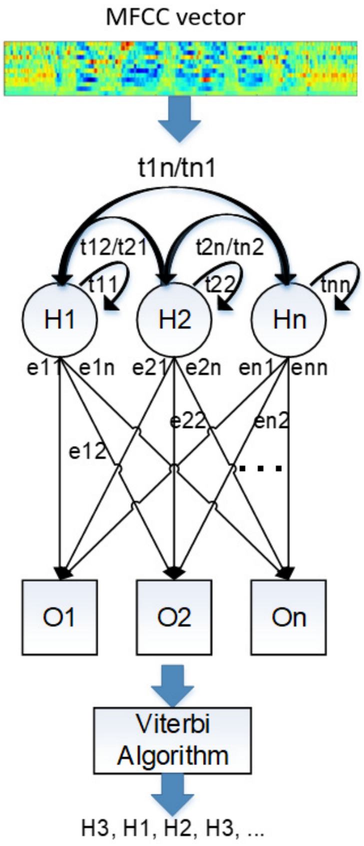

FIGURE 7. The schematic diagram of the state sequence by Viterbi algorithm.

E. SPEECH MODEL ON TESTING

After the model building phase, the proposed system will be operated in the testing phase or in the listening phase because the system will always detect the sound. In the testing phase, system will check an unknown speech segment cut by VAD and then find out the corresponding speaker. We need to calculate the log likelihood of the GMM to find the corresponding speaker. The log-likelihood formula is shown as:

$$\ln p(D|\pi,\mu,\Sigma) = \sum_{n=1}^{N} \ln\left\{\sum_{k=1}^{K} \pi_k N\left(x_n \mid \mu_k,\Sigma_k\right)\right\} \tag{21}$$

where π, μ and Σ is the parameters of GMM, k is component. In order to judge how similar between the two state sequences, we use Levenshtein distance to compare the result. Levenshtein distance is a dynamic algorithm to calculate similarity between two sequences with different length, hereby same length also works. This method is usually used for DNA analyze, spelling checking and speech recognition [42]–[44]. The formula is as:

$$lev_{a,b}(i,j) = \begin{cases} \max(i,j)\,; & \text{if } \min(i,j)=0 \\ \min\begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j-1)+1, \\ where\ (a_i \neq b_j)\,; \end{cases} & \text{otherwise} \end{cases} \tag{22}$$

where the length of sequence a is I and length of sequence b is J. By a matrix calculating, every element of matrix has been determined. The Levenshtein distance value of these two sequences is at lev (I,J). The value is non-negative, and the smaller the more similar between sequences.

If the input GMM log-likelihood is passed and state sequences are similar, the system will be awakened successfully. After users wake up the system successfully, the whole process is done. If the WUW is not correct, the system will not be awakened and continued to listen the next voice input.

IV. EXPERIMENT RESULTS

We set the testing environment with 1.5 meter between speaker and microphone, the angle between the white noise and the speaker's voice is 60 degrees. The experimental environment is shown in Fig. 8.

The whole system is performed and divided into two phases: training phase and testing-comparison phase. The training phase has been constructed and discussed on Section 3. In the testing-comparison phase, VAD and Mel-Frequency Cepstral Coefficients are still used for unknown voice input. Next, this feature will be calculated through the log likelihood of the GMM to find the corresponding speaker, and the Viterbi algorithm is used to calculate the state sequence of the unknown speech through Hidden Markov Model.

The system was tested under the environment in 10dB (A task), 5dB (B task), and 0dB (C task) of SNR, respectively. Due to the difficulty of collecting voice data, we test five people in the laboratory. Next, we choose ten wake-up words: "Hello Smart Light", "Chih ma kai men", "Ni hao pai tu", "Hsiao ai tung hsueh", "Smart Mirror", "OK Google", "Hey Siri", "Hello Jarvis", "Magic Robot" and "My Computer" as training data. All these ten wake-up words include

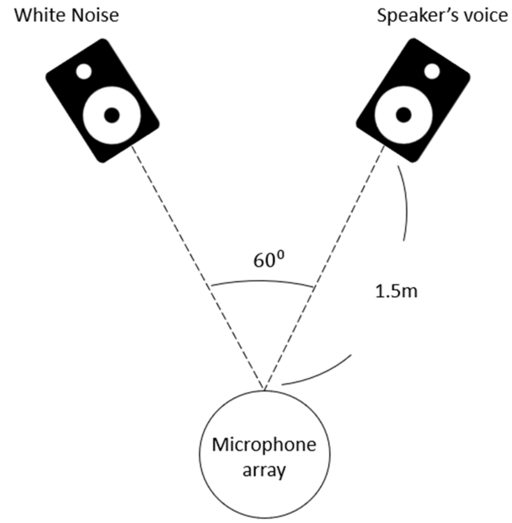

FIGURE 8. Experimental environment of the proposed system.

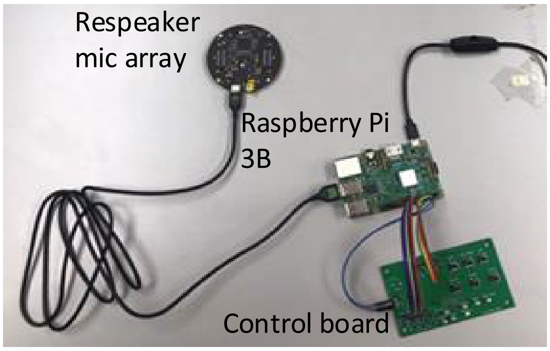

FIGURE 9. The demonstration on embedded system.

English, Chinese and Vietnamese. Every wake-up word is repeatedly played with 1000 times to test the accuracy. Table 1 and Table 2 show the accuracy results of WUWs testing under clean environment and different white noise environments, respectively.

TABLE 1. Ten WUWs testing under clean environment.

	Accuracy
Hello Smart Light	100 %
Chih ma kai men	99.7 %
Ni hao pai tu	99.9 %
Hsiao ai tung hsueh	100 %
Smart Mirror	99.9 %
OK Google	99.8 %
Hey Siri	100 %
Hello Jarvis	99.6 %
Magic Robot	99.8 %
My Computer	99.7 %
Average accuracy	**99.84%**

To evaluate the whole system, the False Reject Rate (FRR) and the False Accept Rate (FAR) are introduced below. FRR is the correct input but judgement failed. In our work, it is represented that the speaker says the correct WUW but the system does not wake up. The FRR is 1-Accuracy. FAR is incorrectly taking the wrong input as true answer. These two indicator scores are very important for deciding performance of a WUW system.

TABLE 2. Ten WUWs testing under three different white noise environments.

	A task	B task	C task
Hello Smart Light	93.4 %	82.8 %	4.2 %
Chih ma kai men	92.6 %	83.0 %	3.6 %
Ni hao pai tu	93.2 %	83.0 %	4.4 %
Hsiao ai tung hsueh	93.9 %	82.2 %	4.3 %
Smart Mirror	92.6 %	82.4 %	2.1 %
OK Google	93.3 %	83.1 %	2.5 %
Hey Siri	93.9 %	82.7 %	4.8 %
Hello Jarvis	93.8 %	82.3 %	2.7 %
Magic Robot	93.5 %	81.3 %	2.4 %
My Computer	92.9 %	81.4 %	2.8 %
Average accuracy	**93.31%**	**82.42%**	**3.38%**

TABLE 3. Average accuracy and false accept rate test.

Accuracy in repeat playing 1000 times	Accuracy in repeat speaking 100 times	False Accept Rate
99.84 %	99 %	3 times in 24hr

TABLE 4. Resource usage of entire system.

Memory Used	CPU Usage	Calculation Time
40 MB	757 MIPS	0.5 sec

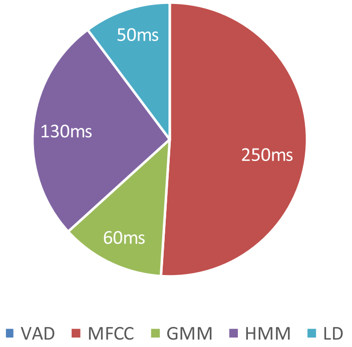

FIGURE 10. Profile of the timing consumption in each module.

We test on actual human speaking with 100 times in experiment and can get the 99 % accuracy. To test FAR, we play the audio wave file from Amazon Alexa open source testing data where it is 24-hour long sequence. To make a fair evaluation, this test sequence should not have any correct WUW. For FAR testing result, it shows three times FAR in 24 hours, as also shows in Table 3.

The design in embedded system is also provided. We construct a real and playable demonstration system on Raspberry Pi 3B board and operated in the real environment as shown in Fig. 9. We use Respeaker microphone array v2 to capture the sound data and use a control board to control the proposed system so that we do not need the keyboard, screen

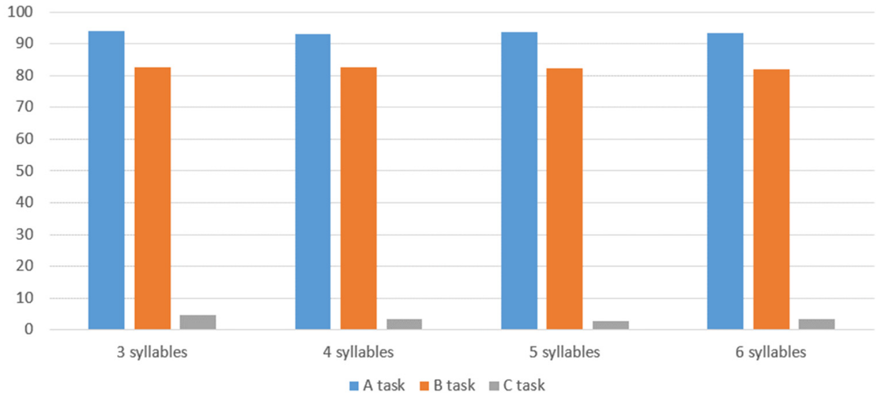

FIGURE 11. Accuracy of WUWs with different numbers of syllables in three different environments.

and mouse such as the product of smart home appliance. In Table 4, the CPU and memory usage of entire system is around 757 MIPS and 40MB, respectively. Fig. 10 shows the profile of the timing consumption in each module. According to the test results, the overall identification time is about 0.5s which meets the real-time requirement. Finally, Fig. 11 shows the histogram of the accuracy of WUWs with different numbers of syllables in three different environments. The x-axis is the syllables and the y-axis is accuracy. This result shows that our system performs well in wake-up words with 3 to 6 syllables.

V. CONCLUSION

This paper presents self-defined wake-up words speaker recognition and its embedded system implementation that can be operated in real time. All of the processes are executed without Internet so that the sound data from users will be in privacy and safety. In the proposed system, VAD technique is used for cutting the voice, and MFCC is used for extracting voice features. We build GMM model to make the speaker's voiceprint database, and Gaussian distributed HMM model to make each sequence model of the phonemes. In detail, we take twenty dimensional MFCC and its first-order differential to make speech feature matrix after cutting voice slides by VAD. We use Levenshtein distance to compare the state sequence. In the training phase, the GMM and the HMM are used to establish the speaker model and the speech model simultaneously. In testing phase, we calculate Gaussian Mixture Model similarity first and use Levenshtein distance to compare dataset state sequence with the unknown speech state sequence. If both of them pass the threshold, then it means a wake-up event is derived successfully. The experimental results are all performed in Raspberry Pi 3B embedded board with different languages as testing. Several kinds of noisy environments are also involved for simulation. The result shows that it can achieve 99.84% accuracy in clean environment, and three times per day of False Accept Rate. The processing time of speaker recognition is around 0.5 sec to meet the real-time requirement.

REFERENCES

[1] C. Xue, "A novel English speech recognition approach based on hidden Markov model," in *Proc. Int. Conf. Virtual Reality Intell. Syst. (ICVRIS)*, Aug. 2018, pp. 1–4.

[2] S. Mao, D. Tao, G. Zhang, P. C. Ching, and T. Lee, "Revisiting hidden Markov models for speech emotion recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 6715–6719.

[3] R. W. Fan, "Positive sequential data modeling using continuous hidden Markov models based on inverted Dirichlet mixtures," *IEEE Access*, vol. 7, pp. 172341–172349, 2019.

[4] A. Zehetner, M. Hagmüller, and F. Pernkopf, "Wake-up-word spotting for mobile systems," in *Proc. 22nd Eur. Signal Process. Conf. (EUSIPCO)*, Lisbon, Portugal, 2014, pp. 1472–1476.

[5] H. Ketabdar, J. Vepa, S. Bengio, and H. Bourlard, "Posterior based keyword spotting with a priori thresholds," in *Proc. Interspeech*, Sep. 2006.

[6] S.-G. Leem, I.-C. Yoo, and D. Yook, "Multitask learning of deep neural network-based keyword spotting for IoT devices," *IEEE Trans. Consum. Electron.*, vol. 65, no. 2, pp. 188–194, May 2019.

[7] A. Tavanaei, H. Sameti, and S. H. Mohammadi, "False alarm reduction by improved filler model and post-processing in speech keyword spotting," in *Proc. IEEE Int. Workshop Mach. Learn. Signal Process.* Boadilla del Monte, Spain: Santander, Sep. 2011, pp. 1–5.

[8] J. Bridle, "An efficient elastic-template method for detecting given words in running speech," in *Proc. Brit. Acoust. Soc. Meeting*, 1973, pp. 1–4.

[9] D. von Zeddelmann, F. Kurth, and M. Müller, "Perceptual audio features for unsupervised key-phrase detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Dallas, TX, USA, Mar. 2010, pp. 257–260.

[10] Y. Zhang and J. R. Glass, "Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams," in *Proc. ASRU*, Dec. 2009, pp. 398–403.

[11] V. Z. Kepuska and T. B. Klein, "A novel wake-up-word speech recognition system, wake-up-word recognition task, technology and evaluatio," *Nonlinear Anal.*, vol. 71, pp. 2772–2789, Dec. 2009.

[12] V. Kepuska and G. Bohouta, "Improving wake-up-word and general speech recognition systems," in *Proc. IEEE 15th Int. Conf. Dependable, Auton. Secure Comput., 15th Int. Conf. Pervasive Intell. Comput., 3rd Int. Conf. Big Data Intell. Comput. Cyber Sci. Technol. Congr. (DASC/PiCom/DataCom/CyberSciTech)*, Orlando, FL, USA, Nov. 2017, pp. 318–321.

[13] R. Chakroun, L. B. Zouari, M. Frikha, and A. B. Hamida, "Improving text-independent speaker recognition with GMM," in *Proc. 2nd Int. Conf. Adv. Technol. Signal Image Process. (ATSIP)*, Mar. 2016, pp. 693–696.

[14] R. Chakroun, L. B. Zouari, M. Frikha, and A. B. Hamida, "A hybrid system based on GMM-SVM for speaker identification," in *Proc. 15th Int. Conf. Intell. Syst. Design Appl. (ISDA)*, Dec. 2015, pp. 654–658.

[15] K. Kaur and N. Jain, "Performance analysis of text-dependent speaker recognition system based on template model based classifiers," in *Proc. Int. Conf. Signal Process., Comput. Control (ISPCC)*, Waknaghat, India, Sep. 2015, pp. 36–39.

[16] P. Punitha and G. Hemakumar, "Speaker dependent continuous Kannada speech recognition using HMM," in *Proc. Int. Conf. Intell. Comput. Appl.*, Mar. 2014, pp. 402–405.

[17] H. Zeinali, L. Burget, and J. H. Černocký, "A multi purpose and large scale speech corpus in Persian and English for speaker and speech recognition: The deepmine database," in *Proc. IEEE Autom. Speech Recognit. Understand. Workshop (ASRU)*, Dec. 2019, pp. 397–402.

[18] S. Laxman and P. S. Sastry, "Text-dependent speaker recognition using speaker specific compensation," in *Proc. Conf. Convergent Technol. Asia–Pacific Region (TENCON)*, Bengaluru, India, 2003, pp. 384–387.

[19] D. Burton, "Text-dependent speaker verification using vector quantization source coding," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-35, no. 2, pp. 133–143, Feb. 1987.

[20] D. D. Thi Thu, L. T. Van, Q. N. Hong, and H. P. Ngoc, "Text-dependent speaker recognition for Vietnamese," in *Proc. Int. Conf. Soft Comput. Pattern Recognit. (SoCPaR)*, Dec. 2013, pp. 196–200.

[21] S. P. Babu and C. K. Jayadas, "GMM based speaker verification system," *Int. J. Eng. Res. Technol.*, vol. 4, no. 4, pp. 1398–1401, Apr. 2015.

[22] T. Mahboob, M. Khanum, M. S. H. Khiyal, and R. Bibi, "Speaker identification using GMM with MFCC," *Int. J. Comput. Sci. Issues*, vol. 12, no. 2, pp. 126–135, 2015.

[23] N. Dave, "Feature extraction methods LPC, PLP and MFCC in speech recognition," *Int. J. Adv. Res. Eng. Technol.*, vol. 1, no. 6, pp. 1–4, 2013.

[24] N. Chauhan, T. Isshiki, and D. Li, "Speaker recognition using LPC, MFCC, ZCR features with ANN and SVM classifier for large input database," in *Proc. IEEE 4th Int. Conf. Comput. Commun. Syst. (ICCCS)*, Feb. 2019, pp. 130–133.

[25] A. Chowdhury and A. Ross, "Fusing MFCC and LPC features using 1D triplet CNN for speaker recognition in severely degraded audio signals," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1616–1629, 2020.

[26] M. Chougala and S. Kuntoji, "Novel text independent speaker recognition using LPC based formants," in *Proc. Int. Conf. Electr., Electron., Optim. Techn. (ICEEOT)*, Chennai, India, Mar. 2016, pp. 510–513.

[27] R. Sharma, R. K. Bhukya, and S. R. M. Prasanna, "Analysis of the Hilbert spectrum for text-dependent speaker verification," *Speech Commun.*, vol. 96, pp. 207–224, Feb. 2018, doi: 10.1016/j.specom.2017.12.001.

[28] M. A. Islam and A.-N. Sakib, "Bangla dataset and MMFCC in text-dependent speaker identification," *Eng. Appl. Sci. Res.*, vol. 46, no. 1, pp. 56–63, 2019, doi: 10.14456/easr.2019.7.

[29] R. K. Das and S. R. M. Prasanna, "Speaker verification from short utterance perspective: A review," *IETE Tech. Rev.*, vol. 35, no. 6, pp. 599–617, 2018.

[30] A. Larcher, K. A. Lee, B. Ma, and H. Li, "Text-dependent speaker verification: Classifiers, databases and RSR2015," *Speech Commun.*, vol. 60, pp. 56–77, May 2014.

[31] A. Muhammad, A. S. Prihatmanto, R. Wijaya, H. A. Rosyid, H. R. Hakim, A. P. Dana, and U. C. A. Himmah, "Distance measurements method for the demite pronunciation assessment," in *Proc. IEEE 8th Int. Conf. Syst. Eng. Technol. (ICSET)*, Oct. 2018, pp. 189–194.

[32] J. Sohn, N. S. Kim, and W. Sung, "A statistical model-based voice activity detection," *IEEE Signal Process. Lett.*, vol. 6, no. 1, pp. 1–3, Jan. 1999.

[33] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-28, no. 4, pp. 357–366, Aug. 1980.

[34] S. R. Madikeri and H. A. Murthy, "Mel filter bank energy-based slope feature and its application to speaker recognition," in *Proc. Nat. Conf. Commun. (NCC)*, Jan. 2011, pp. 1–4, doi: 10.1109/NCC.2011.5734713.

[35] M. Khadkevich and M. Omologo, "Reassigned spectrum-based feature extraction for GMM-based automatic chord recognition," *EURASIP J. Audio, Speech, Music Process.*, vol. 2013, no. 1, p. 15, Dec. 2013.

[36] M. S. Allili, "A short tutorial on Gaussian mixture models," in *Proc. CRV*, 2010, pp. 1–27.

[37] L. Yu, T. Yang, and A. B. Chan, "Density-preserving hierarchical EM algorithm: Simplifying Gaussian mixture models for approximate inference," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 6, pp. 1323–1337, Jun. 2019.

[38] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal Process. Mag.*, vol. 13, no. 6, pp. 47–60, Nov. 1996.

[39] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*, 4th ed. Cambridge, MA, USA: MIT Press, 2001.

[40] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–285, Feb. 1989.

[41] C. Lévy, G. Linarès, and J. Bonastre, "Compact acoustic models for embedded speech recognition," *EURASIP J. Audio, Speech, Music Process.*, vol. 2009, Dec. 2009, Art. no. 806186.

[42] Y. Dai, H. Zhang, Y. Song, H. Du, and T. Jin, "Signal comparing normalized generalized Levenshtein distance-based searching method for modulation period of micro-Doppler signal," *IEEE Sensors J.*, vol. 18, no. 15, pp. 6254–6262, Aug. 2018.

[43] B. Berger, M. S. Waterman, and Y. W. Yu, "Levenshtein distance, sequence comparison and biological database search," *IEEE Trans. Inf. Theory*, vol. 67, no. 6, pp. 3287–3294, Jun. 2021.

[44] B. Ziólko, J. Gałka, and D. Skurzok, "Speech modelling using phoneme segmentation and modified weighted levenshtein distance," in *Proc. Int. Conf. Audio, Lang. Image Process.*, Shanghai, China, Nov. 2010, pp. 743–746.

TSUNG-HAN TSAI (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the National Taiwan University, Taipei, Taiwan, in 1990, 1994, and 1998, respectively.

From 1999 to 2000, he was an Associate Professor of electronic engineering at Fu Jen University. He joined the National Central University, in 2000. Since 2008, he has been a Full Professor with the Department of Electrical Engineering, National Central University, where he is currently the Director of the Intelligent Chip and System Center. He serves as the Principal Investigator for the National Program for Intelligent Electronics. He has been awarded more than 40 patents and 230 refereed papers published in international journals and conferences. His research interests include VLSI signal processing, video/audio coding algorithms, DSP architecture design, wireless communication, and system-on-chip design. He serves as a technical program committee member or the session chair for several international conferences. He received the Industrial Cooperation Award from the Ministry of Education, Taiwan, in 2003. He received the Best Paper Award from the IEEE International Conference on Innovations in Bio-Inspired Computing and Applications (IBICA), in 2011, and IEEE International Conference on Innovation, Communication and Engineering (ICICE), in 2015. His research team has won many international IC related student design contest awards, including TI DSP Asia Design Contest, in 2008, ISSCC, in 2011, and ISOCC, in 2015. He was the General Co-Chair of the IEEE International Conference on Internet of Things 2014 and the General Chair of the IEEE International Conference on Consumer Electronics-Taiwan 2020 (ICCE-TW). He has served as a Guest Editor for special issues of *Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*.

PING-CHENG HAO received the M.S. degree in electrical engineering from the National Central University, Taiwan, in 2019. His research interest includes speech signal processing.

CHIAO-LI WANG received the M.S. degree in electrical engineering from the National Central University, Taiwan. His research interests include speech signal processing and its hardware design.