TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

A PROJECT REPORT ON

OBJECT ORIENTED PROGRAMMING WITH C++

ECHO

SUBMITTED BY:

Bibek Dhungana (078BCT029)

Bishal Panta (078BCT036)

Braj Mohan Neupane (078BCT037)

SUBMITTED TO:

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

PULCHOWK CAMPUS

SUBMITTED ON:

JULY 8, 2023

Acknowledgement

We would like to express our sincerest gratitude towards the individuals who have supported us in completing this project related to object oriented programming in C++. This project would not have been possible without the support and motivation they provided.

First and foremost, we would like to thank Er. Daya Sagar Baral for his invaluable guidance, expertise, and continuous support throughout this project. His deep understanding of OOP concepts and their willingness to provide guidance and feedback have been instrumental in giving shape to our ideas. His teachings, lectures, and assignments have equipped us with the necessary skills and knowledge to undertake this project.

Furthermore, we would like to thank the Department of Electronics and Computer Engineering for granting us this wonderful opportunity to develop a project that has helped us build a strong foundation in programming and OOP principles

We would also like to acknowledge the contributions and support of our classmates and friends who have provided valuable insights, suggestions, and discussions during the development process. Their collaboration and brainstorming sessions have significantly enriched the design and implementation of the software.

Thank you all for your invaluable assistance and for being a part of this journey.

Sincerely,

Bishal Panta - 078BCT036

Braj Mohan Neupane - 078BCT037

Bibek Dhungana - 078BCT029

Abstract

In today's world where we're always talking and using media, our project is all about bringing technology and new ideas together. Our project, "Echo", is all about the fusion of state-of-the-art LLMs and proficient programming to deliver a versatile tool that simplifies access to audio content and enhances user engagement.

The core motivation behind our project is the recognition of the vast potential locked within spoken language. Our endeavor is grounded in the quest to harness this potential by developing an application capable of accurately transcribing speech into text, in real time. And we didn't want to just stop at writing things down – we wanted to add some magic to other things like videos and songs.

With a strong emphasis on object-oriented programming (OOP) principles, we've sculpted a robust foundation for Echo. This foundation not only ensures the project's maintainability and scalability but also serves as an opportunity for us to deepen our understanding of these vital programming paradigms.

As students dedicated to both learning and innovation, we've taken pride in imbuing our project with a sense of practicality. Our journey has been one of exploration, collaboration, and hands-on experimentation. With this report, we endeavor to chronicle our progression, outline our methods, and showcase the outcomes we've achieved.

In the pages that follow, you'll discover a detailed account of our project's objectives, methodologies, implementation, and results. Through this project, we aspire not only to fulfill academic requirements but also to contribute to a more accessible and enriched digital landscape.

Table of Contents

Objectives

This project aims to develop a speech-to-text audio transcriber using OpenAI's Whisper model, focusing on implementing object-oriented programming principles and algorithms to build a robust and user-friendly application. The main objectives of this project can be summarized as follows:

1. To gain practical experience in object-oriented principles
2. To implement key OOP concepts such as encapsulation, inheritance, and polymorphism to create a well-structured and maintainable codebase
3. To apply design patterns and architectural principles to develop a scalable and extensible speech-to-text transcriber solution
4. To gain knowledge and experience in utilizing AI models for natural language processing tasks, such as speech transcription
5. To explore and experiment with different techniques for fine-tuning and optimizing AI models to enhance transcription accuracy
6. To collaborate and learn from team members' implementations of OOP concepts, fostering a deeper understanding and mastery of OOP techniques

Introduction

Echo is an innovative application designed to change the way we interact with audio content. Echo takes a big step forward in turning spoken words into written text. Using Language Models, Echo offers various features like creating song lyrics, subtitles, real-time speech-to-text, and more.

Covering a wide range of functions, Echo aims to provide an all-in-one solution for turning audio into text. It's made to fit different needs and situations. The main things Echo can do are:

1. **Real-time Transcription with Voice Activity Detection (VAD)**: Echo can quickly turn spoken words into text as they're being said. It's smart enough to only focus on the important speech parts by using Voice Activity Detection.

2. **Karaoke Generation from Music Files**: Echo makes it possible to create karaoke-style tracks from music files. This adds a fun and interactive way to enjoy music.

3. **Subtitle Generation for Videos**: Echo employs Language Models to automatically generate hardcoded subtitles for videos. This feature aids in improving accessibility and engagement with video content.

With these cool features, Echo is changing how we deal with audio. By easily turning audio into text that we can work with, it makes things more accessible and lets us dive deeper into understanding and enjoying audio content. Through its many abilities, Echo looks forward to a future where audio becomes simpler to access, understand, and appreciate in different situations.

Application

"Echo" boasts a versatile range of applications that cater to diverse needs and contexts in the modern world. With its transformative features, the application addresses several key areas, each contributing to an enriched user experience.

1. **Real-time Speech Transcription**: The heart of "Echo" lies in its ability to convert spoken words into written text in real time. This feature finds applications in various domains such as note-taking during lectures, transcribing interviews, and generating accurate meeting minutes.
2. **Enhanced Accessibility**: The automatic subtitle generation feature enhances the accessibility of video content, making it more inclusive for individuals who are deaf or hard of hearing. This can prove invaluable in educational settings, online tutorials, and content consumption.
3. **Media Engagement**: "Echo" brings a novel dimension to media engagement. The karaoke generation feature allows users to sing along with their favorite songs by providing on-screen lyrics, fostering a more interactive and immersive musical experience.
4. **Language Learning**: With its subtitle generation capability, "Echo" aids language learners by providing written text alongside spoken words in videos. This facilitates better comprehension and vocabulary acquisition.
5. **Transcription Services**: The real-time transcription feature finds utility in professional settings where accurate and immediate conversion of spoken content into text is essential, such as in legal proceedings, medical dictation, and business meetings.

By addressing these applications, "Echo" emerges as a versatile tool with far-reaching implications. It empowers users to access, understand, and engage with audio content in new and meaningful ways, thereby contributing to an enhanced digital experience

Methodology

The development of "Echo" involved a structured approach that amalgamated both programming proficiency and the integration of LLMs. The project's methodology comprised several key phases, each contributing to the creation of a robust and functional application.

1. **Requirements Analysis**: The initial phase involved a meticulous analysis of project requirements. Understanding the desired features and functionalities laid the groundwork for subsequent development.
2. **Technology Selection**: The choice of OpenAI's Whisper model for speech-to-text conversion formed a cornerstone of the project. Leveraging this large language model aligned with the project's objective of accurate transcription.
3. **Implementation of OOP Concepts**: Object-oriented programming concepts, including encapsulation, inheritance, and polymorphism, were meticulously implemented to ensure code reusability and an organized architecture.
4. **Integration of Large Language Model**: The integration of the Whisper model was a crucial technical undertaking. The team engaged in rigorous testing and fine-tuning to ensure optimal performance and transcription accuracy.
5. **Real-time Transcription with VAD**: The implementation of real-time transcription was fortified by Voice Activity Detection (VAD) technology, enabling the system to focus solely on pertinent speech segments for transcription.
6. **Karaoke Generation and Subtitle Creation**: The development of the karaoke generation and subtitle creation features involved the synchronization of audio and text, enhancing media engagement and accessibility using the well known FFMPEG tool.

7. **Testing and Quality Assurance**: Rigorous testing methodologies were employed to validate the accuracy of transcriptions, the effectiveness of VAD, and the precision of generated subtitles and karaoke tracks.
8. **Iterative Development**: The development process embraced an iterative approach, allowing the team to address challenges, incorporate user feedback, and refine functionalities over multiple cycles.
9. **Collaboration and Knowledge Exchange**: Regular team meetings facilitated collaboration and knowledge exchange, enabling team members to learn from each other's implementations and collectively solve challenges.

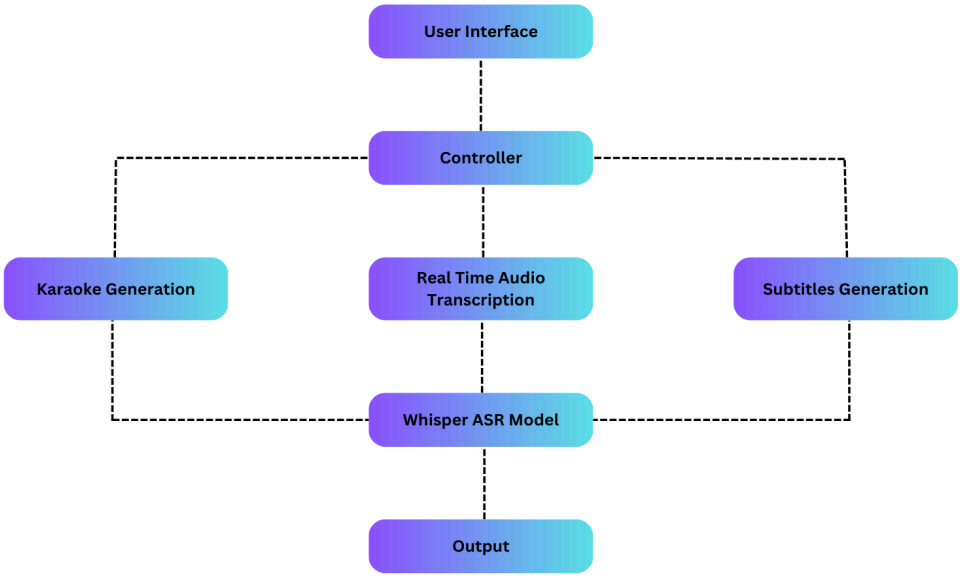
The methodology employed for "Echo" reflects a fusion of systematic planning, programming proficiency, and the integration of cutting-edge technologies. The result is an application that seamlessly merges speech-to-text conversion with media enhancement, embodying the essence of innovation and practicality.

Implementation

The implementation of "Echo" involved the practical realization of its diverse features, driven by the integration of object-oriented programming principles and AI technologies. The process can be broken down into the following components:

1. **Whisper Model Integration**: OpenAI's Whisper model was integrated to handle the speech-to-text conversion. By utilizing the model's capabilities, "Echo" achieved accurate and real-time transcription.
2. **Object-Oriented Architecture**: The application's architecture was meticulously designed using object-oriented programming principles. Class structures were developed to encapsulate functionalities, ensuring a modular and maintainable codebase.
3. **Real-time Transcription with VAD**: The real-time transcription feature was implemented with the aid of Voice Activity Detection (VAD). This allowed "Echo" to recognize when speech was occurring and transcribe only those segments.
4. **Karaoke Generation**: For the karaoke generation feature, audio files were processed into PCM and synchronized with the lyrics and a FFMPEG script was generated using an algorithm. Which was executed to create a synthetic karaoke video.
5. **Subtitle Creation**: Subtitles creation process was similar to karaoke generation with some extra steps involving conversion of mp4 to wav and instead of creating a new synthetic video subtitles were burned into the video.
6. **User Interface**: The user interface was developed to offer intuitive interaction. Users could initiate real-time transcription, upload audio files, and customize preferences for subtitle and karaoke generation.
7. **Testing and Optimization**: Rigorous testing was conducted to ensure the accuracy of transcriptions, proper synchronization of karaoke tracks and subtitles, and smooth user interactions.

i. System Block Diagram



1. The User Interface interacts with the users, providing options for inputting audio files and selecting functionalities.
2. The Controller manages the flow of data and control between the User Interface and the different modules.
3. The Real Time Speech Recognition, Karaoke Generation and Subtitles Generation modules receive instructions from the Controller based on the selected functionality.
4. All three modules converge to the Whisper ASR Model, which processes the input audio and generates the desired output.
5. The Output/Result component displays the transcriptions, converted lyrics, or generated subtitles to the users.

Project Scope

We are aiming to create an interface for everything speech-to-text with the help of OpenAI’s Whisper Automatic Speech Recognition (ASR) model. We used a Port of OpenAI's Whisper model in C/C++ for this project. The Echo Project will focus on all the basic use cases for speech-to-text conversion and implement it with minimal user experience. The main scope of this project is to successfully implement Real Time Speech Recognition.

The Scopes of this Project include:

- Real Time Speech Recognition.
- Burned-in Subtitles Generation
- Karaoke Generation
- Speech extraction from various audio formats (.mp3, .wav)
- Create a Minimal UI to go along with all these features.

Constraints

Our Project shall have some constraints that will have to be overcome. The major constraint is that there is no native support for audio in C++. Our Project relies on audio processing and to have no native library for audio was surely a handicap for us. Another constraint for us was setting up the SFML graphics library for each of our workspace. Working with libraries in C++ is always a hassle, but keeping these constraints in mind we understood that this will be a great learning opportunity and with that belief we will be giving our best to this project.

Problems Faced and Solutions

During the development of "Echo," our team encountered several challenges that required innovative solutions to ensure the application's functionality and performance. Here, we outline some of the key problems faced and the corresponding solutions we devised:

Handling the Resource-Intensive LLM (Large Language Model):

Problem: The computational demands of the Large Language Model (LLM) used for speech-to-text conversion were excessive for our standard PCs, causing performance issues.

Solution: To mitigate this challenge, we implemented Voice Activity Detection (VAD). VAD allowed us to filter out non-speech segments of audio, significantly reducing the processing load on our systems. This optimization ensured smoother real-time transcription.

Concurrency Issues in Real-time Transcription:

Problem: Real-time transcription required simultaneous reading and writing of data, resulting in concurrency conflicts and potential data corruption.

Solution: We implemented a Queue data structure, coupled with buffer management, mutex, and locks. This allowed us to organize the data flow efficiently, ensuring that incoming audio data was processed sequentially and without interference, thereby resolving the concurrency issue.

Optimizing Karaoke and Subtitle Synchronization:

Problem: Achieving precise synchronization between audio, lyrics, and subtitles was a complex task, prone to timing discrepancies.

Solution: We implemented algorithms that analyzed audio signals and text data to ensure accurate synchronization between karaoke lyrics, audio playback, and subtitles. This meticulous synchronization enhanced the user experience, making karaoke and subtitles engaging and coherent.

Conclusion

In culmination, the development of "Echo: Audio Transcriber and Media Enhancer" has been a journey marked by innovation, collaboration, and the practical application of programming principles. The project has succeeded in seamlessly fusing advanced AI technologies, object-oriented programming, and user-centric design to create an application that transcends traditional audio interactions.

Achievements and Contributions:

Throughout the project's lifecycle, we achieved a range of notable accomplishments:

1. **Real-time Transcription**: We transformed spoken words into text in real time, revolutionizing the accessibility and usability of audio content.
2. **Subtitle Generation**: By generating subtitles automatically, we improved the inclusivity of videos and enriched the learning experience.
3. **Karaoke Generation**: The creation of karaoke tracks breathed new life into musical engagement, offering an interactive dimension to audio consumption.
4. **Object-Oriented Approach**: Our commitment to object-oriented programming principles fortified the application's structure, scalability, and maintainability.

Recommendations for Future Enhancements:

As we reflect on our project's evolution, several avenues for future enhancements emerge:

1. **Multilingual Support**: Expanding the application's language capabilities to accommodate multiple languages and accents would enhance its global usability.
2. **Advanced Media Enhancement**: Exploring additional media enhancement features, such as sentiment analysis, summarization, and translation, can provide users with more diverse tools for engaging with content.

In conclusion, "Echo" epitomizes the potential of merging technology with creativity. As a result of our efforts, we have created a versatile tool that enhances the way we interact with audio content. This project has not only deepened our understanding of programming principles and AI integration but has also fostered a sense of accomplishment in delivering practical solutions to modern challenges. As we pass the baton to future developers, we remain excited about the possibilities that lie ahead for "Echo" and the broader field of accessible audio content enhancement..

References

During the development of "Echo: Audio Transcriber and Media Enhancer," we relied on a range of resources for guidance, inspiration, and technical expertise. The following list provides references to the key sources that influenced and informed our project:

1. **OpenAI Whisper**. https://openai.com/research/whisper
2. **Raylib**: https://www.raylib.com
3. **Voice Activity Detection**: https://en.wikipedia.org/wiki/Voice_activity_detection
4. **FFmpeg**: https://ffmpeg.org
5. **PortAudio**: https://www.portaudio.com