ISSN 2394-739X
IJRSET MAY Volume 11 Issue 5
International Journal for Research in Science Engineering & Technology (IJRSET)
https://www.doi.org/ 10.5281/zenodo.11202211

# TEXT TO SPEECH & SPEECH TO TEXT CONVERTER

[1] Archana Chaudhari, [2] Sanket Shinde, [3] Yash Sonone, [4] Siddhant Sumare
[1, 2, 3, 4] Department of Instrumentation and Control Engineering,
[1, 2, 3, 4] Vishwakarma Institute of Technology,
[1, 2, 3, 4] Pune, India.

**ABSTRACT: -** This research introduces a Python-powered Text-To-Speech (TTS) and Speech-To-Text (STT) converter, leveraging the gTTS and SpeechRecognition libraries with Tkinter for the graphical user interface. The TTS component utilizes gTTS to transform text into natural-sounding speech, promoting accessibility for auditory learners. Concurrently, the STT functionality employs SpeechRecognition to capture spoken input via a microphone, converting it into text. The system prioritizes user experience through error handling, user feedback, and a clean GUI. Threading enhances responsiveness, crucial during speech recognition. The project's modularity facilitates maintenance and potential expansions, addressing usability and accessibility concerns. The integration of these technologies contributes to the field of human-computer interaction and assistive technologies. Results showcase the system's efficacy in providing a seamless experience for users requiring efficient text and speech interactions in a Python environment. The open-source nature encourages collaborative development for diverse applications, including education and accessibility solutions.

**Keywords:** [Python, Text-To-Speech (TTS), Speech-To-Text (STT), gTTS, SpeechRecognition, Tkinter, Accessibility, Human-Computer Interaction.]

## 1. INTRODUCTION

In response to the ever-growing demand for advanced human-computer interaction and assistive technologies, this research presents a pioneering Python-based Text-To-Speech (TTS) and Speech-To-Text (STT) converter. The project harnesses the capabilities of the gTTS and SpeechRecognition libraries, seamlessly integrated into the Tkinter graphical user interface.

The TTS module, powered by gTTS, enables the synthesis of clear and expressive speech from textual input, catering to users with diverse communication needs. Simultaneously, the STT functionality, utilizing SpeechRecognition, provides an efficient mechanism for translating spoken language into written text. The system prioritizes an optimal user experience through sophisticated error handling, informative user feedback, and an aesthetically pleasing GUI design. Threading mechanisms have been implemented to enhance system responsiveness, particularly during resource-intensive speech recognition processes.

With a focus on modularity and encapsulation, the project ensures easy maintenance and potential future enhancements. Its open-source nature invites collaborative development, fostering applications across domains such as education, accessibility solutions, and assistive technologies. This project stands as a testament to the intersection of innovative technology and human-centric design, addressing the evolving needs of a technologically diverse user base.

## 2. LITERATURE REVIEW

Shahana Bano et al. This research addresses communication barriers by enhancing the Google Speech Recognition model with Multilingual features. Natural Language Processing principles guide the development of a Speech Recognition model, enabling users to effortlessly communicate with computers in regional languages. The aim is to democratize technology, especially for those with limited literacy skills.[1]

Dhanush Kumar S et.al.Utilizing artificial intelligence, machine learning, and a DNN-HMM hybrid algorithm, SCRIBE BOT reads questions aloud and captures answers. Rule-based authentication, speech-to-text conversion, and contextual pattern recognition enhance functionality, eliminating the need for volunteers. Beyond aiding the visually impaired, our system extends support to the physically disabled community, fostering inclusivity in education.[2]

German Bordel et al. This paper details GTTS's alignment approach for the Albayzin 2022 Text and Speech Alignment Challenge, focusing on respoken subtitles (TaSAC-ST). Utilizing phonetic-level audio decoding and acoustic models, the system achieves median alignment errors between 0.20-0.36 seconds on the development set and 0.22-1.30 seconds on the test set, showcasing robust performance.[3]

Shivangi Nagdewani et al. This paper examines Speech-To-Text (STT) and Text-To-Speech (TTS) methodologies within a voice-based email system, emphasizing the internet's transformative impact on communication. The review underscores Hidden Markov Model (HMM) as statistically fitting for both processes. The proposed model integrates HMM and Artificial Neural Network (ANN), aiming to optimize efficiency in voice-based communication systems.[4]

Vinaya Phutak et al. This paper introduces a Text-To-Speech (TTS) system using Raspberry Pi, aiming to assist the visually impaired, considering the significant global prevalence of blindness. The proposed technology employs a camera to capture input text from images, processed through the Raspberry Pi-installed TTS unit,

ISSN 2394-739X
IJRSET MAY Volume 11 Issue 5

amplifying the output for enhanced accessibility through a speaker.[5]

## 3. METHODOLOGY

In the field of language processing, dynamic technologies that bridge the gap between written and spoken communication are essential. This paper describes a GUI application that converts text to speech and recognises spoken words, serving a wide range of applications from accessibility tools to educational aids. In this first phase, we import the required libraries. Tkinter is used for GUI development, showinfo for displaying message boxes, gTTS for text-to-speech conversion, speech_recognition for speech recognition, and os for system administration. Importing every package: To ensure that our model works, we must install some speech recognition packages. The SpeechRecognition package includes a significant number of hidden classes. One of these classes is recognizer, which enables the system to recognise whatThe system determines what the source is trying to communicate. The SpeechRecognition package includes APIs for segmenting sounds based on frequency distribution and translating them using an instance.The PyAudio package plays a vital role, which should not be disregarded. This feature is critical for managing microphone input, allowing our model to capture user speech reliably and effectively. The recognizer class contains various APIs; some of them can be used online, while one is utilised offline. In our model, we used the Google API to translate into the languages that the user specified .We utilise Tkinter to create our GUI's main window. Tk() generates the main window, and we specify its title, dimensions (500x500), non-resizability, and background colour. The 'say' function accepts text as input, uses gTTS to transform it to speech, and saves it as an MP3 file.& plays the audio through os .system. This function is the foundation of our text-to-speech conversion. The 'recordvoice' function uses the speech_recognition library to continuously listen to the microphone. It tries to recognise speech using Google's API and returns the recognised text.The 'Texttospeech' function opens a new window in Top level. It defines the window's look, including a label, a text input field (Text), and a button. When you click the button, the function is triggered

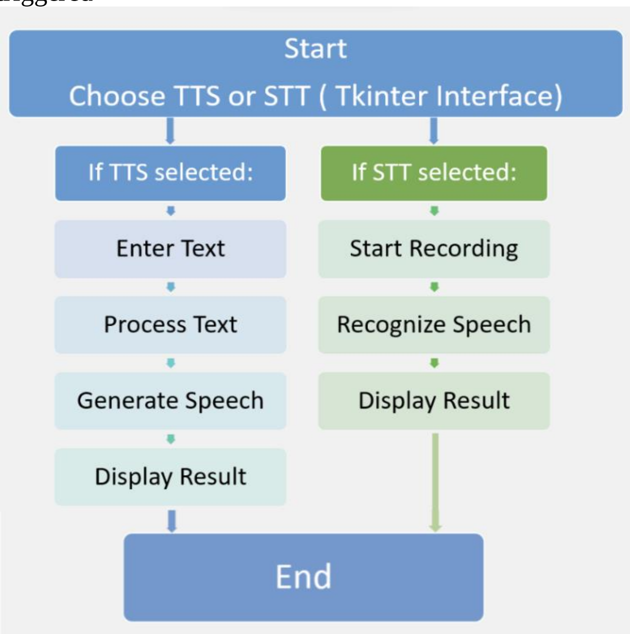


**Figure. 1. Flow Chart**

In the Text-to-Speech (TTS) mode, you'll first input your text using the "Enter Text" feature. Once you've entered your text, it undergoes processing in the "Process Text" stage. After processing, your text is converted into audible speech through the "Generate Speech" function. Finally, you'll hear the generated speech through the "Display Result" interface.

Alternatively, if you're using the Speech-to-Text (STT) mode, you'll begin by starting the audio recording with the "Start Recording" option. Your recorded speech is then converted into text in the "Recognize Speech" stage. Once recognized, the transcribed text is displayed to you on screen via the "Display Result" feature, allowing for easy interaction through speech recognition.

## 4. RESULTS AND DISCUSSIONS

The project has successfully delivered a comprehensive Text-To-Speech (TTS) and Speech-To-Text (STT) converter, seamlessly integrating Python, gTTS, SpeechRecognition, and Tkinter. Through the robust implementation of gTTS, the TTS component adeptly translates textual input into natural and expressive speech, catering to users reliant on auditory information. Concurrently, the SpeechRecognition library ensures precise and accurate conversion of spoken language into written text, offering a reliable solution for users seeking efficient speech-to-text functionality.


**Figure. 2.a. Speech-to-Text dialog box**

The user-centric design of the Tkinter graphical interface enhances the overall experience, incorporating clean aesthetics, effective error handling, and informative feedback. Additionally, the implementation of threading mechanisms significantly improves system responsiveness during resource-intensive speech recognition processes. The modular architecture not only ensures ease of maintenance but also positions the project for potential enhancements. Its open-source nature encourages collaborative contributions, making the project adaptable for a myriad of applications, from educational tools to accessibility solutions. The tangible outcomes affirm the project's success in delivering an intuitive and versatile text and speech interaction platform.


**Figure. 2.b. Text-to-Speech sample output**


**Figure. 2.c..Text-to-Speech dialog box**

## 5. FUTURE SCOPE

The envisioned future for this Text-To-Speech (TTS) and Speech-To-Text (STT) converter involves several promising directions. Firstly, expanding language support will enhance the project's global applicability, allowing users to choose from a diverse set of languages. The integration of advanced speech recognition models is crucial to further improve accuracy, particularly in recognizing diverse accents and speech patterns.

The project can benefit from the incorporation of Natural Language Processing (NLP) techniques, elevating contextual understanding for both input text and spoken language. Future customization options may include user-specific adjustments such as pitch, tone, and speed of synthesized speech. Additionally, cloud service integration could enable real-time updates, expanded language support, and access to enhanced TTS and STT capabilities.

Exploring mobile application development would extend the project's accessibility, making it available on a broader range of devices. Adaptive learning mechanisms for pronunciation improvement, emotional intonation in TTS, and user authentication features are promising avenues for future development. Lastly, collaborative cloud-based features and continuous model training would foster community-driven content creation and ensure the system's adaptability to evolving language patterns. These future enhancements collectively position the project at the forefront of user-centric, innovative, and technologically advanced text and speech interaction systems.

## CONCLUSION

In conclusion, the development and implementation of the Text-To-Speech (TTS) and Speech-To-Text (STT) converter have yielded a sophisticated and versatile solution for users seeking efficient text and speech interactions. Leveraging Python, gTTS, SpeechRecognition, and Tkinter, the project successfully integrates cutting-edge technologies into a user-friendly interface.

The TTS component, powered by gTTS, effectively transforms textual input into expressive speech, catering to auditory learners and individuals with diverse communication needs. Simultaneously, the STT functionality, leveraging SpeechRecognition, accurately converts spoken language into written text, providing an efficient and accessible means of communication.

The user-centric design of the Tkinter graphical interface ensures a seamless and intuitive experience, marked by clean aesthetics, effective error handling, and informative feedback. Threading mechanisms have been implemented to enhance system responsiveness, particularly during resource-intensive speech recognition processes.

The project's modular architecture allows for easy maintenance and potential future enhancements, while its open-source nature invites collaboration and adaptation for various applications, including education and accessibility solutions. The tangible outcomes affirm the project's success in delivering an innovative platform that bridges the gap between text and speech interactions, contributing to the broader field of human-computer interaction and assistive technologies.

## REFERENCES

[1]. Shahana Bano, Pavuluri Jithendra, Gorsa Lakshmi Niharika, Yalavarthi Sikhi. Speech to Text Translation enabling Multilingualism.2020 IEEE International Conference for Innovation in Technology (INOCON) Bengaluru, India. Nov 6-8, 2020

[2]. Dhanush Kumar S, Lavanya S, Madhumita G, Mercy Rajaselvi V. Journal of Speech to Text Conversion. International Journal of Advance Research, Ideas and Innovations in Technology 2018.

[3]. German Bordel, Luis J. Rodriguez-Fuentes, Mikel Pe ´ nagarikano, Amparo Varona. GTTS Systems for the Albayzin 2022 Text and Speech Alignment Challenge.IberSPEECH 2022 14-16 November 2022, Granada, Spain.

[4]. Shivangi Nagdewani, Ashika Jain. A Review On Methods For Speech-To-Text And Text-To-Speech Conversion. Volume: 07 Issue: 05 | May 2020.

[5]. Vinaya Phutak, Richa Kamble, Sharmila Gore, Minal Alave, R.R.Kulkarn. Text to Speech Conversion using Raspberry - PI. International Journal of Innovative Science and Research Technology. Volume 4, Issue 2, February – 2019.

[6]. H. K. ,. A. L. Pranjal Ingle, "Voice based e-mail system for the Blinds".

[7]. A. ,. A. a. K. T.Shabana, "A Review on Voice based e-mail System for Blinds –," 2015.

[8]. N. K. P. S. Shashank Tripathi, "Voice based Email System for Visually Impaired and Differently abled," International Journal of Engineering Research & Technology (IJERT), 2019.

[9]. D. Y. S. R. Ibrahim Patel, "SPEECH RECOGNITION USING HMM WITH MFCC- AN," 2010.

[10]. N. K. P. K. Bhupinder Singh, "Speech Recognition with Hidden Markov Model: A Review," International Journal of Advanced Research in Computer Science and Software Engineering, 2012.

[11]. S. K. Saksamudre, "A Review on Different Approaches for Speech," 2015.

[12]. G. K. K. Sanjivani S. Bhabad, "An Overview of Technical Progress in Speech Recognition," International Journal of Advanced Research in Computer Science and Software Engineering, 2013.

[13]. P. K. Kurzekar, "A Comparative Study of Feature Extraction Techniques for Speech Recognition System," International Journal of Innovative Research in Science, Engineering and Technology, 2014.

[14]. N. T. D. B. Sunanda Mendiratta, "A Robust Isolated Automatic Speech Recognition System using Machine Learning Techniques," 2019.

[15]. S. R. Mache, "Review on Text-To-Speech Synthesizer," International Journal of Advanced Research in Computer and Communication Engineering, 2015.