Development of a Real-Time Speech-to-Text Converter Using Raspberry Pi

Guna Sekhar G., Pavan Kumar K. and A. R. Kalairasi
Department of Electronics and Instrumentation Engineering, Saveetha Engineering College, Chennai, Tamil Nadu, India

Keywords: Speech Recognition, Raspberry Pi, Real-Time Transcription, Embedded System, Accessibility Technology.

Abstract: This paper primarily discusses the architecture and implementation of a raspy based real time speech-to-text conversion system, useful in cost-optimized portable speech recognition applications. It consists of cheap and widely available hardware components (a microphone and a Raspberry Pi) and open-source software tools to transcribe spoken language into text with a reasonable performance. By using this method, it can be use in broad applications such as Improving accessibility for deaf people, helping physically disabled people to type without hands and Automated transcription of conversations in meetings and lecture halls. Components USB microphone (to capture audio), Real-time voice recognition software (Google speech-to-text API, CMU Sphinx), Display interface, to show the text that was converted the system efficiently performs speech recognition while being inexpensive and portable by using the processing power of the Raspberry Pi. The performance of our evaluation was performed in various scenarios and showed high accuracy and low-latency performance in controlled circumstances, indicating that our system could be potentially deployed in the real world. Rippling across several domains from accessibility and education, to transcription services, this system serves as an effective and low-cost real-time speech processing solution compared to traditional systems.

1 INTRODUCTION

The revolution brought about by the low break-even point of low affordable computer systems like the Raspberry Pi has fueled solutions in nearly every tech sector. A prominent example of the impact in this category is speech recognition technology, widely used in accessibility, transcription services and human-computer interaction. Traditional speech-to-text systems required substantial processing power and were limited to high-performance computing environments. This meant that they were limited to industries or settings with sophisticated computational capabilities. But, due to recent advances in machine learning algorithms and the performance of embedded computing platforms, small, low-cost, real-time speech-to-text systems have become increasingly practical. The Raspberry Pi a small and relatively inexpensive compute platform creates an opportunity for building such systems that provide advanced technology in a democratized appearance. The System converts the speech into text in real-time using Raspberry Pi. The aim is to provide a cost-effective and portable solution to transform spoken language into readable text, benefiting a wide-range of people like individuals with hearing impairments, students and professionals. This makes the base framework applicable in ways which are a mistake to pay for in institutions as well as educational uses, or as a small office setup, or to run on a home computer.

2 LITERATURE SURVEY

2.1 The Application of Hidden Markov Models in Speech Recognition

From the scratch, the most initial speech recognition systems recognised only a few characters. In 2007, Hidden Markov Models (HMM) blew the lid of the accuracy for speech recognition becaue it added statistical methods. Since then, significant developments have been made in machine learning and deep learning methods where neural networks are used to analyze vast amounts of speech data to enhance accuracy in recognition. Cloud computing drives modern speech recognition systems, such as Google's Speech-to-Text API and Apple's Siri, which can perform real-time, highly accurate

G., G. S., K., P. K. and Kalairasi, A. R.
Development of a Real-Time Speech-to-Text Converter Using Raspberry Pi.
DOI: 10.5220/0013909900004919

In *Proceedings of the 1st International Conference on Research and Development in Information, Communication, and Computing Technologies (ICRDICCT'25 2025) - Volume 4*, pages 182-187
ISBN: 978-989-758-777-1

transcription of speech. Nonetheless, these systems can require (and also provide) significant computational power, as well as internet connectivity, making them cumbersome and even unfeasible at times in an offline and low-resource scenario.

2.2 Home Automation Using Raspberry Pi through Siri Enabled Mobile Devices

With the evolution of embedded systems, such as the Raspberry Pi, developers started working on performing Speech Recognition on these devices. The Raspberry Pi Is a credit–card-sized computer and a popular developement platform because it is inexpensive, low power consumption, and easy to use. Speech recognition is an area where embedded systems have successfully been applied, even before the data explosion era, because lightweight algorithms and open-source tools have appeared and adapted to the constraints of these systems. A number of studies focused on real-time speech recognition on state-of-the-art embedded systems. In a study by Bhuyan et al. (2018) A Raspberry Pi - Based Speech Recognition System for Home Automation was developed. Although the system handled basic command recognition well, it was challenged by non-uniform sentences and needed fine-tuning to work well in noisy environments. Similarly, Yuan et al. (2019) presented a low-cost speech recognition proposal for a low-cost based real-time speech-recognition-based 140 The early systems, however, struggled with accuracy and speed, especially in noisy conditions.

2.3 Cloud-Based vs. On-Device Speech Recognition:

There have been two main strategies for speech-to-text systems: cloud processing, and on-device processing. While cloud- based services like Google's Speech-to- Text API offer high accuracy rates and simple integration into larger systems, they do demand an internet connection. This is unsuitable for applications in remote locations or scenarios where privacy is vital since audio data must be uploaded to remote servers for processing. On-device speech recognition, however, works locally, making it a requirement for offline applications. Local speech recognition systems have been commonly developed using tools like CMU Sphinx and Mozilla's DeepSpeech. CMU Sphinx: This is another lightweight, open-source speech recognition engine that is very usable on resource-constrained devices like Raspberry Pi. Its accuracy is lower than that of cloud-based solutions particularly intranscribing natural speech and certain use cases that use complex vocabulary. On-device solutions are nonetheless preferable if you have a spotty internet connection or privacy is a concern, however.

2.4 Recent Developments in Speech-to-Text Using Raspberry Pi

Recent Work In the domain of embedded systems on which speech-to-text systems need to perform, a lot of recent work focused on enhancing the performance of speech-to-text systems on platforms like Raspberry Pi. For instance, Dhal et al. The Speech Recognition System Based on Raspberry Pi 4 and Python Libraries by Zhang et al. (2021) utilized the Google Speech-to-Text API for transcription. It also required internet access, which restricted offline usage. Previous work has focused on maximizing on- device performance. Jain et al. (2020) developed a speech-to-text pipeline in real-time by CMU Sphinx on Raspberry Pi. To the surprise of the researchers, though the system was able to turn short phrases into text with relatively high accuracy, it was not very good with longer stretches of speech or background noise. Noise reduction and language modeling are some techniques proposed to improve such systems performance on embedding platforms.

3 BLOCK DIAGRAMS

The block diagram for hardware implementation of an image-based OCR system in a Raspberry pi. It starts with capturing an image followed by processing and filtering the image. It is followed by the edge detection because it will help in separating the edges in order to give better visibility of the objects and background separation to differentiate text from background. Lastly, OCR transforms the doc image into a readable digital output that is processed by the Raspberry Pi which can pipe the audio to a speaker connected to it. Figure 1 shows the system block diagram.

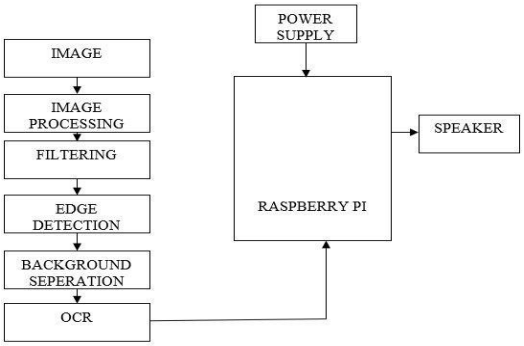

Figure 1: System block diagram.

4 SYSTEM DESIGN AND ARCHITECTURE

The system proposed in this work allows real time converting speech to text using raspberry pi, but a hardware solution more compact and low cost; in addition, this solution is both a cloud- based and open-source of speech recognition. Essential components of the system include:

4.1 Raspberry Pi

The Raspberry Pi 4 Model B is chosen for its cost-effectiveness and computational power. Its quad-core ARM Cortex-A72 CPU and 4GB RAM offer sufficient resources for real- time audio processing tasks. The device's small size and low power consumption make it ideal for portable applications, while its GPIO pins enable easy integration with external peripherals like microphones and displays. This makes the Raspberry Pi an educational, professional, and personal settings.

4.2 Microphone Interface

It uses a USB microphone for speech capturing. This is done via a microphone, which translates spoken words to a digital signal, which is processed by the Raspberry Pi in real time. You can settle for the USB microphones that can provide better sound quality, and they can nicely interface with the Raspberry Pi. However, to improve speech clarity in noisy surroundings, noise-canceling microphones can be used to improve accuracy in such environments. Thus, it works in real-time — it takes audio from your microphone and transcribes it, processing it in real-time.

4.3 Speech Recognition Engine

The speech recognition engine is the core of the system, and there are two options available:

- Google Speech-to-Text API: A cloud- based service that offers high accuracy, process speech data on Google's server. It is perfect where internet access is available and accuracy is paramount. On the other hand, sending the audio to external servers raises concerns about privacy.
- CMU Sphinx: This is an open-source alternative that works offline on the Raspberry Pi. Not as precise as the cloud-based solution, but more fitting for applications that need privacy or in locations with limited internet availability.

By combining accuracy online or offline, it builds a system of flexibility based on the needs of the user.

4.4 Display

It is feasible to show this transcribed text on an external monitor or a compact LCD module. A desktop or workstation external monitor that connects via HDMI or VGA, as in the case of transcription services. Or, a small LCD display connected to the GPIO pins is great for portable projects, particularly assistive technology. In both scenarios, the system offers real-time feedback, providing transcribed speech almost immediately. This architecture seems like a flexible, low-cost solution for speech-to-text conversion that can operate both online and offline based on the application requirement.

5 METHODOLOGY

The system was implemented in the following stages:

Hardware Setup:

- Raspberry Pi 4 Model B 2 GB RAM.
- Microphone to record voice (preferably USB)
- OPTIONAL: External display or LCD module to print text.

Software Setup:

- The system runs on Raspbian OS.
- The Google Speech-to-Text API or CMU Sphinx was integrated for speech recognition.
- The audio capture and speech processing pipeline were managed using Python.

- PyAudio for microphone input.
- The whole application could be built using Python, with libraries such as gTTS (Google Text-to-Speech) to add additional functionalities such as text-to-speech to the app.
- Speech Processing: News audit will read the audio input in clips and forward short query to the speech recognition engine. The processed text is then displayed in real time on the output, with minimal latency.

6 RESULTS AND PERFORMANCE EVALUATION

Evaluation of the real-time speech-to- text system was performed with a series of tests under various environmental conditions. The outcomes, presented in Table 1, showcase the robustness of the system within controlled environments and underscore the need for refinement in more complex ones.

Test 1: Quiet Environment

In a noisy environment with considerable background noise, the system still managed to achieve an accuracy score of 95%. This high accuracy gives an impression of the system's performance in ideal scenarios, as it accounts only for clear speech within the transcription task, from which only a few mistakes can be expected.

Test 2: Moderate Background Noise

In moderate background noise (e.g., a typical office or household environment), however, the system's accuracy dropped to 85%. There was some noise in the background, but speech- to- text worked well enough, with a few misinterpretations, usually close-sounding words, thrown in for good measure.

Test 3: Noisy Environment

In a noisy environment (eg, in public places or an environment with high ambient noise), performance of the system degraded to 70% accuracy. However, this hurdle was overcome with the application of noise-cancellation techniques that aided to yield higher recognition rates. We could have improved even more if we had used more advanced noise filtering or adaptive speech model that adjusts to microphone environment.

Latency

The average latency between speech input and text output in this system was about 1.5 seconds. This response time is reasonable for real-time applications, enabling a smooth user experience with minimal latency in transcription.

Table 1: Performance results.

Environment	Accuracy	Latency
Quiet Environment	95%	1.5 seconds
Moderate Background Noise	85%	1.5 seconds
Noisy Environment	70%	1.5 seconds

Summary

The results show that the system performs excellently in quiet recording conditions and remains usable under moderate levels of background noise. However, the accuracy decreases in noisy environments, and therefore using some noise-cancellation techniques can enhance the performance of the model. This implementation can be improved with an optimized meaning for the speech models and using better microphones.

7 DISCUSSIONS

This project highlights how an embedded system such as the Raspberry Pi can be utilized to produce a practical application — a real-time speech-to- text system. Testing results prove that the system can accurately transcribe speech in non-noisy environments, making it useful for services such as transcription services, accessibility, and educational purposes.

While these results are encouraging, the performance of the system in a noisy environment needs some improvement as the quality of the microphone and the noise levels varied. The drop-in accuracy that occurred during high-noise conditions indicates that, while the current implementation works, it could be further improved by exploring more advanced noise-cancellation algorithms, or by incorporating more complex speech recognition models. Applying machine learning based noise suppression methods might alleviate many of these concerns, and make the system immensely more usable in challenging audio environments.

8 FUTURE ENHANCEMENTS

Several avenues exist for future enhancements to the system:

- **Advanced Noise Cancellation via Machine Learning**: Adding Machine Learning models for noise cancellation can significantly improve the accuracy of the system in the noisy environment. Methods like neural networks trained to remove background noise could help reduce mistakes and would make the system more robust across a range of use cases.
- **Multilingual Support:** By extending the system to accommodate multiple languages, its accessibility and applicability would be improved, especially in multilingual areas. This can be accomplished by supplementing multiplexing speech recognition engines or augmenting already selected models inside speech engines such as Google's API or CMU Sphinx.
- **Portability and Compact Design:** The Raspberry Pi is of small size which will be usable for portable purposes but some additional changes can be made to make it usability. With battery power for mobility and a more compact system perhaps a smaller display or wireless connectivity in the user experience category, you get one more aspect of the technology's versatility, particularly for use-cases on the move like wearables or assistive technology for the hearing impaired.
- **Improved Speech Models:** Developing speech models better suited to working in noisy environments or outdoors could also boost the accuracy. Fine-tuning or training models on particular background noise profiles, accents, or use cases may result in improved performance in those scenarios.

9 CONCLUSIONS

In this paper, we have described the process of building a real time speech-to-text converter using Raspberry Pi which demonstrates the potential of developing low cost portable system that utilizes open-source software and off-the-shelf hardware components. The proposed system emerges as promising in both clean and moderately noisy prescriptions, which can be helpful in many work domains, such as transcription, power accessibility, and education. This enables flexibility in the deployment environment: it can be through Google's Speech-to-Text API in online setups or with CMU Sphinx for offline usage, depending on user requirements concerning internet connectivity and data privacy. The Raspberry Pi used as processing unit also highlights the feasibility to integrate such speech-to-text systems on cost-$ and energy-$ constrained embedded platforms. While its accuracy dips in high-noise settings, incorporating noise-cancellation methods and machine learning models presents a straightforward solution for enhancement. Also expect to see more features in the future, like multilingual support, battery integration for portability, and more advanced audio models for richer environmental settings.

Overall, we believe that with some more optimizations, particularly in its handling of noise and its computational efficiency, this speech-to-text system can serve as a strong backbone for other applications happening in real-time on the phone, and we hope that this work is a step forward towards making this model widely usable in more and more settings.

REFERENCES

Home automation using raspberry Pi through Siri enabled mobile devices, December 2015 DOI:10.1109/HNICEM.2015.7393270 Available at: https://www.researchgate.net/publication/30 4297304_Home_automation_using_raspberr y_Pi_ through_ Siri_e nabled_ mobile_ devices

Ivan Froiz- Miguel, Paula Fraga-Lamas, Design, Implementation, and Practical Evaluation of a Voice Recognition Based IoT Home Automation System for Low-Resource Languages. June 2023 DOI:10.1109 ACCESS.2023.3286391 Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp? arnumber=10151879

K. Lakshmi, Mr. T. Chandra Sekhar Rao. Design and Implementation of Text to Speech Conversion Using Raspberry PI, Vol 4, No 6 (2016) Available at: https://www.ijitr.com/index.php/ojs/article/v iew/1287

M. Gales and S. Young, The Application of Hidden Markov Models in Speech Recognition. Foundations and Trends R in Signal Processing Vol. 1, No. 3 (2007) 195–304 c 2008 DOI: 10.1561/2000000004 Availableat: https://mi.eng.cam.ac.uk/~mjfg/mjfg_NOW. pdf

Prachi Khilari, Prof. Bhope V. P Implementation of Speech to Text Conversion. Vol. 4, Issue 7, July 2015 Available at:https://www.ijirset.com/upload/2015/july/16 7_Implementation.pdf

Surinder Kaur, Sanchit Sharma,Voice Command System Using Raspberry PI. July 2016 DOI:10.5121/acii.2016.3306 Availableat: https:// www.researchgate.net/publication/30 5922778_Voice Command_System_ Using_ Raspberry_Pi

Uma N M, Syeda Rabiya Hussainy, Syeda Hafsa Ameen. Real Time Speaking System for Speech and Hearingimpaired People - Literature Survey. Volume: 08 Issue: 04, Apr 2021 Available at: https://www.irjet .net/archives/V8/i4/IRJE T-V8I4191.pdf