Assessing the Performance of a Speech Recognition System Embedded in Low-Cost Devices

Original Scientific Paper

Fatima Barkani
University Sidi Mohamed Ben Abdellah,
Faculty of Sciences Dhar Mahraz,
Laboratory of Computer Science, Signals,
Automation and Cognitivism
Fez, Morocco
fatimabarkani4@gmail.com

Mohamed Hamidi
Mohammed First University,
Pluridisciplinary Faculty of Nador,
Team of modeling and scientific computing
Oujda, Morocco
mohamed.hamidi.5@gmail

Ouissam Zealouk
University Sidi Mohamed Ben Abdellah,
Faculty of Sciences Dhar Mahraz,
Laboratory of Computer Science, Signals,
Automation and Cognitivism
Fez, Morocco
ouissam.zealouk@gmail.com

Hassan Satori
University Sidi Mohamed Ben Abdellah,
Faculty of Sciences Dhar Mahraz,
Laboratory of Computer Science, Signals,
Automation and Cognitivism
Fez, Morocco
hassan.satori@usmba.ac.ma

Abstract *– The main purpose of this research is to investigate how an Amazigh speech recognition system can be integrated into a low-cost minicomputer, specifically the Raspberry Pi, in order to improve the system's automatic speech recognition capabilities. The study focuses on optimizing system parameters to achieve a balance between performance and limited system resources. To achieve this, the system employs a combination of Hidden Markov Models (HMMs), Gaussian Mixture Models (GMMs), and Mel Frequency Spectral Coefficients (MFCCs) with a speaker-independent approach. The system has been developed to recognize 20 Amazigh words, comprising of 10 commands and the first ten Amazigh digits. The results indicate that the recognition rate achieved on the Raspberry Pi system is 89.16% using 3 HMMs, 16 GMMs, and 39 MFCC coefficients. These findings demonstrate that it is feasible to create effective embedded Amazigh speech recognition systems using a low-cost minicomputer such as the Raspberry Pi. Furthermore, Amazigh linguistic analysis has been implemented to ensure the accuracy of the designed embedded speech system.*

Keywords*: Speech recognition, HMMs, GMMs, Raspberry Pi, Amazigh language*

1. INTRODUCTION

The technology of automatic speech recognition (ASR) enables the transcription of spoken messages and the extraction of their linguistic content, making it applicable to numerous ASR research applications, including various fields such as education, interactive services, messaging, machine and robot control, quality control, data entry, remote access, and more. The Raspberry Pi low-cost minicomputer has also been utilized in studies for speech detection, speaker recognition, user speech interface, and robots [1-2].

Researchers have proposed a speaker recognition method that utilizes Hidden Markov Models (HMMs) and the Google API through a Raspberry Pi board. This method is capable of recognizing and authenticating users, serving as a secure speech recognizer for automated door control, and functioning as a general voice recognizer for controlling various appliances [3]. In another study, a home automation system was developed for motion detection and image capture, utilizing a Raspberry Pi board, a connected camera, and motion sensors [4].

An algorithm for efficient home automation through email on Raspberry Pi was proposed in [5], while a method for remote controlling domestic equipment via an Android application using a Raspberry Pi card was proposed in [6]. Additionally, a speech-controlled system was developed for visually impaired individuals to control computer functions using their voice [7]. In another study, Raspberry Pi was utilized to construct an intelligent control and monitoring system for water

treatment plants [8]. Meanwhile, our lab researchers concentrated on integrating the Amazigh language into ASR systems for diverse applications [9, 10-17]. This research presents an architecture for a low-cost, speaker-independent speech recognition system based on the Amazigh language, implemented on a Raspberry Pi board. The system utilizes Amazigh voice commands to control devices connected to the Raspberry Pi. We have designed an open-source evaluation platform that combines a hybrid model of Hidden Markov Models and Gaussian Mixture Models with the Mel-Frequency Cepstral Coefficients feature extraction technique to determine the optimal values for achieving maximum performance. Our work significantly contributes to the advancement of speech recognition technologies in resource-constrained environments. This, in turn, creates opportunities for enhanced speech recognition capabilities in affordable devices. This paper is organized as follows: In Section 1, an introduction to the topic is presented. Section 2 provides an overview of the architecture and functioning of the automatic speech recognition system. Section 3 discusses the proposed work in detail. In Section 4, a discussion of the findings and their implications is presented. Finally, in Section 5, the conclusion of the study is provided.

2. ASR SYSTEM STRUCTURE

2.1. GENERAL ARCHITECTURE

Automatic Speech Recognition (ASR) is a technology that facilitates the transcription of spoken words into text through pattern recognition techniques, involving feature extraction, pattern matching, and reference model library phases [18], as illustrated in Fig. 1. Our research objective was to develop an ASR system tailored for the Amazigh language.

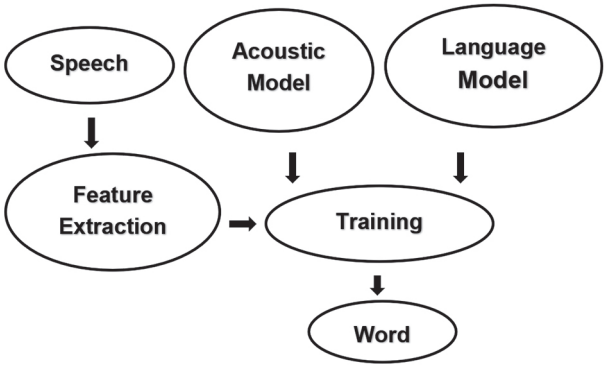

Fig. 1. ASR system architecture

2.2. FEATURE EXTRACTION

The first component of an ASR system is feature extraction, which involves capturing and analyzing speech signals. The system should extract relevant features, such as voice pitch, duration, and energy, from the voice signal. These characteristics are then transformed into a set of numerical values that can be analyzed in more detail. In this work, the feature extraction method used is the MFCC technique [19] with a 16 kHz sampling rate, 16 Kbit sample size, and a 25.6 ms Hamming window.

2.3. HIDDEN MARKOV MODEL

Hidden Markov Models (HMMs) are extensively utilized as a statistical technique for modeling and examining discrete-time series data. Their versatility has made them highly valuable across multiple domains, with a significant presence in the realm of speech processing. Specifically, HMMs have proven to be instrumental in tasks like automatic speech recognition, demonstrating their effectiveness and applicability in capturing the underlying patterns and dynamics of speech signals. The robustness and success of HMMs in these applications have solidified their status as a powerful tool in speech processing research and development and other domains. Fig. 2 presents three states of the Hidden Markov Model topology [19].

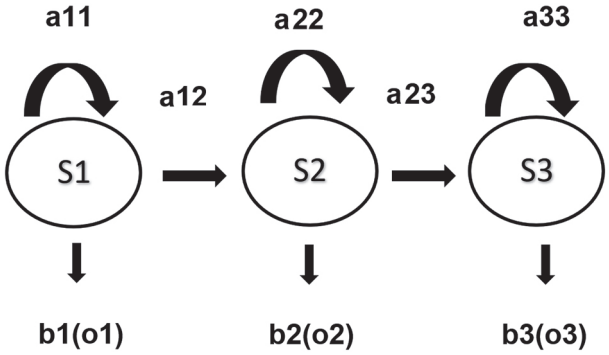

Fig. 2. The 3 states of HMM architecture

2.4. ACOUSTIC MODEL

The acoustic model is an essential element of an ASR system that represents the many sounds and words in a language. It is a statistical model that associates phonemes, syllables, and words with acoustic properties taken from speech signals. The model learns to associate linguistic units with phonemic properties during the training phase. The model is used to match the properties retrieved from the input audio signal to the corresponding linguistic units during the recognition phase. The power and accuracy of the acoustic model greatly affect the accuracy of the ASR system. Hence, one of the major areas of study in acoustics is the construction and improvement of acoustic models. This study employs 3-state and 5-state HMMs to recognize speech data. Additionally, several Gaussian mixture models with 8, 16, and 32 GMMs are used.

2.5. AMAZIGH SPEECH DICTIONARY

The dictionary is a mediator between the Acoustic Model and the Language Model. Our pronunciation dictionary includes all the Amazigh training words along with their corresponding pronunciations. The dictionary used to train our system is presented in Table 1.

Table 1. The used Amazigh Dictionary

AFLLA	AE F L AH
AFOSI	AE F AH S IY
ALNDAD	AE L N D AH D
AMAGGWAJ	AH M AE G W AH JH
ANAKMAR	AH N AE K M AH R
AWAR	AH W AO R
AZLMAD	AE Z L M AH D
DAR	D AA R
DAT	D AE T
DDAW	D AO
ELEM	E L E M
KRAD	K R A D
KOZ	K O Z
SA	S A
SEDISS	S D E SS
SMMUS	S M U S
SIN	S I N
TAM	T A M
TZA	T Z A
YEN	Y E N

Native memory refers to the memory available to a process, such as a Java process, and it is based on physical memory, disks, and other physical devices that are managed by the operating system (OS). The CPU utilizes the memory bus to access the normal memory and also predicts the instructions to execute, storing the results in registers - fast memory components that can hold CPU results. The memories that are accessed depend on the physical address size, which the CPU uses to identify physical memory. For instance, a 16-bit address can access 2^{16} (=65,536) memory locations, while a 32-bit address can access 2^{32} (=4,294,967,296) memory locations. To map the physical memory to the memory that each process can see, an operating system (OS) uses virtual memory [20].

3. THE PROPOSED SYSTEM ARCHITECTURE

The aim of this study is to build an advanced system that can recognize Amazigh speech through voice commands, utilizing a Raspberry Pi. The proposed system architecture is illustrated in Fig. 3.

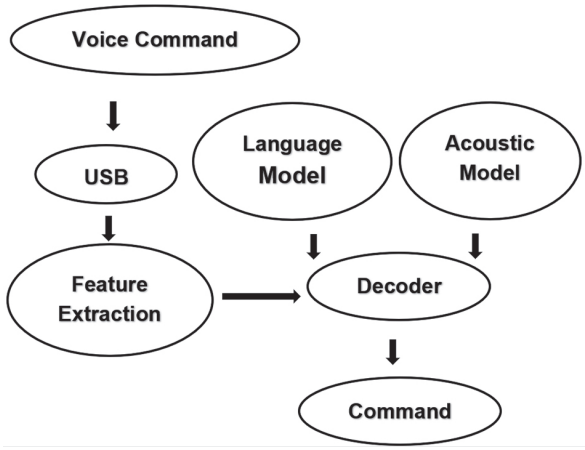

Fig. 3. Proposed System

3.1. THE RASPBERRY PI CARD

Raspberry Pi 3 is a powerful, compact computer that can perform the same tasks as a regular personal computer. In this work, we use a micro SD card compatible with Raspberry Pi, pre-loaded with 1GB of RAM. The design of this system does not include an integrated hard disk or solid-state drive for storage purposes. Instead, it relies on a 16GB SD card to serve as a boot device and internal storage. Since the Raspberry Pi does not have a built-in microphone, we use an external USB microphone in this work.

3.2. CORPUS

The speech data used in this study are unique to the laboratory and have not been previously published or shared. This dataset contains 20 Amazigh words collected from 30 native Tarifit speakers in Morocco. The dataset includes the first ten Amazigh digits (Table 2) and ten other words (Table 3). The speech was recorded using a microphone and the WaveSurfer program and saved in a ".wav" file format. Thirty percent of the data was used for testing, while the remaining 70% was used for training (refer to Table 4 for details).

Table 2. The used Amazigh digits

Amazigh digits	English	Tifinagh	Syllables
ELEM	Zero	ⵖⵎⵏⵍ	VCCV
YEN	One	ⵢⴰⵏ	CVC
SIN	Two	ⵙⵉⵏ	CVC
KRAD	Three	ⴽⵕⴰⴷ	VCCVC
KOZ	Four	ⴽⵓⵣ	CVC
SMMUS	Five	ⵙⵎⵎⵓⵙ	CCCVC
SEDISS	Six	ⵙⴹⵉⵙⵙ	CCVC
SA	Seven	ⵙⴰ	CV
TAM	Eight	ⵜⴰⵎ	CVC
TZA	Nine	ⵜⵣⴰ	CCCV

Table 3. The used Amazigh words

Amazigh Words	English	Tifinagh	Syllables
AFLLA	above	ⴰⵀⵍⵍⴰ	VCCCV
AFOSI	right	ⵀⵓⵙⵉ	VCVCV
ALNDAD	in front of	ⴰⵍⵏⴷⴰⴷ	VCCCVC
AMAGGWAJ	far	ⴰⵎⴰⴳⴳⵯⴰⵊ	VCVCCVC
ANAKMAR	near	ⴰⵏⴰⴽⵎⴰⵔ	VCVCCVC
AWAR	after	ⴰⵡⴰⵔ	VCVC
AZLMAD	left	ⴰⵣⵍⵎⴰⴷ	VCCCVC
DAR	beside	ⴷⴰⵔ	CVC
DAT	before	ⴷⴰⵜ	CVC
DDAW	down	ⴷⴷⴰⵡ	CCVC

Table 4. Corpus characteristics

Recorder type	Number of recorders used for training	Number of recorders used for testing
Amazigh digits	21	9
Amazigh words	7	3

3.3. RESULTS

The designed system was trained and generated using a laptop equipped with an Intel Core i3 CPU running at 2.4 GHz, 4 GB of RAM, and the Ubuntu 14.04 LTS operating system. After training the system, we conducted two test experiments: the first one using the same laptop. Additionally, in the second experiment, we tested the acoustic system on a Raspberry Pi board with 1 GB of RAM and the Raspbian operating system to assess its functionality on a lower-specification device. The data presented in Figs. 4 and 5 illustrate the accuracy of digit recognition achieved by using various GMMs and HMMs on a laptop. The testing corpus subset consisted of 900 tokens for all ten digits. Our results showed that when using 3 HMMs, the system achieved recognition rates of 90.33%, 91.33%, and 88.67% for 8, 16, and 32 GMMs, respectively. Meanwhile, when using 5 HMMs, the system achieved correct rates of 84.11%, 84.22%, and 81.88% for 8, 16, and 32 GMMs, respectively.

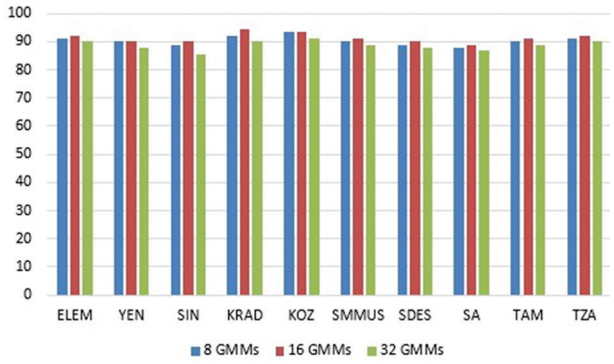

Fig. 4. Digits recognition rates by using a laptop 3 HMMs

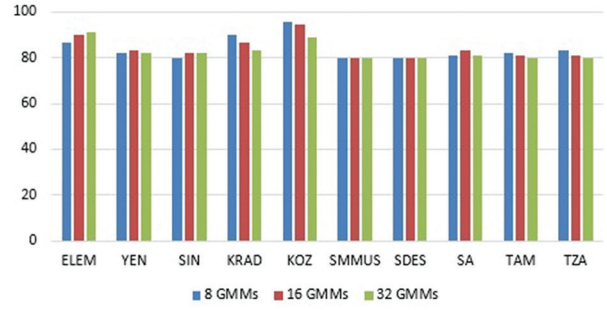

Fig. 5. Digits recognition rates by using a laptop 5 HMMs

The system aims to recognize 300 tokens of 10 Amazigh commands. When using 3 HMMs, the recognition rates achieved were 86.66%, 86.99%, and 85.33% for 8, 16, and 32 GMMs. Figures 6 and 7 present the recognition rates for Amazigh words using various GMMs and HMMs on a laptop. On the other hand, when using 5 HMMs, the system achieved recognition rates of 82.33%, 83.66%, and 83.33% for 8, 16, and 32 GMMs, respectively. The highest recognition rate was achieved using 16 GMMs with 3 HMMs. Based on these results, it can be concluded that the Amazigh digits KRAD and KOZ are the most frequently recognized using 3 and 5 HMMs, respectively. Additionally, the best-recognized Amazigh words using 3 HMMs are AFLLA, AFOSI, AZLMAD, and DDAW, while the best-recognized word using 5 HMMs is DDAW.

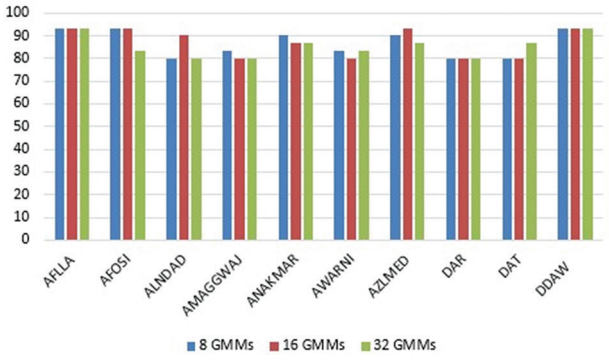

Fig. 6. Commands recognition rates by using a laptop 3 HMMs

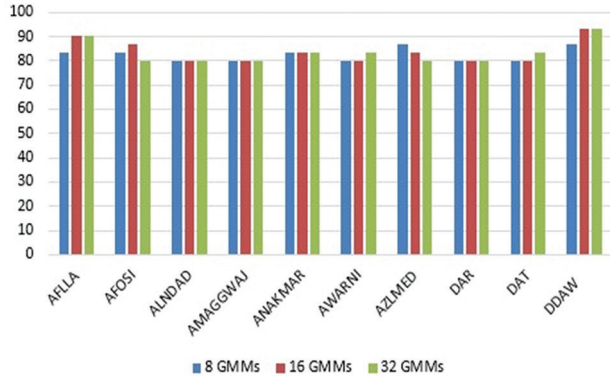

Fig. 7. Commands recognition rates by using a laptop 5 HMMs

Figs. 8 and 9 show the results of digit recognition rates obtained through various GMMs and HMMs on a Raspberry Pi board. The recognition rates obtained for 3 HMMs using 8, 16, and 32 GMMs are 89.00%, 89.67%, and 86.67%, respectively. For 5 HMMs, the recognition rates achieved using 8, 16, and 32 GMMs are 83.66%, 84.11%, and 82.77%, respectively. The highest recognition rate was obtained using 16 GMMs and 3 HMMs.

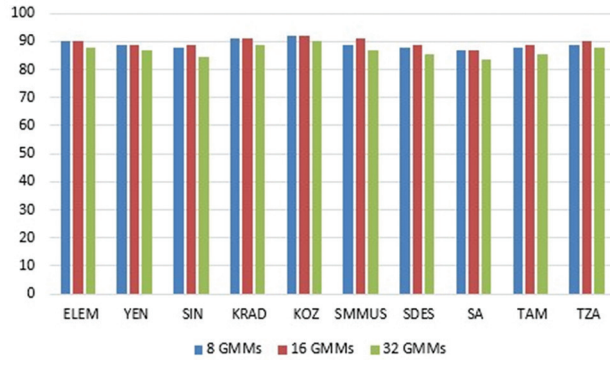

Fig. 8. Raspberry Pi board recognition rates for Amazigh Digits by using 3 HMMs

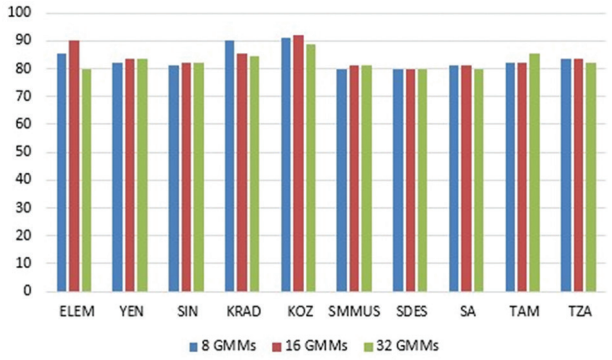

Fig. 9. Raspberry Pi board recognition rates for Amazigh Digits by using 5 HMMs

Figs. 10 and 11 present the recognition rates for Amazigh words achieved through different GMMs using a Raspberry Pi board. For 3 HMMs, the recognition rates obtained were 85.66%, 86.33%, and 83.99% when using 8, 16, and 32 GMMs, respectively. On the other hand, for 5 HMMs, the recognition rates were 82.99%, 82.66%, and 83.33% with 8, 16, and 32 GMMs, respectively. The most optimal recognition rate was achieved by using 16 GMMs with 3 HMMs. The most commonly recognized Amazigh numeral for 3 and 5 HMM is KOZ. For 3 HMM, AFLLA and DDAW are the most frequently recognized Amazigh words, while for 5 HMM, AFFLA and AFOSI are the best recognized Amazigh words. The results indicate that the recognition rates achieved using a Raspberry Pi were lower than those obtained using a laptop.

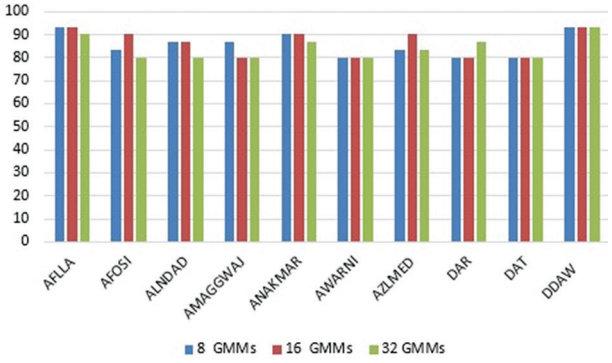

Fig. 10. Raspberry Pi board recognition rates for Amazigh-commands by using 3 HMMs

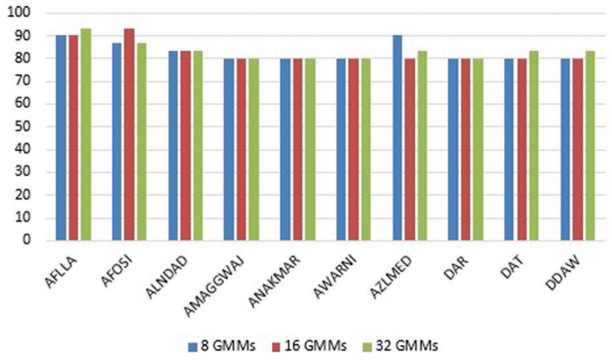

Fig. 11. Raspberry Pi board recognition rates for Amazigh-commands by using 5 HMMs

Figs. 12 and 13 show the memory consumption tests conducted on both a laptop and a Raspberry Pi board for Amazigh commands. It was found that the Raspberry Pi board consumed more memory than the laptop during the tests.

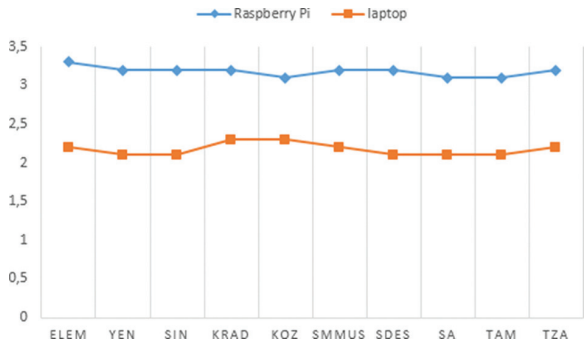

Fig. 12. System memory consumption for digits

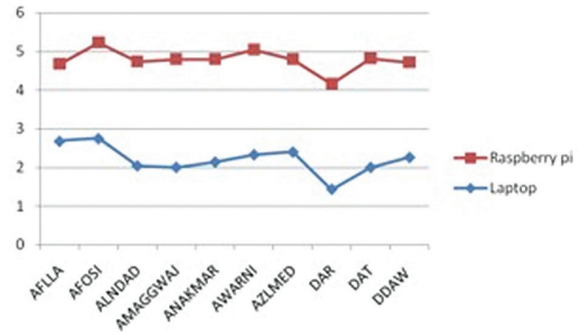

Fig. 13. System memory consumption for Amazigh words

Figs. 14 and 15 illustrate the recognition time for Amazigh digits and words, respectively, using both a laptop and a Raspberry Pi board.

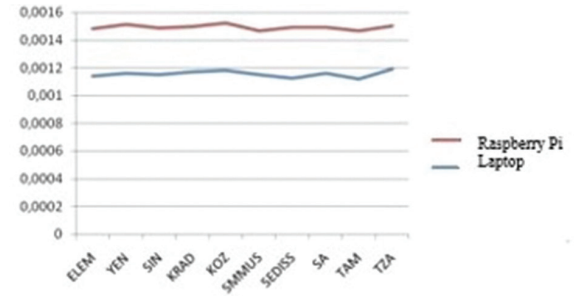

Fig. 14. System time for Amazigh digits

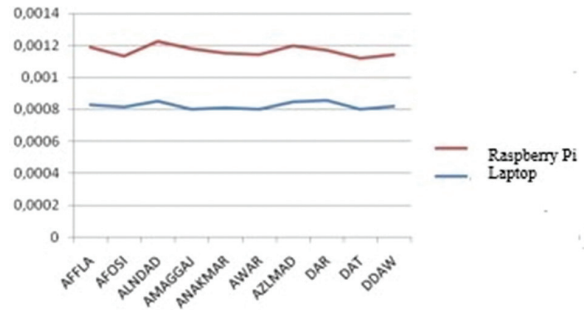

Fig. 15. System time for Amazigh words

The results indicate that the recognition time for the Raspberry Pi board was greater than that for the laptop. These results were expected, given the limitations of the Raspberry Pi hardware. Raspberry Pi has less processing power than a laptop, which can lead to slower and less accurate performance with the ASR system.

We have conducted a comparison of our proposed approach with other existing works. As presented in Table 5, our approach differs significantly from the methods employed by other researchers in several aspects, notably, our accuracy score is 89.16%.

Reference	Year	Methods	Results
[21]	2019	WAT | MFCC | SVM	100%
[22]	2020	CNNs | SVM	95.30% | 72.39%
[23]	2020	PWP | MFCC	99%
[24]	2020	SVM | DTW	97%
[25]	2021	DSP | HMM	80%
Proposed work	2023	GMM | HMM | MFCC	89.16%

4. DISCUSSION

Our tests and analysis show Our tests and analysis show the following:

- The 16 Gaussian mixture distributions yielded the best results.
- The laptop outperforms the Raspberry Pi.
- The laptop has a shorter recognition time compared to the Raspberry Pi.
- The Raspberry Pi board consumes more memory than a laptop.

Based on the results presented in Figs. 12 and 13, it is evident that the number of syllables in Amazigh words significantly impacts memory consumption. The data suggests that when the number of syllables is less than two, memory consumption is lower as well. Most Amazigh commands for computers and Raspberry Pi consist of only one syllable. Analyzing the digit consumption revealed that the KOZ digit consumed the least memory on the Raspberry Pi board, while the YEN and SIN digits consumed the least memory on a laptop. When considering word memory consumption, the DAR word consumed the least memory on both the laptop and the Raspberry Pi board. Therefore, it appears that all Amazigh commands consisting of a single syllable consume less memory. The results displayed in Figures 14 and 15 indicate that the Amazigh language commands "TAM" and "DAT" are recognized more quickly by the system, regardless of whether a computer or a Raspberry Pi is used. These two Amazigh commands have a monosyllabic structure with a consonant-vowel-consonant (CVC) pattern.

5. CONCLUSION

The aim of this research is to investigate the feasibility of utilizing Amazigh speech recognition for controlling external devices. The study involved the creation of an embedded system that utilizes isolated Amazigh words on a Raspberry Pi board. The goal was to develop a portable system that could function effectively within a limited resource environment. To achieve this, the study employed HMMs, GMMs, MFCCs, and parameter optimization to design the speech recognition system. The system was designed to recognize twenty Amazigh words, including ten words and the first ten Amazigh digits, using the open-source CMU Sphinx 4. The findings demonstrated that the optimal performance was achieved with 3 HMMs, 16 GMMs, and 39 MFCC coefficients, resulting in an accuracy rate of 89.16%.

In our future work, our efforts will be directed toward improving the performance of our system through the utilization of hybrid and deep learning techniques. We aim to explore the benefits of combining these approaches to achieve even greater results.

6. REFERENCES:

[1] F. Raffaeli, S. Awad, "Portable low-cost platform for embedded speech analysis and synthesis," Proceedings of the 12th International Computer Engineering Conference (ICENCO), IEEE, 2016, pp. 117-122.

[2] A. Mnassri, M. Bennasr, C. Adnane, "A Robust Feature Extraction Method for Real-time Speech Recognition System on a Raspberry Pi 3 Board," Engineering, Technology & Applied Science Research, Vol. 9, No. 2, 2019, pp. 4066-4070.

[3] S. Suresh, Y. S. Rao, ''Modelling of secured voice recognition based automatic control system". International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE) ISSN, 2015, pp. 0976-1353.

[4] V. Patchava, H. B. Kandala, P. R. Babu, "A smart home automation technique with Raspberry Pi using IoT," Proceedings of the 2015 International Conference on Smart Sensors and Systems (IC-SSS), IEEE, 2015, pp. 1-4.

[5] S. Jain, A. Vaibhav, L. Goyal, "Raspberry Pi based interactive home automation system through E-mail," Proceedings of the 2014 International Conference on Reliability Optimization and Information Technology (ICROIT), IEEE, 2014, pp. 277-280.

[6] H. Lamine, H. Abid, "Remote control of a domestic equipment from an Android application based on Raspberry Pi card," Proceedings of the 15th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), IEEE, 2014, pp. 903-908.

[7] M. S. K. Upadhyay, M. V. N. Chavda, "Intelligent system based on speech recognition with capability of self-learning," International Journal for Technological Research in Engineering, Vol. 1, No. 9, 2014.

[8] S. S. Lagu, S. B. Deshmukh, "Raspberry Pi for automation of water treatment plant", Proceedings of the 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, 2014, pp. 1999-2003.

[9] Y. E. Saady, A. Rachidi, M. Yassa, D. Mammass, "Amhcd: A Database for Amazigh Handwritten Character Recognition Research," International Journal of Computer Applications, Vol. 27, No. 4, 2011, pp. 44-48.

[10] H. Satori, F. ElHaoussi, "Investigation Amazigh speech recognition using CMU tools," International Journal of Speech Technology, Vol. 17, 2014, pp. 235-243.

[11] M. Hamidi, H. Satori, O. Zealouk, K. Satori, "Amazigh digits through interactive speech recognition system in noisy environment," International Journal of Speech Technology, Vol. 23, No. 1, 2020, pp. 101-109.

[12] H. Satori, O. Zealouk, K. Satori, F. ElHaoussi, "Voice comparison between smokers and non-smokers using HMM speech recognition system", International Journal of Speech Technology, Vol. 20, No 4, 2017, pp. 771-777.

[13] M. Hamidi, H. Satori, O. Zealouk, K. Satori, ''Speech coding effect on Amazigh alphabet speech recognition performance", J. Adv. Res. Dyn. Control Syst, Vol. 11, No 2, 2019, pp. 1392-1400.

[14] O. Zealouk, H. Satori, M. Hamidi, N. Laaidi, K. Satori, ''Vocal parameters analysis of smoker using Amazigh language", International Journal of Speech Technology, Vol. 21, 2018, p. 85-91.

[15] O. Zealouk, H. Satori, M. Hamidi, K. Satori, "Voice pathology assessment based on automatic speech recognition using Amazigh digits," Proceedings of the 2nd International Conference on Smart Digital Environment, 2018, pp. 100-105.

[16] M. Hamidi, H. Satori, O. Zealouk, K. Satori, N. Laaidi, "Interactive voice response server voice network administration using hidden Markov model speech recognition system," Proceedings of the Second World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), IEEE, 2018, pp. 16-21.

[17] O. Zealouk, H. Satori, M. Hamidi, K. Satori, ''Speech recognition for Moroccan dialects: feature extraction and classification methods", J. Adv. Res. Dyn. Control Syst, Vol. 11, No 2, 2019, pp. 1401-1408.

[18] X. Zhong, Y. Liang, ''Raspberry Pi: An effective vehicle in teaching the internet of things in computer science and engineering", Electronics, Vol. 5, No 3, 2016, pp. 56.

[19] H. Hermansky, ''Perceptual linear predictive (PLP) analysis of speech", the Journal of the Acoustical Society of America, Vol. 87, No 4, 1990, pp. 1738-1752.

[20] P. Vojtas, J. Stepan, D. Sec, R. Cimler, O. Krejcar, "Voice recognition software on embedded devices". In: Intelligent Information and Database Systems: 10th Asian Conference, ACIIDS 2018, Dong Hoi City, Vietnam, March 19-21, 2018, Proceedings, Part I, Springer International Publishing, 2018, pp. 642-650.

[21] M. Walid, S. Bousselmi, K. Dabbabi, A. Cherif, ''Real-time implementation of isolated-word speech recognition system on raspberry Pi 3 using WAT-MFCC", IJCSNS, Vol. 19, No 3, 2019, pp. 42.

[22] M. S. I. Sharifuddin, S. Nordin, A. M. Ali, ''Comparison of CNNs and SVM for voice control wheelchair". IAES International Journal of Artificial Intelligence, Vol. 9, No 3, 2020, pp. 387.

[23] W. Helali, Z. Hajaiej, A. Cherif, ''Real-time speech recognition based on PWP thresholding and MFCC using SVM", Engineering, Technology & Applied Science Research, Vol. 10, No 5, 2020, pp. 6204-6208.

[24] A. Ismail, S. Abdlerazek, I. M. El-Henawy, ''Development of smart healthcare system based on speech recognition using support vector machine and dynamic time warping", Sustainability, Vol. 12, No 6, 2020, pp. 2403.

[25] A. Abdulkareem, T. E. Somefun, O. K. Chinedum, A. F. Agbetuyi, ''Design and implementation of speech recognition system integrated with internet of things", International Journal of Electrical and Computer Engineering (IJECE), Vol. 11, No 2, 2021, pp. 1796-1803.