RESEARCH ARTICLE OPEN ACCESS

# SPEECH TO TEXT SUMMARIZER FOR LECTURE NOTES

S Senthil[1], Sanjay K[2], Chandru K[3] , Kirthik Varshan M[4] , A M Ram[5] , Santhosh S[6]

[1] Lecturer/Computer Engineering, Murugappa Polytechnic, Chennai Email: sennagavel@gmail.com
[2] Computer Engineering, Murugappa Polytechnic, Chennai Email: sanjaykathirvel717@gmail.com
[3] Computer Engineering, Murugappa Polytechnic, Chennai Email: sakthikchandru06@gmail.com
[4] Computer Engineering, Murugappa Polytechnic, Chennai Email: kiruthikvarshancte1@gmail.com
[5] Computer Engineering, Murugappa Polytechnic, Chennai Email: amram99529@gmail.com
[6] Computer Engineering, Murugappa Polytechnic, Chennai Email: s.santhoshcte2mpc@gmail.com

----------------------------------------************************----------------------------------

## Abstract:

The Speech-to-Text Summarizer for Lecture Notes is an innovative AI solution that merges real-time speech recognition with smart NLP summarization. This system takes lectures and turns them into clear, concise summaries using advanced transformer models like BART and T5, along with Python and various speech recognition libraries. With its easy-to-use graphical user interface (GUI), users can effortlessly transcribe, summarize, and even export their notes as PDFs. What sets this system apart from traditional speech-to-text tools is its ability to work both online and offline, automatically pull out key points, and reduce background noise for better accuracy. It's all about enhancing learning accessibility and productivity by bridging the gap between old-school note-taking and the power of AI automation.

**Keywords:** Speech-to-Text Summarizer featuring a user-friendly GUI for transcription, summarization, PDF export, key point extraction, noise reduction, and AI-driven real-time speech recognition, NLP-based summarization, Python, and transformer models (BART/T5)

----------------------------------------************************----------------------------------

## I.NTRODUCTION

In our fast-paced learning world, it can be tough to keep up with all the important points during lectures. That's where the Speech-to-Text Summarizer for Lecture Notes comes in! This tool uses AI and natural language processing to give you real-time transcriptions and neat summaries, making your note-taking a breeze. By combining speech recognition with smart summarization, it turns spoken words into organized notes. Plus, it has a super user-friendly interface that makes transcription, summarization, and exporting to PDF a piece of cake. Whether you're online or offline, it's designed for flexibility and easy access. With features like noise reduction and speech adaptation, it ensures you get accurate results no matter where you are, making your learning experience smoother and more accessible.

## OBJECTIVE

The main goal of this project is to create a real-time speech recognition system that can accurately transcribe lectures. We're also looking to implement a summarization model based on natural language processing (NLP) to produce clear and concise summaries from longer transcriptions. To make things easier for users, the system will have a user-friendly interface that promotes easy interaction and readability. Plus, it will include features for exporting and saving, like the option to generate PDFs for convenient storage and retrieval of lecture notes. Designed to work both online and offline, this system will be accessible no matter if you have internet access or not. A major focus is on ensuring robustness and accuracy in various audio settings, accommodating different accents, speech speeds, and background noise levels to really optimize performance.

## II. SYSTEM ANALYSIS

### Existing System

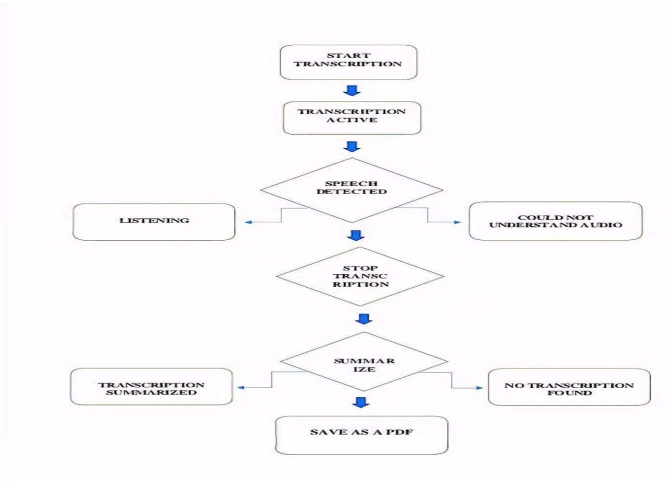
These days, most speech-to-text solutions mainly churn out raw transcriptions without any summarization features. Sure, they do a great job of turning spoken words into text, but they fall short when it comes to helping users pull out the key points from long transcripts. This leaves people with the boring task of sifting through tons of text to find the important bits, which can be a real hassle, especially in academic settings where lectures can cover a lot of ground. On top of that, many of these speech-to-text apps depend heavily on having a solid internet connection to process audio and create transcripts. This reliance can really limit their usefulness in offline situations, making it tough for students and professionals to transcribe and summarize lectures in places where internet access is spotty or nonexistent. In settings like remote classrooms, conference halls, or areas with shaky network coverage, this drawback can stop users from making the most of these tools when they need them the most. So, the absence of built-in summarization and offline capabilities in current solutions really points to the need for a more advanced and flexible speech-to-text summarization tool that can make the process smoother and ensure accessibility no matter the internet situation.

### Proposed System

The system we're proposing takes traditional speech-to-text technology up a notch by adding advanced real-time speech recognition and a Natural Language Processing (NLP)-based summarization model. This means it can not only transcribe spoken words accurately but also create concise, meaningful summaries on the fly. Unlike older methods that force users to wade through long lecture transcripts, this system makes things easier by highlighting key points in a clear and organized way. The user interface is designed to be user-friendly and interactive, allowing users to see both the live transcription and the summarized content simultaneously. Plus, it has a handy feature that lets you export your summarized notes as a PDF, making it simple to store, retrieve, and share lecture notes later on. One of the standout features is its ability to work both online and offline, so you can rely on it even in places with spotty internet. This makes it a fantastic tool for students and professionals who want a smooth note-taking experience in various environments, whether in classrooms, conference rooms, or remote areas. By automating the transcription and summarization processes, the system saves a ton of time and effort compared to manual note-taking, ultimately boosting productivity and enhancing the overall learning experience. With its real-time processing, offline capabilities, and summarization features, it's a practical and efficient solution for anyone looking to effectively capture and manage lecture content.

## III. SYSTEM ARCHITECTURE

The Speech-to-Text Summarizer for Lecture Notes is built on a smart, three-layered system architecture that ensures everything runs smoothly. These layers include the Input Layer, Processing Layer, and Output Layer, each playing a vital role in making speech recognition, summarization, and note management a breeze. Starting with the Input Layer, this is where users interact with the system. It features a user-friendly interface with essential buttons like Start Transcription, Stop, Summarize, and Save as PDF. These controls make it super easy for anyone to kick off, manage, and wrap up the transcription process without needing any technical know-how. The built-in microphone captures live speech in real-time, making it accessible for both students and professionals. By keeping the interface simple yet effective, the system removes any unnecessary complexity, making it adaptable for various educational and professional environments. Now, let’s dive into the heart of the system—the Processing Layer. This is where the magic happens, with speech recognition and summarization taking place. It consists of several modules that work together to ensure everything is accurate and efficient. The Speech Recognition Module uses cutting-edge technologies like Google Speech API, Vosk, or Whisper to turn spoken words into text with impressive precision. These tools are chosen for their ability to handle different accents, speech speeds, and even background noise, ensuring that transcriptions are spot-on in various settings. If the speech is unclear or can’t be recognized, the system keeps users in the loop with a message saying "Could Not Understand Audio," adding a layer of transparency to the process. Once the speech is transcribed, it gets processed and organized in the Text Processing Module, setting the stage for the next important step: the NLP Summarization Module, which is powered by advanced algorithms.




# IV. MODULES

## A. Speech Recognition

The Speech Recognition module is designed to convert spoken language into text in real-time, making it perfect for accurately transcribing lectures and discussions. It leverages the SpeechRecognition library to connect with various engines such as Google Speech API, Vosk, or Whisper, while a built-in microphone captures and processes live audio. To enhance clarity, it employs noise adjustment techniques that minimize background noise, and it has error-handling features to tackle issues like unclear speech, interruptions, and API failures. If the online API encounters a problem, the system smoothly transitions to an offline model like Vosk, ensuring that transcription continues without a hitch. The transcribed text appears instantly on the GUI, adapting to different speech speeds and pauses. With its real-time processing, noise reduction capabilities, and offline compatibility, this module offers a smooth and efficient transcription experience for both students and professionals.

## B. Summarization

The summarization module is all about turning long speech transcripts into short, easy-to-digest summaries. This is super helpful for anyone who wants to get the main points without wading through tons of text. It uses cutting-edge Natural Language Processing (NLP) models like BART and T5 to automatically pull out the most important insights. Before diving into summarization, the system does a thorough cleanup of the text, making sure everything is neat and organized to boost accuracy and clarity. This means getting rid of filler words, fixing punctuation, and standardizing how sentences are structured. The module also employs techniques for extracting and compressing sentences, so it keeps the vital information while cutting out any fluff, ensuring the summary is both informative and straight to the point. Plus, the final output is formatted for easy reading, so users can quickly catch the main ideas. By running the transcription through an NLP pipeline, the system effectively condenses the most significant details while keeping everything logical and coherent. The end result is a well-organized summary that helps users review key points without missing any important context. This method not only makes information more accessible but also simplifies note-taking, making it a fantastic resource for students, professionals, and anyone who deals with a lot of spoken content.

### *C.* Graphical User Interface (GUI)

The Graphical User Interface (GUI) is platform for managing live speech transcriptions and summaries, making the experience smooth and enjoyable for users. It’s built with Tkinter, a popular Python library for creating graphical interfaces, which means it offers a responsive and intuitive environment. This allows users to easily control the transcription and summarization process. As speech is recognized, the GUI dynamically shows real-time transcriptions, so users can keep an eye on the conversion as it unfolds. To make things even easier, the interface comes with essential control buttons that let users start and stop speech recognition with ease, clear transcribed text when necessary, and export the final transcript or summary to a file for future reference. Moreover, the interface includes customization options, enabling users to tweak the font size for better readability and adjust the summary length according to their needs—whether they want a quick outline or a more detailed summary. This adaptability makes the tool perfect for various scenarios, like taking lecture notes, transcribing meetings, or summarizing content. The GUI also has a feature that allows users to turn summarization on or off, giving them the flexibility to either review full transcripts or focus on the key points. Navigation is designed to be straightforward, ensuring that even those with minimal technical skills can use it without any hassle. In addition to displaying and managing live transcriptions, the module makes exporting a breeze, allowing users to save or export their transcripts and summaries in different formats, like text files or PDFs. This means that the processed information can be easily stored, shared, or referred to later. Overall, the GUI is designed with efficiency, clarity, and user-friendliness in mind, making it a powerful tool for real-time speech-to-text applications in various professional and educational settings.

### *D.* Save as a PDF and Export

The "Save as a PDF and Export" feature makes it super easy for users to store, retrieve, and export their transcriptions and summaries whenever they need them. It uses local file storage to keep transcriptions in either text or JSON format and leverages the FPDF library to create nicely formatted PDFs. Plus, the auto-save function kicks in to prevent any data loss by regularly saving your transcription progress, so you won’t lose anything if something interrupts you. On top of that, the user data management system lets you name, categorize, and organize your saved transcriptions for quick access. This module automatically saves both the complete transcription and the summarized version, converts them into PDF format, and gives users the flexibility to open, edit, and manage their saved transcripts with ease.

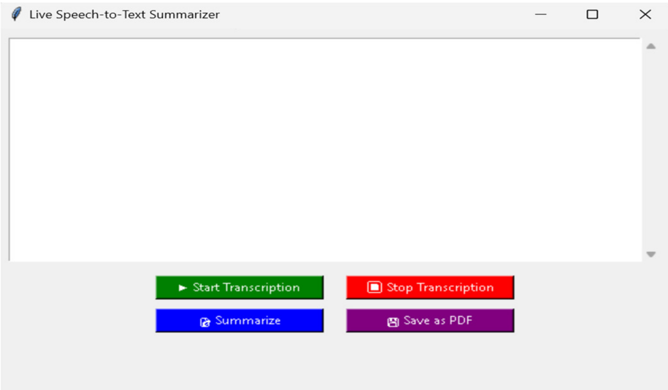
### *E.* API and Backend Processing

The API and Backend Processing Module ensures seamless communication between components of the speech-to-text summarization system. Built on a Flask-based REST API, it efficiently handles speech recognition and summarization requests, linking the frontend with core processing functions. A key feature is server-side speech processing, which offloads audio-to-text conversion to a remote server, reducing the computational burden on the user's device and enabling smooth handling of large transcriptions. To enhance reliability, error handling mechanisms address API failures, network disruptions, and processing errors, ensuring an uninterrupted user experience. The module manages audio input, processes requests, and sends the final output to the frontend in real time while ensuring secure and optimized API communication. This scalable and efficient backend infrastructure makes the system responsive and reliable for various real-world applications.

## PARTIAL OUTPUT

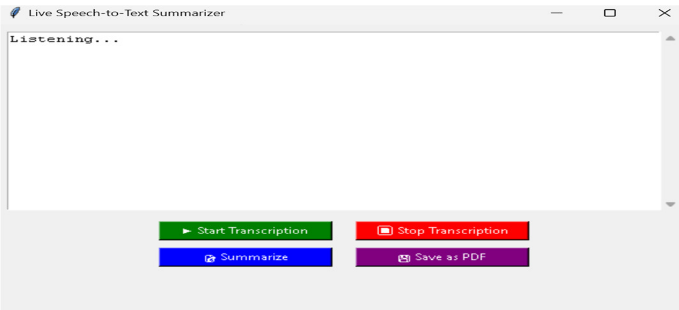
**STEP 1 :**

The user interface of **"Speech to Text Summarizer for Lecture Notes "**



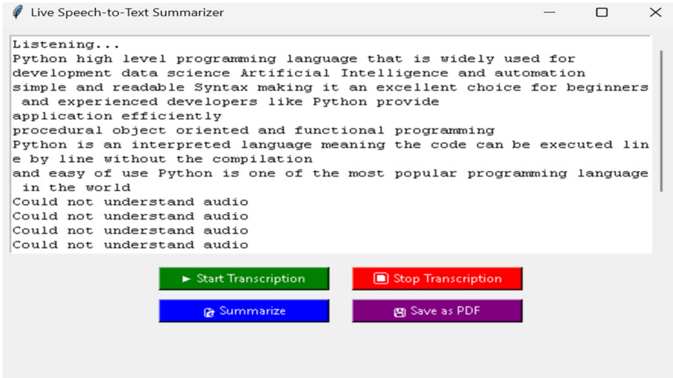

**STEP 2 :**

Click the " **Start Transcription** " in the user interface.



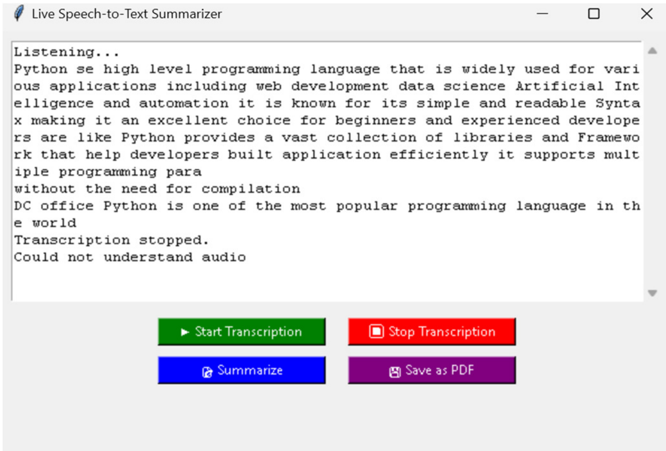

**STEP 3 :**

Start your Speech using **"Start Transcription Button "**



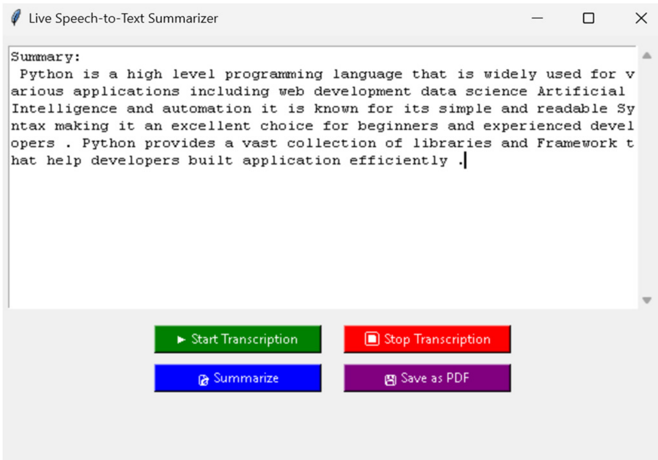

**STEP 4:**

Stop the Speech using **"Stop Transcription Button "**



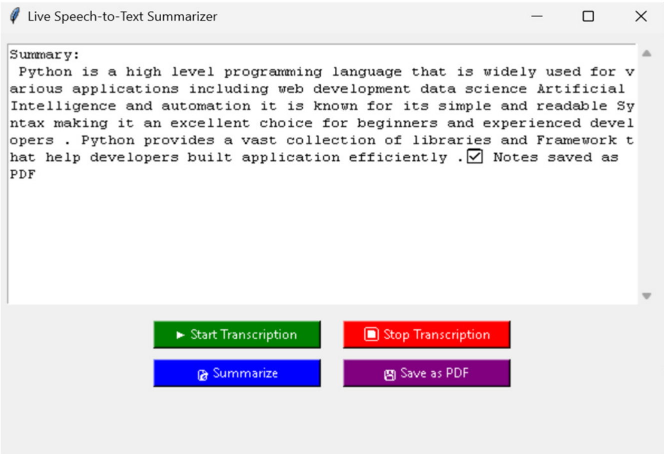

**STEP 5 :**

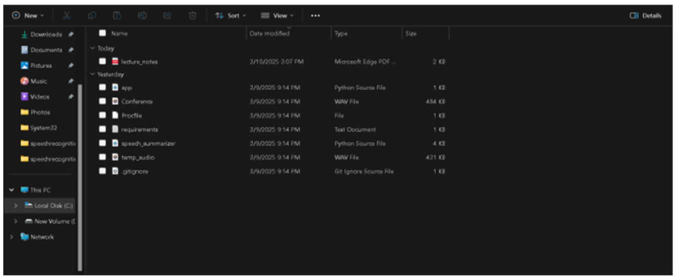
Summarize the speech using **"Summarize Button"**




**STEP 6 :**

Now Click the " **Save as PDF Button** "



## VI. CONCLUSION

The Speech-to-Text Summarizer for Lecture Notes is a thoughtfully designed system that brings together several key modules, each one playing an important part in making it efficient and user-friendly. The Speech Recognition Module does a fantastic job of transcribing spoken words in real-time, so users can easily capture lectures and discussions without a hitch. Then there's the Summarization Module, which takes those transcriptions and uses advanced NLP techniques to create clear and concise summaries, helping users quickly grasp the main points without wading through lengthy texts. The Graphical User Interface (GUI) Module offers an intuitive and interactive experience, featuring real-time text display, customization options, and straightforward navigation. Plus, the Save as a PDF and Export Module allows users to store, retrieve, and manage their transcriptions in an organized way, ensuring everything is accessible and safe from data loss.

Finally, the API and Backend Processing Module keeps everything running smoothly by managing API calls, data flow, and efficiently processing requests between the various components. Together, these modules form a powerful, AI-driven solution that streamlines the note-taking process, boosts accessibility, and enhances productivity, making it an invaluable resource for students, professionals, and anyone in need of effective lecture transcription and summarization.

## VII. ACKNOWLEDGEMENT

We express our gratitude to Murugappa Polytechnic College for providing resources and support for this study. Special thanks to our project guide, S.Senthil for his guidance throughout the research. Additionally, we appreciate the contributions of experts in **speech recognition and natural language processing** who provided critical insights into dataset preparation and model evaluation. Their support and expertise have been instrumental in enhancing the effectiveness and accuracy of our project, **"Speech-to-Text Summarizer for Lecture Notes."**

## VIII. REFERENCES

**[1] Devlin, J., Chang, M., Lee, K., & Toutanova, K.** (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. ACL.

**[2]Lewis, M., et al.** (2020). *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension*. ACL.

**[3]Raffel, C., et al.** (2020). *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer (T5)*. JMLR.

**[4]OpenAI** (2021). *Whisper: Robust Speech Recognition System*. OpenAI Research.

**[5]Python Software Foundation**. (2023).