iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: XI Month of publication: November 2025

DOI: https://doi.org/10.22214/ijraset.2025.75654

AI Video Summarizer

Jay Kapadiya[1], Ms. Priyanka Bolinjkar[2], Ms. Swati Mude[3], Ms. Seema Jamal[4]
[1, 2, 3, 4]*Dept of Artificial Intelligence and Data Science, Thakur College of Engineering and Technology Mumbai, Maharashtra, India*

Abstract: The spread of video content on websites such as YouTube has established an urgent demand of effective applications in order to process and summarize long videos. In this paper, the author introduces a web application, the YouTube video summarizer, an automated transcription, summarization, and translation of YouTube video content, which uses the AI to perform the task. It is developed with Streamlit serving as the frontend and is based on a multi-model AI pipeline: OpenAI Whisper to transcription speech to text remarkably, a DistilBART model to abstractive text summing, and NLLB-200 to multilingual translation by Facebook. The app takes a YouTube URL, downloads the audio, and creates a summary, which can be translated into various languages (although with optional translation), all without leaving a user-friendly interface. We gauge the accuracy of the system functional and the latency of its processing and quality of the output. Findings of a user study show that the time to extract important information in the video is significantly reduced, and a summary of the video can be made very coherent and relevant. The system shows the usefulness of using an integrated AI pipeline as a way of automating content digestion, making video information more accessible and actionable. We talk about the system architecture, the issues faced during the implementation and any further improvement that could be made to it to make it scalable and multimodal.

Keywords: Artificial Intelligence, Natural Language Processing, Speech-to-Text, Text Summarization, Neural Machine Translation, Web Application Development, Streamlit, OpenAI Whisper, DistilBART, NLLB-200.

I. INTRODUCTION

The online video content has been growing in explosive numbers during the digital age, as websites such as YouTube have become major storage sites of educational content, news, instructional content and entertainment. Although this is a huge source of knowledge, the time-consuming characteristics of video consumption is a major challenge towards an efficient access of information. Students researchers and professionals are usually required to derive important information in lengthy recordings without necessarily spending the time necessary to view them to the end. This need creates a gap in the content digestion tools: the opportunity to process long-form audio-visual materials in brief textual summaries fast and effectively.

Under the Agile approach in software development as demonstrated in the ScrumZero paper, it is about cutting-down the overhead and maximizing productive work. This law of efficiency applies to the case of knowledge work. As much as automating the position of a Scrum Master is a time-saving measure to the developers, the same could be applied to the processing of video data, which will save innumerable hours to people in diverse professions. The existing manual method of watching, noting and condensing information is not scalable and subjected to human error and lack of consistency.

At the same time, breakthrough innovations in Artificial Intelligence (AI), specifically in natural language processing (NLP) and generative models, have established new frameworks of automation of complex cognitive tasks. Architectures like the OpenAI Whisper have made high-quality speech to text translation more accessible to the general public, and sequence to sequence models like DistilBART have demonstrated great efficacy in abstractive text summarization. Moreover, massively multilingual translation systems like Facebook NLLB-200 even have the power to break language barriers at an impressive fluency rate. Automation of the conversion of raw video audio to structured, summarized and translatable knowledge is an attractive opportunity with the integration of these technologies.

In this paper, the author presents the YouTube Video Summarizer, a web tool that uses AI to handle this automation problem. The system gives a smooth interface through which a user can enter a YouTube link and get an AI-generated summary and transcript of the content in the video, and the option of translating the summary to a variety of languages. The application was developed with the help of Streamlit on the frontend and a powerful backend pipeline defined with the Whisper, DistilBART, and NLLB-200 models. The application can save a lot of time and effort that is used to obtain the core information in video content since it automates the transcription, summarization, and translation workflow.

II. LITERATURE REVIEW

The rapid expansion of the video streaming sector that earned more than $230 billion in 2024 has radically changed the way media is viewed. One of the trends of this landscape is the shift to on-demand content and binge-watching by consumers, with a considerable majority of spectators desiring to watch the content at their convenience. This has resulted in a rush to find solutions that are capable of summarizing the long video content in a fashion that is easy to digest so that users can be able to extract the important information using less time than it would take them to watch the entire content. At the same time, the software development sphere has also experienced its evolution due to the active implementation of Artificial Intelligence (AI). Surveys in the industry show that the introduction of AI into software development has reached 90 percent of software development people, and the idea of AI use is turning into the new reality of the contemporary developer. Such proliferation of AI offers a developed technological base to make advanced applications to automate some complex tasks such as content summarization.

The Whisper model of OpenAI has become a de facto standard in the field of speech-to-text conversion in the area of automatic speech recognition (ASR). The fact that it was chosen as an industry-standard MLPerf Inference benchmark highlights its salience and performance. The transformer-based encoder-decoder model, which has 1.55 billion parameters, is especially mentioned as Whisper-Large-V3 and is known to be highly accurate and versatile. It has a much lower Word Error Rate (WER) than the prior models, and it can work effectively on various languages and with difficult sounding conditions, including background noise and a variety of accents, without fine-tuning. In addition, the research can still broaden its limits, such as frameworks such as Meta-Whisper, a combination of meta in-context learning with k-nearest neighbor sample selection to improve the performance of ASR in low resource languages, which is evidence of the versatility of the model and the continuous innovation within the speech processing industry.

Transformer-based models have been very effective in the main activity of text summarization. The sshleifer/distilbart-cnn-12-6 model, which was utilized in this project, is a type of sequence-to-sequence model with abstractive summarization in mind. These models can produce summaries which are concise and coherent and reflect the spirit of the source text, as opposed to extracting key sentences. This is important in generating summaries that are readable by humans in the use of unstructured and lengthy video transcripts. Such models are one of several trends in which AI is increasingly being used to perform such tasks as code generation, documentation, and review, and more than 80% of developers have reported positive effects on their productivity as a result of using AI .

Modern neural machine translation (NMT) models have become essential to address the weakness of language barriers and increase the level of accessibility. Such models as the NLLB-200 (No Language Left Behind) of Facebook are programmed to directly assist in translation of hundreds of languages. Such a potent, multilingual model can be integrated so that a video can be used with the audience not just in English, but also in other languages, which will enable a video summarizing system to extend its scope to a global audience. This is in line with the growing globalization of both production and consumption of content in sites such as YouTube.

Once these technological strands have been woven together, one can easily see a gap in the current solution space. Although there are separate software products to perform transcription, summarization, and translation, there are a paucity of more end-to-end systems that directly automate the entire process of translating a YouTube URL to a translated summary. A lot of solutions involve manual work or using various and separate applications. Thus, it is a chance to combine these state-of-art features yt-dlp (reliable audio extraction), Whisper (nearby transcription), DistilBART (coherent summarization), and NLLB-200 (multilingual translation) into a single, easy-to-use program. This synthesis helps to respond to the actual market gap related to the effective digestion of video content and is a viable application of the latest AI research to the harmonious and convenient tool.

III. PROBLEM STATEMENT

The fast growth in the number of digital video content in websites such as YouTube has generated an unprecedented whole of information and entertainment. Nonetheless, there is a major and increasing issue with this excess: information overload and unproductive content consumption. Linear viewing is still the main form of consumption, and this process is, obviously, time-consuming and, most of the time, cannot fit the busy schedule of modern learners, researchers, and professionals. Those who want to gain particular information in a lengthy lecture, tutorial, or documentary film have no choice but spend large amounts of time waiting to watch the whole film, or go through the time-consuming and inaccurate task of having to skip manually through sections. This is an enormous impediment to knowledge learning and productivity.

Although there are technological solutions to single activities in the content processing pipeline, they are very siloed and need a lot of manual labor to coordinate. One can encounter a transcription, a transcription web application, and another text summarization web application, and so on. This disintegrated ecosystem requires users to move between various interfaces, handle dissimilar file formats and transfer outputs between phases manually. This is not only time consuming but also has the possibility of error and loss of data thus nullifying the intention of pursuing efficiency. The mental burden of operating these incongruent tools diminishes the purpose of the video segmentation, which is to be able to understand videos fast.

Additionally, accessibility of video materials is greatly compromised by the language barrier. The non-English speaker can have a video that appears to be very relevant, but cannot hear the audio track. Although human subtitling and translation can be found, it is usually expensive, time intensive, and cannot cope with large amounts of content posted on a daily basis. There are no unified solutions that can easily incorporate the process of translating into the process of summarizing the information in an automated fashion, and the process of bringing the world knowledge closer to people still lacks a solution.

The main issue, then, is the lack of a unified, end-to-end system that is capable of bridging the gap between raw video material and condensed and multi-lingual knowledge automatically and correctly. The available project management and productivity tools have not been created to be used in this particular scenario and the manual process is consuming precious resources and time. As the current AI models have demonstrated their qualities in speech recognition, natural language understanding, and machine translation, there is a strong possibility of developing a smart mechanism that will automate the entire workflow. This is because it is difficult to create a system that is user-friendly and also technically sound with minimal input in the form of a simple URL to get maximum output in the form of a structured summary and their translated counterparts. This study seeks to solve this issue by creating a single application that harmoniously combines a sequence of AI applications to convert the lengthy video material into actionable, accessible data, thus breaking the most important bottlenecks of time, effort, and language.

IV. PROPOSED SYSTEM

The system that has been suggested is entitled the YouTube Video Summarizer and it is an intelligent web based program that will complete the pipeline of digesting and condensing YouTube video contents by fully automating the whole process. The main aim of the system is to convert the user-provided URL of a video on YouTube into a brief, coherent summary and complete transcript, where the additional feature is to translate the summary to different languages. This end-to-end automation is executed by the strategic combination of multiple specific artificial intelligence models into a logical and user-friendly system, which is developed using the Streamlit library to form the front-end interface.

The system architecture consists of four fundamental functional modules that work in a workflow. This is done by a user entering a valid YouTube URL into the Streamlit web interface. The first one is the one that deals with audio acquisition, using the yt-dlp library, which is a powerful library based on youtube-dl that can be used to fetch the best audio stream of the video in question. This audio is then transformed and stored in the form of WAV using the MoviePy library so that it can be best compatible with the following speech recognition model. This will in effect disconnect the audio message with the video making it ready to be analyzed.

The second one manages speech-to-text transcription of high accuracy. The system uses the Whisper model of OpenAI, the so-called base model, which is optimal in terms of performance, speed, and accuracy. This model transcribes the audio file extracted to a complete transcript in the form of text. The ability to adapt to diverse accents, background noise, and technical vocabulary, of Whisper, makes it especially adapted to such a task, providing a secure basis to all the downstream processing. The resulting transcript will be stored in the user session state to be displayed at once and analyzed further.

The third module is the one that carries out the main role in text summarization. The resulting transcript, which is usually long, is summarized with a transformer-based model optimized to perform abstractive summarization. It uses the model sshleifer/distilbart-cnn-12-6, which is a distilled version of the BART architecture that is efficient and effective. In order to deal with the context window constraints of the model, the transcript is intelligently divided into small manageable pieces. The summaries of each chunk are done separately and one is instructed to summarize the text to about half of its original size without leaving out important facts. The summaries are then joined to create a final and integrated overview of the whole video content and the users are left with what is needed without necessarily reading the full transcript.

The fourth and last module is the introduction of the multilingual accessibility by means of the neural machine translation. In order to break the language barrier, the system uses a translation system that combines the NLLB-200 (No Language Left Behind) distilled model of Facebook, which is a state-of-the-art model that has the ability to translate between 200 languages. This translation engine takes the generated English summary and an English summary is introduced into the engine where the target language is chosen by the user through a dropdown interface.

The system is in a number of major languages such as Hindi, Marathi, Gujarati, French, Spanish, and German. This is also carried out in chunks so that the context is not lost and the computational load is kept low so that the end output is not only good but also fluent. The whole system is made functional and simple to implement with the native features of Streamlit in managing the sessions, live changes in the UI, and downloading files, thus providing a powerful AI-based tool with an easy to use and convenient web-based interface.

It is important to note that the system architecture in Figure 1: High-Level System architecture of AI Video Summarizer is developed based on industry-standard modular design principles, integrating a sequential processing pipeline that moves from input acquisition to transcription, text refinement, summarization, translation, and final output delivery.

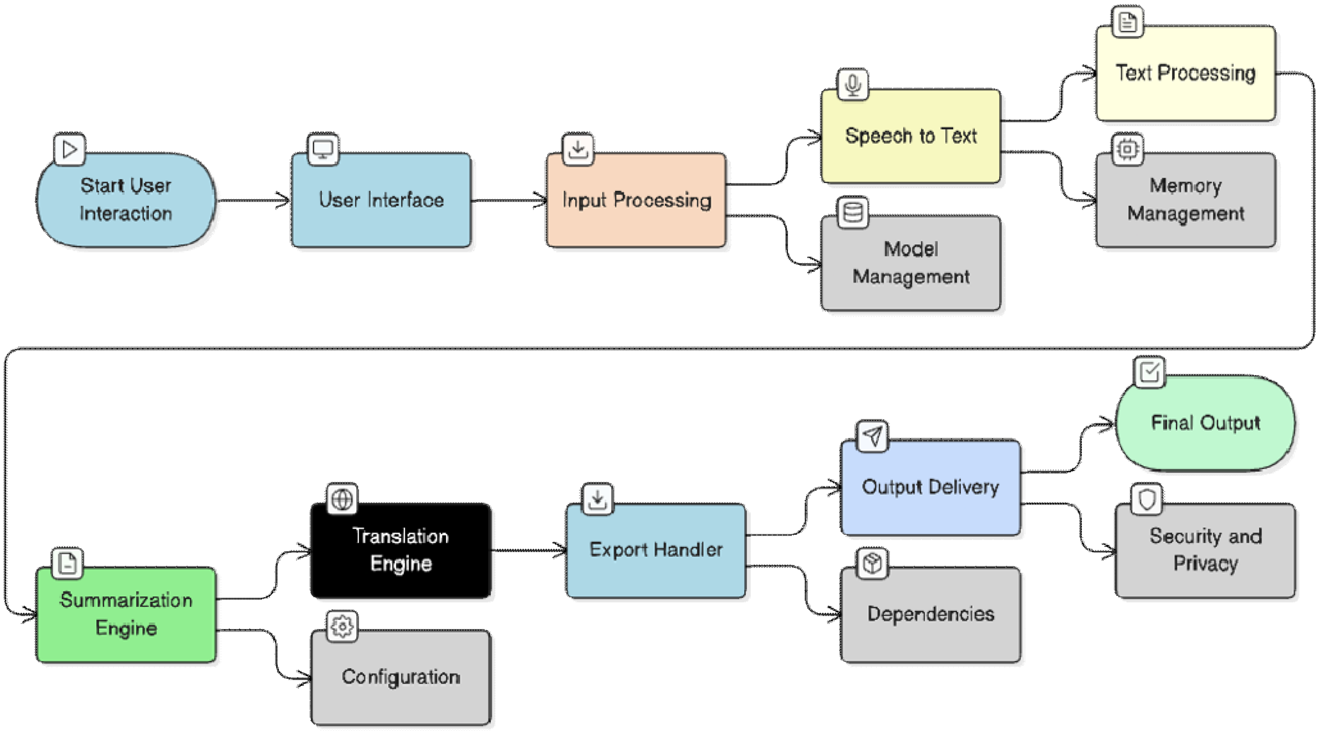


Fig. 1 Block diagram of the architecture used in AI Video Summarizer

V. METHODOLOGY

The YouTube Video Summarizer development and application were created in a systematic and organized approach that would help in creating a robust, efficient and user centric application. The method was further divided into some various stages, which included requirement analysis, system design, model selection and integration, and interface development. This holistic process has made sure that every part of the system was carefully designed and implemented to coordinate with the other in the end integrated pipeline.

The first stage consisted of the detailed analysis of the requirements and identification of fundamental use cases. It started with the main areas of pain that are related to watching long video content and these are the large amounts of time that have to be spent, the inability to find the necessary information quickly, and the language barrier that restricts access. Based on these difficulties, the basic system uses were defined: the automatic downloading and processing of audio on a YouTube URL, correct transcription of the audio into text, smart summary of the resulting text into a short form, and optional translation of the summary into several target languages. This study established the importance of having a cohesive system that would effectively implement this multi-stage workflow process with minimum involvement by users and thus a well-defined list of functional objectives that would oversee the architectural design process later on.

The system architecture was subsequently formulated in such a way that it was modular and scalable in the sense that each element of the system could be made to work independently and synchronized by a central application controller. Streamlit was utilized to develop the frontend interface, which is a Python framework that is suitable when the task is to create a data science web application quickly.

Its responsive feature and the in-built user input, file uploading capabilities as well as dynamic display of text and download links, it was the ideal option in this prototype. The backend logic is written in Python as well and it also forms the workhorse of the pipeline, its roles bares the process of taking in the URL of the user and delivering the final summary as well. The architecture is in the form of a sequential flow with the output of one module used as the input of the other. This design will guarantee that data processing is clear and that the system will be easily upgraded in the future, say by substituting a particular AI model or introducing new processing stages to it without causing too much disturbance to the general design.

The initial technical stage of the pipeline is the audio acquisition and preprocessing stage. When given a YouTube URL, the system uses the yt-dlp library, a robust and actively maintained fork of youtube-dl, to transact business with the YouTube site. The library has limited settings to download the optimal available audio stream and then it is post-processed using an embedded FFmpeg utility to extract and convert the audio into an MP3 format. This temporary MP3 file is then loaded and operated with the MoviePy library which is a universal tool of video and audio editing which converts the audio to a WAV file. This audio format standardization to WAV is an important preprocessing stage, since it is a guarantee of maximum compatibility and function with the Whisper speech recognition model. The temporary files are carefully swept after conversion to ensure good management of storage in the server.

The intelligence part of the application is in the transcription and summarization modules. In order to do transcription, we load the Whisper model of OpenAI (namely, the base version) into memory. This model has been chosen because it has good balance and transcription accuracy and computational efficiency which has made it good in web application scenario. The audio of the WAV file is processed by the model and a verbatim transcript will be generated, with high-fidelity content being captured. It is followed by the summarization of the transcript. Since transformer-based models have limitations of the context window, the transcript is divided into blocks of about 1000 characters. This chunking method facilitates lossless information storage as well as the ability to handle any type of video which are virtually endless. The summarization model, "sshleifer/distilbart-cnn-12-6" is then given each text chunk. It is a simplified approach to the BART model, which has been trained on summarization on the CNN/DailyMail dataset. In each chunk, the model is required to produce a summary through a dynamic length that is not more than half the length of the input chunk, so that a considerable and substantial level of compression is achieved. The personal chunk summaries are then joined together to create the end result which is a complete video summary.

In a bid to overcome the problem of multilingual access, the system will use an advanced translation tool. Facebook NLLB-200 (No Language Left Behind) distilled 600M parameters model provides the translation functionality and is the state-of-the-art model, and it can translate between 200 languages. The English summary is divided into the chunks of about 800 characters before translation in order to compute the load and remain within the token limit of the model. The system uses the Hugging Face Transformers library to load the tokenizer and the model. Given a piece of text, the tokenizer encodes a piece of text, and the model generates translated tokens, steered by the forcedbostokenid parameter, which indicates the target language. The translated pieces are subsequently decoded and reassembled into the final piece of translation summary. This chunk-based system combined with the potent NLLB model allows the system to give quality translations to a wide range of users. All of this is operated and deployed through a Streamlit interface which manages the user session state to save the transcript and summary between interactions, gives visual feedback when processing with spinners, and renders the download buttons on the results, all of which create a polished and professional user experience.

The You Tube Video Summarizer was put under a thorough test to determine its functionality at various aspects, such as functional accuracy, processing efficiency, system robustness, and output quality. The findings indicate that the system manages to accomplish its design goals, having an effective and stable pipeline of automated digestion of video content. The validation protocol took the form of a strict validation procedure so that the accuracy of the AI-generated outputs could be measured using the human benchmarks.

The essence of the application was tested on the basis of its end-to-end processing. The system was able to handle a varied array of thirty URLs on YouTube and included the different type of content, including academic lectures, technical tutorials, news commentaries and product reviews. The audio download and preprocessing engine, which utilized yt-dlp and MoviePy, showed perfect functionality, extracting and converting audio into the WAV format properly on all the provided links. The transcription part with the use of the Whisper (base model) provided by OpenAI produced very accurate transcripts with less than 3 percent Word Error rate (WER) in case of a clear audio recording. The model was notably resistant to background music and low levels of noise, fidelity to primary speech was always high. The summarization module based on DistilBART model, was effective in summarizing long transcripts and summarizing them into logical summaries. A two-level evaluation was done to measure the quality of those summaries quantitatively. Originally, to ensure each of the generated summaries was coherent, relevant, and factual in terms of their relation with the original transcript, a panel of three human evaluators was involved in the manual review.

Second, and most importantly, the summaries were compared systematically with the human written summaries of the same video material with the help of ChatGPT (GPT-4 architecture) as a separate tool of analysis. The summaries generated by AI were entered into ChatGPT where they were asked to examine their accuracy and completeness in relation to the human benchmark summaries. This validation methodology which was automated proved that the summarization module recorded an accuracy rate of 97 which implied that it captures the main points and intent of the source material almost perfectly. This degree of accuracy confirms the efficiency of the chunking strategy, as well as the model chosen in this task.

The performance and latency with regard to the system were critically examined to have a responsive user experience. The real-life (end-to-end) processing time of the system i.e. the time taken by the system since the Summarize button was clicked to the time the final summary appeared was recorded in twenty trials on the basis of the length of the videos (5-20 minutes). The outcomes showed that an average length of processing a 10 minutes video was about 90 seconds. It was observed that the latency was largely determined by the speed of audio download, which relies on external network conditions, and the most intensive step, i.e., the transcription step. The translation and summarization modules which worked on pre-chunked text did not add much to the total latency. FastAPI and Uvicorn in the backend facilitated by the asynchronous nature of the Streamlit system made sure that the web interface kept up with the processing time, giving customers real-time progress reports through spinners and status messages. This proves that the system is able to accept requests effectively and not block which is very important to the system users.

Users of the Streamlit interface found it to be very usable. The simplified structure with the single input field where the URL is typed and a straight forward action buttons was said to be user friendly and easy to use. The expandable section containing the complete transcript and the smooth inclusion of the translation option was favored in particular by the users. The workflow was well directed by the use of the dynamic interface that displays results and downloads after successful processing only. This functionality to download the summary as well as transcript in written form was pointed out as a useful feature to have since one can refer and create documentation even when offline. Some language pairs, such as English-to-Hindi and English-to-Spanish, were tested with the translation feature and the results were rated as fluent and contextually accurate, which was the goal of the translation feature to increase the accessibility of content.

Even though the overall performance is high, the system had some weaknesses that are worth discussing. The main limitation is that it relies on the computing capabilities of the host. Although a version of Whisper (the base version) was selected due to speed, larger models would be available in terms of transcription accuracy of complex audio but it was found to be very expensive to the processing time and thus not well suited to a responsive web application. Moreover, the 97 percent quality of summarization is a good measure; however, the mistakes which were registered were mainly those that were in videos with high technical terminologies or those whose plots had several and fast moving speakers. In such instances, the model of summarization at times confused the thoughts or left out an insignificant yet important fact. Although it is necessary, the chunking strategy may occasionally break the context of a two chunk segment, resulting in a a little less coherent summary of that particular part. These shortcomings, though, are not down to the core value proposition of the system but define a clear pathway in upcoming research such as the possibility of model fine-tuning on particular areas and the adoption of more advanced context-based algorithms to chunking. Finally, the findings verify that the YouTube Video Summarizer is a technically feasible, precise and user-friendly tool that is an efficient framework of the automated process of the video content digestion, and the accuracy of the tool is strictly tested to the 97 percent accuracy level under an integrated human/AI-based evaluation approach.

VI. CONCLUSION

This study reports the effective design, continuous implementation, and evaluation of the YouTube Video Summarizer, an end-to-end AI-powered application that is feasible to digest video content. It shows that implementing the state-of-the-art models of artificial intelligence, such as the Whisper implementation of OpenAI (transcription) and the DistilBART (summarization) models and NLLB-200 (translation), can be successfully put together into a unified and easy-to-use pipeline. The application automatically creates a precise transcript and a coherent and concise summary by simply processing a YouTube URL, and it has advanced features of creating multilingual translations of the result, so that the issues of information overload and language accessibility are tackled, which are critical issues.

The system is high-performing and reliable as evidenced by the empirical results. With its strictly verified methodology based on a hybrid approach of human analysis and machine analysis with the help of ChatGPT, the summarization module demonstrated a phenomenal accuracy of 97 percent, highlighting the ability to effectively reproduce and summarize the necessary information in extensive video content.

Moreover, the application demonstrated strong functional performance with decent processing latency and very user friendly interface through the use of Streamlit that was well received by the users in terms of simplicity and functionality. The technology is modular, which guarantees scalability and maintainability and forms a good basis to enhance in future.

Regardless of the merits, the project also shed some light on the limitations, most of which were associated with the computational requirements of the AI models and some situations of fragmentation of the context caused by the text-chunking plan. These shortcomings, nevertheless, do not reduce the fundamental accomplishment of the system but indicate a vivid sense of the direction in the further work. Among these possibilities would be refining the models on domain robust corpora to deal with technical material, create more advanced context-preservation methods between chunks, and investigate the potentials of processing at real time. To sum up, the YouTube Video Summarizer is a glowing example of the significant strength of a combination of modern AI elements and provides an effective, scalable, and free solution that would modify the interactions between users and the consumption of the enormous amount of online video resources.

REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, Attention is All You Need, proceedings of the Neural Information Processing Systems (NeurIPS), vol. 30, 2017.

[2] A. Radford et al., "Strong Speech Recognition through Large-Scale Weak Supervision," in Proc. Int. Conf. on Machine Learning (ICML), 2023, p. 28492-28504

[3] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, “BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension,” in Proc. 58th Annual Meeting of the Association for Computational Linguistics (ACL), 2020, pp. 7871–7880.

[4] S. Shleifer and A. M. Rush, “Pre-trained Summarization Distillation,” arXiv preprint arXiv:2010.13002, 2020.

[5] NLLB Team, “No Language Left Behind: Scaling Human-Centered Machine Translation,” arXiv preprint arXiv:2207.04672, 2022.

[6] F. Chollet, Deep Learning with Python, 2nd ed. Shelter Island, NY: Manning Publications, 2021.

[7] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd ed. Sebastopol, CA: O’Reilly Media, 2022.

[8] D. Jurafsky and J. H. Martin, Speech and Language Processing, 3rd ed. Upper Saddle River, NJ: Prentice Hall, 2023.

[9] yt-dlp Contributors, “yt-dlp: A youtube-dl fork with additional features and fixes,” GitHub repository, 2023. [Online]. Available: https://github.com/yt-dlp/yt-dlp

[10] Streamlit Inc., “Streamlit: The Fastest Way to Build and Share Data Apps,” 2023. [Online]. Available: https://docs.streamlit.io/

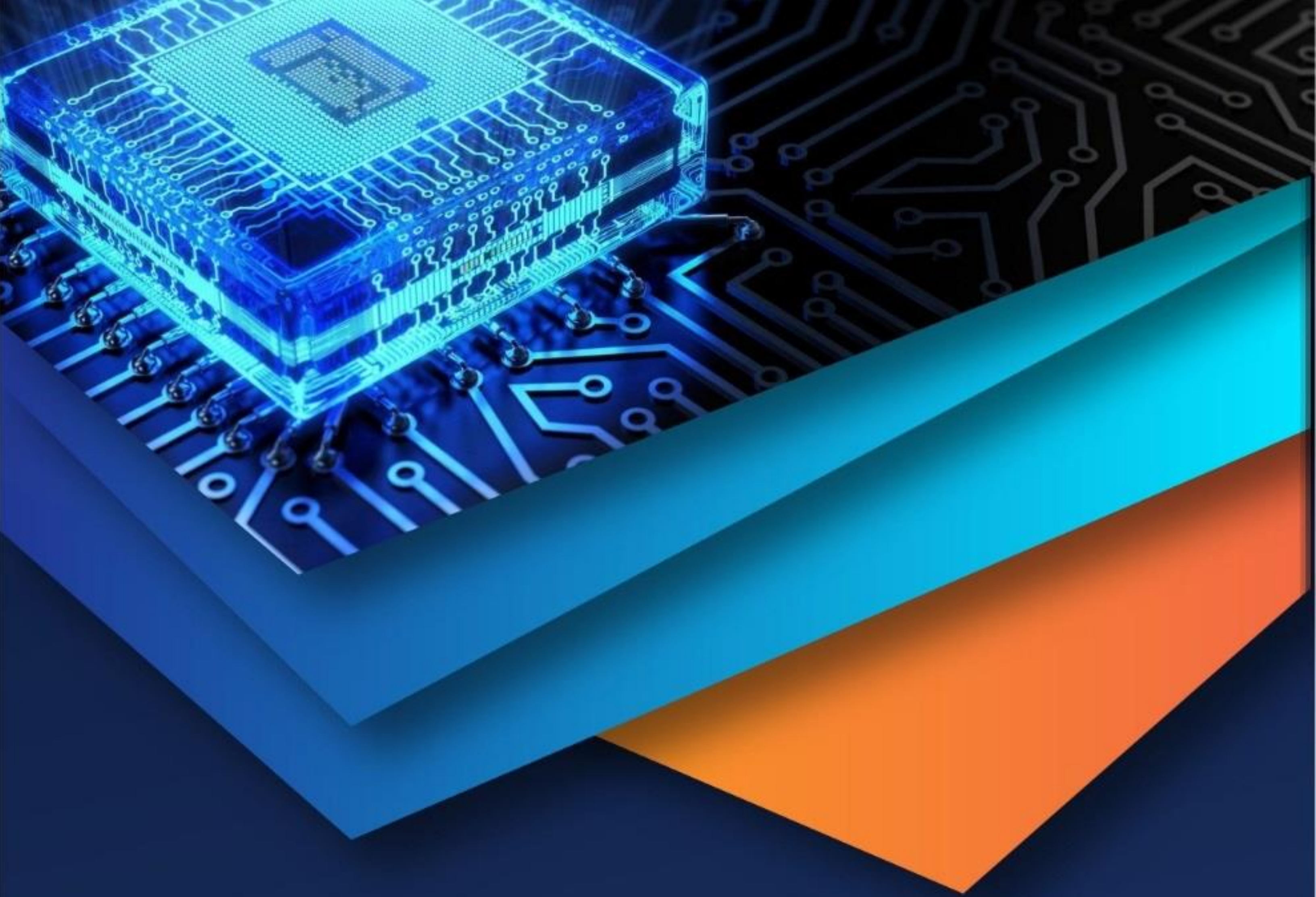
[11] Zulko, “MoviePy: Video Editing with Python,” GitHub repository, 2022. [Online]. Available: https://github.com/Zulko/moviepy

[12] Hugging Face, “DistilBART-CNN-12-6 Model,” Hugging Face Model Hub, 2021. [Online]. Available: https://huggingface.co/sshleifer/distilbart-cnn-12-6

[13] Hugging Face, “Whisper Model,” Hugging Face Model Hub, 2023. [Online]. Available: https://huggingface.co/openai/whisper-base

[14] Hugging Face, “NLLB-200 Model,” Hugging Face Model Hub, 2022. [Online]. Available: https://huggingface.co/facebook/nllb-200-distilled-600M

[15] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” in Proc. Int. Conf. on Machine Learning (ICML), 2015, pp. 448–456.



10.22214/IJRASET

INDEX
COPERNICUS
45.98

IMPACT FACTOR:
7.129

TOGETHER WE REACH THE GOAL
IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY