electronics

MDPI

Article

Hardware-Accelerated, Short-Term Processing Voice and Nonvoice Sound Recognitions for Electric Equipment Control

Wen-Chung Tsai *, You-Jyun Shih and Nien-Ting Huang

Department of Information and Communication Engineering, Chaoyang University of Technology, Taichung 41349, Taiwan

* Correspondence: azongtsai@cyut.edu.tw; Tel.: +886-4-2332-3000-7843

Received: 13 July 2019; Accepted: 21 August 2019; Published: 23 August 2019

Abstract: We proposed and implemented a sound recognition system for electric equipment control. In recent years, industry 4.0 has propelled a rapid growth in intelligent human–machine interactions. User acoustic voice commands for machine control have been examined the most by researchers. The targeted machine can be controlled through voice without the use of any hand-held device. However, compared with human voice recognition, limited research has been conducted on nonhuman voice (e.g., mewing sounds) or nonvoice sound recognition (e.g., clapping). Processing of such short-term, biometric nonvoice sounds for electric equipment control requires a rapid response with correct recognition. In practice, this could lead to a trade-off between recognition accuracy and processing performance for conventional software-based implementations. Therefore, we realized a field-programmable gate array-based embedded system, such a hardware-accelerated platform, can enhance information processing performance using a dynamic time warping accelerator. Furthermore, information processing was refined for two specific applications (i.e., mewing sounds and clapping) to enhance system performance including recognition accuracy and execution speed. Performance analyses and demonstrations on real products were conducted to validate the proposed system.

Keywords: dynamic time warping; field-programmable gate array; mel-scale frequency cepstral coefficients; short-term processing; equipment control; information processing

1. Introduction

Technological advances in industry 4.0 have propelled rapid growth in intelligent human–machine interactions [1]. People are not satisfied with traditional methods for electric equipment control and more convenient approaches have been investigated [2]. In recent years, voice recognition-controlled machines have generated considerable interest. Such human–machine interactions mainly include system parameter adjustments or remote functional operations [3–8]; accordingly, we implemented a sound recognition and control system. The proposed system enables users to control the targeted machine through sound without the use of any hand-held remote control device. In addition to voice recognition, we incorporated nonhuman voice and nonvoice sound detection for two specific applications. The enhancement of recognition accuracy and execution speed for the short-term nonvoice sounds was the major contribution of this study.

In practice, the mel-frequency cepstral coefficients (MFCCs) parameter [9] is commonly used for extracting and recording sound characters [5,6,8,10]. MFCC was used in this study after refinements such as the following: (1) Adjusting sample parameters, (2) checking the frequency variation, and (3) verifying the energy distribution, which can enhance nonhuman voice or nonvoice sound recognition for a specific application. Dynamic time warping (DTW) [11,12] is an efficient method to measure

the similarity between two temporal sequences and has been applied to voice [13] and image [14] character recognition, biometric signature verification [15], and in recent years, data mining analyses for a financial trading system [16] and satellite image time series analyses for earth observation [17]. We implemented the DTW algorithm to function as a hardware accelerator using the Verilog hardware description language (HDL) and executed the algorithm in a field-programmable gate array (FPGA) embedded system [18–20]. Thus, two real-product demonstrations are provided in this paper; the first is a feeder controlled by a nonhuman voice (i.e., a new sound), and the second is a music player operated by a nonvoice sound (i.e., hand clapping). Processing such short-term and nonvoice sounds for electric equipment control requires a rapid response with accurate recognition. In practice, this could result in a trade-off between recognition accuracy and processing performance. This trade-off has been addressed in this paper.

The rest of this paper is organized as follows: In Section 2, a background of sound control systems is explained. The design methodology, including information processing, sound recognition, and a hardware and software co-design architecture, of the proposed system is introduced in Section 3. Experimental results with performance analyses and demonstrations are described in Section 4. A brief discussion is presented in Section 5. Finally, we conclude the paper in Section 6.

2. Background

Possible applications of the sound control system are first reviewed and anticipated. Next, a basic understanding about extraction of sound characters is provided in Section 2.2. In Section 2.3, we propose a preprocessing method for the generated sound characters which is performance-efficient especially for short-term and nonvoice sounds. Finally, the DTW method used for recognizing different characteristic vectors is introduced in Section 2.4; additionally, literature surveys for discussions on hardware accelerations are included.

2.1. Applications of Sound Control System

Sensing and control are crucial components of industry 4.0, which has propelled rapid growth in intelligent human–machine interactions [1]. Such human–machine interactions can be achieved by constructing appropriate sensors and recognizing user biometric sounds. In recent years, voice recognition-controlled machines have generated considerable interest. Kil et al. [3] presented a speech control mechanism for humanoid robots; to increase robustness and noise tolerance, the authors used zero-crossing binaural mask estimation for speech segregation and recognition. Similarly, Zinchenko et al. [4] provided a speech recognition mechanism to control robotic endoscope holders for medical applications. Gałka et al. [5] presented a voice biometric access system that can further support secured controls such as a door lock and safe deposit box. Park et al. [6] improved voice recognition performance by suppressing acoustic interferences for smart television controls. Furthermore, Ding et al. [7] improved the speech recognition performance of Microsoft's Kinect sensor and remotely operated a two-wheel mobile car and multimedia player. Recently, we implemented a sound recognition system to control operations of a toilet and its attached washlet [8]. Voice and nonvoice sounds, such as speaking and hand clapping, were used to provide convenient water flush, nozzle, and seat heating control as demonstrated in [8]. Furthermore, the proposed sound control method was extended to enable the detection of nonhuman voices. Enhanced recognition accuracy and execution speed are the major contributions of this paper.

2.2. Sound Character Extraction

MFCC [9] is a set of parameters commonly used for recording data in sound character extraction [5,6,8,10]. MFCC contains 13 characteristic vectors to identify a voice frame, in which one vector records the voice energy in a logarithmic value and the other 12 mel-scale cepstral coefficients that represent the voice characters. The process to extract an audio frame into MFCC

characteristic vectors includes steps of pre-emphasis, signal framing, hamming windowing, fast Fourier transform, triangular bandpass filter, and discrete cosine transform, as illustrated in Figure 1.

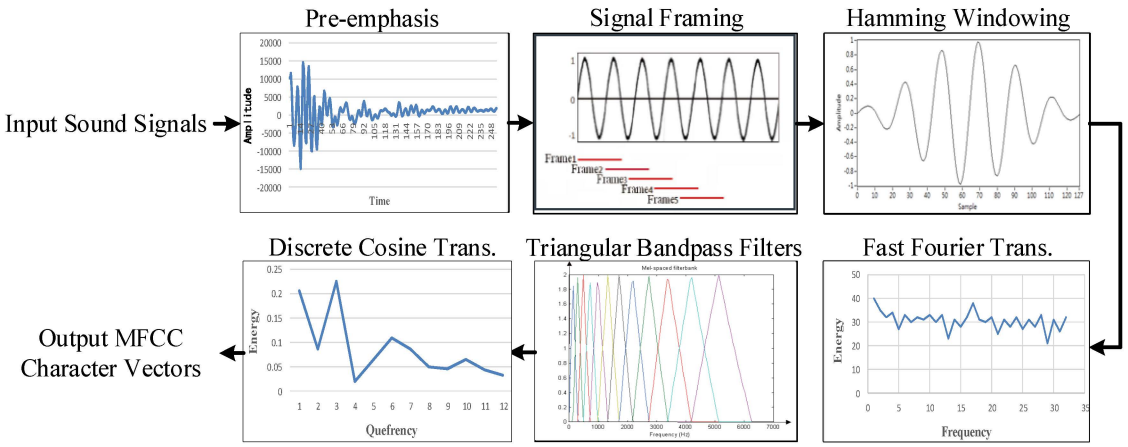


Figure 1. MFCC (mel-frequency cepstral coefficients) characteristic vectors extraction flow.

2.3. Sound Character Preprocessing

Generally, MFCCs are first used to extract characteristic parameters from the raw data of the audio stream. Next, the extracted characters are compared with a set of prerecorded samples using the popular and effective DTW method [12], as depicted in Figure 2a. For example, Muda et al. [10] devised a voice recognition method that used MFCC and then DTW. The DTW algorithm is an efficient method to measure the similarity between two temporal sequences. DTW was first introduced in 1960s [11]. The algorithm has been applied to voice [13] and image [14] recognition, biometric signature verification [15], and in recent years, data mining analyses for a financial trading system [16] and satellite image time series analyses for earth observation [17].

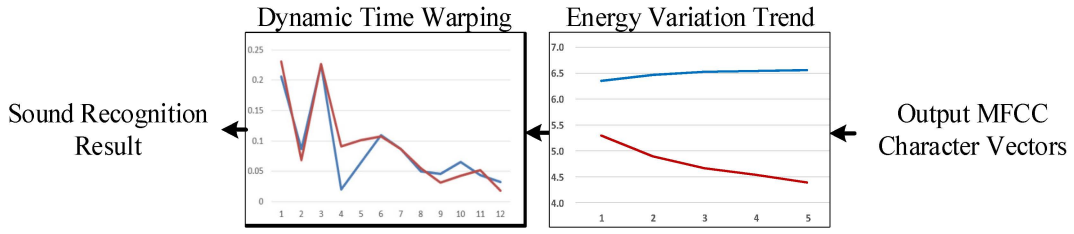


Figure 2. (**a**) DTW (dynamic time warping) character comparison and (**b**) EVT (energy variation trend) energy distribution check.

In addition to the techniques for distinguishing frequency distribution features for general voice recognition, one of the major contributions of this paper is that the proposed method can be applied to increase the accuracy of recognizing short-term, biometric nonvoice sounds such as hand clapping, as illustrated in Figure 2b. The figure shows that because of the short-term property [21], the signal energy distribution of a nonvoice sound (e.g., red line) has a higher degradation than that of a voice sound (e.g., blue line). Therefore, we proposed a signal character recognition scheme that can not only compare the characteristic parameters in the frequency domain (using DTW) but also evaluate the energy distributions in the time domain by the proposed energy variation trend (EVT) method (introduced in Section 4.1.2), which is introduced in more detail in the experimental results section.

2.4. Sound Character Recognition

DTW [12] is a well-known algorithm for measuring similarity between two temporal sequences. In experiments, we applied DTW to calculate an optimal match between two given sequences (i.e., MFCC characteristic vectors). The implemented DTW program compares the characteristic vectors of the input signal and a prerecorded sample, and calculates the distortion amount. For example, suppose we have two characteristic vectors X and Y of lengths n and m, respectively, as follows:

$$\mathrm{X} = x_1, x_1, \ldots, x_n \tag{1}$$

$$\mathrm{Y} = y_1, y_2, \ldots, y_m \tag{2}$$

Consequently, in an n-by-m matrix of vectors X and Y, the two points x_i and y_j exist at a distance $d(x_\mathrm{i}, y_\mathrm{j})$ that can be calculated using the Euclidean distance as follows:

$$d(x_i, y_j) = \sqrt{(x_i - y_j)^2} \tag{3}$$

In DTW, each matrix element (i, j) corresponds to an alignment between the points x_i and y_j. Then, an accumulated distance is measured using a dynamic programming method as follows:

$$D(i, j) = d(i, j) + \min\{D(i-1, j), D(i, j-1), D(i-1, j-1)\} \tag{4}$$

Consequently, the DTW distance between the two characteristic vectors X and Y is the accumulated distance started from the element (0, 0) to the ended element of (n, m) as follows:

$$DTW(\mathrm{X}, \mathrm{Y}) = D(n, m) \tag{5}$$

When comparing time series, a major advantage of using DTW as similarity measure is related to the possibility of applying it for sequences of different lengths (i.e., $n \neq m$). Accordingly, the DTW distance can be more suitable for measuring the similarity between two similar characteristic vectors, which may slightly vary in the time domain. We provided a simple example of computation for the DTW distance of the matrix elements with two sequences, '1,1,1,10,2,3' and '1,1,1,2,10,3'. Figure 3a depicts that the summary of one-on-one element distances is 16 between the two characteristic vectors X and Y, calculated as follows:

$$|1-1|+|1-1|+|1-1|+|10-2|+|2-10|+|3-3| = 16 \tag{6}$$

By contrast, Figure 3b illustrates that the DTW distance between X and Y is a relatively small value of 2, as measured in the following calculation:

$$|1-1|+|1-1|+|1-1|+|1-2|+|10-10|+|2-3|+|3-3| = 2 \tag{7}$$

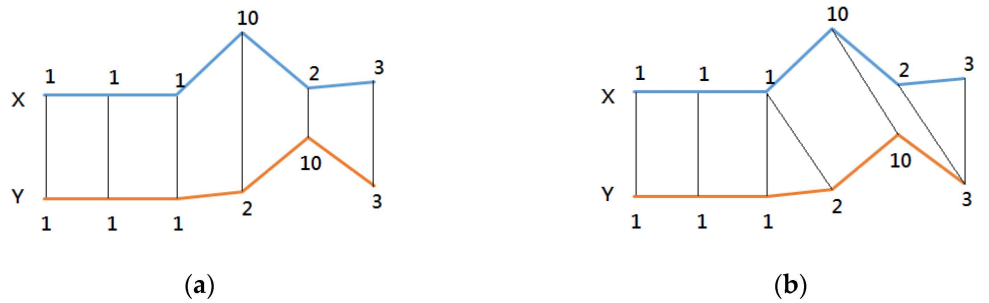


Figure 3. Characteristic distance between vectors X and Y measured using (**a**) one-on-one element distance and (**b**) DTW distance.

Because DTW performs a nonlinear and elastic transformation to calculate the optimal alignment (i.e., minimization of distance) between two points of separated characteristic vectors even if they are out of phase in the time domain, in our opinion, the computational burden of a processor should be reduced using dedicated hardware support, especially for applications that require real-time responses for the recognition results. Although Sart et al. [22] investigated a DTW hardware acceleration using a graphics processing unit for hardware realizations of DTW processing, they did not provide an actual HDL design and its FPGA synthesis result. Tai et al. [23] designed a pure DTW processing unit in very high-speed hardware description language (VHDL); however, the performance was simulated using MATLAB. Mariano et al. [15] implemented the DTW algorithm, which was applied to a FPGA embedded system for biometric online signature verification. Pandey et al. [13] realized the DTW algorithm using VHDL design and Xilinx FPGA [24] to accomplish voice recognition. By contrast, we designed DTW to function as an accelerator using Verilog HDL, which was attached to a system-on-chip bus and evaluated on an Intel FPGA [18]. Moreover, the DTW accelerator was integrated and applied in the proposed sound recognition system for electric equipment control. We introduce the design methodology and provide performance evaluation in following sections.

3. Design Methodology

In this section, we first provide a global view of execution flow of the proposed sound recognition and control system. Next, the execution flow including signal reception, data sampling, and character extraction are introduced in detail (especially, the analyses of the necessary hardware acceleration). Finally, the implemented system architecture and framework are provided.

3.1. Global Sound Recognition and Control System

The execution flow of signal processing and sound recognition of the implemented control system is depicted in Figure 4. In the implemented system, a microphone was used to constantly receive sounds and transmit the analog signals to an audio codec using WM8731 [25], which samples the input signal into digital data for further processing. A volume threshold was set (using programming registers of WM8731) in the proposed design to filter low-volume background noise to the processor and allow the system to enter in the sleep mode to save power and then process data on receiving an interrupt signal from WM8731. Next, the MFCC process was executed using a Nios II microprocessor (μP) [19]; in this step, we provide additional designs to refine the performance for nonhuman voices and nonvoice sounds. Last, the DTW algorithm was designed using Verilog HDL and implemented in an Intel FPGA to considerably accelerate the processing speed of characteristic vector comparisons. Finally, the recognition results are used to operate appropriate electric equipment. Detailed introductions for each execution step are provided in the following sections.

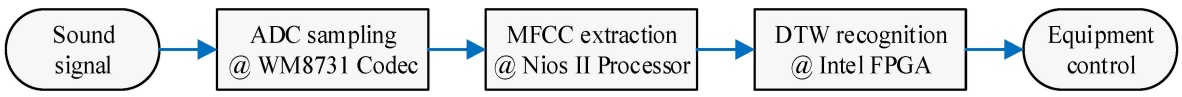
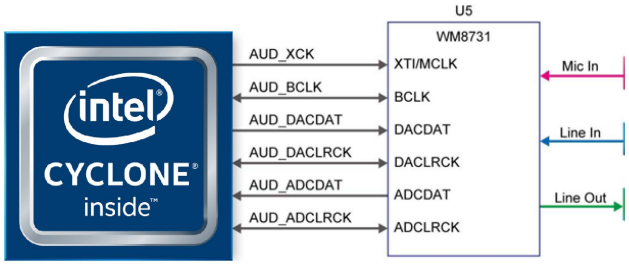


Figure 4. Execution flow of the proposed sound recognition and control system.

3.2. Sound Reception and Signal Conversion

To capture the sound signal, we applied WM8731 [25] as an audio codec connected to the Intel FPGA [18] on the system development board, as illustrated in Figure 5a. The WM8731 codec provides a 24-bit multibit sigma delta analog-to-digital converter (ADC) with oversampling digital interpolation and decimation filters controlled using a two-wire serial interface (i.e., I2C bus), which is used to access registers for functional controls including volume, mute, de-emphasis, and extensive power management, as partially listed in Figure 5b.

Connection with FPGA



Ref. https://www.cirrus.com/products/wm8731/

(**a**)

Control Registers

REGISTER ADDRESS	BIT	LABEL	DEFAULT	DESCRIPTION
0000000 Left Line In	4:0	LINVOL[4:0]	10111 (0dB)	Left Channel Line Input Volume Control 11111 = +12dB . . 1.5dB steps down to 00000 = -34.5dB
	7	LINMUTE	1	Left Channel Line Input Mute to ADC 1 = Enable Mute 0 = Disable Mute
	8	LRINBOTH	0	Left to Right Channel Line Input Volume and Mute Data Load Control 1 = Enable Simultaneous Load of LINVOL[4:0] and LINMUTE to RINVOL[4:0] and RINMUTE 0 = Disable Simultaneous Load
0000001 Right Line In	4:0	RINVOL[4:0]	10111 (0dB)	Right Channel Line Input Volume Control 11111 = +12dB . .1.5dB steps down to 00000 = -34.5dB
	7	RINMUTE	1	Right Channel Line Input Mute to ADC 1 = Enable Mute 0 = Disable Mute
	8	RLINBOTH	0	Right to Left Channel Line Input Volume and Mute Data Load Control 1 = Enable Simultaneous Load of RINVOL[4:0] and RINMUTE to LINVOL[4:0] and LINMUTE 0 = Disable Simultaneous Load

(**b**)

Figure 5. WM8731 audio codec with (**a**) external connections and (**b**) controller registers.

3.3. Signal Processing and Character Recognition

Sound recognition processing is not spontaneous. Whenever a system processes a certain length of audiostream, the processing time is directly proportional to the length of the processing audiostream. Short-term processing (STP) technique [21] performed in both time and frequency domains could shorten the processing time to enable rapid recognition. We sampled the input signal per 20 ms to develop a raw audio frame with 256 sampled data points in the implementation. The STP sound processing and recognition system (Figure 4) was realized using a FPGA embedded system [18,20] and a FPGA-based Nios II μP [19]. In the operation process, MFCC was used to extract characteristic parameters from the audio frames, and then the extracted characteristic parameters were compared with a set of prerecorded characteristic samples using the DTW method. Finally, the result obtained using the DTW distance can be used for sound recognition.

3.4. Execution Time Analyses for Hardware Acceleration Necessary

The DTW algorithm is easy to implement in software; however, its performance in advanced applications of dense character recognitions is limited because of its computation intensive nature with a time complexity of O ($n \times m$), as discussed in Section 2.4. Figure 6 depicts the ratios of execution time of our implemented system, which reveal that the DTW process accounts for 34% of the execution time (including signal processing and character recognition). Therefore, the execution time should be reduced for rapid recognition results for electric equipment control. Thus, hardware-accelerated DTW processing can be beneficial. Results of a study [15] concur with the paragraph: "Note that the DTW stage is the most consuming element for all three embedded systems, representing more than 96% of the total execution time."

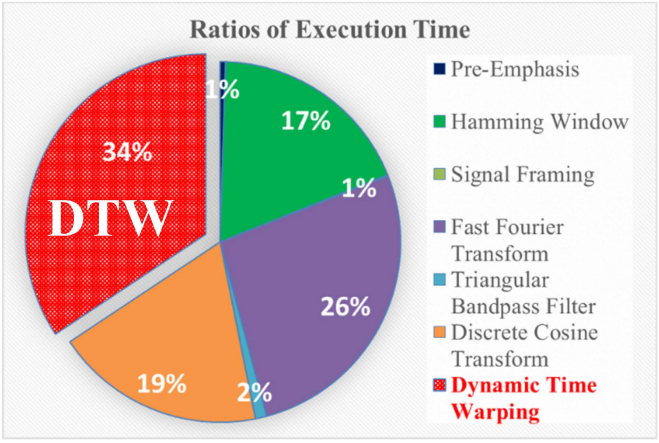


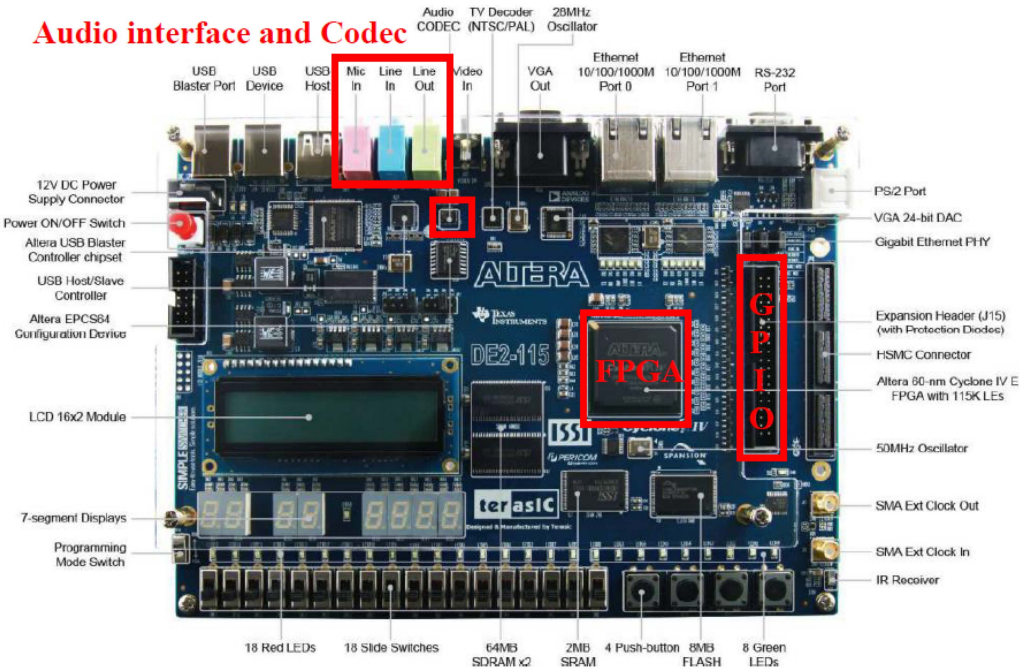
Figure 6. Ratios of execution time of signal processing and sound recognition.

3.5. Implemented System Architecture and Framework

The implemented sound recognition and equipment control system is based on a hardware and software co-design as follows.

3.5.1. System Development Board

The system was implemented on an ALTREA (Intel) FPGA embedded system [20], as shown in Figure 7. In this system, received sounds were processed using hardware and software co-design. Then, the generated control signals were transmitted to the targeted device. The system board depicted in Figure 7 was used to construct a prototype for function and performance evaluation. The implemented design can be replicated in refined system boards for cost-effective mass production.



Ref. https://www.terasic.com.tw/

Figure 7. ALTREA (Intel) DE2-115 development system board [20].

3.5.2. FPGA Design Components

The system was implemented on a FPGA, as depicted in Figure 8a. A Nios II µP [19] provided by Intel [18], was synthesized on the FPGA. Programs were written using the C language and executed over a µC/OS-II [26] operation system. Both the FPGA and µP programming files were stored in the read only memory (ROM, i.e., bootloaders [27]). The audio controller continually receives the sampled frames from the coder/decoder (CODEC) and stores them in the on-chip memory. The DTW accelerator is controlled using the µP to accomplish characteristic vector recognition. Execution times are fetched by accessing a timestamp application interface (API) recorded using a hardware timer implemented in the FPGA. The timestamp is precise because the hardware timer operates simultaneously without additional processing in the µP. Currently, electric equipment is controlled through a general input output interface (GPIO). Moreover, the TCP/IP protocol and network interface were available for further machine-to-machine (M2M) remote control.

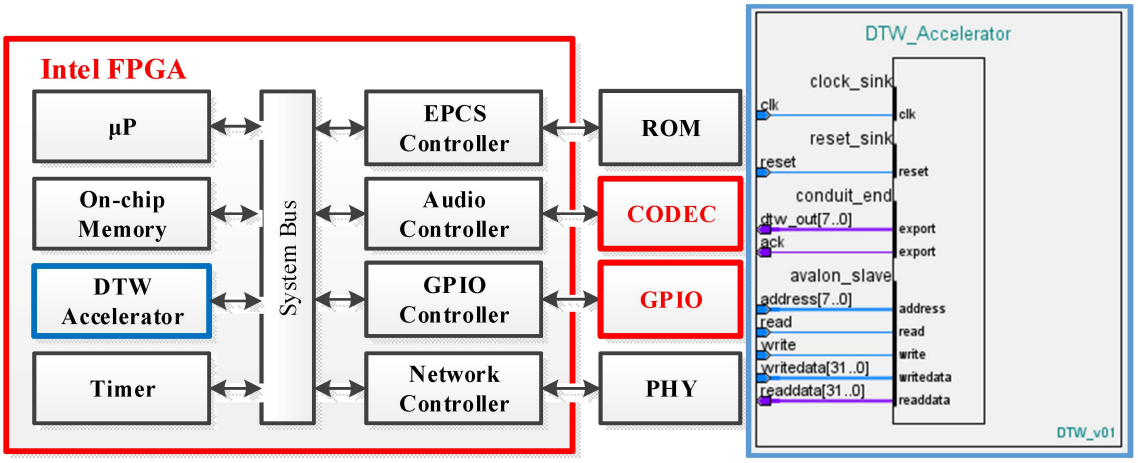
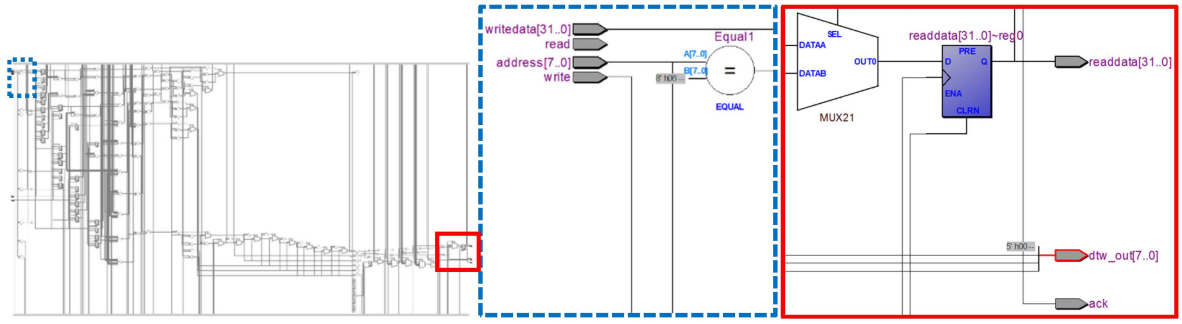


Figure 8. (**a**) Design components in the field-programmable gate array (FPGA) and (**b**) interface of the DTW accelerator module.

3.5.3. DTW Accelerator Design

As introduced in Section 2.4, the DTW accelerator module was designed using Verilog HDL and synthesized as a FPGA design component attached to the system bus, as presented in Figure 8b. Accordingly, the DTW accelerator's capacity can be relatively easy to extend such as bus width and register number for needs in the future. FPGA layout of the DTW accelerator is illustrated in Figure 9a. In more details, Figure 9b,c show the upper-left corner (marked with blue rectangle) and the bottom-right corner (marked with red rectangle) of Figure 9a, they respectively provide input and output interface signals as shown in Figure 8b. In which, except for the system bus interface (i.e., address, read/write data, read/write control), the signal "dtw_out[7..0]" provides an 8-bit DTW distance being valid when the signal "ack" is asserted (at high potential voltage), as shown in the bottom-right corner of Figure 9c.

Table 1 lists the control registers of the designed DTW accelerator module in which we designed eight 32-bit registers at address offset 0x0000 to 0x0007, thus allowing the µP to control the module by accessing the registers to offload the DTW algorithm processing. In the operation flow of DTW recognition (as shown the Figure 4), in practice, some application interfaces (APIs) have been designed to facilitate the software programming.



(**a**) (**b**) (**c**)

Figure 9. (**a**) FPGA layout, (**b**) input interface, and (**c**) output interface of the DTW accelerator module.

Table 1. Control registers of the DTW accelerator module.

Name	Offset *	Description
DtwDistance	0x0000	DTW distance b/w X and Y
MfccCharSetX0	0x0001	1st melcep data vector set X
MfccCharSetX1	0x0002	2nd melcep data vector set X
MfccCharSetX2	0x0003	3rd melcep data vector set X
MfccCharSetY0	0x0004	1st melcep data vector set Y
MfccCharSetY1	0x0005	2nd melcep data vector set Y
MfccCharSetY2	0x0006	3rd melcep data vector set Y
ExecutionStatus	0x0007	0: In progress; 1: Finish

* Address equals the offset adding base address of the DTW module.

3.5.4. Hardware-Accelerated Similarity Measurement

In our implementation, to enable that the DTW-based similarity measure applied for MFCC characteristic vectors, an API function "dtwacc_distcalc" defined in Table 2 can be applied. For example, in the process of MFCC characteristic vectors of our design, each of the 12 mel-scale cepstral coefficients was converted and stored as an 8-bit data. Accordingly, in the function "dtwacc_distcalc", first, at steps 2–4, µP writes the 96-bit MFCC character (12 × 8 bits) of the input signal into the registers "MfccCharSetX0/1/2". Second, at step 5, µP checks the number "NumMfccPreSamp" of the prerecorded samples required to be compared. Third, at steps 7–9, µP writes one of the MFCC characters of the prerecorded samples into the registers "MfccCharSetY0/1/2". Next, in the original design, µP will wait for an interrupt caused by the "ack" signal (ref. Figure 9c) being asserted; however, to avoid the overhead of executing the interrupt service routine (e.g., context switch by the applied multi-processing operation system), in our implementation, we used a C language system function call "usleep(x)" (at step 10) to make the µP sleep for x µs to wait until the completion of the DTW distance calculation. The value of x can be calculated ahead, then assigned as a number of the parameter "DtwAccExTime" defined in Table 2. Besides, using usleep() is better than delay() in the multi-processing system, since during the assigned period (i.e., x), the µP can execute another task or just sleep to save power. Last, at step 11, µP reads the calculation result from the register "DtwDistance". In the "dtwacc_distcalc", steps 7–11 execute repeatedly until all assigned prerecord samples have been compared. Related experimental results are provided in the next section.

Table 2. Notations of function dtwacc_distcalc for hardware-accelerated DTW distance calculation.

Name	Description
MfccInSign	MFCC character of input signal (size: 3 words)
NumMfccPreSamp	Number of MFCC character of prerecorded sample (noted as *i*)
MfccPreSamp*i*	MFCC characters of prerecorded samples (size: NumMfccPreSamp × 3 words)
DtwAccExTime	DTW accelerator execution time (unit: μs)
DtwDistance*i*	Calculated DTW distance (size: 1 byte)

Function Hardware-accelerated DTW Distance Calculation

```
dtwacc_distcalc(*MfccInSign, *MfccPreSamp, NumMfccPreSamp, *DtwDistance)
1:  begin
2:  dtwacc_regwr(0x01,MfccInSign0); //write MfccInSign 1st word to register 0x01
3:  dtwacc_regwr(0x02,MfccInSign1); //write MfccInSign 2nd word to register 0x02
4:  dtwacc_regwr(0x03,MfccInSign2); //write MfccInSign 3rd word to register 0x03
5:  for (i = 0; i = i + 1; i < NumMfccPreSamp) //execute comparison for the assigned number
6:    begin
7:      dtwacc_regwr(0x04, MfccPreSampi0); //write MfccPreSamp 1st word to register 0x04
8:      dtwacc_regwr(0x05, MfccPreSampi1); //write MfccPreSamp 2nd word to register 0x05
9:      dtwacc_regwr(0x06, MfccPreSampi2); //write MfccPreSamp 3rd word to register 0x06
10:     usleep(DtwAccExTime); //make μp sleep for the assigned time period
11:     DtwDistancei = dtwacc_regrd(0x00); //read the calculation result from register 0x00
12:   end
13: end
```

4. Experimental Results

In this section, simulations were run in μC/OS-II [26] with the proposed designed system platform [18–20] coding using the C language. Next, performance analyses of experimental results are discussed in following sub-sections.

4.1. Sound Recognition Enhancement Based on Sound Characters of Applications

In addition to common voice recognition, we further proposed and implemented both enhancements of function and performance, for a nonhuman voice (e.g., mewing sounds) and a biometric nonvoice sound (e.g., clapping), as described in the following sections:

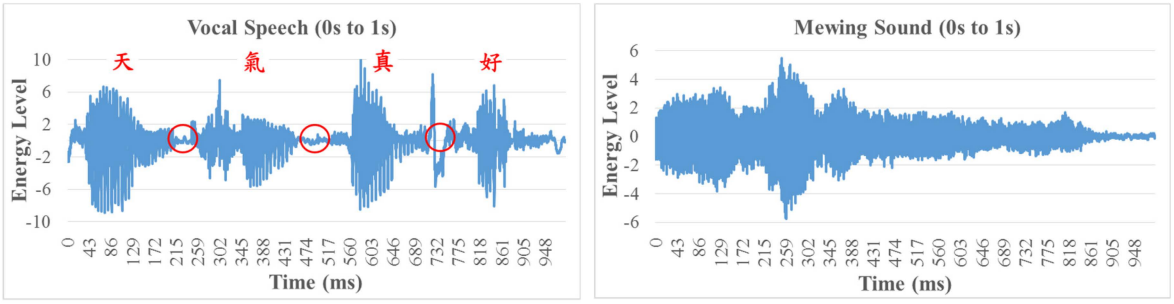
4.1.1. Recognition Performance Enhancement for Nonhuman Voice

In recent years, an increasing number of people keep pets. However, they cannot feed their pets when they leave home for work. To relieve pet owners from the task of having to feed their pets, we implemented a cat voice-controlled smart feeder. The feeding flow of our system, named the mew-mew feeder, is illustrated in Figure 10 and a demonstration video link is provided on YouTube [28].

Figure 10. Feeding flow of a mew-mew feeder.

The mewing sound recognition uses the commonly used voiceprint analyses in both time and frequency domains. Furthermore, to avoid acoustic interferences such as human voices or short-term nonvoice background noises [3,6], we proposed and implemented a sound recognition scheme that is specifically designed for identifying a mewing sound, especially when a cat is hungry. This mewing sound recognition consists of the following steps:

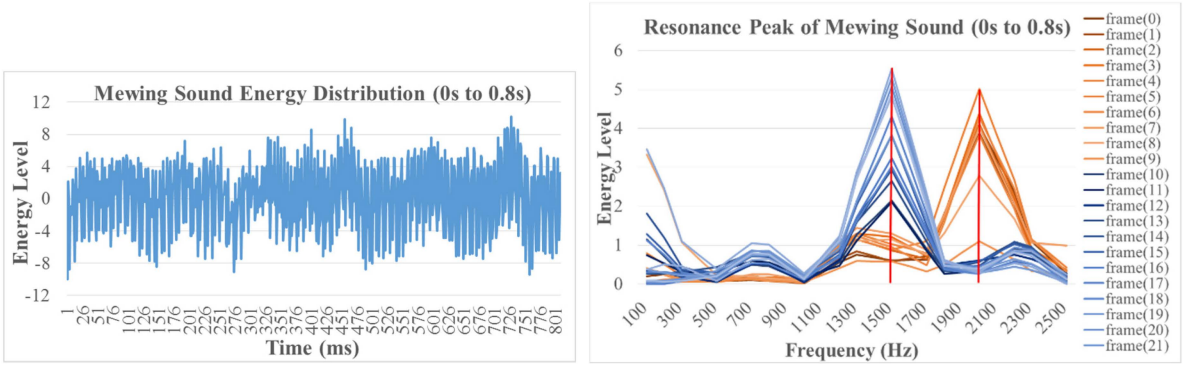
1. Confirming whether the input signal is not broken during a predetermined period: Figure 11a shows a signal variation of a voice " 天氣真好 " (i.e., the weather is fine) in Chinese. We observed that there were less energy amplitudes (marked with red circles) between each word. Figure 11b illustrates another signal recorded from a cat. The signal of the mewing sound is as continuous as singing and is different from people speaking (i.e., breaks exist between words); generally we observed that the cat cannot "speak" but is "singing". To increase the recognition correctness and reduce the processing overhead, a break within 0.8 second of the input sound is regarded as a background noise and excluded from comparison with the prerecorded samples.



(**a**) (**b**)

Figure 11. Voiceprint analyses of (**a**) vocal speech and (**b**) mewing sound in the time domain.

2. Checking whether the input signal has a valid frequency variation: We observed a frequency variation in the continuous mewing sound. Figure 12a,b presents the recorded signal in time and frequency domains, respectively. Two resonance peaks were observed at approximately 1.5 and 2 kHz. The resonance peak moved from 2 to 1.5 kHz progressively, as demonstrated in a video provided by a YouTube link [29]. To increase the recognition correctness and reduce the processing overhead in our design, if the frequency variation degree is less than 300 Hz, the input signal is regarded as a non-mewing sound and excluded from comparison with the prerecorded samples.



(**a**) (**b**)

Figure 12. Voiceprint analyses of mewing sound in the (**a**) time domain and (**b**) frequency domain.

3. Determining whether the characteristics of the input signal are similar to one of prerecorded samples: By using the aforementioned signal processing steps, the system can accurately determine whether the input signal is the mewing sound of a hungry cat. Thus, the feeder can supply cookies to the cat, as depicted in Figure 13. A precise recognition result is a crucial design requirement for the feeder to avoid overfeeding or deprivation in facilitating the good health of the cat.

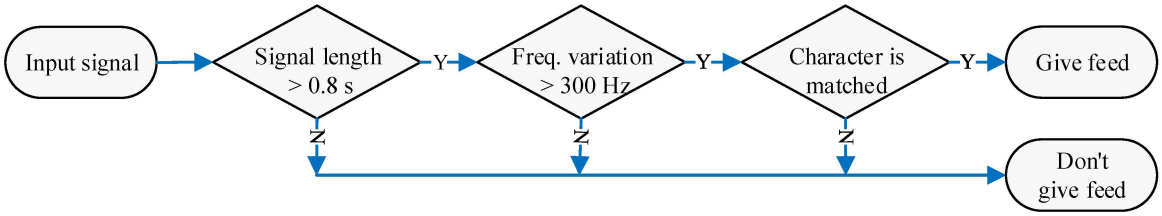


Figure 13. Sound recognition flow of the designed mew-mew feeder.

4.1.2. Recognition Performance Enhancement for Nonvoice Sound

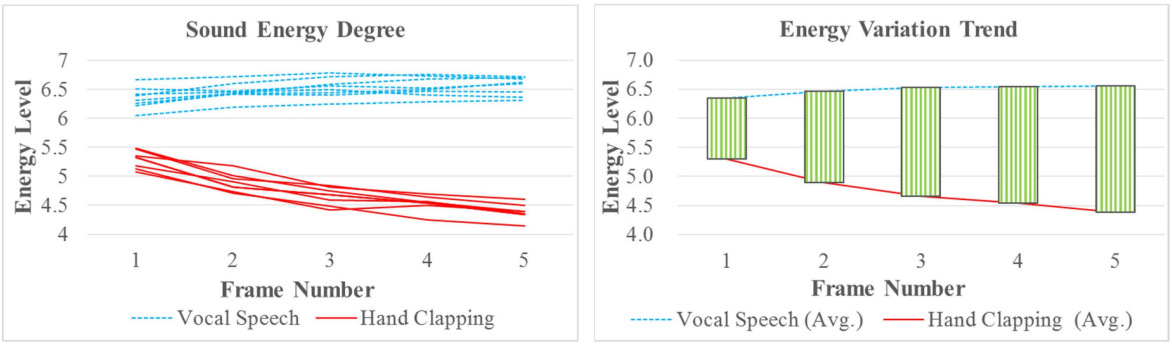
Clapping could be the most convenient method of electric equipment control for people. For example, to play music, in general, we must to push a button either on the control panel or on a remote controller. Therefore, people have to move toward the music player or to search for the (possibly missing) remote controller. Sound control can be used rather than physical touch. That is, when users want to listen music, they could just clap. The music player could identify the sound and play music. In practice, for electric equipment control using nonvoice sounds such as clapping, the most and common acoustic interferences in the environmental background are vocals from people, such as speaking, singing, laughing, and chatting. In addition to the techniques commonly used for distinguishing frequency distribution features for voice recognition, one of the major contributions of this paper is that the proposed STP mechanism can be applied to further increase short-term nonvoice sound recognition accuracy. By analyzing the difference in sound characters between people speaking and clapping, we introduced the following improvements:

- Frequency range: The frequency range of clapping is much lower than that of vocal speech. Accordingly, in experiments, we changed the commonly used mel-frequency filtering range (0–8 kHz) to 0–500 Hz, which led to the error rate of sound recognition of vocal speech being increased because the range of some voice characters is more than 500 Hz. On the other hand, this reduced the error rate of sound recognition of clapping because of the frequency range of clapping sound characters is less than 500 Hz, and this resulted in a higher quality MFCC vector extraction, which contained more and dense clapping sound characters from the used triangular bandpass filters.
- Sample length: In practice, clapping is continuous and short, whereas vocal speech is discontinuous and long. We reduced the sound sampling length from the original 1 s to 0.4 s. In experiments, the error rate of sound recognition of vocal speech was accordingly increased because of the voice character recognition of fewer characters. By contrast, this change did not affect the error rate of sound recognition of clapping. Signal data processing required for sound recognition was considerably reduced to 40% of the original requirement and therefore resulted in a faster response for the recognition result.
- Energy variation: We observed a considerable energy difference between vocal speech and clapping, as depicted in Figure 14a. This experiment extracted the energy levels (ELs) from the generated MFCC characteristic vectors of five frames (from frame #1 to #5) captured in the range 0 s to 0.4 s. We observed that the ELs slightly increased in vocal speech, whereas the ELs considerably degraded in clapping. This is caused by the short-term duration mentioned previously. Figure 14b depicts the average ELs of the eight signals of vocal speech and the eight signals of clapping in Figure 14a. Thus, EL gaps between vocal speech and hand clapping increased from 1.06 (6.35–5.29)

of frame #1, to 2.17 (6.56–4.39) of frame #5. Using measurements of field quantities, we defined such energy level distribution as an energy variation trend (EVT) value between sound frame #*i* to sound frame #*j*:

$$EVT(i, j) = 20 \times \log_{10}\left(\frac{EL(j)}{EL(i)}\right)\text{dB} \quad (8)$$

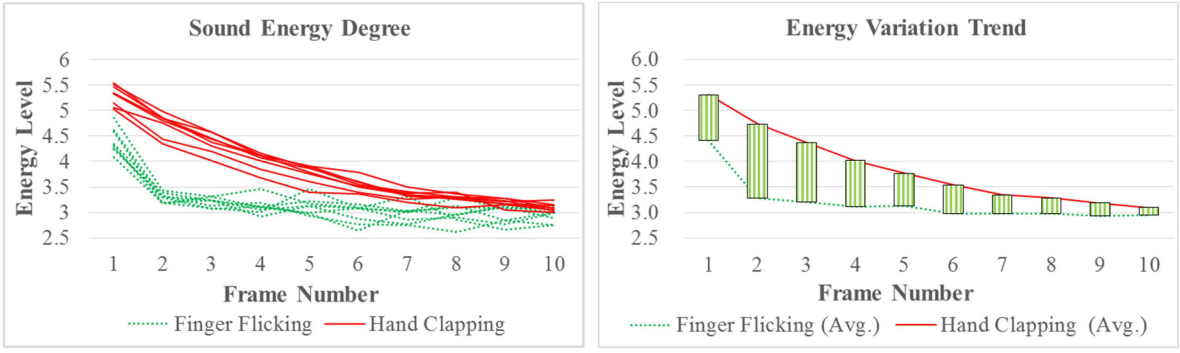
where $EL(j)$ and $EL(i)$ are the energy levels of frame j and frame i, respectively. $EVT(i, j)$ levels indicate a measurement value to express the ratio of $EL(j)$ to $EL(i)$ on a logarithmic scale. For example, in Figure 14b, the $EVT(1, 5)$ of vocal speech is a positive value of 0.28 dB [$20 \times \log_{10}$ (6.56/6.35) dB], and the $EVT(1, 5)$ of hand clapping is a negative value of −1.63 dB [$20 \times \log_{10}$ (4.39/5.29) dB]. We used this EVT feature to increase short-term, nonvoice sound recognition accuracy. A demonstration video link that demonstrates a hand clapping-controlled music player is provided on YouTube [30].



(**a**) (**b**)

Figure 14. Sound energy analyses for vocal speech and hand clapping in (**a**) energy degree and (**b**) variation trend.

To provide more EVT experiments, we compared hand clapping with another nonvoice sound of finger flicking, as depicted in Figure 15a. This experiment showed the ELs of ten frames (from frame #1 to #10) in the range 0 s to 0.8 s. Figure 15b depicts the average ELs of the eight signals of finger flicking and the eight signals of clapping in Figure 15a, in which we observed that the ELs of hand clapping were continuously degrading, as well as that in Figure 14a; by contrast, all of the ELs of finger flicking have a similar distinct drop from frame #1 to #2 in Figure 15a. Besides, from frame #3 to #10 in Figure 15a, the ELs of finger flicking vibrated between level 3.5 and 2.5. Accordingly, as Figure 15b shows, the average ELs of finger flicking from frame #3 to #10 are 3.20 to 2.95, which are approximately 2.96 of the average EL of the environment background noises in our experiments. By analyses, as Figure 15b shows, the maximal EL of finger flicking (4.41 at frame #1) is less than that of hand clapping (5.31 at frame #1); additionally, the $EVT(1, 2)$ of finger flicking is –2.55 dB [$20 \times \log_{10}$ (3.29/4.41) dB], which means that the ELs degraded more than that of hand clapping (−0.99 dB [$20 \times \log_{10}$ (4.73/5.31) dB]). Consequently, from frame #3 to #10 in Figure 15a, the ELs of finger flicking are more susceptible to the background noises than that of hand clapping in the experiments.



(**a**) (**b**)

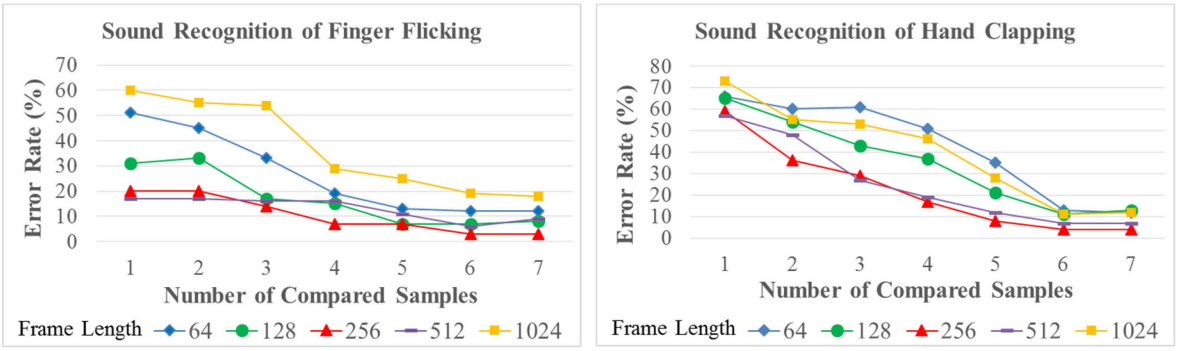
Figure 15. Sound energy analyses for finger flicking and hand clapping in (**a**) energy degree and (**b**) variation trend.

4.2. Sound Recognition Enhancement by Refinements of Processing Resources

Sound recognition for the targeted application can be further increased by refining processing resources, as described in the following sections.

4.2.1. Recognition Performance Enhancement by Adjusting Length of Sampling Frame

In sound signal framing, the selection of frame length (in unit of sample point) affects the result of further signal analyses. Assuming that the applied sample rate and time are fixed, a long frame length results in a small number of captured frames for the identical sample time period. In general, a longer frame could contain numerous sound characters, increasing the difficulty in observing and distinguishing specific characters of the captured frame. By contrast, when the length of the sound frame is short, the number of sample points for analysis is fewer; thus, the result is susceptible to a sudden change and is less representative of the sound signal. Therefore, the length of the sound frame is preferably determined according to the specific application, especially for nonvoice sounds. As depicted in Figure 16, for finger flicking and hand clapping, a frame length of 256 (red line marked with triangle) exhibits the best sound recognition performance (i.e., less error rates) in most testing cases. In addition, a long frame length of 1024 (yellow line marked with square) and a shorter frame length of 64 (blue line marked with diamond) cause higher error rates in sound recognition.



(**a**) (**b**)

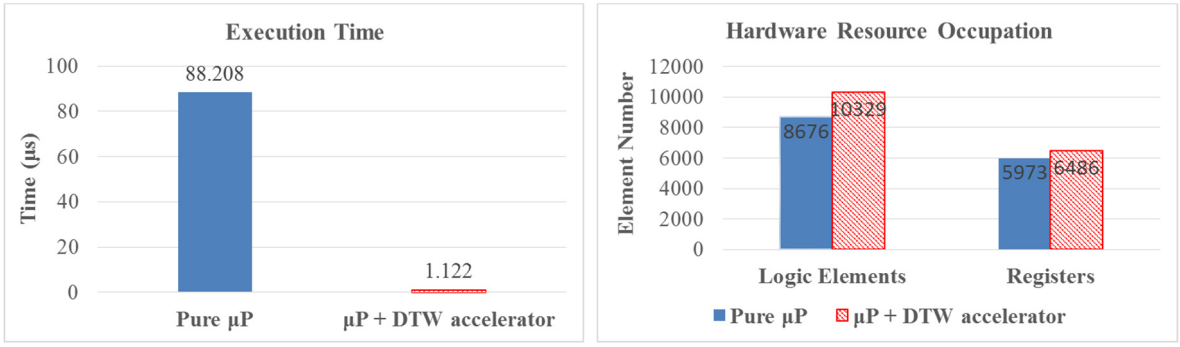
Figure 16. Sound recognition with different sound frame lengths and prerecorded sample numbers for distinguishing (**a**) finger flicking and (**b**) hand clapping.

4.2.2. Recognition Performance Enhancement by Increasing the Number of Compared Samples

We can enhance recognition accuracy for hardware-accelerated DTW processing by comparing multiple samples in a relatively short time period. That is, rather than using just one prerecorded sample to compare with the input signal, multiple samples were applied as essential targets to be compared. In experiments, we attempted the recognition of two similar biometric nonvoice sounds made by a user. The first sound was finger flicking, and the second was hand clapping. In practice, sounds of finger flicking and hand clapping made by the same user can be more easily distinguished; however, the recognition of sounds of finger flicking (or hand clapping) made by different users can be more difficult when just a sound of finger flicking (or hand clapping) has been recorded. As there is no gold standard for a biometric nonvoice sound such as finger flicking, finger flicking sounds made by a user or different users can differ considerably from case to case. Accordingly, we prerecorded seven different sounds of finger flicking made by the same user in the experiment as shown in Figure 16a, in which the number on the horizontal axis is the quantity of compared samples, and when the input signal matches any one of the prerecorded samples, the sound recognition is regarded as correct. Figure 16a depicts that the error rate of finger flicking sound recognition for a frame length of 256 (red line marked with triangle) can be reduced from 20% (one sample) to 3% (seven samples) using more samples. As depicted in Figure 16b, for recognizing the sound of hand clapping, the error rate can be considerably reduced from 59% (one sample) to 4% (seven samples) because the difference between the sounds of hand clapping made by users is more than that of finger flicking in the test cases of our experiments. In practice, sounds of finger flicking made by humans are all possible targets to be compared, which leads to a big challenge in implementation; a large number of prerecorded samples is better. Consequently, the time required for comparisons between sound characters increases; on the contrary, for electric equipment control, the response time for sound recognition should be fast. Therefore, hardware-accelerated DTW processing can satisfy this condition (described in the following sections).

4.2.3. Recognition Performance Enhancement Using a Hardware Processing Accelerator

To achieve a quick response time, we implemented a DTW accelerator, as detailed in Sections 2.4 and 3.5. In this section, real performances of the designed DTW accelerator were measured using the total execution times for operations by using "pure μP" and "μP with DTW accelerator", which were executed on the applied FPGA development board (Figure 7). C language software program coding was run on an Eclipse IDE for C/C++ developers kit provided by Intel [18,19]. As Figure 17a illustrates, it takes 88.208 μs to process a DTW frame recognition using a conventional μP pure software approach. In the proposed DTW accelerator, the execution time was considerably reduced to 1.122 μs, which is merely 1.27% (1.122/88.208 × 100%) of the pure software approach. This can considerably benefit big-data analyses in STP, as discussed in Section 4.2.2. Additionally, we analyzed the hardware overhead of the implemented FPGA system. As Figure 17b shows, the implemented hardware DTW accelerator required an additional 19.05% of logic elements and 8.59% of registers compared with that of the primitive design excluding the DTW accelerator (Figure 8a).



(**a**) (**b**)

Figure 17. DTW processing performance comparisons of (**a**) execution time and (**b**) hardware resource occupation between "pure μP" and "μP with the DTW accelerator".

5. Discussion

For short-term sound recognition processing, we list the major contributions of this paper:

1. Verifies that the software-implemented DTW processing is computation intensive and can be a performance bottleneck for embedded systems, especially for further big-data analyses;
2. Provides the FPGA-based system platform to validate that the proposed hardware and software co-design framework can increase the recognition accuracy by comparing additional samples in a relatively short time period using the designed DTW accelerator;
3. Applies the proposed method and implemented system to a nonhuman voice (mewing sounds) and nonvoice sounds (hand clapping) recognition where only limited research has been conducted; and
4. Implements two real products (the mew-mew feeder and a music box) with demonstration video links on YouTube.

6. Conclusions

To ensure short-term biometric nonvoice sound recognition for electric equipment control, we implemented a FPGA-based, STP sound control system, which can rapidly process specific sound characters with a high recognition correction. The experiments on real products demonstrated that the proposed method can not only detect nonvoice sounds, but also enhance performance. Currently, the proposed sound control method is realized in the applied evaluation board. Next, the design can be optimized as a small embedded system to reduce manufacturing cost and power consumption, allowing for further mass production. Besides, the DTW accelerator can be refined as an intellectual property merging into a system-on-chip design for developments of other applications that require rapidly processing pattern recognition with big data. These will be our future work.

Author Contributions: W.-C.T. designed the framework and wrote the manuscript. Y.-J.S. implement the design of signal processing. N.-T.H. analyzed the performance and realized the productions.

Funding: This work is partially supported by the MOST, ROC, under grant number of MOST 106-2221-E-324-007-MY2, MOST 108-2221-E-324-011, and Chaoyang University of Technology (CYUT) and Higher Education Sprout Project, Ministry of Education, Taiwan, under the project: "The R&D and the cultivation of talent for Health-Enhancement Products".

Conflicts of Interest: The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Wan, J.; Tang, S.; Shu, Z.; Li, D.; Wang, S.; Imran, M.; Vasilakos, A.V. Software-defined industrial Internet of Things in the context of industry 4.0. *IEEE Sens. J.* **2016**, *16*, 7373–7380. [CrossRef]

2. Caranica, A.; Cucu, H.; Burileanu, C. Speech recognition results for voice-controlled assistive applications. In Proceedings of the International Conference on Speech Technology and Human-Computer Dialogue, Bucharest, Romania, 6–9 July 2017; pp. 1–8.
3. Kil, R.; Kim, Y. Zero-crossing-based speech segregation and recognition for humanoid robots. *IEEE Trans. Consum. Electr.* **2009**, *55*, 2341–2348.
4. Zinchenko, K.; Wu, C.Y.; Song, K.T. A study on speech recognition control for a surgical robot. *IEEE Trans. Ind. Inform.* **2017**, *13*, 607–615. [CrossRef]
5. Galka, J.; Masior, M.; Salasa, M. Voice authentication embedded solution for secured access control. *IEEE Trans. Consum. Electr.* **2014**, *60*, 653–661. [CrossRef]
6. Park, J.S.; Jang, G.J.; Kim, J.H.; Kim, S.H. Acoustic interference cancellation for a voice-driven interface in smart TVs. *IEEE Trans. Consum. Electr.* **2013**, *1*, 244–249. [CrossRef]
7. Ding, I.J.; Lin, S.K. Performance improvement of Kinect software development kit–constructed speech recognition using a client–server sensor fusion strategy for smart human–computer interface control applications. *IEEE Access* **2017**, *5*, 4154–4162. [CrossRef]
8. Tsai, W.C.; Lian, Y.R.; Hsu, S.H.; Zheng, Q.X.; Su, Y.C.; Chen, J.X. An implementation of voice recognition and control system for electric equipment. In Proceedings of the International Symposium on Computer, Consumer and Control, Taichung, Taiwan, 6–8 December 2018; pp. 356–359.
9. Vergin, R.; Shaughnessy, D.O.; Farhat, A. Generalized mel frequency coefficients for large vocabulary speaker independent continuous speech recognition. *IEEE Trans. Acoustic Speech Signal Process.* **1999**, *7*, 525–532. [CrossRef]
10. Muda, L.; Begam, M.; Elamvazuthi, I. Coice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques. *J. Comput.* **2010**, *2*, 138–143.
11. Bellman, R.; Kalaba, R. On adaptive control processes. *IRE Trans. Autom. Control* **1959**, *4*, 1–9. [CrossRef]
12. Berndt, D.J.; Clifford, J. Using dynamic time warping to find patterns in time series. In Proceedings of the AAAI-94 Workshop on Knowledge Discovery in Databases, Seattle, WA, USA, 31 July–1 August 1994; pp. 359–370.
13. Pandey, D.; Singh, K.K. Implementation of DTW algorithm for voice recognition using VHDL. In Proceedings of the International Conference on Inventive Systems and Control, Coimbatore, India, 19–20 January 2017; pp. 1–4.
14. Rath, T.M.; Manmatha, R. Word image matching using dynamic time warping. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Madison, WI, USA, 18–20 June 2003; pp. 521–527.
15. Mariano, L.G.; Rafael, R.L.; Oscar, M.H.; Enrique, C.N. Embedded system for biometric online signature verification. *IEEE Trans. Ind. Inform.* **2014**, *10*, 491–501.
16. Kim, S.H.; Lee, H.S.; Ko, H.J.; Jeong, S.H.; Byun, H.W.; Oh, K.J. Pattern matching trading system based on the dynamic time warping algorithm. *Sustainability* **2018**, *10*, 4641. [CrossRef]
17. Radoi, A.; Burileanu, C. Retrieval of similar evolution patterns from satellite image time series. *Appl. Sci.* **2018**, *8*, 2435. [CrossRef]
18. Intel FPGAs. Intel Corporation. Available online: https://www.intel.com/content/www/us/en/products/programmable/fpga.html (accessed on 12 July 2019).
19. NIOS II Processor: The World's Most Versatile Embedded Processor. Intel Corporation. Available online: https://www.intel.com/content/www/us/en/products/programmable/processor/nios-ii.html (accessed on 12 July 2019).
20. Altera DE2-115 Development and Education Board. Terasic Inc. Available online: https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=139&No=502 (accessed on 12 July 2019).
21. Nandhini, S.; Shenbagavalli, A. Voiced/unvoiced detection using short term processing. *Int. J. Comput. Appl.* **2014**, *2*, 39–43.
22. Sart, D.; Mueen, A.; Najjar, W.; Keogh, E.; Niennattrakul, V. Accelerating dynamic time warping subsequence search with GPUs and FPGAs. In Proceedings of the International Conference on Data Mining, Sydney, NSW, Australia, 13–17 December 2010; pp. 1001–1006.
23. Tai, J.S.; Li, K.F.; Elmiligi, H. Dynamic time warping algorithm: A hardware realization in VHDL. In Proceedings of the International Conference on IT Convergence and Security, Macao, China, 6–18 December 2013; pp. 1–4.

24. Xilinx FPGA. Xilinx Inc. Available online: https://www.xilinx.com/ (accessed on 12 July 2019).
25. WM8731 Codec with Headphone Driver. Cirrus Logic, Inc. Available online: https://www.cirrus.com/products/wm8731/ (accessed on 12 July 2019).
26. µC/OS, RTOS and Stacks. Silicon Labs. Available online: https://www.micrium.com/rtos/kernels/ (accessed on 12 July 2019).
27. EPCS Bootloaders. Intel Corporation. Available online: https://fpgawiki.intel.com/wiki/EPCS_bootloaders (accessed on 12 July 2019).
28. Demo of Mew Mew Feeder. Available online: https://youtu.be/8ol5aJ79Xgs (accessed on 12 July 2019).
29. Frequency Variation of Mewing Sound (0s to 0.8s). Available online: https://youtu.be/D27AVfRSj6M (accessed on 5 August 2019).
30. Music Box Controlled by Hand Clapping. Available online: https://youtu.be/kShZE4_udDU (accessed on 12 July 2019).