

Performance Comparison between Lab-VIEW and MATLAB on Feature Matching-based Speech Recognition System

Edita Rosana Widasari¹, Barlian Henryranu Prasatio², Dian Eka Ratnawati³

Dept. of Computer Engineering, Faculty of Computer Science, Brawijaya University, Malang, Indonesia^{1,2}

Dept. of Informatics Engineering, Faculty of Computer Science, Brawijaya University, Malang, Indonesia³

Abstract—Speech recognition systems are widely used for smart applications. The smart application-based speech recognition system has different requirements for processing the human voice. The most common performance in the speech recognition system is essential to observe, since it is necessary to design smart application-based speech recognition systems for people's needs. Moreover, feature matching is the principal part of speech recognition systems since it plays a key role to authenticate, separate one human voice from another, and their different articulation. Therefore, this work proposes a performance comparison of speech recognition systems based on feature matching using Lab-VIEW and MATLAB. The feature extraction involves calculation of Mel Frequency Cepstral Coefficients (MFCC) for each frame. For the matching process, the system was tested 100 times for each five speeches by making changes in articulation with the same vocal cords. This matching process uses DTW (Dynamic Time Warping), and then the testing is based on the most common performance in the speech recognition system's comparison between Lab-VIEW and MATLAB such as accuracy rate, execution time, and CPU usage. Based on experimental results, the average accuracy rate of MATLAB is better than Lab-VIEW. The execution time testing indicates that Lab-VIEW has a shorter execution time than MATLAB. On the other hand, Lab-VIEW and MATLAB have almost the same CPU usage. This result indicates that the performance comparison is able to be used according to the requirements of smart application-based speech recognition systems.

Keywords—Speech; articulation; feature matching; Lab-VIEW; MATLAB

I. INTRODUCTION

In general, the instrument for creating the human voice can be confined into three parts, i.e., lungs, vocal strings, and verbalization [1]. The combination of the vocal strings with verbalization can create an assortment of discourse. The discourse quality, counting unforgiving, tense, breathy, or whispery voice, can be influenced by emotion and temperament [2]. In the last decade, there has been an automated method of identifying words spoken by the human voice and converting them into readable text. This automated method is called speech recognition. Furthermore, the human voice can also be utilized in computer technology by using the speech recognition system.

Speech recognition is utilized to change over talked shape into content to help people needs [3]. The speech recognition

system is widely used for smart applications, e.g., intelligent wheelchair, Google assistant, Alexa, Cortana, Siri, and home assistant [4], [5]. Each smart application-based speech recognition system has different requirements for processing the human voice. Typically, the speech recognition system works through four stages, i.e., speech analysis, feature extraction, modeling, and testing techniques (matching process) [6]. In the speech analysis stage, speech data contains different types of information that appear a human voice identity. The next stage is feature extraction, which takes on features that might be used to match the digital signal of the human voice to a particular pattern. Then, the modeling stage is used to generate speaker models using speaker-specific feature vectors. In the last stage, the speech-recognition system matches a detected word to a known word using testing techniques (matching process). Feature matching is the most important stage of speech recognition since it plays an important role in authenticating and separating one human voice from another and their different articulation. The matching results then identify similarities [7].

In the last decade, some works have observed the performance of speech recognition system separately [8], [9], [10], [11]. However, since the most common performance of speech recognition systems based on feature matching was not observed at all. Thus, it is needed to observe of all the most common performance of speech recognition systems based on feature matching.

To this end, this work proposes a system that can identify a speech using features matching the most common performance of speech recognition systems. The work is aimed to recognize the speech built in two programming languages Lab-VIEW and MATLAB. They were selected since they have been widely used for designing smart application-based speech recognition systems [12], [13]. In Lab-VIEW programming, the structure of the graphical piece of a program chooses the execution stream in which the computer program design interfacing center points by drawing wires [14]. Furthermore, Lab-VIEW programming is able to combine the virtual instrumentation technology and speech recognition system; and also provided password authentication [15]. On the other hand, MATLAB permits framework control, plotting of a work and information and calculation usage. In spite of the fact that numerically nuanced, a tool compartment that employments a typical machine permits get to the computer logarithmic capabilities [16]. Moreover, in

order to address the analysis and testing issue an appropriate software tool is developed using MATLAB that enabled unified framework for tracking the performance of all necessary functions of speech recognition system [17]. These programming language performances were compared when identifying speech to determine the parameters used for processing human voice.

This work focuses on the method of how the system can recognize the speech based on previously stored voice features sequence as the reference signal. This work implemented the voice recognition based on feature matching with changing articulations. Structurally, the speech recognition system requires a dataset used to train the system's sensitivity to speech patterns. In this work, the stored training data is called the dictionary. This dictionary is used as the database in matching the spoken word.

This work organizes the rest of this paper as follows. Section II presents a review of related works. Section III describes the detail process of feature matching. In Section IV express the materials and system design includes the detail description of the data used in this work and the design of proposed system. Section V presents the result and discussion of the performance comparison between Lab-VIEW and MATLAB on feature matching-based speech recognition system. Finally, Section VI is dedicated to conclusion and further works.

II. RELATED WORK

The speech recognition system is widely used for smart applications. A study [18] estimated that speech recognition needs to achieve close to a 90% accuracy rate for designing smart homes assistants. One of the factors needed in its design is data communication between devices so as to provide security and convenience that meet people's needs [19]. Moreover, the slower or faster time execution of an intelligent wheelchair is according to their current resistance [20]. Afterwards, the voice assistant technologies (such as Google

Assistant, Alexa, Cortana, Siri, etc.) require addressing restrictions like CPU and memory limitations [21]. This is to achieve an efficient on-device streaming speech recognition system. Thus, this work considers that the accuracy rate, time execution, and CPU usage could represent the most common performance in the speech recognition system. Therefore, the consideration of these parameters is important to observe.

According to A. A. A. Zamil, et al. [8], the extricated highlights of the obscure discourse and after that compared them to the put away extricated highlights for each diverse speaker in arrange to distinguish the obscure speaker using a voting mechanism. However, the key process of selecting the extracted features is minimizing the difficulty of speech recognition system computing for matching processes [9]. Therefore, another study has observed the performance of speech recognition system computing [10], [11]. Nevertheless, the most common performance of speech recognition systems based on feature matching was not observed at all. Therefore, the observation of all the most common performance of speech recognition systems based on feature matching (i.e., the accuracy rate, time execution, and CPU usage) is needed.

III. FEATURE MATCHING

Before the recognition process, pre-emphasis was applied to the voice signal [22]. Pre-emphasis was aimed to suppress high-frequency parts during the production mechanism of the human voice before performing the framing and windowing process. The framing and windowing process was intended to split the speech signal into smaller parts [23] because it needs to be assumed as a stationary signal. Finally, energy detection was performed on each frame to detect the existence of pronunciation in that frame [24].

Structurally, speech recognition consists of four stages: (1) speech analysis, (2) feature extraction, (3) modeling, and (4) testing techniques or matching process. The block diagram of feature matching, which includes all stages of speech recognition, is shown in Fig. 1.

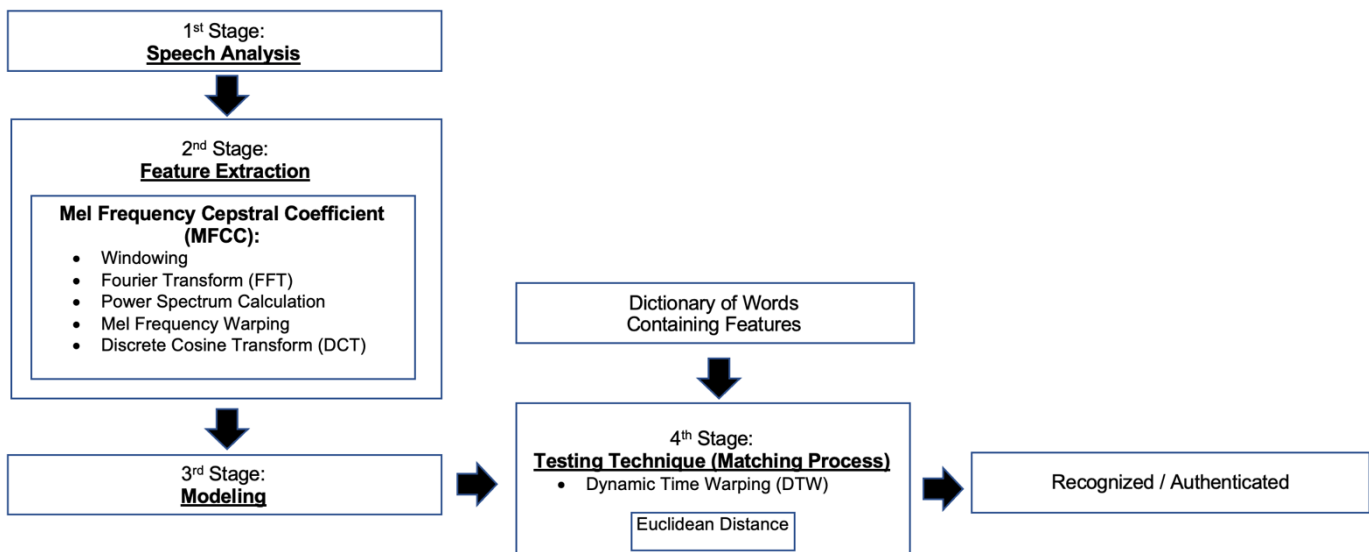


Fig. 1. Block Diagram of Feature Matching.

The first stage is speech analysis, which contains different types of information that shows a human voice identity. After the pronunciation has been detected, this work needs to extract the speech feature on each frame in the second stage. The feature extraction involves calculation of Mel Frequency Cepstral Coefficients (MFCC) for each frame. By using the MFCC, it is able to develop the features from the speech signal which can be used for speech recognition system [25]. The MFCC consist of the following steps: Windowing, Fast Fourier Transform (FFT), power spectrum calculation, Mel Frequency warping, and Discrete Cosine Transform (DCT). The block diagram of the feature extraction process of MFCC is shown in Fig. 2.

Since the effect of frame blocking the speech signal becomes discontinuity, windowing is required in the first process of second stage. Then, FFT is applied to transform speech signal to frequency domain in each frame. Furthermore, the power spectrum for each frame is calculated. However, it is having a lot information which is not needed for feature matching process. Thus, Mel Frequency Warping is used for filtering in the form of a filter bank to determine the size of the power spectrum of a certain frequency band and convert the frequency into mel. Finally, DCT is used for producing a mel spectrum to improve recognition quality [25], [26].

In the third stage, the feature extraction process was done on a set of words and then stored the feature vector sequences [27] as the dictionary. The dictionary function performs feature extraction, which will be stored and used as a reference. The dictionary will be used as a matching reference with the recorded voice speech features. Thus, in the early stages, this work needs to save the voice that will be used as a dictionary dataset. The data were stored in the dictionary and formatted as an array. There were 25 feature vector data sets in the dictionary, which consisted of voice left (5 sets), right (5 sets), up (5 sets), down (5 sets) and stop (5 sets).

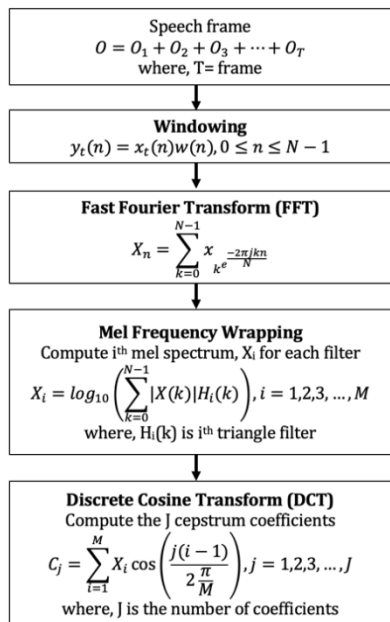


Fig. 2. The Flowchart of Feature Extraction Process of MFCC.

Typically, people have different speaking speeds and characteristics [28]. Dynamic Time Warping (DTW) is used to normalize these differences. Moreover, feature vectors of the voice test sequence were also compared with each word in the dictionary set using the DTW algorithm and the best match in each set was outputted in the final stage. DTW is an algorithm used to measure similarity between two sequences which may vary in time or speed. Thus, it can find the best alignment between two different sequences of signals. Fig. 3 shows the flowchart to implement the DTW algorithm.

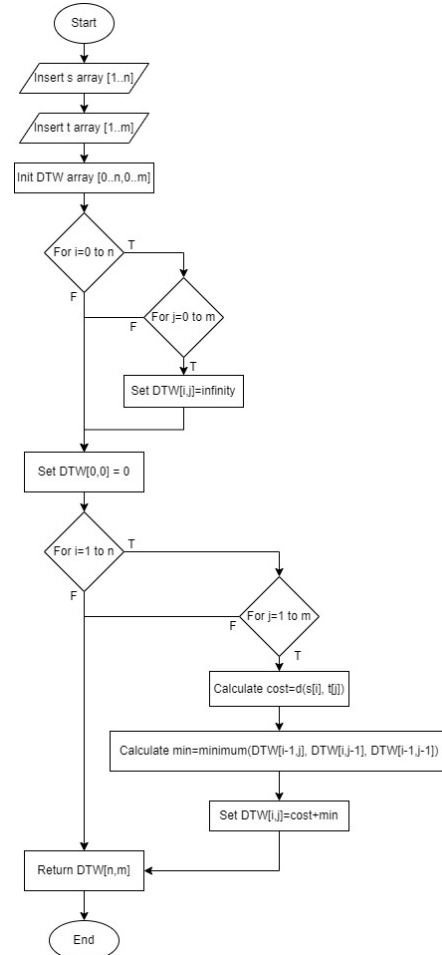


Fig. 3. The Flowchart of DTW Algorithm.

The DTW algorithm works mainly by calculating the Euclidean distance between two points [8], which is the point of the test and reference points for recognizing or authenticating each frame [29]. The threshold was set so that the random noise signal is not generated by the matching speech. Therefore, the DTW algorithm can also be used to find the best matching between voice data test and dictionary data. Then the Euclidean distance function is called when calculating the DTW. After that the matching process between the dictionary voice features with the voice data that has been recorded is performed. Fig. 4 shows the flowchart for calculating Euclidean distance. Finally, this distance will use for comparing voice and reference voice to recognize or authentication process. The detail process of recognize or authentication is described in Section IV below.

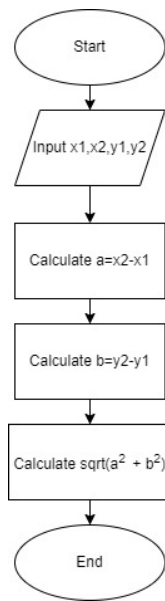


Fig. 4. The Flowchart of Euclidean Distance Calculation.

IV. MATERIAL AND SYSTEM DESIGN

This work used the TIMIT Acoustic-Phonetic Continuous Speech Corpus dataset. This is a standard dataset used for evaluation of automatic speech recognition systems. It consists of recordings of 630 speakers of 8 dialects of American English each reading 10 phonetically-rich sentences. It also comes with the word and phone-level transcriptions of the speech. Furthermore, to obtain the work’s objective, the five sets of voice dictionaries (such as left, right, up, down, and stop) is used for matching processes by making changes in articulation.

The system design consists of three parts: Dictionary, DTW, and Euclidean Distance. The Dictionary was used to store five words: left, right, up, down and stop. Each word was repeated 5 times. The voice test sequence will be matched to all words in the first dictionary set and then the word with the least distance will be extracted. Then, the voice test sequence will be matched to all words in the second dictionary set and the matched word will be extracted. The same will be done for the next three sets. Thus, the dictionary will give 5 matched outputs from the dictionary set. The most repeated word in the output will be regarded as the best match.

The DTW can normalize the speed difference of speaking [30]. It can find best-alignment between two different signals. This function will call the Euclidean distance function to calculate the distance between the test signals with reference signals. After that, the results of feature extraction were matched with the dictionary that has been recorded as the reference. This function will do the matching features by calling DTW to find the best alignment between the two sets of sequences.

The voice received by the system compared to reference voice that has been stored in the dictionary. The distance of both speeches was calculated using the Euclidean Distance algorithm. The closest distance to the set reference speech is output for recognizing or authentication process.

V. RESULT AND DISCUSSION

In this section, the block diagram and flowchart in the previous section have been implemented in the programming language using Lab-VIEW and MATLAB on a desktop of Intel i7 4-core CPU and 8 GB RAM. Furthermore, this work describes the performance comparison results in the speech recognition system.

A. Experimental Result

The testing phase was performed by connecting a microphone to a computer that had installed Lab-VIEW and MATLAB program code. Then, while the speaker speaks a voice, the signals were acquired by the Lab-VIEW for 2 seconds duration at 11025 Hz sampling rate. This work tested the system using 5 different voice speeches consisting of “Left”, “Right”, “Up”, “Down”, and “Stop”. Each of these words was repeated 100 times with different articulations. The detailed information about the amount of data for the matching process can be seen in Table I.

The recognized outputs were observed to calculate the system performance. The first test is the success level of the system in identifying what is spoken. The details of the first testing result of accuracy rate can be seen in Table II.

The average accuracy rate for Lab-VIEW and MATLAB are: 85,6 % and 89,6 %. The accuracy results are obtained by ratio between true positive (TP) and true negative (TN) to TP, TN, false positive (FP), and false-negative (FN) as shown in (1) [31]. TP presents the number of voices that are labeled correctly and TN for the number of voices that are correctly identified as not corresponding to the words spoken. FP indicates the number of voices that are incorrectly labeled. FN denotes the number of voices that are unidentified in words spoken.

TABLE I. DETAILED INFORMATION ABOUT DATA FOR MATCHING PROCESS

| Word | The number of data |
|--------------|--------------------|
| Left | 100 |
| Right | 100 |
| Up | 100 |
| Down | 100 |
| Stop | 100 |
| Total | 500 |

TABLE II. THE TESTING RESULT OF ACCURACY RATE IN THE SPEECH RECOGNITION SYSTEM

| Word | Lab-VIEW Accuracy Rate (%) | MATLAB Accuracy Rate (%) |
|----------------|----------------------------|--------------------------|
| Left | 81 | 91 |
| Right | 85 | 85 |
| Up | 88 | 87 |
| Down | 85 | 92 |
| Stop | 89 | 93 |
| Average | 85,6 | 89,6 |

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

The second test is to test the running time required to execute the program. The execution time is calculated from the voice received by the system until the system finishes identifying the voice. The testing is done 100 times on each word. The second test result of execution time can be seen in Table III, which shows that the average execution time for Lab-VIEW and MATLAB are: 723 ms and 969,4 ms.

TABLE III. THE EXECUTION TIME RESULT OF SPEECH RECOGNITION SYSTEM

| Word | Lab-VIEW Execution Time (ms) | MATLAB Execution Time (ms) |
|----------------|------------------------------|----------------------------|
| Left | 752 | 1044 |
| Right | 883 | 1091 |
| Up | 617 | 953 |
| Down | 698 | 847 |
| Stop | 665 | 912 |
| Average | 723 | 969,4 |

The third test is CPU usage. This test is quite important in building the system so that resources can be used optimally. The testing is done 100 times on each word. The third test result of CPU usage can be seen in Table IV, which shows that the average of CPU usage for Lab-VIEW and MATLAB are: 1,55 % and 1,57 %.

TABLE IV. THE CPU USAGE RESULTS OF SPEECH RECOGNITION SYSTEM

| Word | Lab-VIEW CPU Usage (%) | MATLAB CPU Usage (%) |
|----------------|------------------------|----------------------|
| Left | 1,56 | 1,41 |
| Right | 1,84 | 1,42 |
| Up | 1,41 | 1,74 |
| Down | 1,63 | 1,67 |
| Stop | 1,29 | 1,62 |
| Average | 1,55 | 1,57 |

B. Discussion

Based on experimental results, the average accuracy rate of MATLAB is better than Lab-VIEW since the value of TP and TN in the MATLAB results is higher than Lab-VIEW. On the other hand, Lab-VIEW has a shorter execution time than MATLAB since Lab-VIEW has simple code that is assessed according to the number of elements or lines used. The faster the code can be updated or debugged the better surveyable the program is [32].

In order to verify the performance comparison results, a box-plot method or also called Box-and-Whisker plot method was conducted as presented in Fig. 5. From the figures, this work can find that average accuracy rate and execution time are significantly different between Lab-VIEW and MATLAB using t-test at the level of significance (ρ) of 0.05. This result indicates that the average accuracy rate of MATLAB is actually better than Lab-VIEW. Then, Lab-VIEW actually has a shorter execution time than MATLAB since both of them

are significantly different. It is also in accordance with the experimental results.

CPU usage becomes an essential metric to determine how well an application is using the cores. CPU usage refers to a program's usage of processing resources, or the amount of work handled by a CPU. Since the amount and type of managed computing tasks are the same, Lab-VIEW and MATLAB have almost the same CPU usage. Based on the statistical analysis, the CPU usage is not significantly different between Lab-VIEW and MATLAB using t-test at the level of significance (ρ) of 0.05. This result indicates that the Lab-VIEW and MATLAB have the same CPU usage since both of them are not significantly different and in accordance with the experimental result.

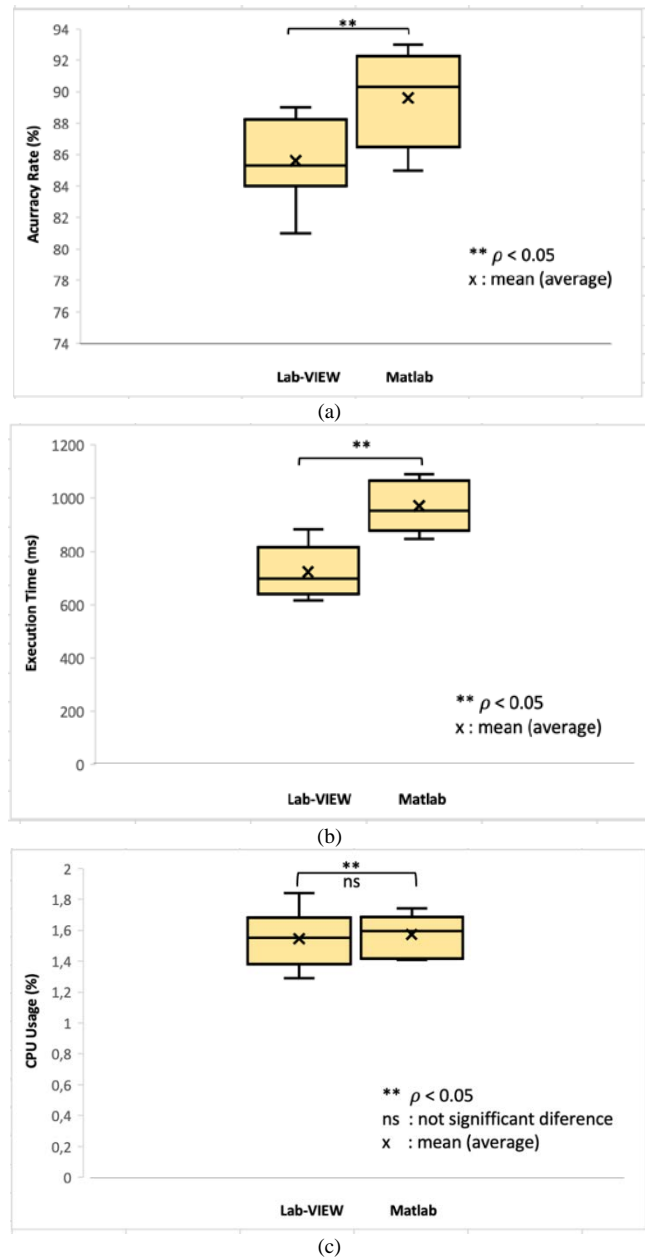


Fig. 5. Box-and-Whisker Plots of Average (a) Accuracy Rate, (b) Execution Time and (c) CPU usage.

VI. CONCLUSION

This work has designed a speech recognition system using feature matching on Lab-VIEW and MATLAB. The feature was extracted using MFCC and calculating distance using DTW from the speech. Before performing the speech, this work saved a set of the features from speech voice as a training set in a dictionary. The matching process was performed between the feature of the voice and the feature which had been saved in the dictionary. In the testing phase, this work tested five speech words and each word was repeated 100 times. The system experimented using the most common performance in the speech recognition system i.e., accuracy rate, execution time, and CPU usage. The performance comparison results show that the average accuracy rate of Lab-VIEW is 85.6% and MATLAB is 89.6%. The execution time testing of Lab-VIEW is 723 ms and MATLAB is 969.4 ms. While, the Lab-VIEW and MATLAB have almost the same CPU usage which is around 1.5%.

Speech recognition has wide smart applications and includes voice-controlled appliances fully featured speech-to-text software, automation of operator-assisted services and voice recognition aids. This work's result indicates that the performance comparison is able to be used according to the requirements of smart application-based speech recognition systems. Hence, the performance comparison results in improving many speech recognition system applications can be further extended, which can make the process more robust and effective.

REFERENCES

- [1] Z. Zhang, "Mechanics of human voice production and control," *J. Acoust. Soc. Am.*, vol. 140, no. 4, pp. 2614–2635, 2016. DOI: 10.1121/1.4964509.
- [2] S. Bhatt, A. Jain, and A. Dev, "Acoustic Modeling in Speech Recognition: A Systematic Review," *Int. J. Adv. Comput. Sci. Appl. Sci. Inf. Organ.*, 2020. DOI: 10.14569/IJACSA.2020.0110455.
- [3] S. Yang, "Listener's ratings and acoustic analyses of voice qualities associated with English and Korean sarcastic utterances," *Speech Commun.*, vol. 129, pp. 1–6, 2021. DOI: 10.1016/J.SPECOM.2021.02.002.
- [4] M. K. Luka, F. Ibikunle, and O. Gregory, "Neural network based Hausa language speech recognition," *Int. J. Adv. Res. Artif. Intell. Sci. Inf. Organ.*, vol. 1, no. 2, pp. 39–44, 2012. DOI: 10.14569/IJARAI.2012.010207.
- [5] G. Dharmale, V. M. Thakare, and D. D. Patil, "Implementation of Efficient Speech Recognition System on Mobile Device for Hindi and English," *Int. J. Adv. Comput. Sci. Appl. Sci. Inf. Organ.*, 2019. DOI: 10.14569/IJACSA.2019.0100212.
- [6] K. Gupta and D. Gupta, "An analysis on LPC, RASTA and MFCC techniques in Automatic Speech recognition system," in *2016 6th international conference-cloud system and big data engineering (confluence)*, 2016, pp. 493–497. DOI: 10.1109/CONFLUENCE.2016.7508170.
- [7] S. Karpagavalli and E. Chandra, "A review on automatic speech recognition architecture and approaches," *Int. J. Signal Process. Image Process. Pattern Recognit.*, vol. 9, no. 4, pp. 393–404, 2016. DOI: 10.14257/ijcip.2016.9.4.34.
- [8] A. A. A. Zamil, S. Hasan, S. M. D. J. Baki, J. M. D. Adam, and I. Zaman, "Emotion detection from speech signals using voting mechanism on classified frames," in *2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, 2019, pp. 281–285. DOI: 10.1109/ICREST.2019.8644168.
- [9] K. Jermsttiparsert *et al.*, "Pattern recognition and features selection for speech emotion recognition model using deep learning," *Int. J. Speech Technol.*, vol. 23, no. 4, pp. 799–806, 2020. DOI: 10.1007/s10772-020-09690-2.
- [10] A. Hussein, "Analysis of Voice Recognition Algorithms using MATLAB," *Int. J. Eng. Res. & Technol.*, vol. 4, no. 8, pp. 273–278, 2015.
- [11] W. Liu, Q. Liao, F. Qiao, W. Xia, C. Wang, and F. Lombardi, "Approximate designs for fast Fourier transform (FFT) with application to speech recognition," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 66, no. 12, pp. 4727–4739, 2019. DOI: 10.1109/TCSI.2019.2933321.
- [12] S. Pandita and S. K. Sharma, "Speech-enhancement techniques and review on the role of labview in speech-enhancement techniques," *INROADS-An Int. J. Jaipur Natl. Univ.*, vol. 8, no. 1and2, pp. 31–34, 2019. DOI: 10.5958/2277-4912.2019.00005.5.
- [13] F. Barkani, H. Satori, M. Hamidi, O. Zealouk, and N. Laaidi, "Comparative Evaluation of Speech Recognition Systems Based on Different Toolkits," *Embed. Syst. Artif. Intell.*, pp. 33–41, 2020. DOI: 10.1007/978-981-15-0947-6_4.
- [14] N. Berezowski and M. Haid, "How to Use a Graphical Programming Language in Functional Safety, using the Example of Lab VIEW," in *2020 IEEE International RF and Microwave Conference (RFM)*, 2020, pp. 1–4. DOI: 10.1109/RFM50841.2020.9344729.
- [15] S. Pleshkova, Z. Zahariev, and A. Bekiarshi, "Development of Speech Recognition Algorithm and LabView Model for Voice Command Control of Movable Robot Motion," in 2018 In International Conference on High Technology for Sustainable Development (HiTech), 2018, pp. 1-5. DOI: 10.1109/HiTech.2018.8566257.
- [16] L. Keviczky, R. Bars, J. Hetthéssy, and C. Bányász, "Introduction to MATLAB," in *Control Engineering: MATLAB Exercises*, Springer, 2019, pp. 1–27.
- [17] I. McLoughlin. *Speech and Audio Processing: A MATLAB®-based Approach*. University Printing House, Cambridge CB2 8BS, United Kingdom, 2016.
- [18] M. Alzantot, B. Balaji, and M. Srivastava, "Did you hear that? adversarial examples against automatic speech recognition," *arXiv Prepr. arXiv1801.00554*, 2018.
- [19] B. H. Prasetio and D. Syaury, "Implementation of Lossless Voice Data Communication using Network Streams on Embedded System," *JITECS (Journal Inf. Technol. Comput. Sci.)*, vol. 2, no. 2, 2017. DOI: 10.25126/jitecs.20172225.
- [20] A. Ghorbel, N. Ben Amor, and M. Jallouli, "Towards a Power Adaptation Strategy in Multi-core Embedded Devices. A Case Study: a HMI for Wheelchair Command Technique," *Ada User J.*, vol. 38, no. 2, 2017.
- [21] N. Hossain and M. Nazin, "Emovoice: Finding my mood from my voice signal," in *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, 2018, pp. 1826–1828. DOI: 10.1145/3267305.3277832.
- [22] A. S. S. Kyi and T. Shimamura, "Pre-Emphasis and Linear Prediction Error Filters for Quality Improvement of Bone-Conducted Speech," in *2020 IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE)*, 2020, pp. 481–484. DOI: 10.1109/WIECON-ECE52138.2020.9397998.
- [23] B. H. Prasetio, H. Tamura, and K. Tanno, "Deep time-delay Markov network for prediction and modeling the stress and emotions state transition," *Sci. Rep.*, vol. 10, no. 1, pp. 1–12, 2020. DOI: 10.1038/s41598-020-75155-w.
- [24] F.-N. Landini, "A Pronunciation Scoring System for Second Language Learners," *Univ. Buenos Aires Fac. Ciencias Exactas y Nat. Dep. Comput. Buenos Aires*, 2017.
- [25] D. Prabakaran and S. Sriuppili, "Speech Processing: MFCC Based Feature Extraction Techniques- An Investigation," *J. Physics*, vol. 1717, no. 1, pp. 1-8, 2021. DOI: 10.1088/1742-6596/1717/1/012009.
- [26] A. Sithara, A. Thomas, and D. Mathew, "Study of MFCC and IHC Feature Extraction Methods With Probabilistic Acoustic Models for Speaker Biometric Applications," *Proc. Comput. Sci.* vol. 143, pp. 267-276, 2018. DOI: 10.1016/j.procs.2018.10.395.
- [27] K. K. Sabor, A. Hamou-Lhadj, and A. Larsson, "Durfex: a feature extraction technique for efficient detection of duplicate bug reports," in

- 2017 IEEE international conference on software quality, reliability and security (QRS), 2017, pp. 240–250. DOI: 10.1109/QRS.2017.35.
- [28] J.-S. Bae, H. Bae, Y.-S. Joo, J. Lee, G.-H. Lee, and H.-Y. Cho, “Speaking Speed Control of End-to-End Speech Synthesis using Sentence-Level Conditioning,” *arXiv Prepr. arXiv2007.15281*, 2020.
- [29] B. H. Prasetyo, H. Tamura, and K. Tanno, “Emotional variability analysis based i-vector for speaker verification in under-stress conditions,” *Electronics*, vol. 9, no. 9, p. 1420, 2020. DOI: 10.3390/electronics9091420.
- [30] A. Labied, M.; Belangour, “Automatic Speech Recognition Features Extraction Techniques: A Multi-criteria Comparison,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 8, pp. 177–182, 2021. DOI: 10.14569/IJACSA.2021.0120821.
- [31] B. H. Prasetyo and D. Syauqy, “Design of Speaker Verification using Dynamic Time Warping (DTW) on Graphical Programming for Authentication Process,” *JITeCS (Journal Inf. Technol. Comput. Sci.)*, vol. 2, no. 1, pp. 11–18, 2017. DOI: 10.25126/jitecs.20172124.
- [32] G. Kurniawati and O. Karnalim, “Introducing a practical educational tool for correlating algorithm time complexity with real program execution,” *JITeCS (Journal Inf. Technol. Comput. Sci.)*, vol. 3, no. 1, pp. 1–15, 2018. DOI: 10.25126/jitecs.20183140.