2025 3rd International Conference on Intelligent Systems, Advanced Computing and Communication (ISACC) | 979-8-3315-2389-3/25/$31.00 ©2025 IEEE | DOI: 10.1109/ISACC65211.2025.10969233

ClipXpert: Automated Clip Mining from Video Data for High-Demand Content

Rajdeep Chatterjee
School of Computer Engineering
KIIT Deemed to be University
Bhubaneswar, India
cse.rajdeep@gmail.com

Sudip Chakrabarty
School of Computer Engineering
Kalinga Institute of Industrial Technology
Bhubaneswar, India
sudipchakrabarty6@gmail.com

Pappu Bishwas
School of Computer Engineering
Kalinga Institute of Industrial Technology
Bhubaneswar, India
pappuovi8@gmail.com

***Abstract*—ClipXpert is a new system that automates the extraction of relevant clips from YouTube videos using user-defined keywords or frequently used word or comment analysis. The system uses advanced transcription models to search a pre-existing database for lines associated with the provided keywords. If the database doesn't contain the keywords, it dynamically identifies and stores relevant content for future use. ClipXpert also automates identifying frequently used nouns, adjectives, and adverbs to generate focused content highlights without user intervention. It also performs sentiment analysis on YouTube comments to understand audience engagement and reception. ClipXpert enhances video content extraction efficiency and accuracy, catering to the growing demand for targeted, high-value clips in the digital media landscape.**

***Index Terms*—Clip mining, video segmentation, keyword based video extraction, high-demand content.**

I. Introduction

The explosion of video content on platforms like YouTube has made it both a valuable resource and a daunting challenge for users seeking specific information. With millions of videos being uploaded daily, the task of finding precise, relevant content within vast video archives has become increasingly difficult. Users often face the frustration of sifting through lengthy videos, spending significant time and effort to locate segments that are most relevant to their needs.

In high-demand contexts, where time is of the essence, the ability to quickly and accurately extract meaningful video clips is crucial. Traditional methods of video navigation, such as manual searching and skimming, are not only time-consuming but also often ineffective in delivering the desired results. This has sparked a need for innovative solutions that can streamline the process of retrieving key video segments.

A. Contribution

ClipXpert emerges as a solution to this problem by offering an automated approach to video clip mining, specifically designed to cater to the demands of users who require quick access to targeted content. Rather than summarizing entire videos, ClipXpert focuses on extracting clips based on user-specified keywords. The system employs advanced transcription models to convert video speech into text, which is then analyzed to locate the exact segments that match the user's query. A key feature of ClipXpert is its ability to maintain a dynamic database that stores previously extracted content. This ensures that future searches for the same keywords can be processed more rapidly, enhancing the system's efficiency over time. Additionally, ClipXpert automates the identification of frequently used parts of speech, such as nouns and adjectives, to create content highlights, further reducing the need for user intervention. Moreover, ClipXpert incorporates sentiment analysis of YouTube comments, offering insights into how the content is perceived by viewers. This additional layer of analysis provides valuable context and aids in refining the selection of video clips.

This paper outlines the development and evaluation of ClipXpert, highlighting its potential to transform the way users interact with and extract the high-demand contents from the original video.

B. Organization:

The paper contains six sections. Section II provides a discussion of the related works. The proposed pipeline has been discussed in Section III. Section IV explains the methodologies. Section V analyses the performance of the proposed pipeline. Finally, We conclude the work and discuss the future scopes in Section VI.

II. Related Works

Video segmentation is crucial for various applications. Recent deep learning approaches have improved performance. This research uses natural language processing to automate text summarization in YouTube videos. The system uses the term frequency inverse document frequency (TF-IDF) method to extract important keywords. The goal is to summarize long videos quickly and efficiently, benefiting students and researchers who have limited time to spend on long videos. The results were evaluated using the Rouge method on a CNN-dailymail-master data set [1]. This article discusses temporal video segmentation techniques for automatic annotation of digital video, comparing performance, merits, and limitations of uncompressed and compressed methods. It also reviews shot boundaries detection algorithms and camera operation recognition [2]. The study explores methods for skimming

Amygdala AI, is an international volunteer-run research group that advocates for *AI for a better tomorrow* http://amygdalaai.org/.

979-8-3315-2389-3/25/$31.00 ©2025 IEEE

video streams into semantically consistent segments, focusing on spatial-temporal segmentation. It also explores temporal segmentation of multidimensional time series, reducing data processing time. The study uses multidimensional time series analysis theory to identify shots with homogeneous characteristics, using VAR models, exponential smoothing, and predictive models [3]. The study presents a temporally distributed network for fast video semantic segmentation, utilizing temporal continuity in videos. It extracts sub-features from a single sub-network, recomposes full features using an attention propagation module, and introduces a knowledge distillation loss. Experiments show state-of-the-art accuracy, faster speed, and lower latency [4]. Automatic temporal video scene segmentation is an open problem with no definite solutions. Multimodal techniques, such as early fusion or late fusion, show better results. Convolutional neural networks (CNN) have been used to extract features from multiple data sources, but they struggle to learn temporal cues. Recurrent neural networks (RNN) can help. This paper proposes a new multimodal approach that combines CNN and RNN capabilities, achieving better efficacy results on a public video dataset [5].

We introduce a simple pipeline to segment video temporally based on the most frequent words/phrases used in the audio of the same video. We develop the pipeline using SOTA, an open-source deep learning model. We also store the extracted temporal information in a database to minimize the time required to provide a solution in the future.

III. Proposed Pipeline

The proposed ClipXpert system follows a structured approach to generate keyword-based highlighted videos efficiently.

The ClipXpert system initiates its process by accepting a YouTube video link, at which point it prompts the user to provide specific keywords of interest. Upon receiving the link, the system transcribes the video's audio into text. If the user does not provide keywords, the system automatically identifies key terms from the transcript by analyzing the most frequently used words or by extracting sentiment-weighted terms from the comment section. This allows the system to generate relevant keywords that capture the essence of the video content.

Once the keywords are identified, whether supplied by the user or generated by the system, the video ID, keywords, and relevant transcript lines are stored in a database to ensure efficient access in future interactions. The system then searches the transcript for the specified or identified keywords. If the keyword already exists in the database, the process is significantly expedited through the use of pre-stored data, resulting in faster performance. The final step involves retrieving the relevant lines from the transcript and generating a highlighted video that focuses on the content associated with the selected keywords. This streamlined approach ensures that users receive a concise and relevant summary of the video content in a highly efficient manner (refer to Algorithm 1).

Algorithm 1: Step-by-Step Procedure for Generating Highlighted Video

Input: YouTube video URL u, User-specified keywords set $K = \{k_1, k_2, ..., k_n\}$
Result: Highlighted video H

```
/* Check if we've processed this video before */
Let D be the database, v be the video id;
if (v, K) ∉ D then
    /* Process new video */
    A ← ConvertToAudio(u) ;   /* A is WAV format audio */
    T ← Transcribe(A) ; /* T is the transcript */
    if K ∩ WordSet(T) ≠ ∅ then
        /* User-specified keywords found */
        H ← GenerateHighlight(u, K, T);
        D ← D ∪ {(v, K, T, H)};
    else
        /* User keywords not found, try alternative methods */
        if |WordSet(T)| > 0 then
            /* Use frequently occurring words as keywords */
            W ← TopFrequentWords(T, 3);
            H ← GenerateHighlight(u, W, T);
            D ← D ∪ {(v, W, T, H)};
        else
            /* Analyze comments if no keywords can be derived */
            C ← RetrieveComments(u);
            S ← SentimentAnalysis(C);
            W ← IdentifyKeywords(S);
            H ← GenerateHighlight(u, W, T);
            D ← D ∪ {(v, W, T, H)};
        end
    end
else
    /* Video already processed, retrieve existing data */
    (K', T', H') ← D[v];
    H ← GenerateHighlight(u, K ∪ K', T');
end
return H /* Highlighted video is now ready for viewing */
```

The overall system flow is illustrated in the block diagram (shown in Fig. 1), providing a visual representation of the ClipXpert process from begining to the generation of the highlighted video.

IV. Methodology

In this section, we detail the methodology employed by the ClipXpert system to efficiently generate highlighted videos based on user-defined or system-identified keywords. The process is designed to be both dynamic and scalable, ensuring

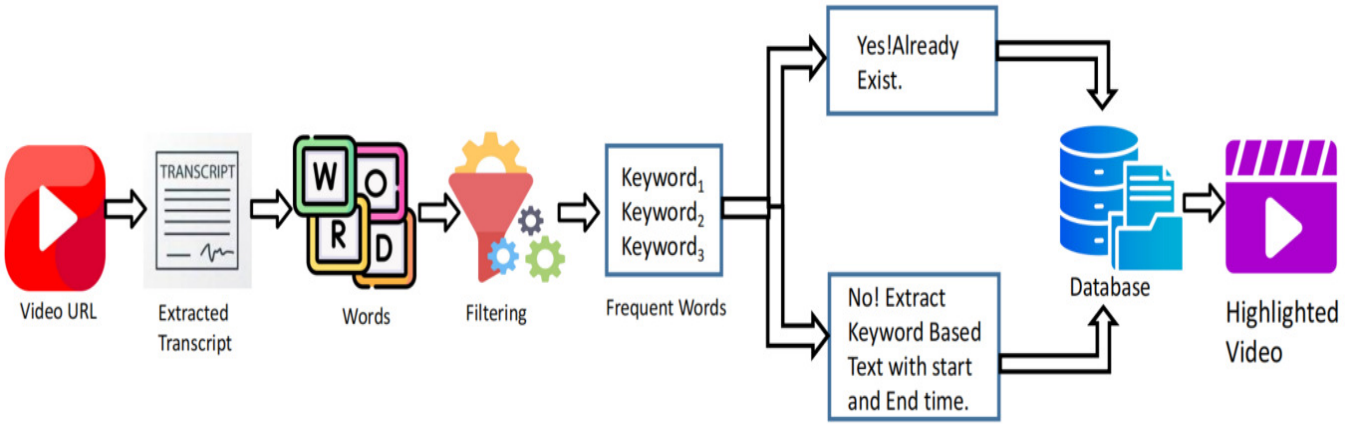


Fig. 1: Block Diagram of the proposed solution, showcasing the process from video input to the generation of a highlighted video based on user-defined keywords

that users receive content tailored to their specific interests while also leveraging advanced transcription and keyword analysis techniques. To provide a clear understanding of the process, we present a flowchart illustrating the step-by-step procedure of the methodology (shown in Fig. 2).

A. User Input and Keyword Selection

The process begins when the user provides a YouTube video link. Following this, the system prompts the user to specify keywords of interest.

B. Transcription Generation

After receiving the video URL, the system first extracts the audio track from the video. This audio is then saved in WAV format to ensure high-quality processing. Once extracted, the WAV file is ready for further processing, including transcription and keyword analysis. The system employs the advanced OpenAI Whisper[1] model to convert the audio into a textual transcript. The transcript is temporarily stored in a JSON file, serving as an intermediate step before further analysis. This transcript is a crucial step in identifying relevant content based on the provided video URL, enabling precise keyword extraction and accurate video highlights.

C. Choosing Appropriate Keywords

In cases where the user's provided keyword is not found within the full transcript, an error message is displayed to inform the user. To enhance the process, the system also performs additional analysis: it extracts the most frequent words from the transcript to identify potential keywords and conducts sentiment analysis on comments associated with the video. This dual approach ensures that even if the initial keywords are missing, relevant content can still be highlighted based on prevalent terms and user sentiment.

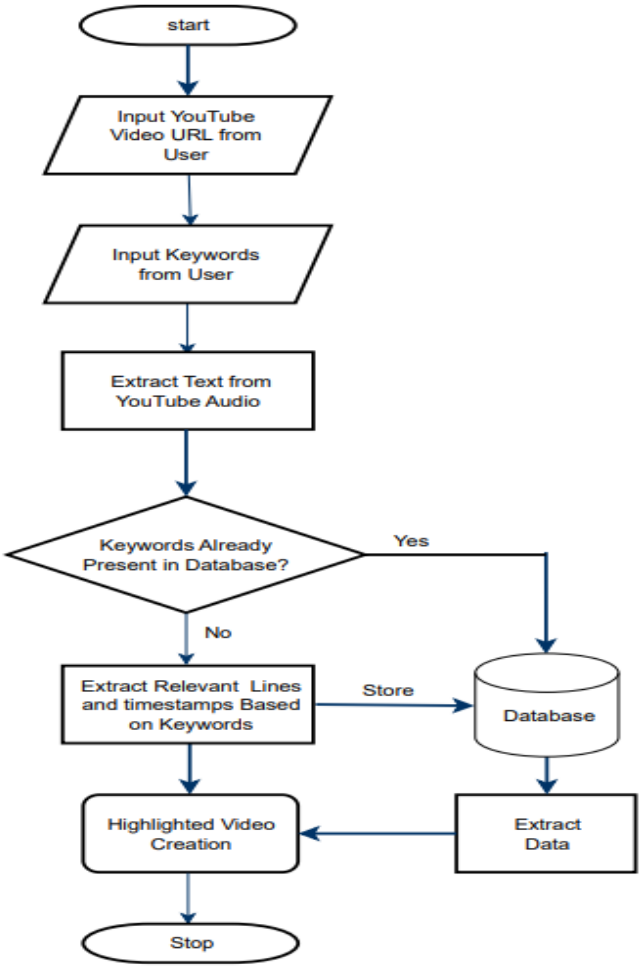


Fig. 2: Flowchart outlining the key stages of the proposed pipeline.

[1]OpenAI Whisper: https://github.com/openai/whisper

In addition to the keyword identification process, the system utilizes the Natural Language Toolkit (NLTK)[2] library to perform advanced text analysis. Specifically, NLTK is used for tokenization and part-of-speech tagging to extract and analyze frequent words from the comments section [6]. This approach not only refines the keyword extraction process but also enhances the system's ability to generate a more accurate and relevant highlighted video by focusing on significant terms and their context.By employing NLTK for part-of-speech tagging, the system effectively classifies and analyzes the text from user comments. This classification helps distinguish between meaningful terms such as nouns, verbs, adjectives, and adverbs, and less relevant parts of speech.Irrelevant words, including auxiliary verbs and specific adverbs, are systematically removed to ensure that the analysis centers on the most impactful and contextually significant terms. This targeted approach guarantees that the extracted keywords are not only relevant but also enhance the overall quality of the highlighted video. By focusing on essential terms and excluding less informative ones, the system maintains the integrity and relevance of the content, ultimately leading to a more precise and engaging final output.

D. Frequent Words Extraction

If the specified keywords are not found in the transcript, the system employs a secondary strategy to ensure relevant content is still highlighted. It starts by analyzing the transcript to determine the frequency of each word, focusing specifically on nouns, adjectives, adverbs, and verbs. This selective focus helps in capturing the essential elements of the content. The system then ranks these words based on their frequency of occurrence, selecting the top three most prevalent terms. This method helps identify significant content segments that may not have been directly addressed by the initial keywords but are still likely to be of interest to the user. By doing so, the system ensures that the generated video highlights portions of the transcript that are contextually relevant and informative.

Furthermore, to enhance the relevance of the highlighted content, the system excludes keywords with high frequency but low contextual relevance. This step ensures that the final highlighted video maintains a high quality of content, effectively reflecting the user's interests and improving the overall utility of the video summarization process.

E. Sentiment Analysis on Comments

In addition to extracting frequent words from the transcript, the system also performs sentiment analysis on the comments associated with the video. This analysis helps to understand the general sentiment and emotions expressed by viewers. By identifying keywords and topics that resonate with the audience, the system can incorporate these insights into the video highlighting process. This ensures that the content highlighted not only matches prevalent terms but also aligns with user sentiment and feedback [7], [8], [9], [10].

To extract YouTube comments, we use the Playwright[3] library for browser automation. First, we install and set up Playwright, which allows for headless browser interactions. We then create an asynchronous function to open a Chromium browser and navigate to the YouTube video URL. The function automatically scrolls through the page multiple times to load additional comments, since YouTube's comment section loads comments dynamically as the page is scrolled. Comments are extracted by querying the page's HTML for text elements containing comment content. This approach ensures that all visible comments are captured, even those loaded later during the scrolling process. Finally, the collected comments are saved to a text file, each labeled with its order in the sequence. This method effectively handles YouTube's dynamic content and provides a complete set of comments for further analysis.

F. Database Storage and Retrieval

In our study, we compared three database systems—XML, JSON, and MongoDB—to evaluate their effectiveness in handling video metadata and keyword information.

1) XML: XML provides a structured and hierarchical format with strict schema definitions, making it ideal for applications needing well-defined data structures but potentially cumbersome due to its verbosity [11].

2) JSON: JSON is a lightweight, human-readable format known for its simplicity and ease of use, particularly in web applications. Its flexible nature allows for straightforward data manipulation and integration, making it a popular choice for modern, data-driven environments [12], [13].

3) MongoDB: MongoDB, a NoSQL database, excels in managing large volumes of unstructured data with high performance and scalability. Its document-oriented model supports dynamic schemas, enabling efficient querying and retrieval of complex data structures. This flexibility makes MongoDB well-suited for applications requiring rapid access and scalability [14], [15].

The databases are designed to facilitate faster processing for future queries by storing and the lines associated with the identified keywords.The storing process of data in the database is shown below in Fig. 3.

G. ST (Start Time)

Refers to the beginning timestamp of a segment within the video that is of interest. The duration is measured in seconds. This timestamp is crucial for identifying when a specific segment starts.

H. ET (End Time)

Denotes the ending timestamp of a segment, marking the conclusion of the segment of interest.

[2]NLTK: https://github.com/nltk/nltk

[3]Playwright: https://github.com/microsoft/playwright

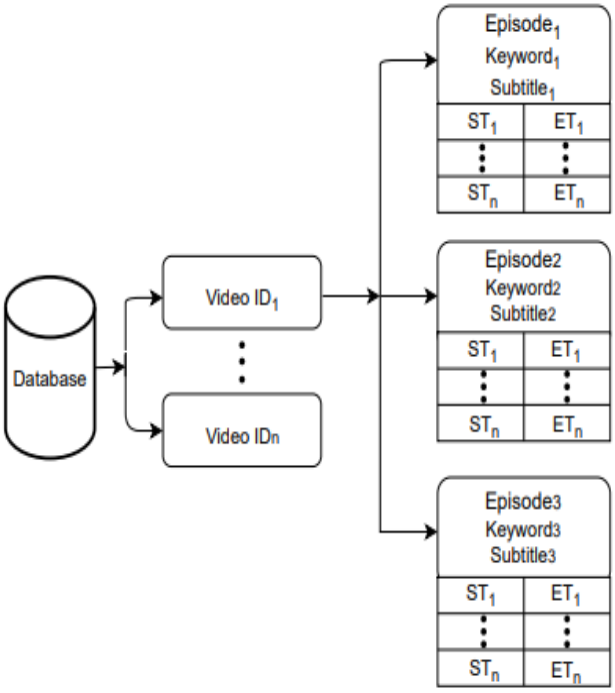


Fig. 3: Database Structure

I. highlighted video creation

After retrieving data from the database, which includes the relevant timestamps (start and end times) and corresponding subtitles based on the specified keywords, a highlighted video is created. This process involves assembling the video segments that match the keywords, ensuring that the final output accurately reflects the desired content. In cases where the specified keyword is not found in the database, the system first generates the highlighted video from the segments corresponding to the provided keywords. Following this, the newly discovered keyword, along with its associated start and end times, is added to the database. This continuous updating of the database with new keywords and their respective time-frames not only enriches the existing data but also improves the system's efficiency in handling future queries and generating relevant video highlights. By incorporating new data, the system progressively enhances its capability to quickly and accurately respond to user needs, ensuring an increasingly robust and responsive video analysis tool. In Fig. 4, several notations are used to denote specific elements related to video processing:

J. SOV (Start of Video)

Represents the starting point of the given video. This is the initial point from which the video is considered for processing, highlighting, or analysis.

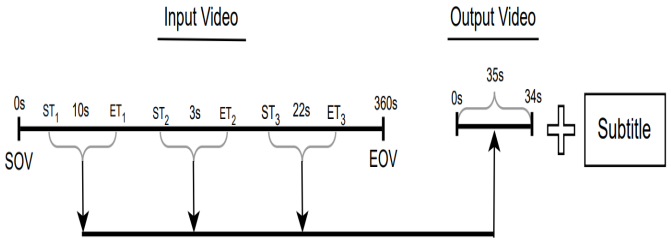


Fig. 4: Highlighted Video Creation Process

K. EOV (End of Video)

Indicates the endpoint of the video. This timestamp signifies where the video ends, encompassing all relevant content between SOV and EOV.

During the creation of the highlighted video, multiple timestamps are collected based on the keywords. These timestamps represent various segments of the video where the keywords are relevant. The system aggregates all these start and end times to produce the final highlighted video.

Additionally, corresponding subtitles are generated for each segment identified by the timestamps. This consolidation ensures that all significant content, including both video segments and their subtitles, is included in the final output. By combining these segments and their subtitles, the system ensures that the final video effectively captures and highlights the key content based on the specified criteria.

V. Performance Analysis

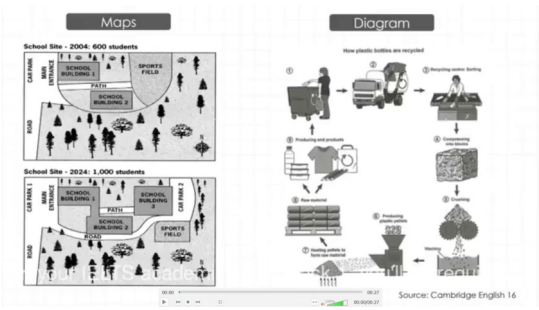
In our performance analysis, we assessed the efficiency of three database types—XML, JSON, and MongoDB—by measuring the total time required for the entire process from start to finish. This included both storing and retrieving data, with a focus on two specific scenarios: when a keyword is not found and when it is present. For each database, we recorded the time taken to extract audio from a video, process and store the data, and subsequently retrieve and process the relevant information. In cases where the keyword was not found, the time taken was generally longer due to the need to handle larger volumes of data and perform extensive searches. Conversely, when the keyword was present, the time to complete the process was notably reduced, reflecting the database's efficiency in retrieving and processing the necessary information. Additionally, while long videos can be processed using the system, they require significantly more processing time and storage, making the task computationally complex. However, this effort is valuable, as the processed data can be reused in subsequent searches by other users, enhancing the system's utility over time. This comprehensive evaluation not only allowed us to compare the performance of XML, JSON, and MongoDB but also provided insights into the overall efficiency and effectiveness of the solution, demonstrating how

each database type impacts the speed and performance of the system (see Table I).

TABLE I: Results obtained from the proposed model database storing and retrieval process.

Storage Types	Time Taken for New Video (seconds)	Time Taken for Existing Keyword in Database (seconds)
XML	308.94	57.67
JSON	430.88	63.15
MongoDB	417.31	66.01

We have successfully examined the proposed pipeline on multiple Youtube videos. However, here we want to demonstrate it one at least one such video "The Ultimate Guide to IELTS Academic Writing Task 1"[4]. The proposed pipeline provides the set of most frequent words: one of which is *IELTS*. The automated extracted clip is given in *https://www.youtube.com/watch?v=TPN4vs6UqFU*.



(a) Screenshot at 00 seconds

(b) Screenshot at 22 seconds

Fig. 5: Automated extracted clip of 27 seconds from the original video of 17:19 minutes; keyword: "IELTS" (video source: FASTRACK IELTS)

VI. Conclusion

ClipXpert is an automated system for video data mining, focusing on high-demand content. It uses advanced models and techniques to efficiently process YouTube videos, ensuring accuracy and user convenience. The system uses transcription, keyword-based extraction, and sentiment analysis to identify relevant segments. It operates seamlessly with minimal user input, maintaining a dynamically updated database for faster processing. Future work could focus on optimizing the database management system, exploring additional NLP models, and expanding the system's capabilities.

The current system effectively highlights video content based on keywords and user sentiment but could benefit from advanced natural language processing, real-time processing, and machine learning models for improved sentiment analysis and keyword extraction. Future developments will focus on optimizing the system to handle live-streamed content efficiently, enabling real-time analysis with minimal latency while maintaining high accuracy in sentiment detection and keyword identification.

[4]Sample Input Video: https://www.youtube.com/watch?v=VQluL1IRDbY

References

[1] R. A. Albeer, H. F. Al-Shahad, H. J. Aleqabie, and N. D. Al-shakarchy, "Automatic summarization of youtube video transcription text using term frequency-inverse document frequency," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 26, no. 3, pp. 1512–1519, 2022.

[2] I. Koprinska and S. Carrato, "Temporal video segmentation: A survey," *Signal processing: Image communication*, vol. 16, no. 5, pp. 477–500, 2001.

[3] S. Mashtalir and V. Mashtalir, "Spatio-temporal video segmentation," *Advances in Spatio-Temporal Segmentation of Visual Data*, pp. 161–210, 2020.

[4] P. Hu, F. Caba, O. Wang, Z. Lin, S. Sclaroff, and F. Perazzi, "Temporally distributed networks for fast video semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8818–8827.

[5] T. H. Trojahn and R. Goularte, "Temporal video scene segmentation using deep-learning," *Multimedia Tools and Applications*, vol. 80, no. 12, pp. 17 487–17 513, 2021.

[6] A. Kao and S. R. Poteet, *Natural language processing and text mining*. Springer Science & Business Media, 2007.

[7] H. Deng, D. Ergu, F. Liu, Y. Cai, and B. Ma, "Text sentiment analysis of fusion model based on attention mechanism," *Procedia Computer Science*, vol. 199, pp. 741–748, 2022.

[8] M. Wankhade, A. C. S. Rao, and C. Kulkarni, "A survey on sentiment analysis methods, applications, and challenges," *Artificial Intelligence Review*, vol. 55, no. 7, pp. 5731–5780, 2022.

[9] R. Hu, L. Rui, P. Zeng, L. Chen, and X. Fan, "Text sentiment analysis: A review," in *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*. IEEE, 2018, pp. 2283–2288.

[10] D. Wang, X. Guo, Y. Tian, J. Liu, L. He, and X. Luo, "Tetfn: A text enhanced transformer fusion network for multimodal sentiment analysis," *Pattern Recognition*, vol. 136, p. 109259, 2023.

[11] S. Banzal, *XML Basics*. Mercury Learning and Information, 2020.

[12] F. Pezoa, J. L. Reutter, F. Suarez, M. Ugarte, and D. Vrgoč, "Foundations of json schema," in *Proceedings of the 25th international conference on World Wide Web*, 2016, pp. 263–273.

[13] D. Petković, "Json integration in relational database systems," *Int J Comput Appl*, vol. 168, no. 5, pp. 14–19, 2017.

[14] K. Banker, D. Garrett, P. Bakkum, and S. Verch, *MongoDB in action: covers MongoDB version 3.0*. Simon and Schuster, 2016.

[15] S. Palanisamy and P. SuvithaVani, "A survey on rdbms and nosql databases mysql vs mongodb," in *2020 International Conference on Computer Communication and Informatics (ICCCI)*. IEEE, 2020, pp. 1–7.