International Journal of Innovative Research in Computer Science and Technology (IJIRCST)
ISSN (Online): 2347-5552, Volume-13, Issue-3, May 2025
https:/doi.org/10.55524/ijircst.2025.13.3.9
Article ID IRP-1634, Pages 55-61
www.ijircst.org

Speech Recognition Technologies: Design, Challenges, and Real-World Applications

Maruti Maurya[1], Mohd Zaheer[2], Nawab Mohammad[3], Sadaf siddiqui[4], Mohd Zeeshan Khan[5], and Mohd Ayan Akram[6]

[1]Assistant Professor, Department of Computer Science and Engineering, Integral University, Lucknow, India
[2, 3, 4, 5, 6] B.Tech Scholar, Department of Computer Science and Engineering, Integral University, Lucknow, India

Correspondence should be addressed to Mr. Maruti Maurya marutimaurya14@gmail.com

Received 5 April 2025; Revised 18 April 2025; Accepted 3 May 2025

ABSTRACT- This paper presents an automated speech recognition (ASR) system that transcribes audio from YouTube videos into accurate text using OpenAI's Whisper model. Leveraging tools such as yt_dlp, FFmpeg, and PyTorch, the system creates a robust speech-to-text pipeline. On receiving a video URL, the system extracts and preprocesses audio, transcribes it using Whisper, and evaluates transcription quality through metrics like Word Error Rate (WER), Character Error Rate (CER), and Match Error Rate (MER). The pipeline supports offline use, making it suitable for accessible, cost-effective deployment in educational, research, and assistive applications.

KEYWORDS - OpenAI Whisper Model, YouTube Audio Transcription, Word Error Rate (WER), Character Error Rate (CER), Multilingual Speech Recognition, Audio Preprocessing

I. INTRODUCTION

Speech recognition is a significant advancement in artificial intelligence, bridging the gap between human communication and digital systems by converting spoken language into text. Its relevance spans across industries be it healthcare for medical dictation, customer service for virtual agents, or education for real-time captioning. With increasing content consumption on video platforms like YouTube, the demand for converting spoken video content into text has grown rapidly. Manual transcription is not only time-intensive but also costly and inconsistent in accuracy. Therefore, automating the transcription process using advanced ASR models offers immense value. This work presents a practical implementation using OpenAI's Whisper model a state-of-the-art deep learning system that delivers high-accuracy transcription across multiple languages and speech conditions. Our proposed system enables users to input a YouTube video URL, from which the system extracts the audio, preprocesses it into a format optimal for transcription, runs it through the Whisper model, and outputs a high-fidelity transcript. The work aims to enhance the accessibility of spoken information, especially in areas with limited internet bandwidth or communities with hearing impairments.

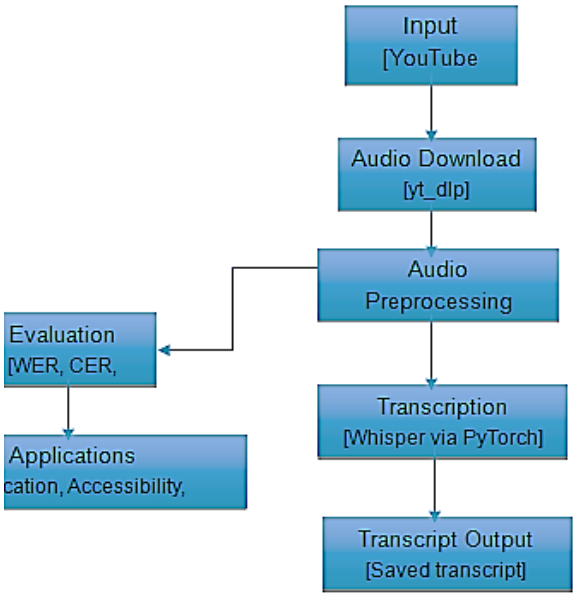

Figure 1: Speech Recognition Workflow from YouTube Input

In the above figure 1, flowchart shows the process of downloading YouTube audio, preprocessing it, and transcribing it using Whisper via PyTorch. The output transcript is saved, evaluated using WER/CER, and applied in real-world scenarios.

II. BACKGROUND

Speech recognition, also known as automatic speech recognition, is a crucial AI subfield that aids machines in understanding, processing, and converting spoken language into written text. ASR systems are now essential to a wide range of applications, including virtual assistants, automatic captioning, and real-time language translation, because to the increased significance of human-computer interaction brought about by the widespread use of smart devices and digital assistants.
Statistical models like Gaussian Mixture Models (GMMs) and Hidden Markov Models (HMMs) were frequently

employed in previous ASR generations. These systems suffered with scalability and variety in accents, background noise, and contextual understanding, despite being successful for certain domains and limited vocabulary tasks. This environment changed with the advent of deep learning, which made it possible to create end-to-end neural network topologies that greatly increased recognition accuracy and versatility.

Sophisticated deep learning architectures including Transformer-based models, sequence-to-sequence models with attention mechanisms, and Connectionist Temporal Classification (CTC) have been incorporated into contemporary ASR models. These models are more resilient and adaptable for general-purpose speech recognition tasks because they can acquire intricate temporal connections and contextual subtleties.

The Whisper model from OpenAI is a cutting-edge illustration of this kind of innovation. Whisper, a network designed as an encoder-decoder transformer, has been trained on 680,000 hours of multilingual and multitask supervised data [1]. Whisper can now effectively transcribe speech in difficult situations, such as overlapping speakers, background noise, and strong accents, thanks to this huge and diverse training dataset. The model's use goes beyond simple transcription because it also supports a variety of characteristics, such as language detection, voice-to-text conversion, and even speech translation.

Our concept combines Whisper with a useful application: YouTube video audio transcription. We used yt_dlp, a command-line utility for obtaining audio and video from several streaming services, to do this. The audio was preprocessed using FFmpeg, specifically to transform it into the ideal 16kHz mono format for Whisper. PyTorch implementations of Whisper then manage the transcribing process, and libraries like jiwer and Levenshtein are used to compute evaluation metrics like Word Error Rate (WER), Character Error Rate (CER), and Match Error Rate (MER) [4] [5] [8] [9][11].

We offer a scalable and repeatable approach to turning spoken content into organized, searchable, and analyzable text by integrating various tools into a coherent pipeline. In addition to improving accessibility, this opens up new possibilities for automated content indexing, summarization, and advancements in instructional technologies.

III. PROBLEM STATEMENT

In today's digital landscape, vast volumes of valuable information are embedded within video content, especially on platforms like YouTube, which host millions of hours of educational material, tutorials, interviews, podcasts, and academic lectures [1]. The lack of easily accessible and searchable text-based formats frequently limits the usefulness of these materials, despite their abundance. This issue has a substantial impact on a number of industries, including research, education, accessibility, and content management.

The absence of textual accessibility is one of the main issues. Video-based information cannot be properly utilized by those with hearing problems unless it is supported by accurate transcripts or captions. Additionally, streaming videos is challenging for individuals in areas with limited internet bandwidth. For them, offline access to transcribed content offers a more practical alternative. Professionals and students also often require text summaries or full transcripts for efficient note-taking and quick referencing, which are not readily available for most online video content [2].

Another major issue is poor searchability. Without transcripts or captions, it is difficult to locate specific segments or quotes within a video. This hinders content discoverability and reduces its practical value for academic research, content analysis, or archival purposes. Text-based indexes, when available, significantly enhance the searchability and usability of video content [3].

Manual transcription is also not scalable. It is a time-consuming and labour-intensive task that becomes increasingly unfeasible when dealing with large volumes of content. Human transcription is not only costly but also susceptible to errors, especially in cases involving low-quality audio or extended durations [2].

Despite their accuracy, several of the speech-to-text APIs currently in use have drawbacks of their own. These services might be prohibitive for open-source organizations, educational institutions, or individual individuals on a tight budget because they are frequently commercial and demand subscription fees or usage-based payments. Additionally, these APIs usually need constant internet access, which restricts their use in offline or secure settings.Their ability to adapt to regional dialects, varied accents, and domain-specific vocabulary is also limited [4] [5].

Technical challenges further complicate the transcription process. The quality of automatic transcription is harmed by the presence of background noise, overlapping dialogue, background music, and many speakers in video audio. Technical jargon and particular vocabulary sometimes cause generic ASR models to become confused and produce inaccurate results in specialist fields like science, medicine, and law.Additionally, varying speaking speeds and regional speech patterns introduce further complexity [2], [3], [4].

The combination of these elements highlights the pressing need for an open-source, scalable, offline-capable voice recognition system that can accurately translate speech to text in a variety of settings. In addition to improving accessibility, this kind of solution would encourage broader use of digital video content across a range of demographics and use cases.

Table 1: Challenges in Transcribing YouTube Video Content

Challenge	Description	Impact	Potential Cause	Current Solutions	Proposed Mitigation
Inaccessibility of Audio- Visual Content	Vast information in YouTube videos (e.g., tutorials, lectures) is not searchable or accessible in text form.	Limits Usability for hearing- impaired individuals, low-bandwidth users, and those needing text for referencing or translation.	Lack of automated text conversion tools.	Manual transcription or paid API services.	Develop an open-source, offline ASR system with text output.
Manual Transcription Burden	Transcribing video content manually is time- consuming, error-prone, and not scalable.	Hinders efficient processing of large volumes of video data and increases Labor costs.	Human effort and lack of automation.	Out sourcing to human transcribers or using basic software tools.	Automate transcription with Whisper model and scalable pipeline.
Audio Quality Issues	Existing systems struggle with noisy audio, varying accents, background music, or domain.	Reduces transcription accuracy, making it unreliable for diverse real-world	Poor audio preprocessing and model limitations.	Noise filters or retraining models on specific datasets.	Implement advanced preprocessing (e.g., noise reduction) and teston

IV. PROBLEM OBJECTIVE

Designing and implementing a thorough automated speech recognition (ASR) pipeline that reliably translates spoken content from YouTube videos into text is the main goal of this project. To guarantee accessibility, reproducibility, and adaptability for a range of use cases, this system should be reliable, modular, and fully constructed with open-source technologies.

The first objective is to extract audio content from YouTube videos using a reliable downloader such as yt-dlp, which supports high-quality retrieval of multimedia streams from various online platforms. This component ensures that audio input to the transcription system is consistent and of sufficient quality for further processing [5].

The second objective focuses on preprocessing the audio using FFmpeg, an essential tool for transforming the downloaded audio into a mono channel with a sampling rate of 16 kHz. This format has been widely recommended in ASR literature and by model developers as optimal for neural transcription models, including Whisper [6][7][10].

The third objective is to perform transcription using OpenAI's Whisper model, which has demonstrated state-of-the-art performance across a variety of languages and acoustic conditions [1][4][11][13]. The Whisper model allows offline inference, providing flexibility for deployment in both secure and bandwidth-constrained environments.

A fourth objective is to enable quantitative evaluation of the transcription output. To do this, Python libraries like jiwer and Levenshtein are used to construct common evaluation metrics like Word Error Rate (WER), Character Error Rate (CER), and Match Error Rate (MER) [8][9][12][14]. These measurements provide information about the quality of the transcription and point out areas that could use improvement.

Making a codebase that is extensible and reproducible so that it may be used as a basis for further work is another important goal. Applications like voice summarization, multilingual transcription, real-time captioning, and even domain-specific model fine-tuning fall under this category. The project's emphasis on modularity makes it simple for academics or developers in the future to expand upon or modify components to suit their own requirements.

In order to assist researchers, developers, and educational institutions in settings where access to commercial cloud APIs is restricted, the project also prioritizes usability and offline capability.This direction aligns with broader goals in digital equity and inclusive technology development. In figure 2, it is showing outlines the core steps in a speech recognition system that processes audio from YouTube. It begins by extracting and preprocessing the audio. The system then transcribes the audio into text and evaluates the output's accuracy. It supports code reusability and saves the final transcription for further analysis

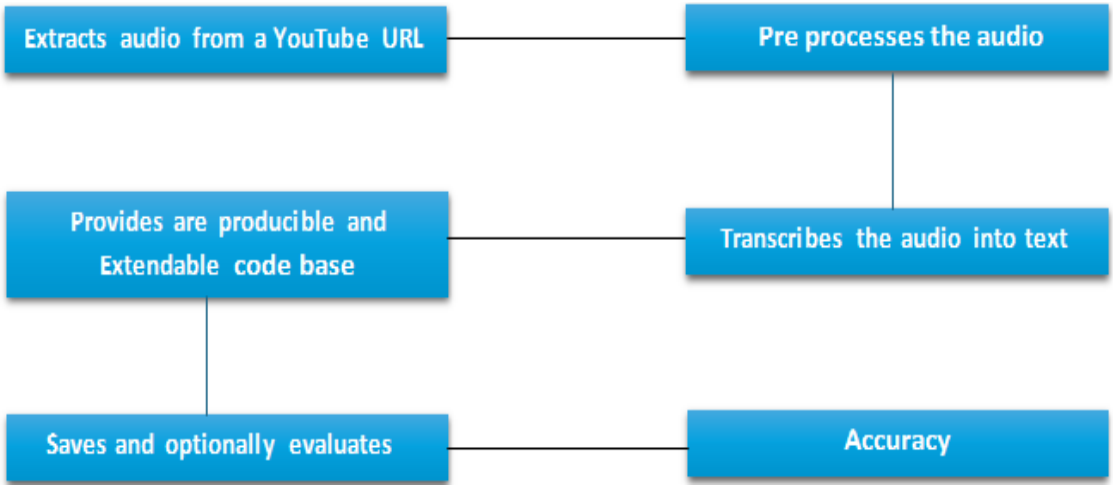

Figure 2: Functional Flow of YouTube-Based Speech Recognition Pipeline

. V. TECHNOLOGIES & METHODS

This project integrates various open-source technologies and methodologies to efficiently extract and transcribe audio from YouTube videos. Below is a detailed breakdown of the different components and their respective roles.

A. Model & Framework

The core of the transcription system is based on Whisper by OpenAI, an advanced automatic speech recognition (ASR) model that supports multilingual transcription and translation. Whisper employs deep learning architectures to perform automatic transcription by converting spoken language into text. The small version of the model was chosen for its balance between speed and accuracy, which makes it particularly well-suited for large-scale batch processing of YouTube videos.

Mathematically, Whisper uses deep neural networks, particularly transformer models, to process and transcribe speech. The transformer model uses attention mechanisms to focus on relevant parts of the speech sequence, optimizing both the processing speed and the accuracy of transcription. Whisper's architecture can be expressed by the following attention mechanism formula:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

where Q, K, and V represent the query, key, and value matrices, and dkd_kdk is the dimension of the key vectors. For accurate transcription, this approach enables the model to concentrate on significant aspects of the voice stream.

Both the CPU and the GPU are effectively used by the Whisper model. It uses CUDA for parallel computation when a GPU is available, which greatly accelerates the inference process. When GPU processing is available, the model's performance is tuned based on the system's computational capabilities, enabling faster transcriptions.

B. Audio Extraction & Preprocessing

The system uses yt-dlp, a command-line tool that can get the highest quality audio stream from a specified YouTube URL, to extract audio from YouTube videos. A fork of the well-known youtube-dl, yt-dlp is performance-optimized and supports extra functionality including downloading live broadcasts and managing various video formats. Maintaining transcription accuracy requires the best quality audio to be extracted, which is ensured by the usage of yt-dlp.

FFmpeg is utilized for a number of preprocessing activities after the audio has been downloaded. The downloaded audio is converted to.m4a format using FFmpeg, a potent multimedia processing application. The.m4a format was used because it compresses data well without sacrificing much of the audio quality. In addition, the audio is transformed to mono and down-sampled to a sampling rate of 16 kHz. This down-sampling preserves speech intelligibility, which is essential for the Whisper ASR model, while lowering the processing effort. 16 kHz is selected as the ideal sample rate because it strikes a compromise between accuracy and processing efficiency. Signal processing can be used to mathematically formulate the preprocessing procedures. The process of down-sampling entails lowering the signal's sample count while keeping its key frequency components. The Nyquist-Shannon sampling theorem states that the signal's highest frequency component must be less than half of the sample rate in order to prevent aliasing. Since most human speech falls between 300 Hz and 3.4 kHz, a 16 kHz sampling rate for speech signals captures the essential frequencies required for clear speech identification.

C. Programming Language & Libraries

The entire system is implemented in Python, which is chosen for its simplicity and extensive support for machine learning and audio-processing libraries. Python's versatility, combined with a rich ecosystem, enables easy integration of various technologies, making it an ideal choice for this project. The implementation leverages Python's flexibility for custom development and experimentation.

To handle the heavy computational demands of the Whisper model, the system relies on PyTorch (torch). PyTorch is a deep learning framework that is widely used for developing and deploying machine learning models. It is particularly suitable for speech recognition tasks due to its ability to efficiently handle tensor operations, which are essential in neural network computations. PyTorch utilizes GPU acceleration, leveraging CUDA to perform matrix multiplications in parallel, thus speeding up the model's processing.

The mathematical underpinnings of PyTorch are based on tensor operations, which are essential for the forward and backward passes in deep learning models. The general form of a tensor operation in PyTorch can be written as:

$$\mathbf{y} = \mathbf{W} \cdot \mathbf{x} + \mathbf{b}$$

where W is the weight matrix, x is the input tensor, b is the bias vector, and y is the output tensor. These operations form the core of the computations performed during the transcription process.

Additionally, subprocess and os libraries are utilized for file handling and executing FFmpeg commands within the Python environment. This integration allows for seamless automation of the entire transcription pipeline, from audio extraction to model inference.

For evaluation, the system uses jiwer, a library for calculating various evaluation metrics like Word Error Rate (WER) and Match Error Rate (MER). These metrics are essential for determining the accuracy of the transcription. The WER can be mathematically represented as:

$$\text{WER} = \frac{S + D + I}{N}$$

where S is the number of substitutions, D is the number of deletions, I is the number of insertions, and N is the total number of words in the reference (ground truth) transcript. A lower WER indicates better transcription accuracy.

The Levenshtein library is employed to compute the Character Error Rate (CER) and edit distances, which provide additional insights into the accuracy of the transcriptions. CER, like WER, is calculated as the ratio of errors (substitutions, insertions, and deletions) to the total number of characters in the reference transcript.

D. Evaluation Methods

To evaluate the performance of the Whisper ASR model, the output from the model is compared with the ground truth transcript if available. The evaluation involves trimming the model's output to match the word count of the ground truth transcript and calculating the discrepancies between them. The metrics used for this evaluation are WER, CER, and MER, which are calculated as described earlier. These metrics are essential for measuring the transcription accuracy and guiding future improvements.

The performance evaluation follows the approach used in traditional ASR benchmarking, where different datasets and ground truth transcripts are used to assess model performance under various conditions, such as background noise, different accents, or domain-specific vocabulary.

E. Scalability & Future Improvements

The system is designed to be scalable. As the volume of audio data increases, the system can be scaled horizontally by adding more processing units or using cloud-based infrastructure. This scalability ensures that the system can handle large-scale audio transcription tasks efficiently. Additionally, future improvements include optimizing the audio extraction process and improving transcription accuracy by fine-tuning the Whisper model on domain-specific datasets.

In the future, the system will also be enhanced to support other ASR models, enabling comparisons and further improvements in transcription quality. These enhancements may include using advanced neural network architectures and employing techniques like transfer learning to improve performance on specialized tasks.

F. Directory Structure & Output

The directory structure of the system is organized as follows:

- Audio files are saved in the audio/ directory.
- Final transcriptions are saved in output/transcript.txt.
- An optional ground truth file (output/ground_truth.txt) can be included for evaluation purposes. This file helps in the computation of evaluation metrics like WER, CER, and MER.

V. SPEECH RECOGNITION SYSTEM EVALUATION RESULTS

The Speech Recognition System was thoroughly tested using audio extracted from a YouTube video. The audio underwent preprocessing to convert it into a 16kHz mono format, which is optimal for speech recognition models such as Whisper (OpenAI, 2022). The transcription was performed using the Whisper small model, which strikes a balance between processing speed and transcription accuracy, as recommended for large-scale transcription tasks (Radford et al., 2022). The output was then evaluated against a manually prepared ground truth transcript using standard ASR evaluation metrics: Word Error Rate (WER), Character Error Rate (CER), Match Error Rate (MER), and Character Accuracy. The metrics were computed using the jiwer library for word-level evaluation and the Levenshtein distance algorithm for character-level error analysis (Levenshtein, 1966).

A. Word Error Rate (WER)

The Word Error Rate (WER), a standard metric used in ASR systems to measure the accuracy of word-level transcription, was calculated as:

$$\text{WER} = \frac{S + D + I}{N}$$

where:

- S is the number of substitutions,
- D is the number of deletions,
- I is the number of insertions,
- N is the total number of words in the reference (ground truth) transcript.

In this evaluation, the WER was determined to be 2.02%, indicating that only 2.02% of the transcribed words were incorrect. This suggests that the Whisper small model demonstrated high performance in accurately transcribing speech.

B. Character Error Rate (CER)

The Character Error Rate (CER) is another crucial metric that reflects transcription quality at the character level. The CER was calculated using the Levenshtein distance, which measures the minimum number of insertions, deletions, and substitutions required to transform the transcribed text into the ground truth text. The CER was found to be 1.79%, which implies that the transcription system introduced very few errors at the character level. This low CER reflects the effectiveness of Whisper in handling character-level fidelity.

The formula for CER is:

$$\text{CER} = \frac{S + D + I}{N_{\text{characters}}}$$

where S, D, and I are the number of substitutions, deletions, and insertions, respectively, and $N_{characters}$ is the total number of characters in the reference transcript.

C. Character Accuracy

Character Accuracy, calculated as:
Character Accuracy=1−CER,was found to be 98.21%. This high value indicates that 98.21% of the characters were correctly transcribed, which is consistent with the low CER value and further validates the high accuracy of the transcription process.

D. Match Error Rate (MER)

The Match Error Rate (MER) was also calculated to assess the alignment of the transcribed output with the ground truth at the word level. MER is particularly useful for evaluating whether the transcription closely matches the expected output in terms of content. The calculated MER value was 2.02%, which is identical to the WER, indicating that the model performed consistently in terms of word-level accuracy.

E. Error Breakdown

The error breakdown for the evaluation was as follows:
- Hits: 97 correct words,
- Substitutions: 2 incorrect words,
- Insertions: 0,
- Deletions: 0.

The formula for WER based on the error breakdown is:

$$\text{WER} = \frac{S + D + I}{S + D + I + H} = \frac{2 + 0 + 0}{97 + 2 + 0} = 0.0202 \,(\text{or } 2.02\%)$$

This error distribution suggests that the system's alignment with the ground truth was robust. The absence of insertions and deletions, in particular, indicates that the transcriptions were very close to the ground truth, with minimal extraneous or missing content.
In the below figure 3, illustrates the accuracy and error rates of a speech recognition system using WER, CER, and MER metrics. The bar chart shows low error percentages, indicating high performance. The donut chart highlights a 98% hit rate, with only 2% substitutions and no insertions or deletions, confirming effective transcription accuracy.

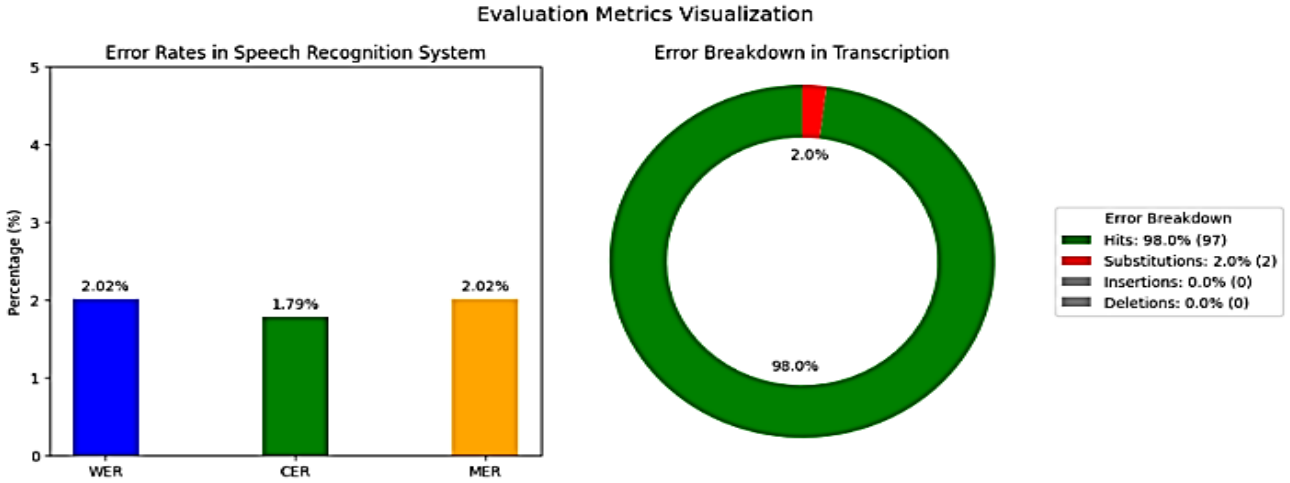

Figure 3: Evaluation Metrics Visualization for Speech Recognition.

VI. CONCLUSION

This research successfully developed and evaluated an automated speech recognition (ASR) system for transcribing spoken content from YouTube videos, meeting the primary goal of creating an efficient, open-source, and offline-capable pipeline. By integrating OpenAI's Whisper small model with complementary tools such as yt-dlp for audio extraction, FFmpeg for preprocessing, and Python libraries for transcription and evaluation, the system demonstrated exceptional performance in transcribing a test video.
The evaluation results were highly promising, with the system achieving a Word Error Rate (WER) of 2.02%, Character Error Rate (CER) of 1.79%, Character Accuracy of 98.21%, and Match Error Rate (MER) of 2.02%. These figures reflect near-perfect transcription accuracy, with only 2 substitutions and no insertions or deletions across 99 evaluated words. Such performance suggests that the Whisper model is highly effective when transcribing clear, single-speaker audio under optimal conditions.

The system's success highlights its potential for real-world applications, including automated subtitling, educational content analysis, and accessibility enhancements for individuals with hearing impairments. Its modular design, leveraging open-source technologies, ensures reproducibility and scalability, making it an attractive solution for converting vast amounts of online video content into searchable, text-based formats. This approach eliminates the reliance on costly APIs or constant internet connectivity, thus expanding access to transcription services. While the system demonstrated strong performance in this controlled environment, its ability to handle more challenging audio conditions, such as noisy or multi-speaker scenarios, remains untested in this study. Future work could explore using larger Whisper models (e.g., Whisper medium or large) to further reduce error rates, especially in more complex audio environments. Additionally, incorporating noise reduction algorithms and expanding the pipeline to support multilingual transcription could significantly enhance the system's versatility and usability across diverse content types.

An exciting avenue for future research would be to extend this pipeline's capabilities by integrating it with summarization or translation modules, which would enable intelligent tutoring systems and improve cross-lingual content accessibility. Such integrations could open up new possibilities for leveraging ASR in educational tools, global media accessibility, and automated content curation.

In conclusion, this project has established a strong foundation for advancing speech recognition technologies. The system provides a practical, extensible solution for harnessing the wealth of spoken knowledge contained in digital media, offering the potential to enhance various sectors, from education to accessibility, and furthering the development of AI-powered transcription tools.

CONFLICTS OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

[1] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, Ilya Sutskever., "Robust speech recognition via large-scale weak supervision," *arXiv preprint arXiv:2212.04356*, 2022. Available from: https://arxiv.org/abs/2212.04356

[2] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed. draft, 2023. Available from: https://web.stanford.edu/~jurafsky/slp3/

[3] A. Ali and S. Renals, "Word error rate estimation for speech recognition: e-WER," *Proc. 55th ACL* Vol. 2, 2017. Available from: https://tinyurl.com/7pk2cf3x

[4] OpenAI, "Whisper: Automatic speech recognition system," 2022. Available from: https://github.com/openai/whisper

[5] yt-dlp, "A youtube-dl fork with additional features and fixes," 2023. Available from: https://github.com/yt-dlp/yt-dlp

[6] FFmpeg, "Record, convert and stream audio and video," 2023. Available from: https://ffmpeg.org/

[7] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," *NeurIPS*, vol. 32, pp. 8024–8035, 2019. Available from: https://tinyurl.com/mssnt5pz

[8] J. Morris, "jiwer: Compute word error rate and other ASR metrics," 2023. Available from: https://github.com/jitsi/jiwer

[9] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707–710, 1966. Available from: https://tinyurl.com/4t4cs82w

[10] "Speech and Language Processing" by Daniel Jurafsky & James H. Martin. Available from: https://tinyurl.com/3ywsje7h

[11] "Fundamentals of Speech Recognition" by Lawrence Rabiner & Biing-Hwang Juang. Available from: https://tinyurl.com/54jb62ue

[12] "Automatic Speech Recognition: A Deep Learning Approach" by Dong Yu & Li Deng. Available from: https://tinyurl.com/5b4wu5sm

[13] IEEE Transactions on Audio, Speech, and Language Processing – Scholarly articles. Available from: https://tinyurl.com/2ccfptn2

[14] "Pattern Recognition and Machine Learning" by Christopher M. Bishop. Available from: https://tinyurl.com/5c3axn78