

PAPER • OPEN ACCESS

Voice-Based Malay Commands Recognition by Using Audio Fingerprint Method for Smart House Applications

To cite this article: M Abu *et al* 2020 *IOP Conf. Ser.: Mater. Sci. Eng.* **767** 012053

View the [article online](#) for updates and enhancements.

You may also like

- [Bridging the digital divide: a survey on development of spoken language technology, a northeast India perspective](#)
Deepshikha Gogoi and Yang Saring
- [Arduino based voice controlled wheelchair](#)
Tan Kian Hou, Yagasena and Chelladurai
- [Array of vector vortex beam for audio security](#)
Allarakha Shikder, Sudheesh K Rajput, Naveen K Nishchal *et al.*



The Electrochemical Society
Advancing solid state & electrochemical science & technology



249th
ECS Meeting
May 24-28, 2026
Seattle, WA, US
Washington State
Convention Center

Spotlight Your Science

**Submission deadline:
December 5, 2025**

SUBMIT YOUR ABSTRACT

Voice-Based Malay Commands Recognition by Using Audio Fingerprint Method for Smart House Applications

M Abu¹, A Amir¹, N A H Zahri¹, R Ngadiran¹

School of Computer and Communication Engineering
Universiti Malaysia Perlis
Perlis, Malaysia

amizaamir@unimap.edu.my

Abstract. Voice-based command recognition is commonly used in security systems, phones, household appliances and hardware designed for handicapped people. Most of the current research study the voice command recognition for the smart home in English. Lack of study for voice command recognition in Malay makes it challenging to apply the voice command services for the smart home in Malaysia. Also, voice recognition is a non-trivial task in natural language processing. This project is to identify the command used for smart home appliances using Malay and design the algorithm for this system. Then, the proposed algorithm will be deployed on a Raspberry Pi to see the performance of Malay command in accuracy and the suitability of the algorithm to be deployed on low cost embedded devices. Light, fan, and television had been chosen as electrical appliances to build the command. An algorithm that previously used to recognize songs, the robust quad algorithm, is used in this project for voice command recognition. The proposed method has two main processes, known as feature extraction and voice recognition. In the feature extraction process, the audio fingerprint will extract data from the command spectral peak. For voice recognition, audio fingerprint matching will be used to analyze the audio commands. The outcome of this project is when the voice command is given by the user by activate or deactivate the target home appliance. The second outcome is the background noise that affects the system is reduced by using robust quad algorithm and increase the accuracy of the system. The results of this project have shown that the proposed algorithm is suitable to be implemented on a Raspberry Pi and achieve a high recognition rate with 87%. In the presence of noise with 15 dB, the proposed algorithm can maintain the high recognition rate with 82%.

Keywords—peak detection, audio fingerprint method, noise resistance

1. Introduction

Human voice had been used in many technologies for example voice dialling, data entry, ID verification, detect identity in the crowd, smart house, and security [1], [2]. Voice recognition for controlling house appliances has been widely studied, but most of the available applications are developed in English. This limits the applicability of these applications for non-English speakers. Hence, the voice recognition system based on other languages is vital to ensure the scalability of the application is widely used in different communities. Recently, research in voice recognition using different languages such as Japanese and Thai have also attracted attention among research communities[3].

XXX-X-XXXX-XXXX-X/XX/\$XX.00 ©20XX IEEE



Content from this work may be used under the terms of the [Creative Commons Attribution 3.0 licence](https://creativecommons.org/licenses/by/3.0/). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

This paper proposes the Malay language as a command since it can help those critically ill, aged and disabled people in Malaysia who cannot speak English. The Malay-based voice recognition is significant to ensure more users from Malaysia can use the voice-based control smart home systems. Lately, most of the research and project for voice recognition are based on English speech recognition, and in comparison, other language speech recognition is still limited [4]–[6]

Based on Aripin and Othman (2014) research, voice-based command recognition is susceptible to detect unnecessary noise like screaming, dog barking and crying sound[2]. Leechor *et al.* (2010) stated that voice recognition accuracy and response speed could be affected by environment, because when the commands are given in a noisy environment, low recognition rate will occur, and when the command is given in a quiet environment, a high recognition rate will occur. Therefore, using a suitable microphone and a noise filter or noise handling algorithm can improve the accuracy rate for the proposed voice commands recognition [7], [8].

Oulkar *et al.* (2017) [9], Gyulyustan and Enkov (2017) [10], and Upadhye and Khan (2017) [11] choose Raspberry Pi to implement their system on board. The authors built their system in English and tested the performance of their systems in Raspberry Pi. Oulkar et al. stated that by using Raspberry Pi and connecting it with a suitable microphone, the accuracy could achieve 99% based on the ideal environment. Abd Manan (2006) developed his Malay-based voice commands in MATLAB but not implement it on board. The implementation of a Malay-based voice command system on a Raspberry Pi is desirable since to apply such system in a smart home system requires it to be implemented on an embedded device [9]–[11].

There are voice command recognition for smart house that have been proposed by other researchers. For example, Aripin and Othman (2015) chose fan and light for their “on” and “off” command [2]. Samah and Osman (2015), stated that light, fan, air conditioner, television, and radio are suitable for “on” and “off” command [12]. These commands built by the researchers is in English. Therefore, in this project, three devices —fan, light, and television— are chosen to build Malay commands for smart houses application.

2. Literature Review

Hidden Markov models (HMM) used a statistical approach to characterize the spectral properties frames of speech in the context of statistical methods. The advantage of HMM is provides a natural and highly reliable way of recognizing speech for a wide variety of applications as a stochastic modelling tool. HMM can integrates well into systems incorporating information for both acoustics and syntax [3], [13].

Artificial Neural Networks (ANN) often used as a classifier for tasks in speech recognition, and have several advantages over parametric classifiers. However, there are disadvantages in terms of the amount of training data required and length of training time. Second disadvantage, the length of time required to train the networks can present problems, particularly when investigating a variety of feature sets to represent speech data. ANN has three architecture, Feedforward Perceptrons Trained with BackPropagation, Radial Basis Function Networks and Learning Vector Quantization. Based on three architectures, Feedforward Perceptrons Trained with BackPropagation is the most popular model [3].

Support Vector Machine (SVM) have the ability to transform data to high dimensional space implicitly and to construct a linear binary classifier in this high dimensional space without perform any computations in the high dimensional space. The disadvantages of using SVM is the number of training samples it can deal with is limited. Second disadvantages, the formulation of SVM can only work with vectors of fixed dimension input and it does not give a reliable measure of the probability for correctness during classification. This can affect the recognition performance because, without a concrete value of probability, some algorithms cannot carry out to search for the most probable sequence of recognition units [5].

Differently, this research investigates the potential of an audio fingerprint approach, the Robust Quad-Based Audio Fingerprint Method [14] for voice command recognition. Audio fingerprint algorithm is usually used for music recognition. The advantages of using this algorithm are noise and distortion resistant. This algorithm is also able to recognize a short audio sample of music that had been broadcast that means timing differences between train samples and test samples will not affect. The time to recognize a large database is faster compared to another recognition method [14]. The robustness to noise may not be the main concern for this algorithm to recognize audio material that was modified in tempo or pitch. The other advantages of using audio fingerprint method is it can efficiently identify audio in a large database. It is also robust to noise and audio quality degradation, as well as to severe distortions of speed, tempo, and frequency [15].

3. Methodology

This project studies the performance of voice-based Malay command recognition in the normal environment by using Robust Quad-Based Audio Fingerprint Method. Firstly, All the voice command will be recorded using a mobile phone. Each speaker will record their voice three times for each command. The command will be divided into 80:20 ratio for the training and testing datasets. The recorded voice command will undergo feature extraction to extract the audio fingerprint. The feature extraction will be done for both training and testing datasets. Then, the extracted audio fingerprint for the testing dataset will be matched together with training dataset to get the result. The performances of the system will be tested on PC; then it will be deployed in Raspberry Pi. The time-efficiency of Raspberry Pi and computer will be compared.

3.1. Survey to Identify Common Malay Voice Commands for Smart Home

Firstly, we selected three electrical appliances: fan, light, and television. In previous studies[5], the usual commands are 'Buka Kipas', 'Tutup Kipas', 'Buka Lampu', and 'Tutup Lampu'. In addition to these commands, this paper investigates the common Malay commands to turn on and turn off a television and control television channel. To decide the usual commands for these tasks, a survey has been conducted by asking the respondent what are the common Malay commands used to control televisions. The survey questions were generated by using Google form and shared with 55 respondents through WhatsApp messaging system and Facebook.

For turn on and turn off the television, from 55 respondents 45 (85%) respondents chose 'Buka TV' and 'Tutup TV', respectively to turn on and to turn off the television. Only 10 (18%) of the respondents selected 'Buka Televisyen' and 'Tutup Televisyen', which shows that these commands are not commonly used for the task to turn on and turn off the television. For control television channel, from 55 respondents 48 (87%) chose 'Siaran 1', 'Siaran 2', and 'Siaran 3' to control the television channel. Only 4 (7%) of the respondents chose 'Siaran Naik' and 'Siaran Turun' and the other 3 (5%) of the respondents chose others as voice command that not commonly used for controlling television channel. Others are the voice command suggested by the respondent. Hence, 'Buka TV', 'Tutup TV', 'Siaran 1', 'Siaran 2', and 'Siaran 3' is used for data collection. Hence, nine commands are selected to control for the electrical appliances.

3.2. Audio Data Collection

For audio data collection, 24 speakers from different backgrounds and genders were asked to record their voices in the normal environment, for example, a quiet room with fan and light are activated. From the 24 speakers, 12 males and 12 females with age ranging between 20 to 30 years old recorded their voice based on the command given in Table 1. Each speaker recorded their voice three times for each command, so there are 648 samples overall. The samples were divided into 80:20 ratio for the training and testing datasets. The distance between the speaker and the phone is 0.5 meters, based on a previous study [16]. The time range for the speaker to record their voice depends on their timing. The voice

command sample rate is 8 kHz because this sample rate is adequate for human speech. Recording channel for the device is mono. Figure 1 shows an example of the voice command signal.

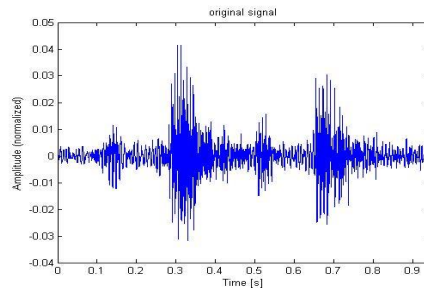


Figure 1. An example of a voice command signal

3.3. Software Design

Based on the flowchart as shown in Figure 2, in feature extraction, voice command is converted into a spectrogram, and the local maxima (spectral peak) will be searched to build quads. Two types of filters are applied in robust quad-based audio to obtain the actual spectral peak in the spectrogram. Quad is a continuous geometric hash representation of quadruples of a point. The quad will be converted into the hash and stored as an audio fingerprint. Then, the audio fingerprint for training and the testing dataset is used during classification model for decision-making.

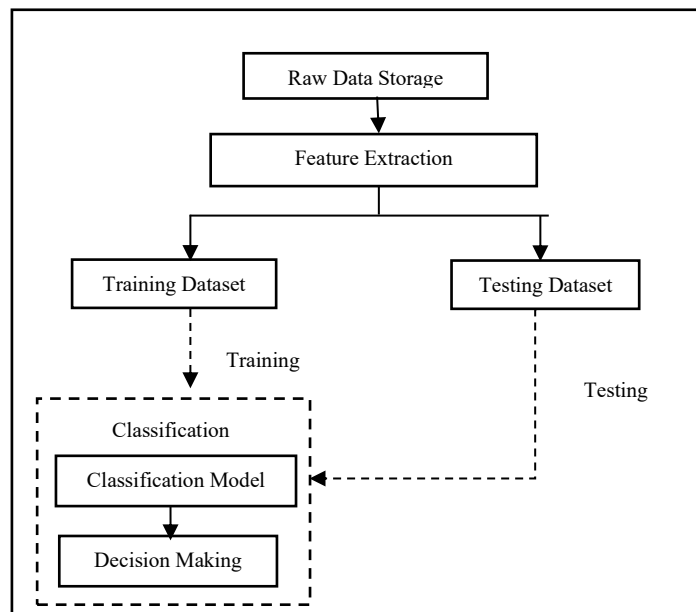


Figure 2. Flowchart of voice-based Malay command recognition for smart house applications

The detail to obtain the hashes for the matching process is explained in Robust Quad Based Audio. Figure 3 shows the training process and Figure 4 shows the classification process.

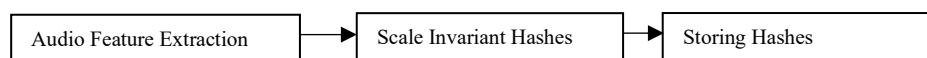


Figure 3. The training process

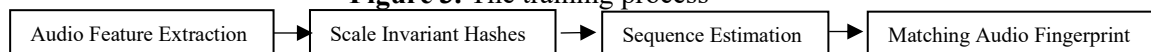


Figure 4. The classification process

Audio Feature Extraction

To extract audio fingerprint, voice command quads need to be constructed as shown in Figure 5. This algorithm works on translation and scale invariant geometric hashes by combinations of spectral peaks. To produce quad the command signal will be converted into spectrogram in two-dimensional time-frequency. Then, the local maxima of voice command need to be found and will be shown in the scatter plot. The local maxima are represented as time for the x-axis and the peak of frequency for the y-axis in spectrogram space. The spectrogram is using a Hann window of size 1024 samples and hops size of 32 samples. The advantage of using windows is that the frequency can be seen as a narrow peak in the spectrogram [18]. To find the quad, max filter and min filter will be used. The local maxima that are detected by both filter will be removed from the spectrogram.

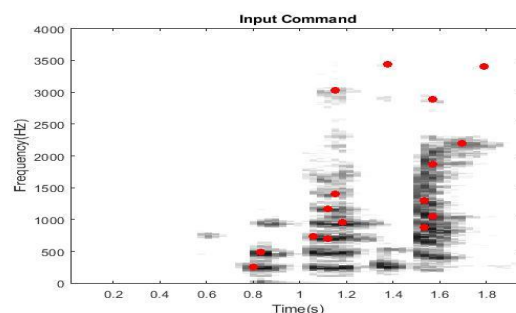


Figure 5. Examples of the quad for voice command.

Each command shows a different quad pattern because voice command is extracted based on frequency and time. Pre-processing is removed in this algorithm because the spectral peak of the speaker is higher than the background noise and it is neutral for the speaker to speak louder than the noise. In this paper, the noise is stationary, but the timing of the command is nonstationary so the system will start to detect the spectral peak as the signal is fluctuating. The offset time for this system to detect the peak spectral is within 0.3 seconds because the command time is too short, and the system will give output even though only part of the command is given.

Scale-Invariant Hashes

Each command will produce different hash because of frequency and the tempo of the speakers' speak. The quads will be converted into Translation and Scales Invariant after it has been extracted from each command. To make sure the test samples quad match the train samples quad, the quad needs to be summarized by converting it into hashes. To convert quads into a hash, the property of rotational invariant and the resulting mode is released to compute translation and scale-invariant to the hash.

Storing Hashes

The hashes will be stored in four data structures *peakfile*, *refrecords*, *fidindex* and *searchtree*. *Peakfile* contains the continuous of two-dimensional coordinates of spectral peaks that obtained from train samples. *Refrecords* stores quads and quad hashes along with the file-ID of all train samples in the reference collection. A quad record consist of spectral peak and where denotes the height and width of the quad in spectrogram space. *Fidindex* will map each voice command file for train samples to a unique file-ID. The number of extracted peaks and quads also is stored by *fidindex* along with other metadata. Given a specific voice command file-ID, the *fidindex* is used to find the corresponding record range in the *peakfile*. *Searchtree* is used to perform an efficient fixed-radius near neighbour searches of quad hashes. Memory-bounded tree construction is simple to establish by using tree variant.

Audio Fingerprint Classification

There are three stages to matching test sample with train samples: the first stage performs the selection of match candidates, followed by a filtering stage to remove the false positive candidate. Second is the match sequence estimation stage, using the result from the first stage, the sequences are searched within matched candidates. The last stage is the verification step. This step is to maintain high identification precision on extensive database collections, especially if the voice command is repeated many times.

Match Candidate Selection and Filtering

A fixed-radius near neighbour search in the *searchtree* is performed for each test sample quad hash. This lookup returns a set of raw match candidates that consists of those quad records with hashes that are similar to the test sample quad-hashes. It is called the set of raw candidates because it will most likely be a mixture of true positives and a large number of false-positive matches. For matching audio fingerprint, all the process will be done on spectral quads.

Sequence Estimation and Match Verification

This stage is on a per-file-ID basis, so the match candidates are group by file-ID, and the groups will be sorted by the number from the most match candidates to less match candidate. Per file-ID, by following the step in Shazam's algorithm, the matching process is done by using the histogram method [17]. The method is adapted such that the test sample time is scaled according to the uncovered time scale factor. The file-ID for the largest histogram bin is returned together with the match position. Finally, if match sequences are found for a given file-ID, and their number of matched candidates is larger than a threshold value, these sequences will be verified match-by-match. If the testing samples match, from the beginning of the training dataset matching feature, it will occur at the similar relative offset. In one of the training samples, sequences of hashes should occur in matching training samples with the same relative time sequences. Figure 6 shows a diagonal line is performed within the quad.

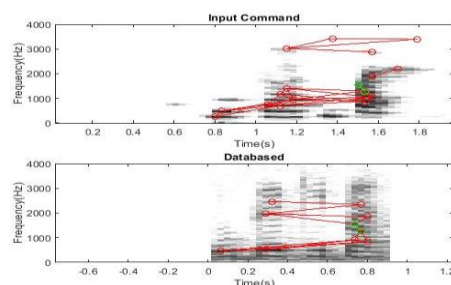


Figure 6. The diagonal line is performed within the quad.

3.4. Hardware Design

The algorithm will be deployed in Raspberry Pi (as shown in Figure 7) to test its accuracy and efficiency of the algorithm when it is deployed on board. Raspberry Pi is chosen because its size is small and easy to carry compared to the computer. For real-time test, the Raspberry Pi will be connected to the microphone to record the Malay command given by the user. The LED that is connected with the Raspberry Pi will show the output.

Besides using the recorded dataset, Raspberry Pi also can connect online and offline with Google Speech Recognition API. In this case, not all Malay commands can be obtained in the Google Speech API. If the command given by the user matches with any of the recorded datasets or matches with the API, the LED will be switched on for 'Buka' command and switched off for 'Tutup' command. For controlling the television channel, RGB LED is used so the colour of the LED will be changed according to the number of channels chosen by the user. Command 'Siaran 1' blue colour, 'Siaran 2' green colour and 'Siaran 3' red colour.

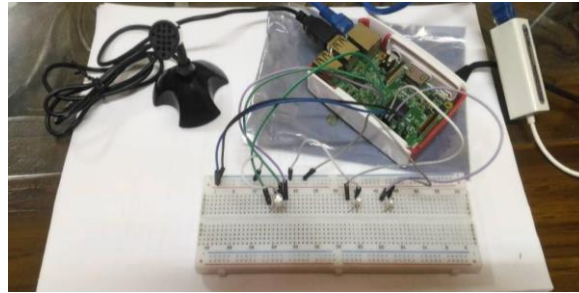


Figure 7. The connection of Raspberry Pi

4. Result and Discussion

The voice-based Malay command recognition will be tested with the different signal of noise ratio (SNR) in range of -10 dB to 30 dB (noise that been implement in MATLAB). Then, the system will be deployed in Raspberry Pi, and the performance of the system to will be compared between computer and Raspberry Pi. The accuracy and performance of each command will be recorded.

4.1 Accuracy Results

The accuracy of the system will be measured based on its recognition rate. To get the recognition rate, the number of commands had been recognized by the system will be divided by the total number of testing samples. The recognition rate of voice-based command recognition is defined as follows Equation 1:

$$\text{Recognition Rate} = \frac{\# \text{correctly recognized}}{\text{total number of testing samples}} \times 100\% \quad (1)$$

The precision of this system also will be recorded. Precision is the proportion of cases out of all cases where the system claim to have identified the reference, where the prediction is correct. Given tp is number of cases where the commands are correctly recognized and fp is the number of cases when the command are incorrectly recognized, the precision for the system is defined in Equation 2:

$$\text{Precision} = (tp / tp + fp) \times 100\% \quad (2)$$

The recognition rate and precision of this system are obtained by calculating the mean of the 5-fold cross-validation process. The recognition rate of this system is 87%. The recognition rate of each command are 86% for 'Buka Kipas', 80.56% for 'Tutup Kipas', 83.33% for 'Buka Lampu', 83.33% for 'Tutup Lampu', 87.50% for 'Buka TV', 87.50% for 'Tutup TV', 91.67% for 'Siaran 1', 90.28% for 'Siaran 2', 90.28% and 'Siaran 3'. The recognition rate for 'Tutup Kipas' and 'Tutup Lampu' is low because their command signal has a similar pattern but different amplitude. The command signal for 'Lampu' is slightly smaller than 'Kipas'.

The precision for voice command is 88.22%. The precision for each command are 87.32% for 'Buka Kipas', 82.86% for 'Tutup Kipas', 84.51% for 'Buka Lampu', 86.96% for 'Tutup Lampu', 88.93% for 'Buka TV', 91.30% for 'Tutup TV', 91.67% for 'Siaran 1', 90.28% for 'Siaran 2', 90.28% and 'Siaran 3'. Precision was calculated because there is some command that cannot be predicted by the system. To see the performance of the algorithm in the noisy background, each command is tested with different levels of noise. Table 1 shows the recognition rate and the precision for each command before adding SNR.

Table 1. The recognition rate and the precision for each command before add SNR.

| Malay Command | Recognition Rate | Precision |
|---------------|------------------|-----------|
| Buka Kipas | 86.57 | 87.32 |
| Tutup Kipas | 80.56 | 82.86 |
| Buka Lampu | 83.33 | 84.51 |
| Tutup Lampu | 83.33 | 86.96 |
| Buka TV | 87.5 | 88.73 |
| Tutup TV | 87.5 | 91.30 |
| Siaran 1 | 91.67 | 91.67 |
| Siaran 2 | 90.28 | 90.28 |
| Siaran 3 | 90.28 | 90.28 |

Then, the background noise with signal to noise ratio (SNR) from range -10 dB to 30 dB is added with test samples. The mixed test sample will be compared again with the training dataset to see the recognition rate and precision of voice-based Malay command recognition. The recognition rate of voice-based Malay command begins to drop 50% at approximately -5 dB SNR because the noise is too loud. When the mixed sound is played, only some part of the command is heard. Figure 8 shows the recognition rate and the precision for all command.

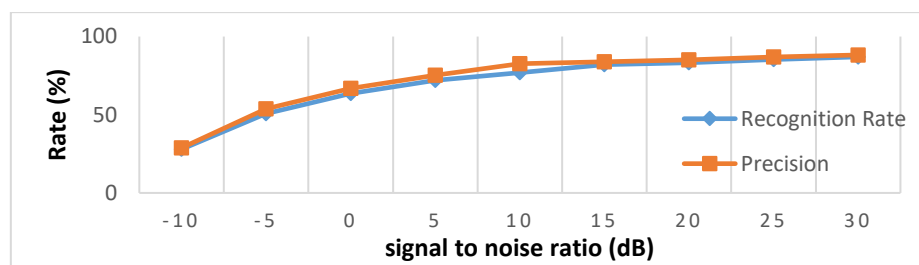
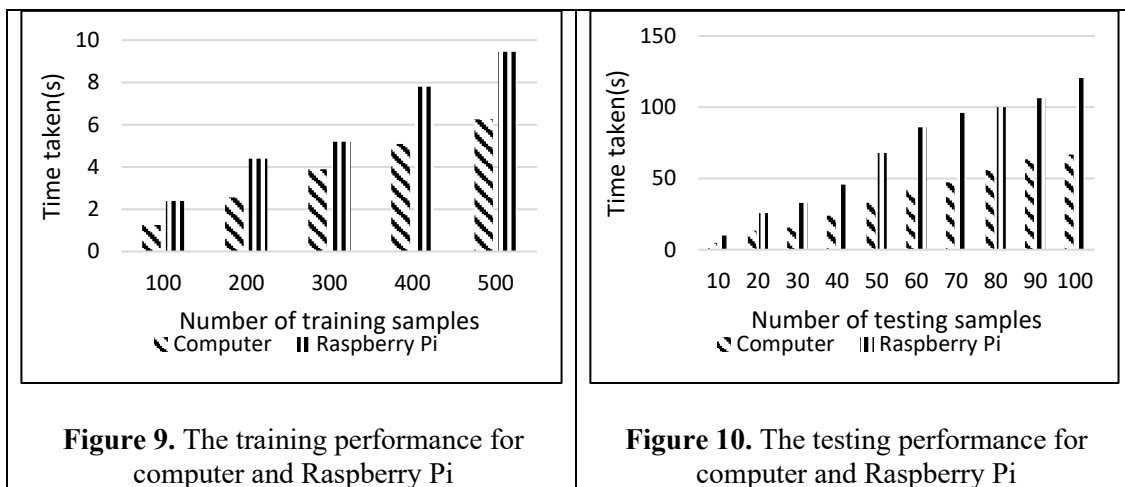


Figure 8. The recognition rate and the precision for all command

After that, the algorithm will be deployed in Raspberry Pi for real-time test. To test in real-time, the speaker needs to train their voice three times in the system first before test on the system five times. The accuracy of real-time recognition is 68.5%, and the accuracy starts to decrease when the SNR reaches 15 dB approximately. The accuracy of real-time is less than the already recorded voice command because the tempo for the speaker to record their voice is different. There also some disturbance while recording the command in real-time, for example, other speakers voice and their movement. The device used to record the voice command also can be the reason why the accuracy of real-time in low. The time taken to record voice command in real-time is 2 seconds compare to recorded voice most of the command is recorded in 1 second because 1 second is not enough time for real-time. Some speaker starts late, so some of the commands are cropped.

4.2 Recognition Time

The efficiency of this system will be tested on the computer and Raspberry Pi based on the performance of the system to recognize the voice command in a different platform by changing the number of train and test the sample. The specification used for the PC is Intel core i5, RAM 6 and the operating system is Windows 8.1. The training dataset and the testing dataset is divided into 80:20 ratio by choosing the dataset randomly. The samples will be trained from 100 samples until 500 samples. It takes approximately 1 second to train 100 samples, 2 seconds to train 200 samples, 3.8 seconds to train 300 samples, 5 seconds to train 400 samples and 6 seconds to train 500 samples. Figures 9 and 10 show the performance for computer and Raspberry Pi.



The training performance for Raspberry Pi is approximately two times the performance of the computer. For testing, the samples will be test from 10 samples to 100 samples. The performance of the system for testing dataset also increases as the samples of testing dataset increase. It only take approximately 4 seconds to train 10 samples, 13 seconds to train 20 samples, 15 seconds to train 30 samples, 24 second to train 40 samples, 33 second to train 50 samples, 43 second to train 60 samples, 47 seconds to train 70 samples, 56 second to train 80 samples, 63 second to train 90 samples and 67 second to train 100 samples. As shown in Figure 9 and Figure 10, the training time and testing time of the proposed algorithm on the Raspberry Pi is higher than the computer due to the higher computational capability of the computer. However, the training and testing time of the algorithm on Raspberry Pi is linear with the number of samples. These results show the efficiency of the algorithm to be deployed on a Raspberry Pi.

4.1. Comparison MFCC+NN and Audio Fingerprint Method

In this paper, comparison had been made between robust quad based audio fingerprint method and the neural network (NN) method with Mel-frequency Cepstral Coefficient (MFCC). The accuracy for audio fingerprint method is 86% higher than the neural network method only 40%. The accuracy of the NN method low because the time taken for the speaker to record their voice is difference, so the timing for the training data and testing data are different. For the NN method, the timing for all training and testing datasets must be the same. The audio fingerprint method focus on the spectral peak from the frequency and time domains so time differences and background noise can be ignore. Even in real-life application, the louder the noise, the louder voice command given. Besides that, the time taken for the NN method to train the data is longer than the audio fingerprint method. The time taken for the neural network to train 500 samples is approximately 17 seconds while audio fingerprint method only takes approximately 5 seconds.

5. Summary

Voice-based Malay command recognition for smart house applications has very limited Malay commands compared to English. Therefore, a preliminary work on voice-based Malay commands recognition was designed by using an audio fingerprint method. Three electrical appliances (light, fan, and television) had been chosen as target appliances for smart house applications. In this paper, an audio fingerprint method that detects the highest peak of frequency was used. By detecting the highest peak, the background noise can be ignored because human voice will become louder if the noise is louder. Then, the algorithm was deployed on a Raspberry Pi board and we evaluate its performances based on accuracy and efficiency. The performance was evaluated in terms of recognition time by running the

system on the computer and by deploying it on Raspberry Pi. The experiment result shows that besides using low pass filtering and high pass filtering, peak detection also can be used to increase the accuracy of voice command recognition. This paper has shown that the proposed algorithm is lightweight, and it can be implemented on a Raspberry Pi. From all experiment that has been done, some further works can be added to see the performances of this method. In this paper, the test sample is added with SNR and the highest accuracy is 87%. Besides adding background noise to the voice command, by using the same algorithm the voice command can be tested in different distances. This algorithm is originally used for music recognition. Nonetheless, it has the potential to be applied for detecting a long speech recognition.

Acknowledgements

Firstly, I would like to dedicate my acknowledgement of gratitude toward my supervisor, Dr Amiza Amir, for her guidance and support as I make my research paper. I also want to thank my parents, family and friends, who provide advice and financial support during my research.

References

- [1] A. Kumar, "Voice Based Control Command Signal Generation for Intelligent System," *2015 IEEE UP Sect. Conf. Electr. Comput. Electron.*, pp. 1–6, 2015.
- [2] Norhafizah and M. B. Othman, "Voice control of home appliances using Android," *Proc. - 2014 Electr. Power, Electron. Commun. Control Informatics Semin. EECCIS 2014. conjunction with 1st Jt. Conf. UB-UTHM*, pp. 142–146, 2014.
- [3] U. Shrawankar and V. Thakare, "Techniques for Feature Extraction in Speech Recognition System : A Comparative Study," 2013.
- [4] H. F. Ong and A. M. Ahmad, "Malay Language Speech Recogniser with Hybrid Hidden Markov Model and Artificial Neural Network (HMM / ANN)," *Int. J. Inf. Educ. Technol.*, vol. 1, no. 2, pp. 114–119, 2011.
- [5] Abd Manan, "The Design and Development of a System for Controlling Automotive Functions Using Speech Recognition," 2006.
- [6] P. Barua, K. Ahmad, A. Anam, S. Khan, and M. Sanaullah, "Neural Network Based Recognition of Speech Using MFCC Features," *3rd Int. Conf. Informatics, Electron. Vis. 2014*, pp. 1–6, 2014.
- [7] P. Leechor, C. Pompanomchai, and P. Sukklay, "Operation of a radio-controlled car by voice commands," *ICMEE 2010 - 2010 2nd Int. Conf. Mech. Electron. Eng. Proc.*, vol. 1, no. Icmee, pp. 2–5, 2010.
- [8] H. L. Zhe Gong, "Voice Controlled Home Automation System," no. 13, 2014.
- [9] S. Oulkar, R. Bamane, S. Gulave, and P. Kothawale, "Voice Controlled Home Automation Using Raspberry Pi 3," *Int. J. Recent Innov. Eng. Res.*, vol. 2, no. 1, pp. 28–32, 2017.
- [10] H. Gyulyustan and S. Enkov, "Experimental speech recognition system based on Raspberry Pi 3," *IOSR J. Comput. Eng.*, vol. 19, no. 3, pp. 107–112, 2017.
- [11] Upadhye and R. A. Khan, "Raspberry Pi Home Automation with Voice Control," *Int. J. Adv. Res. Ideas Innov. Technol.*, vol. 3, no. 6, pp. 1297–1301, 2017.
- [12] A. Samah and A. A. Osman, "Controlling Home Devices for Handicapped People via Voice Command Techniques," *Int. Conf. Comput. Control. Networking, Electron. Embed. Syst. Eng.*, pp. 374–378, 2015.
- [13] L. Pan, "Research and simulation on speech recognition by Matlab," 2013.
- [14] A. L.-C. Wang, "An Industrial Strength Audio Search Algorithm.," *Proc. 4th Int. Soc. Music Inf. Retr. Conf. (ISMIR 2003), Balt. Maryl. (USA), 26-30 Oct. 2003*, pp. 7–13, 2003.
- [15] R. Sonnleitner and G. Widmer, "Robust quad-based audio fingerprinting," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 24, no. 3, pp. 409–421, 2015.
- [16] N. Mulhern, N. Mccaffrey, N. Beretta, E. Chabot, and Y. Sun, "Designing Android Applications using Voice Controlled Commands For Hands free interaction with Common Household Devices," *2013 39th Annu. Northeast Bioeng. Conf. Des.*, pp. 265–266, 2013.