**SPEECH TRANSCRIBER**

# Mini Project Report

Submitted in partial fulfilment of the requirement of University of Mumbai

For the Degree of

**Bachelor of Engineering**

|  |  |
| --- | --- |
| By |  |
| **1. Mayur Giri** |

### Index

TABLE OF CONTENTS

**Sr. No. Title Page No.**

**7**

**Abstract**

**List of Figures 8**

**List of Tables** 9

Chapter 1 Introduction 10

1.1 Motivation

1.1.1 Need of the Project

1.2 Scope of the project

1.3 Aim

Chapter 2 Problem Statement 12

2.1 Problem statement

2.2 Features

2.3 Objectives

Chapter 3 Literature Survey 13

Chapter 4 Design and Implementation 16

4.1 Software Requirements

4.2 Explanation of Flowchart

Chapter 5 Results & Discussion 19

Chapter 6 Conclusion and Future Work 23

6.1 Conclusion

6.2 Future Scope

References 24

## Abstract

The Speech-to-Text Extractor Project provides an efficient solution for automatically converting spoken content from video files into accurate text transcriptions. With the growing demand for accessible video content and text-based analytics, this project implements a streamlined workflow that bridges the gap between multimedia content and textual data processing.

The system architecture follows a clear sequential process, beginning with video uploading through a user-friendly web interface. The frontend handles file transfers asynchronously via AJAX technology, while a Flask-based backend manages the secure storage and processing of uploaded content. The core functionality leverages ffmpeg for precise audio extraction, followed by specialized audio formatting that optimizes the sound data for speech recognition.

A key technical component is the integration with OpenAI's Whisper API, which applies advanced machine learning algorithms to transform audio signals into text. The system processes the API's JSON responses to extract high-quality transcriptions that maintain the semantic integrity of the original spoken content. Users can view the transcribed text directly in the application and download it as a portable text file for further use.

This project addresses significant challenges in video content management, including the processing of large files, maintaining audio quality through format standardization, and ensuring accurate speech recognition across various accents and speaking patterns. The solution is particularly valuable for content creators, educational institutions, media organizations, and accessibility services that require efficient methods to transform spoken content into searchable, analysable text.

By automating the transcription process, this system significantly reduces the manual effort traditionally required for transcribing video content, improves content accessibility, and enables new possibilities for content analysis and knowledge extraction from video-based information sources.

## List of Tables

Sr. No. Name of Table Page No.

1. 3.1 Literature Survey 13

## List of Figures

Sr. No. Name of Figure Page No.

1. 4.1 Flowchart 17

1. 5.1 Transcribe Text 19

### Chapter 1 Introduction

**1.1. Motivation:**

The motivation behind the Speech-to-Text Extractor project stems from the growing volume of video content being created and consumed across various platforms. Videos contain valuable information that often needs to be accessed in text format for searchability, analysis, and accessibility purposes. Traditional methods of transcription are time-consuming, labour-intensive, and costly, creating a significant barrier to making video content more accessible. With the exponential growth of multimedia content, there is an urgent need for automated solutions that can efficiently convert spoken words into text without extensive manual intervention. The advancement in speech recognition technologies, particularly cloud-based AI services like Google Speech Recognition API, presents an opportunity to develop systems that can accurately transcribe speech from videos with minimal human effort. By leveraging these technologies, we can create tools that democratize access to information locked within video content, enabling better indexing, searching, and analysis of spoken information across educational, business, and entertainment sectors.

**1.1.1 Need of the problem:**

The need for an efficient Speech-to-Text Extractor arises from several critical challenges in managing and utilizing video content:

Accessibility is a primary concern, as individuals with hearing impairments require text alternatives to access video content. Without proper transcriptions, a significant portion of the global population is excluded from valuable information and entertainment. Additionally, modern content strategy demands that videos be searchable and indexable, which is only possible when the spoken content is available in text format. Search engines cannot effectively crawl audio content, making videos without transcriptions essentially invisible in search results for specific spoken terms.

Content creators and educational institutions frequently need to repurpose video content into other formats such as articles, study materials, or documentation. Manual transcription creates a significant bottleneck in this workflow. Furthermore, in sectors like legal, healthcare, and business, accurately documenting spoken communications from recordings is essential for compliance, record-keeping, and knowledge management purposes.

The manual transcription process is not only time-consuming but also prone to errors and inconsistencies. A solution that automates this process while maintaining high accuracy would address a significant pain point across multiple industries and use cases.

**1.2. Scope of the project:**

The Speech-to-Text Extractor project encompasses the development of a comprehensive system for converting spoken content in videos to text format through an intuitive web-based interface. The scope includes:

1. Creating a user-friendly frontend interface for video uploads and transcript display
2. Developing a robust backend system using Flask to handle video processing and storage
3. Implementing video processing modules that extract high-quality audio from uploaded videos
4. Building audio formatting functionality to optimize sound data for speech recognition
5. Integrating with Google Speech Recognition API for accurate transcription services
6. Providing transcript editing capabilities within the web interface
7. Enabling transcript export in various formats, primarily .txt files
8. Ensuring the system can handle videos of various formats, lengths, and quality levels
9. Implementing proper error handling for issues such as upload failures or API timeouts
10. Optimizing the system for processing efficiency, particularly for larger video files

The project does not include real-time transcription of ongoing recordings or multilanguage translation capabilities in its initial phase, though these represent potential areas for future expansion.

**1.3 Aim:**

The Speech-to-Text Extractor project aims to develop an efficient system that automatically converts video speech content into accurate text transcriptions, eliminating manual transcription work while making video content more accessible and searchable.

### Chapter 2 Problem Statement

**2.1 Problem statement:**

Video content contains valuable spoken information that remains inaccessible, unsearchable, and difficult to analyse in its audio format. Manual transcription is timeconsuming, expensive, and prone to errors, creating a significant barrier to transforming spoken content into usable text. The Speech-to-Text Extractor project addresses these challenges by developing an automated system that efficiently extracts and transcribes audio from videos, making the content accessible, searchable, and available for further processing.

**2.2 Features**:

* Streamlined Video Upload Interface
* Automated Audio Extraction and Processing
* High-Quality Speech Recognition
* Interactive Transcript Display
* Text File Export Functionality
* Format Standardization
* Error Handling and Recovery

**2.3 Objectives:**

* Develop an End-to-End Transcription System: Create a complete pipeline from video upload to text download that automates the entire transcription process with minimal user intervention.
* Optimize Audio Processing: Implement high-quality audio extraction and formatting procedures that prepare sound data optimally for speech recognition accuracy.
* Ensure Usability and Accessibility: Design an intuitive interface that allows users of various technical backgrounds to easily upload videos and obtain accurate transcriptions.
* Maximize Transcription Accuracy: Leverage advanced speech recognition technology to achieve the highest possible accuracy in converting spoken content to text across different speech patterns and contexts.

### Chapter 3 Literature Survey

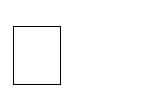
|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Paper Details Authors,  “Title”, Resource Name, Year, page nos. | Problem Addressed | Methodology Used |
| 1. | Authors: Sai Teja  Ramacharla, Vustepalle  Aniketh, Dr. M. Senthil  Kumaran    Title: Speech to Text  Transcription  Resource: IEEE  explore  Page No.: 09 | * Existing systems require manual file conversion, which is time-consuming and inefficient. * Lack of integration between different tools for processing audio and video. | * Developed a webbased application using Flask (Python) for user interaction. * Backend built with: * Speech Recognition   library for real-time transcription.   * PyDub for converting MP3 to WAV. |
| 2. | Authors: Jarrah Sladek,  Andrew Zschorn, Ahmad  Hashemi-Sakhtsari  Title: Speech-to-Text  Transcription in Support of  Pervasive Computing  Resource: IEEE explore  Year: 2003  Pg No.: 16 | Difficulty in capturing, transcribing, and organizing  spoken communication in collaborative environments like meetings and interviews. | AuTM (Automatic  Transcriber of Meetings)  Dragon NaturallySpeaking (DNS) |
| 3. | Authors: Eric Benhaim,  Hichem Sahbi, Guillaume  Vitte  Title: Continuous Visual  Speech Recognition for  Audio Speech Enhancement  Resource: IEEE explore Year:2015, Pg No.:08 | Traditional audio-only speech enhancement methods perform poorly in noisy environments. There is a lack of efficient frameworks to combine visual and audio information | Visual Speech Recognition (VSR)  Audio Speech Enhancement |

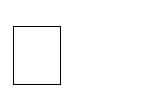
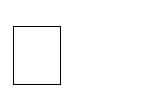
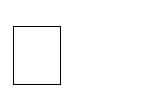
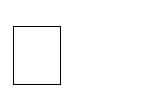
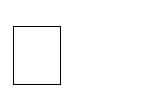
|  |  |  |  |
| --- | --- | --- | --- |
| 4. | Authors: Piotr Kozierski,  Talar Sadalla, Szymon  Drgas, Adam Dąbrowski,  Joanna Ziętkiewicz  Title: The Impact of  Vocabulary Size and Language Model Order on the Polish Whispery Speech  Recognition  Year:2017, Pg no.:06 | Traditional speech recognizers struggle with whispered speech. Lack of Polish whispered speech corpora.  Need to evaluate effect of vocabulary size and language model order on ASR accuracy. | Created a new whisper speech dataset.  Used Kaldi toolkit with MFCC features.  Language models trained with SRILM and tested with 5k to 300k word vocabulary sizes and 1st to 4th order  LM. |
| 5. | Authors: Guoyun Lv, Dongmei Jiang, Rongchun Zhao, Xiaoyue Jiang, H.  Sahli  Title: Multi-Stream  Asynchrony Dynamic  Bayesian Network Model for  Audio-Visual Continuous Speech Recognition Year: 2023. | Traditional models (HMM, MSHMM) don’t effectively handle asynchrony between audio and visual speech. Need for better modeling to improve recognition accuracy, especially in noisy environments. | Proposed the MS-ADBN model, a multi-stream DBN where audio and visual streams are synchronized only at word level, not phone level. Used Gaussian Mixture Models (GMMs)  and tested on a digit audiovisual database. Visual features extracted using Bayesian Tangent Shape Model. |
| 6. | Authors: Lantian Li, Dong  Wang, Chenhao Zhang,  Thomas Fang Zheng  Title: Improving Short  Utterance Speaker  Recognition by Modeling Speech Unit Classes  Year:2015, Pg no.:06 | Short utterances in speaker recognition systems cause performance degradation due to mismatched model priors. | • Proposed a subregion modeling approach using speech unit classes derived through both knowledge-based and data-driven clustering. |
| 7. | Authors: L.R. Bahl, P.V. de  Souza, P.S. Gopalakrishnan,  D. Nahamoo, M.A. Picheny  Title: A Fast Match for  Continuous Speech Recognition Using  Allophonic Models  Resource: IEEE explore, Pg  No.: 27 | Traditional phonetic models for large vocabulary continuous speech recognition are too slow for real-time use and lack the needed accuracy in continuous speech due to context variation. | Replaces phonetic models with allophonic models using decision trees. Built treestructured word models. Uses vector quantization. |
| 8. | Authors: Vinnarasu A., Deepa  V. Jose  Title: Speech to Text  Conversion and  Summarization for Effective  Understanding and  Documentation  Resource: Int. Journal of  Electrical and Computer Engineering (IJECE), Year:  2021 | Manual note-taking from long speeches or lectures is timeconsuming and prone to loss of key information. Speech recognition often lacks sentence boundary clarity, affecting summarization quality. | Speech-to-text conversion using Google API with custom logic for sentence boundaries (adding periods and question marks). Summarization done using NLTK and frequency-based sentence ranking. Comparisons made with Gensim library for performance. |

*Table 3.1 Literature Survey*

### Chapter 4 Design and Implementation

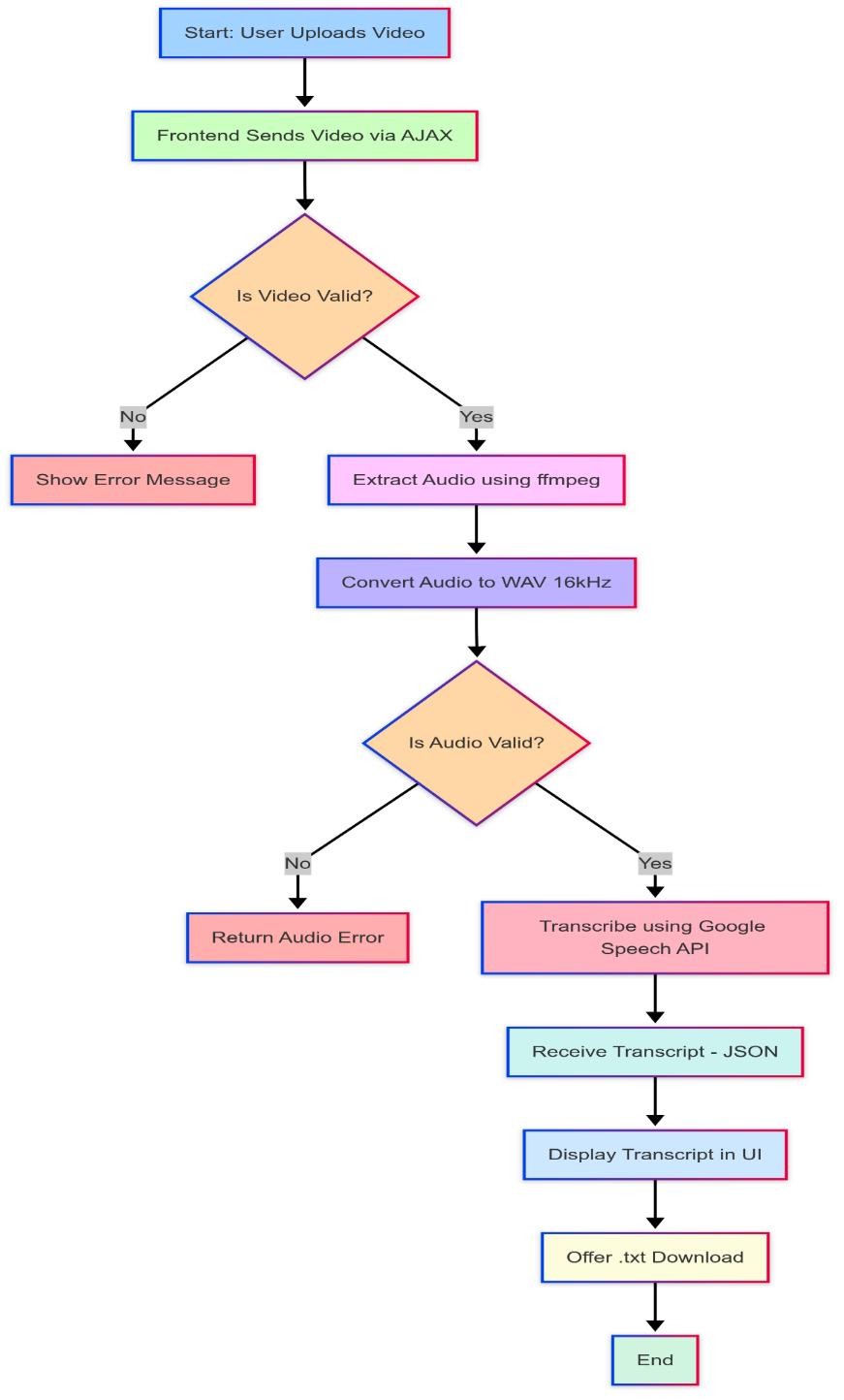
**4.1 SOFTWARE REQUIREMENTS:**

 **Languages and Frameworks:**

* Python (Backend)
* Flask (Web Framework)
* HTML/CSS/JavaScript (Frontend)
* Bootstrap (UI Framework) **Libraries and Tools:**
* FFmpeg: For video and audio processing
* Speech Recognition: OpenAI Whisper API
* Werkzeug: Handling file uploads securely
* Font Awesome: For icons
* Subprocess: For executing FFmpeg commands **File Structure:**
* app.py: Main Flask application with routes and upload handling
* main.py: Entry point for the application
* utils/video\_processor.py: Contains functions for extracting audio from videos
* utils/transcription.py: Contains functions for transcribing audio to text
* templates/base.html: Base template with layout and common elements
* templates/index.html: Main page template
* static/css/style.css: Custom CSS styles
* static/js/app.js: Frontend JavaScript for form handling and UI updates **Features:**
* Supports multiple video formats (MP4, AVI, MOV, WMV, FLV, MKV)
* Real-time progress updates
* Error handling at multiple stages
* File size limitation (100MB)
* Download functionality for the transcript
* Automatic cleanup of temporary files **API Integration:**
* OpenAI Whisper API for speech-to-text conversion
* REST API principles for structured data exchange
* JSON for data formatting and transmission **Deployment:**
* Gunicorn for production WSGI HTTP server
* Docker for containerization and consistent deployment
* Nginx as reverse proxy for production environments

This implementation architecture ensures a smooth, end-to-end process flow from video upload to transcript generation while maintaining modularity and separation of concerns between different system components.

**4.2 Explanation of Flowchart:**



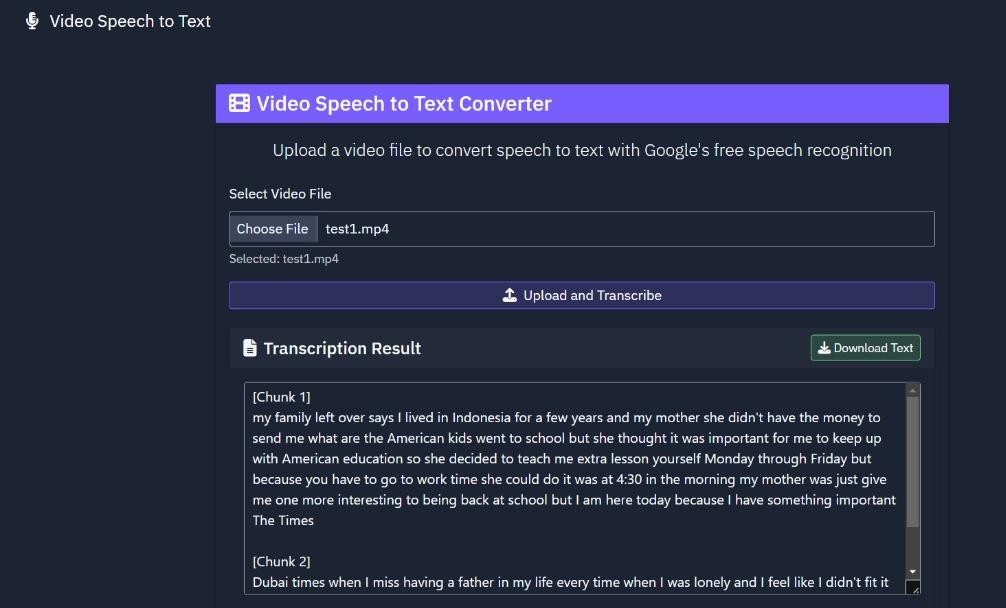
*Fig 4.1 Flowchart*

The flowchart above depicts a video-to-text transcription system workflow, showing the complete process from video upload to transcript download. Explanation of each step in detail is as follows:

1. **Start: User Uploads Video (blue)** 
   * The process begins when a user uploads a video file to the system.
2. **Frontend Sends Video via AJAX (light green)** 
   * The user interface sends the uploaded video to the backend server using AJAX (Asynchronous JavaScript and XML) technology.
3. **Is Video Valid? (orange)** 
   * The system checks if the uploaded video meets necessary requirements (likely format, size, or content validation).
   * If No: Show Error Message (process stops here with an error notification)
   * If Yes: Continue to the next step
4. **Extract Audio using ffmpeg (pink)** 
   * The system uses ffmpeg (a multimedia framework) to separate the audio track from the video file.
5. **Convert Audio to WAV 16kHz (purple)** 
   * The extracted audio is converted to WAV format with a 16kHz sampling rate, which is optimal for speech recognition.
6. **Is Audio Valid? (orange)** 
   * The system verifies if the extracted audio meets quality standards for transcription.
   * If No: Return Audio Error (process stops with an audio-specific error)
   * If Yes: Continue to the next step
7. **Transcribe using Open AI Whisper** 
   * The system sends the prepared audio to Google's Speech-to-Text API for transcription.
8. **Receive Transcript - JSON** 
   * The system receives the transcription results from Google in JSON format.
9. **Display Transcript in UI (blue)** 
   * The generated transcript is shown to the user in the interface.
10. **Offer .txt Download (cream)** 
    * The system provides an option for the user to download the transcript as a text file.
11. **End** 
    * The process concludes after the transcript is displayed and download option provided.

**Chapter 5**

### Results and Analysis



*Fig 5.1 Transcribe Text*

The image shows the user interface for the Speech-to-Text Extractor project, specifically the initial upload screen. The interface features:

1. A clean, dark-themed design with a purple accent colour for key elements
2. A prominent header reading "Video Speech to Text" at the top of the page
3. A secondary heading "Video Speech to Text Converter" with a film icon in a purple banner
4. Clear instructional text that reads: "Upload a video file to convert speech to text with Google's free speech recognition"
5. A file selection area with:

o A label "Select Video File" o A "Choose file" button with status indicator showing "No file chosen" o A helpful note listing supported video formats: "Supported formats: MP4, AVI, MOV, WMV, FLV, MKV"

1. A prominent purple "Upload and Transcribe" button with an upload icon at the bottom

This interface represents the entry point of the application workflow, where users begin the transcription process by selecting and uploading their video files. The design emphasizes simplicity and clarity, making it immediately obvious what the application does and how to use it, while providing necessary technical information about supported file formats.

**Chapter 6**

### Conclusion and Future Scope

**6.1 Conclusion**

In conclusion, the proposed speech transcriber system successfully automates the process of converting spoken content from videos into accurate and readable text. By integrating audio processing tools like FFmpeg and utilizing the OpenAI Whisper API, the system ensures high-quality transcription. It handles input validation, audio conversion, and real-time transcription efficiently. The final output, which is displayed on the UI and made available for download, simplifies tasks like note-taking, documentation, and content summarization. This project enhances productivity, accessibility, and user convenience in various domains such as education, media, and corporate communication.

**6.2 Future Scope**

To enhance its utility and reach, the speech transcriber should incorporate several key features. Firstly, multi-language support is crucial to extend its transcription capabilities to both regional and international languages, catering to a wider user base. Secondly, speaker identification, through the integration of speaker diarylation, will allow the tool to differentiate between multiple speakers in an audio recording, significantly improving clarity and understanding of conversations. For users in areas with limited or no internet connectivity, offline functionality should be enabled by utilizing local speech models. To improve accuracy in challenging acoustic environments, the application needs robust noise reduction capabilities employing advanced filtering techniques. Furthermore, the addition of real-time summarization, powered by AI, will provide users with instant extraction of key points from the transcribed text. Enhancing the readability of transcripts is paramount, necessitating punctuation and formatting enhancements that automatically add proper punctuation, paragraphing, and formatting. To ensure accessibility on various devices, mobile integration through the development of a mobile-friendly version is essential for on-the-go use. Finally, cloud storage integration, allowing users to save and access their transcripts via popular cloud platforms like Google Drive or Dropbox, will provide convenience and data security.

### References

1. “Speech to Text Transcription”, Sai Teja Ramacharla, Vustepalle Aniketh, Dr. M.

Senthil Kumaran, Computer Science and Engineering SCSVMV, Assistant Professor,

Dept of CSE, SCSVMV University Kanchipuram, Tamil Nadu, India, 2025

1. “Speech-to-Text Transcription in Support of Pervasive Computing”, Jarrah Sladek,

Andrew Zschorn, Ahmad Hashemi-Sakhtsari, Human Systems Integration Group

Command and Control Division Defence Science Technology Organisation, Adelaide

5111, South Australia,2003

1. “Continuous Visual Speech Recognition for Audio Speech Enhancement”, Eric

Benhaim, Hichem Sahbi, Guillaume Vitte, Telecom ParisTech CNRS-LTCI 46 rue

Barrault, 75013 Paris, France, 2015

1. “The Impact of Vocabulary Size and Language Model Order on the Polish Whispery

Speech Recognition”, Piotr Kozierski, Talar Sadalla, Szymon Drgas, Adam Dąbrowski,

Joanna Ziętkiewicz, oznan University of Technology, Piotrowo street 3a, 60-965

Poznana, Poland, 2017

1. “Multi-Stream Asynchrony Dynamic Bayesian Network Model for Audio-Visual Continuous Speech Recognition”, Guoyun Lv, Dongmei Jiang, Rongchun Zhao, Xiaoyue Jiang, H. Sahli, School of Computer Science, Northwestern Polytechnical University Xi'an 710072, Shaanxi Province P.R. China Vrije Universiteit Brussel, Department ETRO, Pleinlaan 1050 Brussels, Belgium, 2007
2. “Improving Short Utterance Speaker Recognition by Modeling Speech Unit Classes”, Lantian Li, Dong Wang, Chenhao Zhang, Thomas Fang Zheng, Tsinghua National

Laboratory for Information Science and Technology, Tsinghua University, Beijing,100084,China,2015

1. “Speech to Text Conversion and Summarization for Effective Understanding and Documentation”, Vinnarasu A., Deepa V. Jose, Department of Computer Science,

CHRIST (Deemed to be University), India, 2019.