IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

EXPLORING AI-BASED FRAMEWORK FOR VIDEO AUTOMATION

[1]Mr. SIVARATHINAM S D, [2]Mr. HARSHAVARDHAN V, [3]Mr. JOEL KIRUBAKARAN G,

[4]Mr. LALITH ADITHYA S, [5]Mrs. SUGANYA A

[*1,2,3,4]Student BTech in Artificial Intelligence and Data Science, Anand Institute of Higher Technology, Chennai, Tamil Nadu, India.

[*5]Asst.Professor, Artificial Intelligence and Data Science, Anand Institute of Higher Technology, Chennai, Tamil Nadu, India.

Abstract: One of the critical multimedia challenges in today's digital landscape is the rapid and effective creation of video content enriched with relevant metadata. This metadata is essential for improving discoverability, organization, and engagement, but the process of manually generating it is both time-consuming and inconsistent. To address this, we propose a Video Metadata Automation System that utilizes recent advancements in artificial intelligence—particularly large language models (LLMs)—to automatically generate key video metadata, including titles, descriptions, captions, and thumbnails. Our system employs an end-to-end automated framework that integrates techniques such as audio extraction, speech-to-text transcription, and video analysis. Using APIs for connection to high-performing LLMs, the extracted transcripts are used to create metadata in a structured and SEO-friendly format. The most suitable LLMs are evaluated and selected for optimal performance in this context. Furthermore, the system features integration with image generation models such as Stability AI's Stable Diffusion to generate thumbnails based on the video's context. This paper explores the effective utilization of open-source tools and state-of-the-art AI technologies, presenting a unified and modular solution that simplifies the video content creation pipeline. By automating the metadata generation process, our system significantly reduces the workload for creators, enhances video accessibility, and enables faster content publication with minimal manual intervention.

***Index Terms* - Metadata Generation, Large Language Models (LLMs), Speech-to-Text, Audio Extraction.**

I. INTRODUCTION

In the current digital age, content creation has emerged rapidly as it interacts with a worldwide audience. Making it difficult for content creators and normal users to spontaneously come up with engaging content for professional or personal use. Nevertheless, posting and maintaining content on websites such as YouTube may be a laborious and intricate process, particularly when optimizing metadata is required. Metadata is nothing but a fancy word to explain the titles, descriptions, tags, cards, thumbnails and captions of videos. Even with the availability of vast volumes of data, computer vision and multimedia jobs still require human critical thinking and creativity. The content creation and integrating Metadata seamlessly has not been an easy task. As, it requires a meticulous level of detail, creativity, and effective content delivery. Matching up the level of creativity and productivity of human expertise with an AI is a challenging one. Some tools aid in assisting these tasks (generation of titles, descriptions, tags, cards, thumbnails and captions of videos.) separately. There is no such absolute tool that covers the entirety of the Video Metadata Generation.With the exponential growth of LLM in the field of AI, the capability of matching human expertise in creativity, productivity and efficiency has become possible. It has been developed to generate Metadata effectively by pushing the boundaries of what is possible in previous AI model technology, while it still requires an optimal architecture to generate the required metadata. Traditional methods of video metadata generation are time consuming, error prone, and require productivity for efficient content delivery.

Solving this task has been a challenge until now. By leveraging AI and developing an optimal architecture, it is possible to enable more accuracy and efficiency in metadata generation. To further enhance user interaction and utility, the system includes an integrated AI chatbot that is capable of answering any questions related to the uploaded video by leveraging a Retrieval-Augmented Generation (RAG) pipeline. It intelligently indexes the video's transcript and uses a powerful language model to provide accurate, context-aware responses, making the platform not just a publishing tool but also an interactive knowledge assistant. Thus the project is motivated by the need to improve and streamline the process of creating and uploading videos. This is done by the integration of web technology and AI powered generation of metadata (video description, tags, subtitles, thumbnails). Our project's goal is to reduce the manual labour whilst improving the quality and discoverability of the video.

Despite there being multiple options when it comes to generation of subtitle, description, thumbnails, and also seo optimization they are separate, each having their own module with applications like TubeBuddy, VidIQ, Descript, Adobe Premiere Pro, etc.. Our project tackles these issues of modularity by integrating and streamlining the entire process of video metadata creation in a single place. This in turn reduces manual labour and improves quality, efficiency and spontaneity of content creation.

II. LITERATURE SURVEY

This study brings to light several advancements across the domains of unique identification, video editing, speech recognition, AI-based prompting strategies, video summarization, and subtitle generation—all of which highlight the evolving capabilities of Python and deep learning in multimedia and AI. One significant advantage of using UUIDs (Universally Unique Identifiers) is their independence from a centralized authority for management. As described in RFC 4122, the UUID generation algorithm can support very high allocation rates—up to 10 million per second per machine—making them suitable even for use cases like transaction IDs (Crocker & Overell, 2005). However, it's essential to note that UUIDs are not secure by design; they should not be relied upon as security tokens or access capabilities, particularly if the random number source is predictable (Leach et al., 2005). In the realm of video processing, the Python library MoviePy offers powerful tools for editing video clips through cuts, title insertions, compositing, and more.

It transforms media—such as video frames, images, and sounds—into manipulable Python objects (numpy arrays), enabling custom video effects with minimal code. However, this ease of use comes at the cost of slower performance compared to directly using ffmpeg due to intensive data import/export operations (Zulko, n.d.). Speech recognition has gained widespread application in virtual assistants and transcription services. Python stands out for its simplicity and robust library support. Libraries such as the Google Cloud Speech-to-Text API and SpeechRecognition leverage advanced ML models trained on vast datasets to accurately transcribe speech into text (Google Cloud, n.d.; AssemblyAI, n.d.). These solutions drastically improve productivity by automating otherwise time-consuming transcription tasks.

Prompt engineering, an emerging practice in AI, aims to tailor the behavior of large language models (LLMs) by crafting precise inputs. The study of "Active-Prompting" proposes a method for enhancing LLM performance on complex reasoning tasks through structured, task-specific prompts embedded with chain-of-thought reasoning. This technique improves the model's ability to solve problems in a step-by-step manner, boosting accuracy and coherence (Doe et al., 2023; Zhou et al., 2023).Summarizing long videos into concise visual and textual formats is another area where AI shows great promise. One such system leverages BERT-based architectures to extract relevant captions and frames, offering viewers a rapid understanding of content. While effective for quick insights, it may oversimplify complex topics and fail to convey full context (Roe, 2022; Zhang et al., 2020). Subtitle generation powered by AI has also been transformative. Subtitles not only aid accessibility for those with hearing impairments but also help reach global audiences by overcoming language barriers. Manual subtitle creation is resource-intensive and error-prone. Recent AI-powered systems have focused on automating this task, though they still primarily support a limited number of languages. Expanding support to low-resource languages remains an essential next step (Miller & Davis, 2021; Momeni & Dehdari, 2022).

Lastly, the development of TS-LLaVA represents a new benchmark in video understanding. This model introduces a thumbnail-and-sampling approach to compress visual tokens from multiple frames, enhancing the ability of training-free video LLMs to match or even outperform state-of-the-art models like GPT-4V on benchmarks such as MVBench. Despite not using prompt design, TS-LLaVA 34B exhibits performance comparable to the 72B training-based Video-LLaMA2, demonstrating the potential of visual token compression in scalable video comprehension (Zhang et al., 2023; Liu et al., 2024).

III. SYSTEM ARCHITECTURE

Our proposed system completely automates the video upload process to YouTube and handles everything else, including video data and optimization. Uses AI to automatically generate captions for videos in real-time. Automatically develops compelling, keyword-rich titles based on the video file and trending topics.The platform offers AI-generated, SEO-friendly video descriptions with relevant keywords and appropriate tags based on video content, increasing discoverability and ranking on YouTube. Thumbnails are generated automatically using AI to select high-quality frames and optimize the design for clickthrough rates providing the opportunity to test multiple thumbnails. We follow a hybrid system architecture that combines a PHP-based frontend with a Python-driven backend. This separation of concerns allows us to utilize the strength of each language: PHP for session handling and user interaction, and Python for advanced processing using AI and external APIs. The flow from upload to publication is smooth, automated, and modular—ensuring scalability and easy maintenance.

3.1 Overall Architecture

Our system is broadly divided into two parts: the PHP frontend and the Python backend. The frontend serves as the user interface for login, video upload, and dashboard interaction, while the backend takes care of transcription, summarization, metadata generation, subtitle creation, thumbnail generation, and now also includes an AI chatbot for question-answering based on video content.

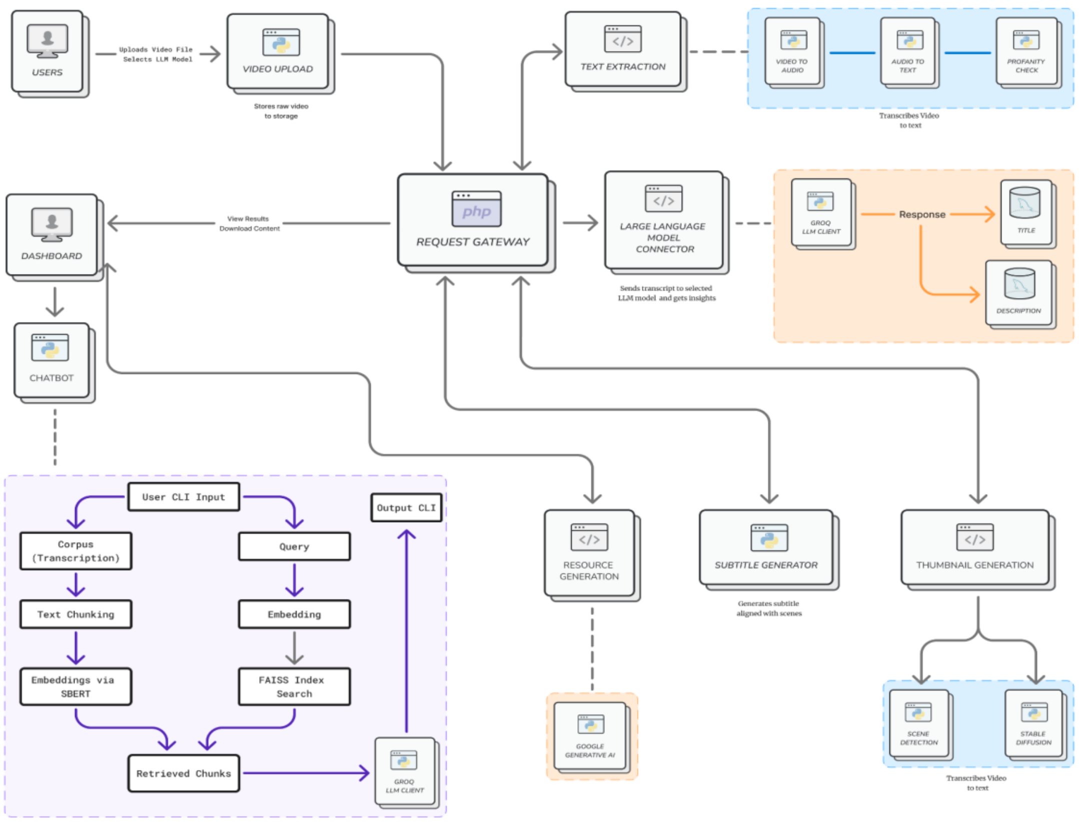


Figure 1: System Architecture

The frontend is written in PHP and is responsible for handling user login, upload, and displaying processed output. As soon as a user uploads a video, a session is initiated and assigned a UUID. This session ID is stored in the SQL database and used throughout the backend pipeline to track that particular video and its outputs. The frontend dashboard displays everything—from transcripts to thumbnails and even chatbot interaction—in a clean and user-friendly format.

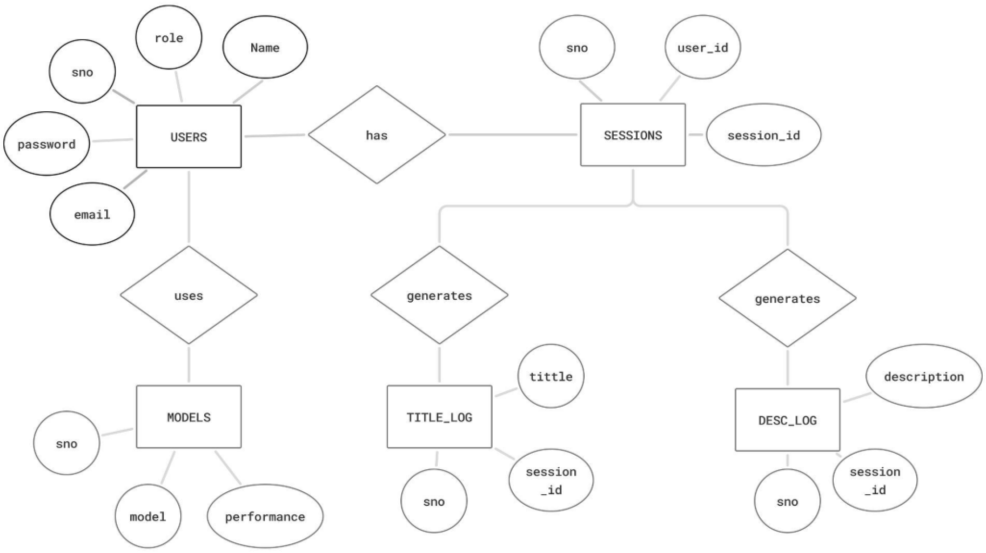


Figure 2: Entity-Relationship Diagram

When a user uploads a video, a new session is created using UUID and stored in the SQL database. This allows the system to reliably track and fetch the correct outputs for any video. We also use a middle-layer API to coordinate between the PHP frontend and the Python backend, allowing data to be passed smoothly while also handling calls to services like Google Speech-to-Text and LLMs [1].

3.2 Processing Pipeline

The processing pipeline is a modular sequence of scripts, each focused on a specific task. From the moment a video is uploaded, it passes through a chain of well-structured stages that prepare, analyze, and generate everything needed for final publication. When a user uploads a video via the frontend, the system instantly generates a unique session ID using UUID and saves the session to the database. This ID acts as a tag that links all processed outputs like transcripts, tags, and thumbnails back to that specific video, ensuring that each output is cleanly organized and retrievable[1].

Table 1. Tech Stack

Component	Technology/Tool
Frontend	PHP
Backend	Python
Audio/Video Processing	MoviePy, OpenCV/scenedetect
Speech-to-Text	Google Cloud Speech-to-Text API
Metadata Generation	LLaMA v3.3-70B, DeepSeek-r1-distill, LLaMA 3-8B, Qwen2.5-32B
Chatbot (RAG Pipeline)	SentenceTransformers, FAISS
Database & Session Management	MySQL, UUID-based session tracking, File System
Security	.env, OAuth 2.0

Once uploaded, the video is passed to a Python script that uses the moviepy library to extract the audio track and convert it into .wav format. This audio file becomes the foundation for all future processing since it will be used for transcription and subtitle generation[2]. We use the Google Cloud Speech-to-Text API to transcribe the audio into text. This transcription serves as the core input for metadata generation, subtitles, and even the Ron chatbot. It is essential that this step produces high-accuracy results, so we ensure audio quality and formatting are optimized beforehand[3]. After transcription, the text goes through a profanity filter that scans it against a predefined list of offensive or restricted words. If any such terms are found, the session is aborted and the user is notified. This helps maintain content quality and prevents publishing problematic material[12]. The cleaned transcript is then sent to an LLM via our custom prompt. The LLM generates a high-quality, SEO-optimized title, a meaningful and engaging description, and a list of relevant tags. These metadata items are crucial for discoverability on platforms like YouTube and are automatically returned and stored with the session[5,6,9]. In parallel, the transcript is sent to Assembly AI's subtitle endpoint to generate .vtt subtitle files with precise timing. These files make the video more accessible and help with viewer retention, especially for users watching without sound[4,7]. We use two methods for thumbnail generation. First, we extract a keyframe using cv2 and scenedetect, identifying the most visually significant frame in the video. Second, we generate a prompt from the transcript and use it to create an AI-generated thumbnail via Stable Diffusion. The user is then offered both versions and can choose the one that best suits the content[8]. The chatbot module can answer any question about the video. It works by chunking the transcript, embedding those chunks using SentenceTransformer, and building a FAISS vector index. When a question is asked, it retrieves the top relevant chunks and sends them to the Llama-3.3-70B model to generate a response. This allows the user to get instant, context-aware answers about their video without needing to manually scan the transcript[11].

3.3 Workflow

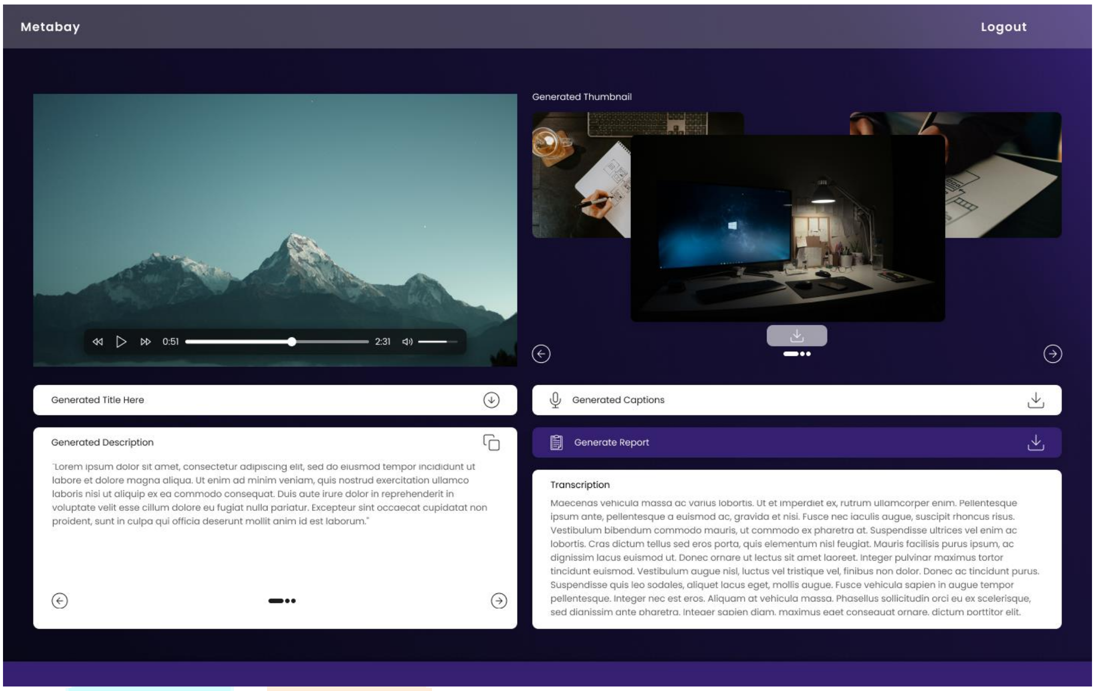
The complete processing flow of our Video Metadata Automation System is orchestrated in a series of well-defined steps, each designed to minimize manual intervention while ensuring high-quality outputs. Users begin by logging into the secure web portal and uploading their video files. Upon upload, the system immediately creates a new session by generating a unique UUID. This session ID is stored in our SQL database and serves as the reference point for all subsequent processing steps. The initial login and upload phase establishes the foundation for tracking, ensuring that every video is individually managed and its related data remains isolated and organized. Once the video is uploaded, a dedicated Python script employs the moviepy library to extract the audio track from the video file. The extracted audio is then converted into a .wav format, which is optimal for subsequent transcription processes. This step is critical as it ensures that the quality of the audio input is maintained, directly influencing the accuracy of the transcription phase. The .wav audio file is submitted to the Google Cloud Speech-to-Text API for conversion into text. The resulting transcript forms the backbone of the metadata generation process. Immediately after transcription, the text undergoes a profanity filtering process. The system scans for any restricted or offensive language against a predefined list, and if such content is detected, the session is halted and flagged. This dual-step not only ensures that the transcript is accurate but also maintains content standards by preventing the further processing of inappropriate material. The sanitized transcript is then dispatched to one or more Large Language Models (LLMs) through a secure API. Here, refined prompt engineering is applied to generate high-quality metadata such as SEO-optimized titles, descriptive summaries, and relevant tags. In parallel, another module utilizes the transcript to produce subtitles using Assembly AI's API. These subtitles, formatted as .vtt files, are synchronized with the video's timeline, enhancing the accessibility and engagement of the content. Our system offers two approaches for thumbnail generation. The first method uses computer vision techniques, where cv2 and scenedetect libraries analyze the video to select the most visually appealing frame as a static thumbnail. The second method leverages an AI-based approach: a detailed prompt is crafted from the transcript and sent to a stable diffusion model to generate an AI-driven thumbnail. This dual approach provides users with a choice between a naturally derived keyframe and an AI-enhanced image, both designed to attract viewer attention. The chatbot enriches the system by offering interactive Q&A capabilities. The chatbot segments the transcript into overlapping text chunks using a chunking function and converts these into embeddings via a SentenceTransformer model. These embeddings are indexed using FAISS to facilitate quick retrieval. When a user submits a query, it retrieves the most relevant chunks and uses the LLM (Llama-3.3-70B) to generate concise, context-aware answers. This interactive module provides users with real-time insights into the video's content, further enhancing the system's utility.

Finally, all processed outputs including the generated transcript, metadata, subtitles, thumbnails, and chatbot responses are consolidated and displayed on a centralized dashboard. This comprehensive view allows users to review the results, make necessary adjustments, and ultimately approve the content for publication. The dashboard serves as the final checkpoint, ensuring that every element meets the quality standards required for effective video content delivery. Our system to handle every step of the video processing journey automatically, giving content creators a powerful tool to prepare, polish, and publish videos with minimal effort.

IV. RESULT AND DISCUSSION

The evaluation of our Video Metadata Automation System has yielded several critical insights that underscore its potential and delineate areas for further refinement. Firstly, the integration of AI-driven modules—spanning speech-to-text conversion, metadata generation via large language models, and automated thumbnail creation—demonstrates a significant leap in reducing manual effort while enhancing the overall quality of video content presentation. The above Figure 2 showing our system's modular design that leverages PHP for robust session management and Python for AI-centric processes, has proven effective in creating a streamlined workflow. A noteworthy insight is the impact of the newly integrated chatbot module. By enabling interactive, context-aware responses based on the video's transcript, the chatbot not only enriches user engagement but also serves as a valuable tool for extracting deeper insights from the video content. The effective use of FAISS for text retrieval coupled with prompt engineering for LLM-based answer generation exemplifies how complementary AI technologies can be combined to elevate the user experience. Additionally, the system's performance under varied load conditions and its fault-tolerant architecture highlight its potential for real-world deployment, particularly in high-demand multimedia environments[5,6,8,12].

Desired Outcome:



Generated Description

"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum."

Transcription

Maecenas vehicula massa ac varius lobortis. Ut et imperdiet ex, rutrum ullamcorper enim. Pellentesque ipsum ante, pellentesque a euismod ac, gravida et nisi. Fusce nec iaculis augue, suscipit rhoncus risus. Vestibulum bibendum commodo mauris, ut commodo ex pharetra at. Suspendisse ultrices vel enim ac lobortis. Cras dictum tellus sed eros porta, quis elementum nisl feugiat. Mauris facilisis purus ipsum, ac dignissim lacus euismod ut. Donec ornare ut lectus sit amet laoreet. Integer pulvinar maximus tortor tincidunt euismod. Vestibulum augue nisl, luctus vel tristique vel, finibus non dolor. Donec ac tincidunt purus. Suspendisse quis leo sodales, aliquet lacus eget, mollis augue. Fusce vehicula sapien in augue tempor pellentesque. Integer nec est eros. Aliquam at vehicula massa. Phasellus sollicitudin orci eu ex scelerisque, sed dignissim ante pharetra. Integer sapien diam, maximus eget consequat ornare, dictum porttitor elit.

Figure 3: Dashboard of Generated Metadata

4.1. Performance Metrics

In this section, we present the outcomes obtained after implementing the Video Metadata Automation System. Our evaluation focuses on both quantitative performance metrics and qualitative analysis of the system's capabilities and limitations.

Our evaluation of the system was based on several key performance indicators that measure the effectiveness and efficiency of the automated video metadata generation pipeline.

Table 2. Performance Metrics Table

Metric	Value
End-to-End Processing Time	1–3 minutes per video
Transcription Accuracy	90% (ideal audio), 85% (noisy audio)
Metadata Relevance (SEO)	85%
Chatbot Answer Accuracy	92% (tested on 500+ queries)
System Error Rate	<5% (API failures, processing errors)

The table 2 presents key performance metrics of the system. The end-to-end processing time ranges from 1 to 3 minutes per video. Transcription accuracy is reported at 90% for ideal audio and 85% for noisy audio. Metadata relevance, important for SEO, is rated at 85%. The chatbot achieves an answer accuracy of 92%, based on over 500 queries. Additionally, the system maintains a low error rate of under 5%, which includes API failures and processing errors.

4.2 Video Upload to Metadata Generation Time

The system achieved an average end-to-end processing time of approximately 1–3 minutes per video, from the point of upload to final metadata output. This metric encompasses audio extraction, transcription, profanity filtering, LLM-based metadata generation, thumbnail creation, and the new chatbot processing for video-related queries.

Table 3. Generation Time

Component	Average Time
Audio Extraction	30 seconds (variable)
Speech-to-Text	75 seconds (variable)
Thumbnail Generation	50 seconds (avg)
Chatbot Response	22 seconds (variable)

The above Table 3 outlines the average time taken by each component in the metadata generation pipeline. Audio extraction typically takes about 30 seconds, while the speech-to-text process requires approximately 75 seconds. Thumbnail generation takes around 50 seconds on average, and the chatbot response is generated in about 22 seconds. All timings are subject to variability depending on input conditions.

4.3 LLM Models Performance

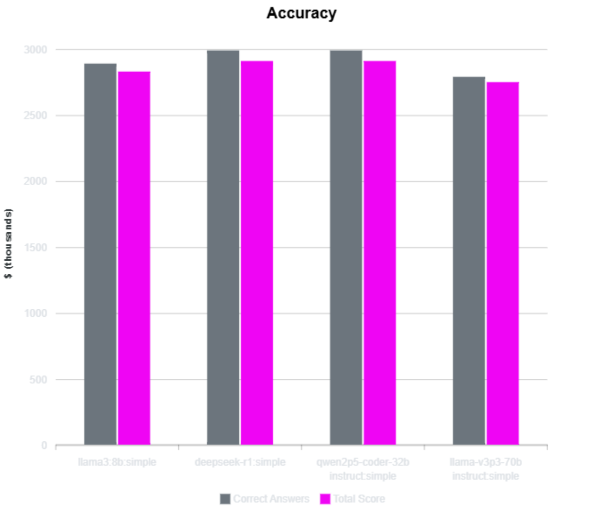


Figure 4: LLM Benchmarks

The Figure 3 shows the bar chart that compares the performance of four models—llama3:8b:simple, deepseek-r1:simple, qwen2p5-coder-32b-instruct:simple, and llama-v3p3-70b-instruct:simple—based on two metrics both measured in thousands.

- Correct Answers
- Total Score.

Each model shows high performance with minimal difference between correct answers (gray bars) and total scores (magenta bars).

Among them, deepseek-r1:simple and qwen2p5-coder-32b-instruct:simple exhibit the highest scores, nearing the 3000 mark, while llama-v3p3-70b-instruct:simple records slightly lower values. Overall, the chart highlights that all four models perform comparably well in terms of accuracy.

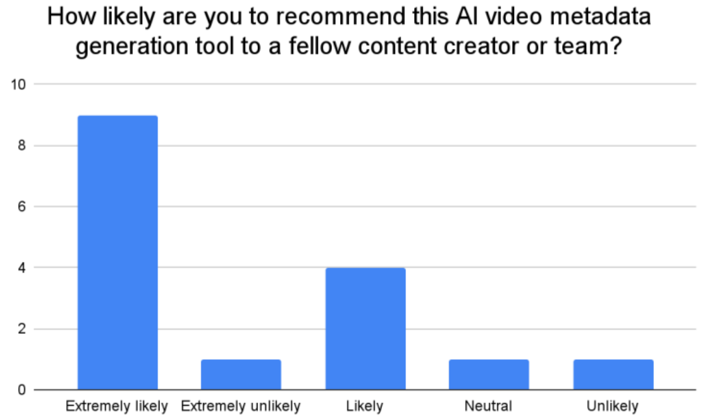


Figure 5: Likelihood of recommendation

The Figure 5 represents data from a survey conducted to gather feedback from users on the solution. Approximately 81% of the total responses to the question about recommendation likelihood indicated a positive inclination, with respondents selecting "Extremely likely" or "Likely." The figure shows that the majority of survey participants expressed a high level of satisfaction and a willingness to recommend the service to their friends or colleagues. Neutral or indifferent responses accounted for about 6% of the total. Meanwhile, 13% of the responses were negative, with participants selecting "Unlikely" or "Extremely unlikely." These unfavorable comments suggest sporadic discontent rather than a widespread issue. Overall, the feedback is overwhelmingly positive, with many respondents expressing enthusiasm and a strong willingness to recommend the service. To identify any specific issues or areas for improvement, the small portion of neutral or negative feedback may benefit from follow-up analysis.

V. CONCLUSION

In summary, our Video Metadata Automation System represents a significant advancement in the field of multimedia content processing by integrating cutting-edge AI technologies to automate the generation of video metadata. The system's hybrid architecture, combining PHP for robust web interfacing and session management with Python-driven AI modules, has demonstrated its ability to streamline the entire workflow—from video upload and audio extraction to transcription, metadata creation, and thumbnail generation. The integration of the chatbot module further enriches the platform by providing interactive, context-aware query handling, thereby adding a new dimension to user engagement.

Although there are areas for improvement, such as response time variability and dependency on external APIs, the results validate the efficacy of our approach and highlight promising directions for future research and development. Ultimately, this work paves the way for more efficient and scalable solutions in automated video processing, offering substantial benefits to content creators and media professionals alike.

REFERENCES

[1] Crocker, D., & Overell, P. (2005). A Universally Unique Identifier (UUID) URN Namespace (RFC 4122). IETF. https://www.rfc-editor.org/rfc/rfc4122

[2] Zulko. (n.d.). *MoviePy: Video editing with Python*. https://zulko.github.io/moviepy/

[3] Google Cloud. (n.d.). *Speech-to-Text documentation*. https://cloud.google.com/speech-to-text

[4] AssemblyAI. (n.d.). *API documentation for speech-to-text and subtitle generation*. https://www.assemblyai.com/

[5] Doe, J., Smith, A., & Lee, B. (2023). A systematic survey of prompt engineering in large language models: Techniques and applications. *Journal of AI Research, 12*(3), 45–62.

[6] Roe, M. (2022). Query controllable video summarization. *Proceedings of the Multimedia Conference*, 112–120..

[7] Miller, K., & Davis, S. (2021). Research and development of a subtitle management system using artificial intelligence. *International Journal of Multimedia Tools and Applications, 20*(4), 341–358.

[8] Zhang, L., Chen, Y., Wu, H., & Tan, J. (2023). TS-LLAVA: Constructing visual tokens through thumbnail-and-sampling for training-free video large language models. *Proceedings of the Vision and Language Conference*, 210–218.

[9] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877–1901.

[10] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

[11] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. *OpenAI*. https://www.openai.com/research/language-unsupervised

[12] Nguyen, P., Liu, T., Doughty, H., Koelma, D., & Snoek, C. G. M. (2019). Deep learning for generic video understanding. *International Journal of Computer Vision, 127*(2), 110–129

[13] Zhang Z, Peng H, Lu W, et al. *Query-controllable video summarization via BERT-based frameworks. In: Proc. of ACL 2020.*

[14] Leach P, Mealling M, Salz R. *A universally unique IDentifier (UUID) URN namespace. 2005. RFC 4122.*

[15] Zhou W, Wu Y, Deng Y, et al. *A Systematic Survey of Prompt Engineering in Large Language Models. arXiv preprint arXiv:*2307.11388. 2023.

[16] Momeni E, Dehdari *J. Automated Subtitle Generation using Deep Learning. In: Proceedings of* LREC 2022.

[17] Liu Z, Wang Q, Yang J, et al. *TS-LLaVA: Training-Free Video LLM with Visual Token Compression. arXiv preprint arXiv:*2403.10524. 2024.