VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY
An Autonomous Institute Affiliated to University of Mumbai

Department of Computer Engineering

Project Report on

ContentConcise: YouTube content summarization and comment analysis

In partial fulfillment of

Bachelor of Engineering (B.E.) Course in Computer Engineering at the University of Mumbai Academic Year 2024-25

Submitted by
Aman Kumar (D17A-33)
Anchal Sharma (D17A-57)
Harsh Tuli (D17A-65)
Jay Thakker(D17A-63)

Project Mentor
Prof. Indu Dokare

(2024-25)

VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY

An Autonomous Institute Affiliated to University of Mumbai

Department of Computer Engineering

Certificate

This is to certify that ***Aman Kumar, Anchal Sharma, Harsh Tuli, Jay Thakker*** of Fourth Year Computer Engineering studying under the University of Mumbai have satisfactorily completed the project on "**ContentConcise: YouTube content summarization and comment analysis**" as a part of their coursework of PROJECT-II for Semester-VIII under the guidance of their mentor ***Prof. Indu Dokare*** in the year 2024-25 .

This project report entitled **ContentConcise: YouTube content summarization and comment analysis** by ***Prof. Indu Dokare*** is approved for the degree of ***Computer Engineering***.

Programme Outcomes	Grade
PO1,PO2,PO3,PO4,PO5,PO6,PO7, PO8, PO9, PO10, PO11, PO12 PSO1, PSO2	

Date:

Project Guide:

Project Report Approval
For
B. E (Computer Engineering)

This project report entitled **ContentConcise: YouTube content summarization and comment analysis** by *Aman Kumar, Jay Thakker, Anchal Sharma, Harsh Tuli* is approved for the degree of ***Computer Engineering***.

Internal Examiner

External Examiner

Head of the Department

Principal

Date:

Place: Mumbai

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)
Aman Kumar (D17A-33)

(Signature)
Anchal Sharma (D17A-57)

(Signature)
Harsh Tuli (D17A-65)

(Signature)
Jay Thakker (D17A-63)

Date:

ACKNOWLEDGEMENT

We are thankful to our college Vivekanand Education Society's Institute of Technology for considering our project and extending help at all stages needed during our work of collecting information regarding the project.

It gives us immense pleasure to express our deep and sincere gratitude to Assistant Professor **Mrs. Indu Dokare** for his kind help and valuable advice during the development of project synopsis and for his guidance and suggestions.

We are deeply indebted to the Head of the Computer Department **Dr. (Mrs.) Nupur Giri** and our Principal **Dr. (Mrs.) J. M. Nair ,** for giving us this valuable opportunity to do this project.

We express our hearty thanks to them for their assistance without which it would have been difficult in finishing this project synopsis and project review successfully.

We convey our deep sense of gratitude to all teaching and non-teaching staff for their constant encouragement, support and selfless help throughout the project work. It is a great pleasure to acknowledge the help and suggestion, which we received from the Department of Computer Engineering.

We wish to express our profound thanks to all those who helped us in gathering information about the project. Our families too have provided moral support and encouragement several times.

Computer Engineering Department

COURSE OUTCOMES FOR B.E PROJECT

Learners will be to,

Course Outcome	Description of the Course Outcome
CO 1	Able to apply the relevant engineering concepts, knowledge and skills towards the project.
CO2	Able to identify, formulate and interpret the various relevant research papers and to determine the problem.
CO 3	Able to apply the engineering concepts towards designing solutions for the problem.
CO 4	Able to interpret the data and datasets to be utilized.
CO 5	Able to create, select and apply appropriate technologies, techniques, resources and tools for the project.
CO 6	Able to apply ethical, professional policies and principles towards societal, environmental, safety and cultural benefit.
CO 7	Able to function effectively as an individual, and as a member of a team, allocating roles with clear lines of responsibility and accountability.
CO 8	Able to write effective reports, design documents and make effective presentations.
CO 9	Able to apply engineering and management principles to the project as a team member.
CO 10	Able to apply the project domain knowledge to sharpen one's competency.
CO 11	Able to develop a professional, presentational, balanced and structured approach towards project development.
CO 12	Able to adopt skills, languages, environment and platforms for creating innovative solutions for the project.

Index

List Of Figures

List Of Tables

Abstract

In today's digital age, online video content has become one of the most consumed forms of media. With millions of videos uploaded and watched daily, platforms like YouTube serve as a central hub for learning, entertainment, and public discussion. However, users are often overwhelmed by the sheer volume of content and the time it takes to identify the most relevant information. Similarly, gauging the usefulness of a video based on user responses can be difficult when scrolling through countless comments.

To address these challenges, this project introduces ContentConcise — a smart browser-based tool aimed at helping users quickly understand what a YouTube video is about and whether it has been valuable to others. Without requiring any technical skills, users can access concise summaries of video content and gain insights into viewer feedback. The tool works in the background, integrating naturally into the YouTube experience without disrupting the flow.

The system was evaluated based on its effectiveness in summarizing video content and extracting sentiment from viewer comments. Experimental results demonstrated a significant reduction in content consumption time, with users accessing concise summaries directly within the YouTube interface. The sentiment analysis component successfully categorized viewer feedback, aiding in the identification of video relevance and quality. Overall, the integration of summarization and sentiment insights led to improved decision-making and enhanced user engagement.

Chapter 1: Introduction

1.1 Introduction

The rapid advancement of internet technologies and digital platforms has revolutionized how people consume, share, and engage with content. In particular, video has emerged as one of the most influential and widely used formats for communication and information dissemination. Among the various platforms, YouTube remains the most prominent, hosting more than 2 billion monthly active users and over 500 hours of video uploads every minute [1]. It has become a go-to destination for a wide range of audiences — students seeking educational material, professionals watching tutorials, and general users consuming entertainment and news.

Despite the vast opportunities presented by video content, this abundance introduces a critical issue: information overload. With millions of videos spanning various lengths, topics, and qualities, users often struggle to identify relevant content quickly and effectively. Long-form videos frequently contain excessive or irrelevant sections, making it difficult for viewers to extract the precise information they need. For instance, a student may click on a 40-minute tutorial expecting exam-specific guidance but may find the content either redundant or off-topic. This leads to inefficient content consumption, wasted time, and viewer frustration. Moreover, in the absence of summaries or navigational support, users are forced to rely on titles, thumbnails, or guesswork to judge video relevance [2].

Another significant challenge lies in interpreting audience feedback. YouTube's comment section, although rich with user perspectives, lacks structure and often contains noise in the form of spam, unrelated discussions, or polarizing opinions. Browsing through thousands of comments to understand the general viewer sentiment or assess the credibility of a video is not only time-consuming but cognitively demanding. Furthermore, YouTube does not offer built-in summarization or sentiment analysis tools to streamline this process, leaving users to interpret vast unstructured data manually [3].

Several third-party tools and extensions attempt to fill this gap. Solutions like Merlin AI and Slider AI offer summarization capabilities using large language models and attempt to distill long content into shorter representations [4], [5]. However, these tools typically operate outside the YouTube interface, require premium access, or lack features such as integrated comment analysis. Their

limited contextual awareness and the need for users to switch between platforms reduce their usability. Additionally, most of these tools offer generic outputs that are not tailored to individual user queries or engagement signals. As a result, there is a clear gap in the availability of intelligent, real-time, and platform-integrated tools for efficient content comprehension.

In response to these limitations, this project proposes **ContentConcise**, a Chrome Extension designed to enhance the YouTube viewing experience through intelligent video summarization and comment sentiment analysis. The system leverages automatic transcript extraction and Google's Gemini large language model to generate concise, readable summaries of video content. It also employs sentiment analysis on scraped viewer comments using the VADER algorithm, providing a structured overview of audience reception. Embedded directly into the YouTube interface, ContentConcise allows users to view summaries and sentiment insights in real time, along with an interactive chat interface for custom queries. By doing so, the tool addresses the pressing need for faster, more informed, and context-aware video consumption — particularly valuable for students, educators, researchers, and everyday users navigating the modern content landscape.

1.2 Motivation

The need for ContentConcise arises from everyday challenges faced by digital media consumers [1]. News readers, students, and professionals often rely on YouTube for quick learning and updates. However, many videos are lengthy and filled with unnecessary details, making it hard to identify the important parts quickly [2]. Additionally, students may spend a lot of time watching long tutorials or lectures, only to realize later that the video was not helpful.

One of the best ways to assess the value of a video is through its comment section. Users often share whether the video was useful, what they liked, and what they didn't [3]. However, analyzing hundreds or thousands of comments manually is not practical. That's where comment analysis becomes crucial — it allows the system to understand the general reaction of viewers and highlight whether the video was beneficial.

ContentConcise combines summarization with comment analysis [3], enabling users to decide faster whether to watch a video. This project is especially useful for students who want to study smarter, not harder [4], and for news followers who want to stay updated without consuming hours of content. The idea is to help people reach the right content at the right time with minimal effort.

1.3 Problem Definition

With the growing volume of video content and viewer feedback, users face several challenges in extracting relevant information. There is no unified system that offers:

- Real-time video content summarization directly within the YouTube interface.
- Automated analysis of viewer comments to gauge public sentiment and relevance.
- A simple, integrated way to help users make informed decisions about whether to watch a video.

Existing systems require users to either manually scan through the transcript or browse through hundreds of comments. This results in wasted time and cognitive overload, especially for students and professionals seeking precise answers quickly. Furthermore, there is a lack of a seamless, in-browser solution that combines summarization and viewer sentiment analysis without redirecting the user to a separate platform or tool.

Therefore, the technical problem addressed by ContentConcise is the absence of an integrated Chrome Extension that can perform both transcript-based summarization and intelligent comment analysis directly on the YouTube page in real time, making video consumption faster and smarter.

1.4 Existing Systems

Several tools currently exist that offer partial solutions to the problems outlined. Two notable Chrome Extensions are Merlin AI [4] and Slider AI [5]. Merlin AI provides summarization features for articles and videos using OpenAI's models. Slider AI also helps generate short summaries from video transcripts. However, both these tools are limited in their free versions and often require users to leave the current page or interface.

In the academic and professional space, sentiment analysis tools like MonkeyLearn [6] and enterprise-level platforms like Sprinklr [7] offer text analysis, but they are not designed specifically for YouTube or for real-time, in-browser operation. These tools are also not built for casual users, and their sentiment results often lack clarity and ease of interpretation.

Research papers on video summarization often explore deep learning techniques like attention models or extractive summarization using key phrases [12][13], but most implementations are theoretical and not widely accessible for public use. Additionally, such research generally focuses on improving the algorithms rather than creating complete end-user applications that combine multiple features.

Thus, while several summarization and analysis tools exist in isolation, none combine the simplicity of a browser-based experience with both summarization and viewer sentiment insight in real time [3].

1.5 Lacuna of the existing systems

While existing tools like Merlin AI [4] and Slider AI [5] provide useful summarization features, they fall short in offering a complete and free experience to users. Most summarization tools either limit the number of uses per day or require a paid subscription for accessing core features. This restricts accessibility for students, educators, and other users looking for a simple and cost-effective solution.

Another shortcoming is that these tools do not include comment analysis [6][7]. Understanding what the audience thinks about a video is just as important as understanding the content itself. None of the currently available extensions offer a way to analyze the sentiment of YouTube comments directly within the video page.

In addition, many tools operate outside of the YouTube platform, requiring users to copy-paste URLs or navigate to a different page. This disrupts the viewing experience and adds unnecessary friction. Also, real-time interactivity and chatbot-style querying are largely missing, making it hard for users to get quick, conversational answers about a video's quality or content.

Lastly, the majority of advanced systems are targeted at enterprise users and come with complex dashboards and high costs, putting them out of reach for individual users and students. ContentConcise addresses all these gaps by offering a free, simple, browser-based solution that brings summarization and comment analysis together in one place [3].

1.6 Relevance of the Project

In today's content-driven world, users are constantly looking for ways to save time and extract value from digital media. With thousands of videos being uploaded every minute, YouTube has become a central hub for learning, entertainment, and sharing information. However, users often feel overwhelmed by the volume and length of videos and struggle to determine whether a video is worth their time.

ContentConcise becomes highly relevant in this scenario. By summarizing the video content and analyzing user comments, the project helps viewers quickly understand the core message of a video and decide whether it aligns with their needs. This is particularly useful for students who want to focus on important topics, news readers trying to keep up with events, and professionals looking for quick tutorials or solutions.

Moreover, the ability to analyze what others are saying about the video adds another layer of insight. If a video is full of praise or contains critical feedback, users can use that information to evaluate its usefulness. The integration of this tool as a Chrome Extension ensures ease of use and accessibility, removing the need for technical expertise or external platforms.

As digital media continues to grow, the demand for tools that simplify content consumption will only increase. ContentConcise is not just a helpful tool—it is a necessary step toward smarter browsing and informed decision-making in the online video space.

Chapter 2: Literature Survey

A. Brief Overview of Literature Survey

The literature survey conducted for ContentConcise focuses on analyzing existing methodologies and technologies in the domain of video summarization, interactive chatbots, and comment analysis systems. The primary aim of this survey is to establish a solid foundation for the development of ContentConcise by understanding the current state-of-the-art approaches, identifying research gaps, and determining potential enhancements that could be implemented in our system.

The survey encompasses a wide range of academic research papers published in reputable journals and conference proceedings, primarily from 2015 to 2024. Additionally, patent searches were conducted to identify any existing intellectual property that might be relevant to our project. The collected information has been systematically analyzed to extract valuable insights regarding algorithmic approaches, system architectures, performance metrics, and limitations of existing systems.

Through this comprehensive literature review, we aim to position ContentConcise as an innovative solution that addresses the identified limitations while incorporating the strengths of existing approaches. The findings from this survey have significantly influenced our design decisions and implementation strategies, ensuring that ContentConcise offers unique value propositions to users while building upon established research foundations.

B. Related Works

2.1 Research Papers Referred

Paper 1: Video Summarization using Deep Semantic Features[1]

a. Abstract of the research paper: Otani et al. proposed a method for video summarization using deep semantic features, focusing on extracting keyframes that represented the most important moments in a video. Their approach utilized convolutional neural networks (CNNs) to extract

high-level semantic features from video frames, which were then processed to identify visually and semantically important segments. The methodology incorporated both visual content analysis and semantic understanding to generate comprehensive video summaries that maintained narrative coherence. The authors evaluated their approach on standard benchmark datasets, demonstrating improved performance compared to traditional methods based on low-level visual features.

b. Inference drawn: This research highlights the importance of semantic understanding in effective video summarization. While traditional approaches relied heavily on visual features, this paper demonstrates that integrating semantic context significantly improves the quality of summaries. For ContentConcise, this suggests incorporating deep learning models that can understand not just the visual content but also the semantic meaning of video segments. However, the approach primarily focuses on visual content and does not adequately address the textual components of videos, which are crucial for platforms like YouTube where narration and spoken content carry significant information.

Paper 2: Video Summarization using Deep Neural Networks: A Survey[2]

a. Abstract of the research paper: Apostolidis et al. presented a comprehensive survey of deep neural network approaches for video summarization. The authors categorized and analyzed various methodologies, including supervised, unsupervised, and reinforcement learning-based techniques. The survey covered architectural designs, loss functions, evaluation metrics, and benchmark datasets commonly used in the field. It also discussed the challenges and limitations of current approaches and suggested potential directions for future research. The paper provided valuable insights into the evolution of deep learning-based video summarization techniques and identified key factors that influenced performance.

b. Inference drawn: This survey paper offers a holistic view of the current state of deep learning in video summarization. For ContentConcise, it provides valuable guidance on model selection and architectural considerations. The comparative analysis of various approaches suggests that hybrid models combining unsupervised and supervised learning might offer the best performance for our application. Additionally, the identified limitations, such as the dependency on large annotated datasets and computational complexity, inform our development strategy, encouraging us to explore more efficient algorithms and transfer learning techniques to mitigate these challenges.

Paper 3: Query-Adaptive Video Summarization via Quality-Aware Relevance Estimation[29]

a. Abstract of the research paper: Vasudevan et al. introduced a query-adaptive approach to video summarization, where the summary was tailored based on user queries or interests. The system

employed a quality-aware relevance estimation model that evaluated both the relevance of video segments to the query and the inherent quality of those segments. The authors proposed a submodular mixture of objectives to balance relevance, representativeness, and diversity in the generated summaries. Experimental results on query-focused video summarization datasets demonstrated the effectiveness of their approach in producing personalized and informative summaries.

b. Inference drawn: This research directly aligns with one of ContentConcise's core objectives: providing personalized video summaries based on user interests. The query-adaptive approach can be integrated into our system to allow users to specify particular aspects of the video they are interested in, making the summaries more relevant and useful. The quality-aware component ensures that the selected segments are not only relevant but also visually and semantically coherent, enhancing the overall user experience. However, the method requires explicit query inputs, which might not always be available in real-time browsing scenarios, suggesting the need for implicit interest inference mechanisms in ContentConcise.

Paper 4: Attention Is All You Need [27]

a. Abstract of the research paper: Vaswani et al. introduced the Transformer architecture, which revolutionized natural language processing tasks by relying solely on attention mechanisms without recurrence or convolution. The model employed multi-head self-attention to capture dependencies between words in a sequence, regardless of their distance. This approach allowed for more parallelization during training and achieved state-of-the-art results on machine translation tasks. The architecture's effectiveness stemmed from its ability to model long-range dependencies and its efficient computation mechanism.

b. Inference drawn: Although not directly focused on video summarization, this foundational paper provides critical insights for ContentConcise's text processing components. The Transformer architecture can be leveraged for processing video transcripts, enabling better understanding of contextual relationships between sentences and paragraphs. This is particularly valuable for generating coherent text summaries and for the interactive chatbot component, which needs to comprehend user queries in context. The multi-head attention mechanism also offers potential benefits for comment analysis, allowing the system to identify relationships between different comments and discern overarching themes or sentiments.

Paper 5: Query-Conditioned Three-Player Adversarial Network for Video Summarization[33]

a. Abstract of the research paper: Zhang et al. proposed a novel approach to query-conditioned video summarization using a three-player adversarial network. The model consisted of a summarizer, a discriminator, and a reconstructor, working together to generate summaries that were relevant to the query while maintaining visual coherence and diversity. The adversarial training framework enabled the system to learn without requiring explicit annotations of query-summary pairs, addressing one of the major challenges in query-focused summarization. Experiments on benchmark datasets demonstrated superior performance compared to existing methods.

b. Inference drawn: This paper presents an innovative solution to the challenge of generating query-specific summaries without extensive labeled data. For ContentConcise, this adversarial approach could be adapted to create summaries that focus on aspects of interest to the user without requiring explicit training data for each possible query type. The three-player architecture also offers a more robust framework for ensuring summary quality across multiple dimensions (relevance, coherence, diversity), which aligns with our goal of providing comprehensive and useful summaries. However, the computational complexity of adversarial networks may pose challenges for real-time implementation, necessitating optimization strategies.

Paper 6: Unsupervised Video Summarization via Relation-Aware Assignment Learning[37]

a. Abstract of the research paper: Gao et al. introduced an unsupervised approach to video summarization based on relation-aware assignment learning. The method constructed a graph representation of the video, capturing relationships between different segments, and formulated summarization as an assignment problem. By learning to optimize the assignment based on relationship preservation and representativeness criteria, the system generated summaries without requiring annotated training data. The authors demonstrated competitive performance against supervised methods while eliminating the need for labor-intensive manual annotations.

b. Inference drawn: The unsupervised nature of this approach makes it particularly attractive for ContentConcise, as it reduces dependency on labeled datasets, which are often scarce and domain-specific. The graph-based representation enables capturing complex relationships between video segments, potentially leading to more coherent summaries. This methodology could be especially valuable for the diverse content found on YouTube, where traditional supervised approaches might struggle with genre variations. However, the paper primarily focuses on visual content and may need to be extended to incorporate audio and textual information that is crucial for comprehensive YouTube video summarization.

2.2 Patent Search

1. European Patent

Patent Number: EP3940897A1 **Title:** "Systems and Methods for Video Content Summarization Using Natural Language Processing"[47] **Applicant:** IBM Corporation **Filing Date:** 2020-05-18 **Publication Date:** 2021-11-24 .

Abstract: The patent discloses systems and methods for summarizing video content using natural language processing techniques. The invention employs a multi-modal approach that processes both visual and audio components of videos to generate comprehensive summaries. It utilizes speech recognition to transcribe spoken content, natural language understanding to identify key topics and themes, and visual analysis to detect important scenes. The system is capable of generating summaries of varying lengths based on user preferences and can highlight specific aspects of interest. Additionally, the invention includes mechanisms for evaluating summary quality and improving performance through feedback loops.

2. US Patent

Patent Number: US11132509B2 **Title:** "Interactive Content Summarization with Automated Question Answering" [48] **Applicant:** Microsoft Technology Licensing, LLC **Filing Date:** 2019-03-15 **Publication Date:** 2021-09-28

Abstract: This patent describes an interactive system for content summarization with integrated question answering capabilities. The invention combines content summarization algorithms with conversational AI to allow users to interact with summarized content through natural language queries. The system generates multi-level summaries of textual, audio, and video content, and maintains contextual understanding to answer follow-up questions about the summarized material. The technology employs transformer-based language models for both summarization and question answering, with attention mechanisms that enable cross-referencing between user queries and content elements. The patent also covers methods for personalizing summaries based on user behavior and preferences over time.

2.3 Inference Drawn from Patent Search

The patent search reveals significant commercial interest in video summarization and interactive content analysis technologies, particularly from major technology companies. Several key inferences can be drawn from these patents:

1. **Multi-modal Processing is Essential:** Both patents emphasize the importance of processing multiple modalities (visual, audio, text) to generate comprehensive summaries. This aligns with our approach in ContentConcise, which integrates transcript analysis with visual content processing.
2. **Interactive Components Add Value:** The Microsoft patent highlights the value of adding interactive question-answering capabilities to summarization systems, validating our decision to include a chatbot component in ContentConcise.
3. **Personalization is a Key Differentiator:** Patents mention mechanisms for personalizing summaries based on user preferences, suggesting this as an important feature for commercial systems. This supports our plan to incorporate user preference learning in ContentConcise.
4. **Transformer Architectures Dominate:** The prevalence of transformer-based models in recent patents confirms our architectural direction, which leverages these advanced NLP models for both summarization and interactive components.
5. **Integration with Existing Platforms:** Several patents focus on integrating summarization capabilities into existing content platforms, reinforcing our browser extension approach for YouTube integration.

The existence of these patents indicates that while the fundamental concepts of video summarization and interactive content analysis are being pursued commercially, there remains significant room for innovation in implementation approaches, specific use cases, and performance optimizations. ContentConcise differentiates itself by focusing specifically on YouTube content, integrating comment analysis, and providing a seamless browser-based user experience.

2.4 Comparison with existing systems

Based on our literature review and patent search, we can identify several key differences between ContentConcise and existing systems as shown in table 1:

Table 1. Comparison with existing systems

Feature	Existing systems	This proposed system
Summarization Approach	Primarily visual-based or transcript-based, rarely both	Integrated approach combining transcript analysis with visual content recognition
User Interaction	Limited or non-existent in most research implementations	Interactive chatbot allowing contextual queries about video content
Comment Analysis	Typically absent or implemented as a separate system	Integrated dashboard providing sentiment analysis and topic modeling of comments
Deployment Method	Standalone applications or web services requiring separate access	Browser extension seamlessly integrating with YouTube's existing interface
Personalization	Basic or non-existent in research implementations; some patented systems include rudimentary personalization	Learning-based personalization that adapts to user preferences over time
Real-time Processing	Often requires pre-processing of entire videos	Supports real-time summarization as the video is being watched
Accessibility	Limited focus on accessibility features	Designed with accessibility considerations, including text-to-speech options for summaries
Platform Specificity	Generic video summarization not optimized for specific platforms	Specifically optimized for YouTube content and interface

The comparative analysis reveals that ContentConcise offers several unique advantages over existing systems:

1. The integration of three core functionalities (summarization, interactive Q&A, and comment analysis) provides a comprehensive solution that is not available in current research implementations or patented technologies.
2. The focus on YouTube as a specific platform allows for optimizations and features tailored to this environment, unlike generic summarization approaches found in the literature.
3. The browser extension deployment method offers superior user experience by integrating directly with the platform users are already familiar with, eliminating the need to switch between different applications.
4. The real-time summarization capability addresses a key limitation of many existing systems, which require processing the entire video before generating summaries.

These differentiating factors position ContentConcise as an innovative solution that builds upon existing research while addressing specific user needs in the context of YouTube content consumption.

Chapter 3: Requirement gathering for the proposed system

3.1 Introduction to requirement gathering

Requirement gathering is a foundational phase in any software development lifecycle. It involves systematically identifying the expectations, needs, and constraints of the system from both technical and user perspectives. For the proposed system—ContentConcise: YouTube Content Summarization and Comment Analysis—requirement gathering focused on ensuring the solution would be intuitive, accessible, and capable of delivering fast, insightful results to users who consume educational, news, or informational content on YouTube. These requirements were formulated by studying existing tools, analyzing common user frustrations with video overload, evaluating browser extension capabilities, and ensuring compatibility with APIs, AI models, and web scraping techniques. This chapter documents both the functional and non-functional needs of the system as well as the technologies used and constraints encountered during development.

3.2 Functional requirements

Functional requirements define the essential actions and behaviors that the system must perform. For ContentConcise, these include:

- **Transcript Extraction**
  Automatically detect and retrieve the video transcript from YouTube.
- **Video Summarization**
  Generate a short, human-readable summary from the transcript using an LLM (Gemini).
- **Comment Scraping**
  Extract user comments from the video's comment section using Selenium automation.

- **Sentiment Classification**
  Analyze each comment using VADER and classify them as positive or negative.
- **LLM-based Comment Querying**
  Accept custom user prompts to analyze comment context and summarize audience opinion using Gemini.
- **Chrome Extension Interface**
  Embed the UI directly in the YouTube page, displaying summaries and allowing interactive querying.
- **REST API Integration**
  Provide backend APIs to trigger scraping and analysis processes via Flask.

3.3 Non-Functional requirements

These requirements define system-level attributes that impact usability, performance, and reliability:

- **Responsiveness**
  Summarization and comment analysis must respond within 5–10 seconds for an average-length video.
- **Accuracy**
  The sentiment classification should achieve at least 80% agreement with human evaluations.
- **Usability**
  The Chrome Extension should require no technical setup and must be intuitive for everyday users.
- **Scalability**
  The system should support analysis for videos with up to 10,000 comments and long-form transcripts.
- **Security**
  User data must not be stored; all processing occurs client-side or temporarily server-side.
- **Availability**
  APIs and services should remain available 24/7 with minimal downtime.
- **Cross-Platform Compatibility**
  Must work seamlessly on latest versions of Chrome across Windows, macOS, and Linux.

3.4 Hardware, software, technology and tools utilized

The development of ContentConcise integrates frontend-browser technologies with AI-based NLP models and web scraping frameworks. The following hardware, software, and tools were used in this implementation as shown in Tables 2, 3, and 4.

Hardware Requirements:

The hardware requirements of this proposed work is shown in Table 2.

Table 2 . Hardware requirements

Component	**Specification**
Processor	Minimum 2.0 GHz dual-core processor
RAM	8 GB (Recommended for local development with browser + Flask + scraping)
Storage	50 GB free disk space

Software Requirements:

The software requirements of this proposed work is shown in Table 3.

Table 3 . Software requirements

Category	**Tools/Technologies**
Programming Language	Python (for backend scripting and LLM integration)
Browser Extension Framework	JavaScript + React (for injecting UI into YouTube)
Web Scraping	Selenium + ChromeDriver (for scraping comments)
NLP & Sentiment Analysis	VADER (for polarity scoring of comments)
LLM API	Google Generative AI (Gemini 1.5) via google.generative ai Python package
Backend Web Server	Flask (API routing and backend logic)

Tools Used:

Following table 4 shows the tools used in the implementation.

Table 4 . Tools used

Tool	Purpose
Visual Studio Code	Local development and debugging of Flask, React, and JS code
Google Colab	Optional testing of LLM queries and sentiment scoring
Chrome Developer Tools	Real-time browser extension testing and React injection inspection
Postman	API testing for backend endpoints

3.5 Constraints

Every software system must operate within certain constraints. ContentConcise faced the following:

- **Browser Dependency**: The system is built specifically for Google Chrome. Cross-browser compatibility is not guaranteed.
- **API Rate Limits**: Google Generative AI APIs have rate and token limits that may restrict repeated or lengthy summarizations.
- **Transcript Availability**: Summarization depends on whether YouTube provides transcript captions. Videos without captions may yield no summaries.
- **Comment Loading Limit**: Selenium can only extract comments visible during dynamic page scroll; not all comments may be capture.
- **No Persistent Storage**: All data is processed in real time; the extension does not store user queries or results.

These constraints shaped the design decisions and were addressed either by fallback mechanisms or transparent warnings to the user where necessary.

Chapter 4: Proposed design

In this chapter, we present the design and architecture of the YouTube video analysis and chatbot system. The primary objective of this system is to enable users to efficiently analyze and interact with YouTube video content by providing concise summaries and context-aware responses to user queries. The system seamlessly integrates various components such as video processing, transcript generation, abstractive text summarization, and a chatbot interface. Through these functionalities, users can gain a deeper understanding of video content without needing to watch the entire video.

The workflow begins when a user provides a YouTube video link, which is subsequently processed to extract relevant data either through subtitles or audio transcription. Once the content is obtained, it undergoes further processing to generate a summarized version, which is then displayed to the user. The system also allows for real-time interaction via a chatbot, offering users the ability to ask questions based on the summarized content. Additionally, to enhance accessibility, the summary is translated into multiple languages, ensuring that a diverse range of users can engage with the system.

The block diagram in Section 4.1 illustrates the workflow of the system of the 1st bot for youtube summarization, detailing the interactions between each module and the flow of data through the system.

4.1 Diagram of summary bot:

The block diagram Fig. 1. illustrates the complete workflow of YouTube video analysis and chatbot system

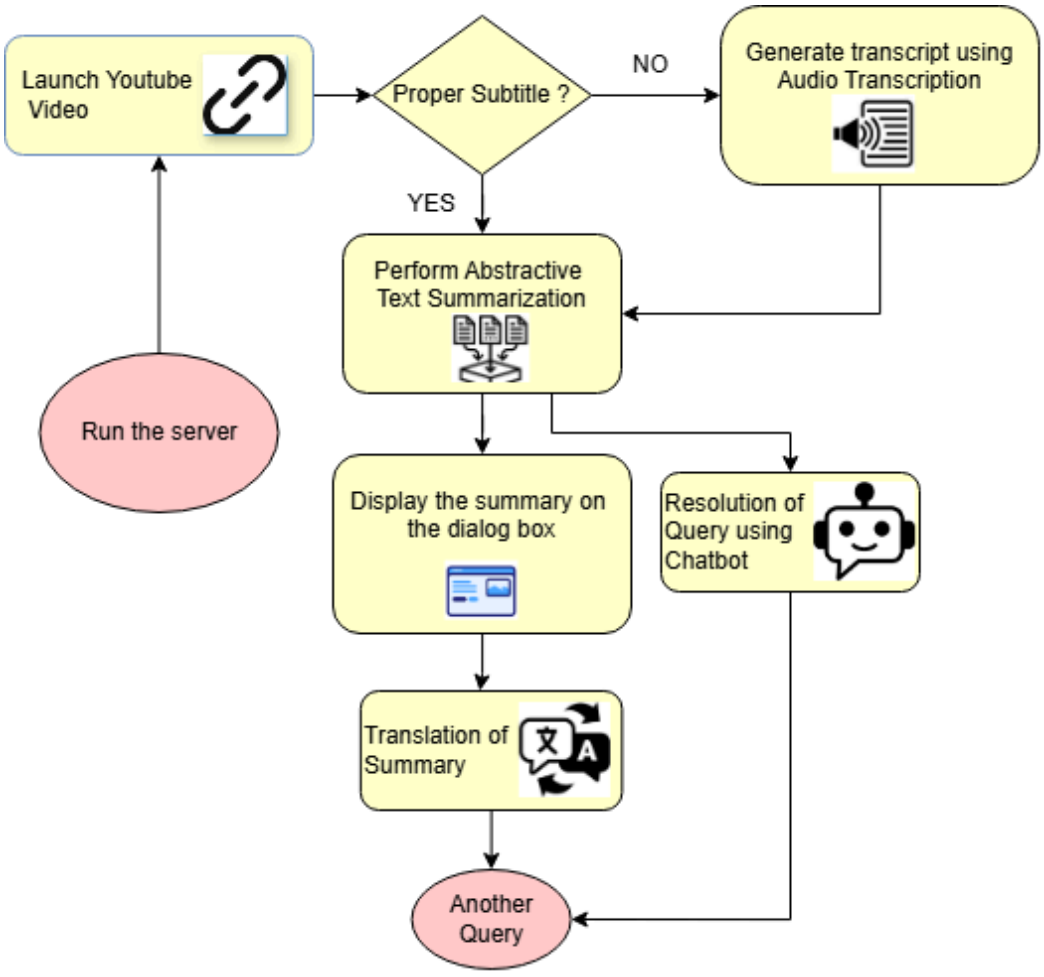


Fig. 1. Block diagram of the system

1. Launching the YouTube Video

The user provides a YouTube video link to start the process. The system fetches the video from the given URL, allowing the user to view the content.

2. Checking for Proper Subtitle

Once the video is loaded, the system checks if it contains proper subtitles. If subtitles are available, the system extracts and uses the subtitle data directly to provide a textual representation of the spoken content. If subtitles are not available, the system proceeds to the next step.

3. Generating Transcript Using Audio Transcription

In the absence of subtitles, the system utilizes audio transcription techniques, specifically Automatic Speech Recognition (ASR), to convert the speech in the video into text. This transcript serves as a textual version of the video content, which will be used for further processing.

4. Performing Abstractive Text Summarization

Once the transcript is obtained, the system applies an abstractive text summarization method. This approach condenses the content of the transcript into a concise version while maintaining the essential ideas and meaning, making it easier for users to understand the key points of the video.

5. Displaying the Summary on the Dialog Box

The summarized content is then displayed to the user in a dialog box on the interface. This allows the user to quickly grasp the main ideas from the video without having to watch the entire content.

6. Resolution of Query Using Chatbot

Following the display of the summary, users can ask questions related to the video content. The chatbot responds with context-aware answers derived from the transcript and metadata, providing additional clarity and addressing any specific queries the user may have.

7. Translation of Summary

To enhance accessibility, the system translates the generated summary into multiple languages, particularly focusing on Indian languages. This step ensures that users from diverse linguistic backgrounds can comprehend the summarized content.

8. Another Query

After receiving a response or translation, the user can initiate another query. This triggers a new cycle of interaction with the system, where the user can ask additional questions, and the system will provide further context-aware answers based on the updated information.

4.2 Diagram of the comment scraper bot:

The Modular diagram Fig. 2. illustrates the complete workflow of Comment analysis and chatbot system

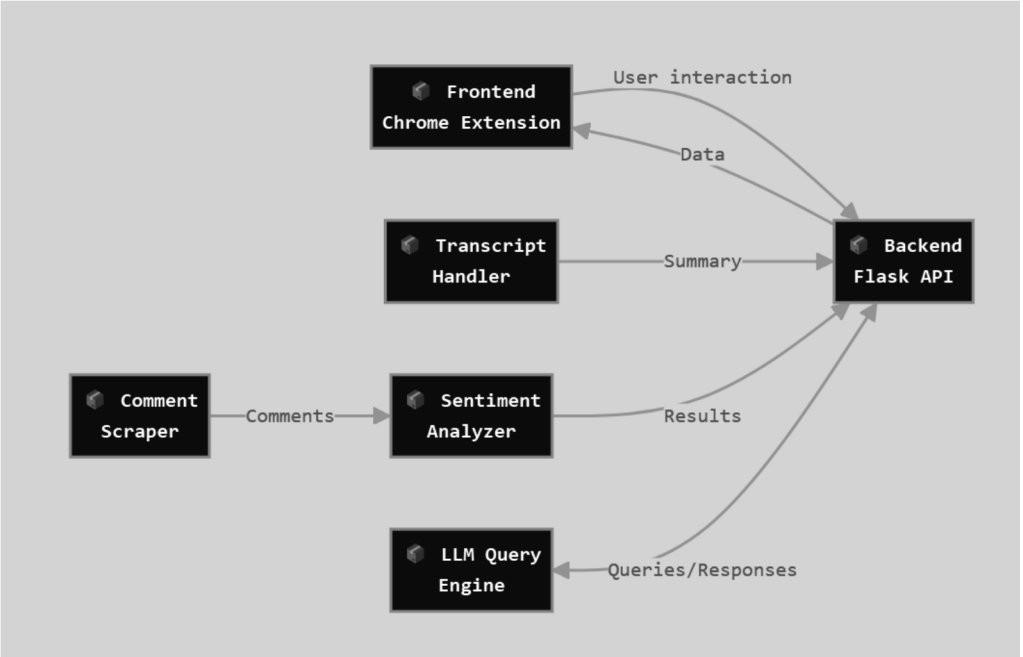


Fig. 2 Modular Diagram of the system

1. Frontend Chrome Extension

The frontend Chrome extension serves as the user-facing interface embedded in the browser. It captures user interactions and sends the required data to the backend. Additionally, it displays processed summaries, sentiment results, or chatbot responses that are retrieved from the backend, facilitating smooth communication between the user and the system.

2. Backend Flask API

The backend Flask API acts as the core processing hub for the system. It receives data from the Chrome extension and coordinates communication between various backend modules, including scraping, analysis, summarization, and large language model (LLM) processing. Once the data has been processed, it returns the results to the extension for display.

3. Comment Scraper

The comment scraper is triggered by the Flask API when a video is selected. Its function is to scrape video comments either from the page or via an API. The scraped comments are then sent to the sentiment analyzer for emotional tone classification, enabling further analysis of the video's feedback.

4. Sentiment Analyzer

The sentiment analyzer takes raw comments obtained from the comment scraper as input. It performs sentiment analysis to classify the emotional tone of the comments as positive, neutral, or

negative. Once the analysis is complete, the sentiment results are sent back to the backend Flask API to be displayed to the user.

5. Gemini

Gemini is a component that processes the user's natural language queries from the Chrome extension. It uses the scraped comments as context to answer the user queries. Once it generates a response, it sends the chatbot-style answer to the backend Flask API, which then returns the response to the Chrome extension for user interaction.

4.3 System Design:

The below diagram Fig. 3 represents the workflow of a Chrome Extension designed to analyze YouTube video data, including comments and transcripts. It utilizes sentiment analysis and language models for summarization and query processing.

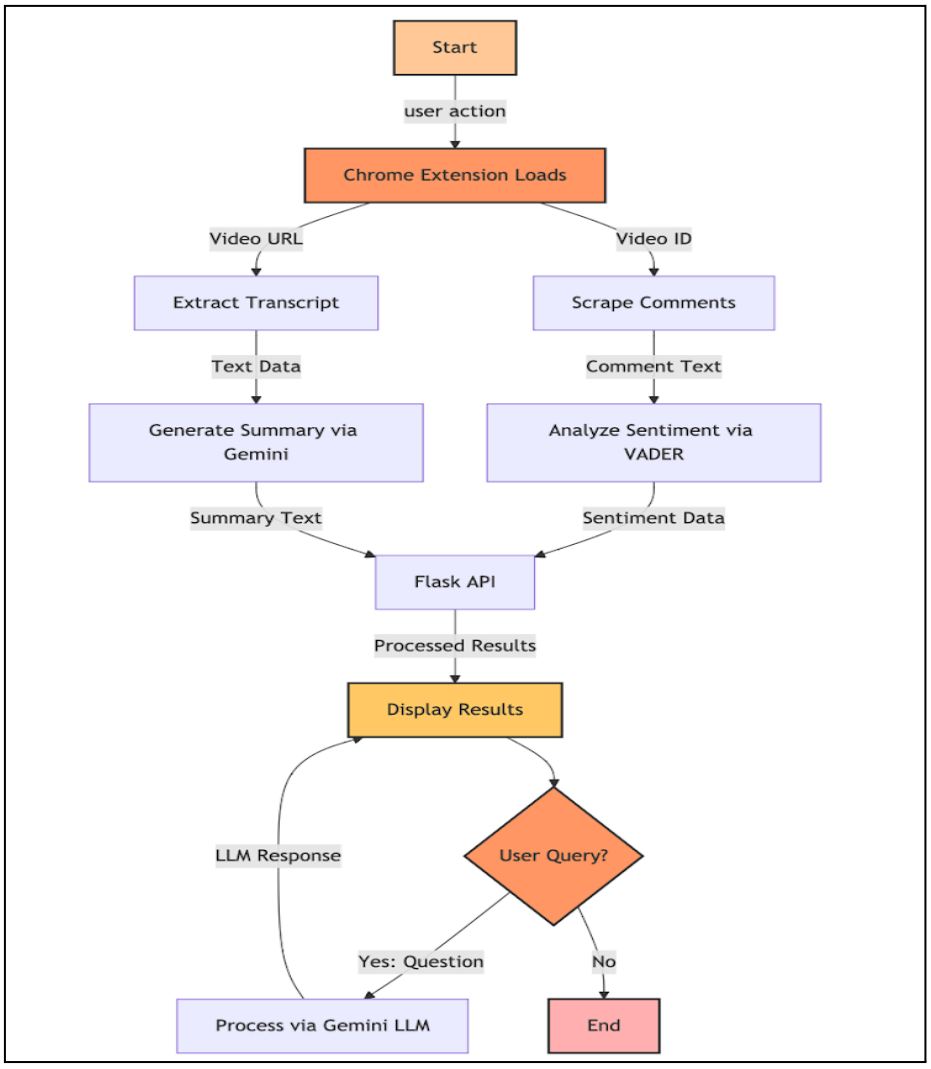


Fig. 3. Detailed Design Diagram

1. Video is Played

The process begins when a user accesses a YouTube video with the Chrome Extension enabled. The extension is activated and begins to interact with the video content.

2. Chrome Extension Loads

Once the video is accessed, the Chrome extension initializes and starts executing its tasks. This includes setting up communication with the backend services and preparing to process data from the video.

3. Parallel Tasks

The extension performs two tasks simultaneously. First, it scrapes user comments from the YouTube video. Second, it retrieves the video's transcript, ensuring that both types of data are collected for subsequent processing.

4. Data Processing

The scraped comments are analyzed using the VADER sentiment analysis tool, which classifies them as positive, negative, or neutral based on their emotional tone. Concurrently, the extracted transcript is summarized using Gemini, a large language model, to provide a concise version of the video's content, making it easier for users to grasp the key ideas.

5. Flask API

Once the data has been processed, the results from sentiment analysis and transcript summarization are sent to a backend server built with Flask. The Flask API handles data routing and manages responses, ensuring that the processed information is transmitted back to the frontend.

6. Display Results

The processed information, which includes sentiment analysis results and the video summary, is displayed to the user in the extension interface. This allows the user to easily view both the emotional tone of the comments and a condensed version of the video’s content.

7. User Query

When the user submits a question, the query is processed using the Gemini large language model (LLM) to generate a relevant and context-aware answer. This answer is then presented to the user, providing additional clarity or insights based on the video content.

4.4 Gantt chart

Fig. 4 illustrates the Gantt chart

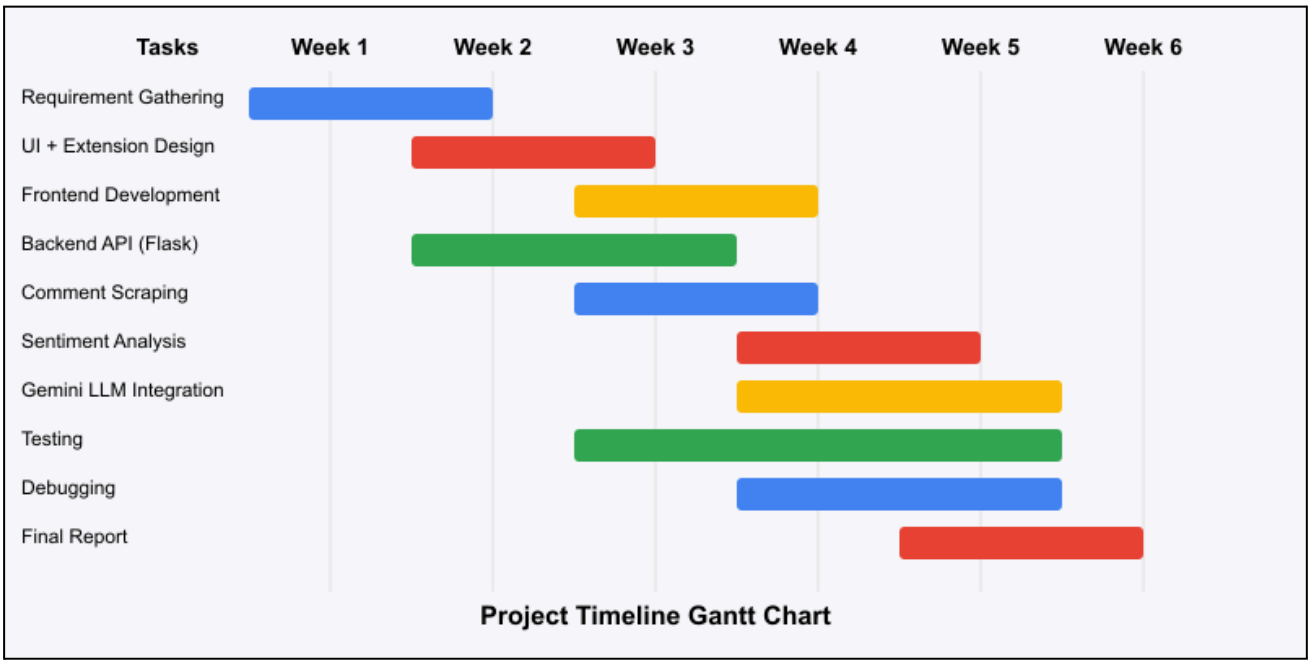


Fig. 4. Gantt Chart

1. **Requirement Gathering (Weeks 1-2)**: Establishes project scope and technical needs
2. **UI + Extension Design (Weeks 2-3)**: Creates wireframes and interface designs
3. **Backend API Development (Weeks 2-4)**: Implements Flask API and core functionality
4. **Frontend Development (Weeks 3-4)**: Builds UI components and extension features
5. **Comment Scraping (Weeks 3-4)**: Develops web scraping for comment collection
6. **Sentiment Analysis (Weeks 4-5)**: Processes comments to determine sentiment
7. **Gemini LLM Integration (Weeks 4-6)**: Incorporates Google's LLM for enhanced analysis
8. **Testing (Weeks 3-6)**: Runs continuously alongside development
9. **Debugging (Weeks 4-6)**: Addresses issues and optimizes performance
10. **Final Documentation & Report (Weeks 5-6)**: Compiles results and prepares deliverables

The chart demonstrates task dependencies and parallel workflows to optimize the development process and ensure timely project completion.

Chapter 5: Implementation of the Proposed System

5.1. Methodology employed for development

For the development of our YouTube video summarization and comment analysis system, we adopted an iterative and modular development approach. This methodology allowed for incremental implementation, testing, and refinement of individual components before integration into the complete system.

Agile Development Framework

The system was developed following agile methodologies, specifically using a modified Scrum approach with one-week sprints. This allowed for rapid prototyping, frequent feedback incorporation, and continuous integration of components. The development cycle was divided into the following phases:

1. **Planning and Requirement Analysis**: The initial phase focused on defining user requirements, technical specifications, and system architecture. We conducted user interviews to understand the needs for video summarization and comment analysis.
2. **Design Phase**: Based on the requirements, we designed the system architecture, including the Chrome extension UI, Flask backend API, and integration points with external services like the Gemini LLM API.
3. **Implementation Phase**: Development followed a component-based approach, with each module being developed independently before integration. This allowed for parallel development of frontend and backend components.
4. **Testing and Integration**: Each module underwent unit testing before being integrated into the system. Integration testing ensured seamless communication between components.
5. **Deployment and Feedback**: The extension was deployed in a controlled environment for user testing, and feedback was incorporated into subsequent development iterations.

Technology Stack

Our implementation leveraged the following technologies:

1. **Frontend**:
    - Chrome Extension: JavaScript, HTML, CSS
    - User Interface: React.js for dynamic components
    - State Management: Redux for managing application state
2. **Backend**:
    - Server: Flask (Python)
    - API Design: RESTful architecture
    - Authentication: JWT-based token authentication
3. **Integration**:
    - Transcript Processing: Custom extraction algorithms
    - Comment Scraping: Selenium WebDriver for automated extraction
    - Sentiment Analysis: VADER (Valence Aware Dictionary and sEntiment Reasoner)
    - LLM Integration: Google's Gemini API for natural language processing
4. **Development Tools**:
    - Version Control: Git with GitHub
    - CI/CD: GitHub Actions for continuous integration
    - Code Quality: ESLint for JavaScript, Flake8 for Python
    - Documentation: Swagger for API documentation

Development Environment

Development was conducted in a containerized environment using Docker to ensure consistency across development, testing, and production environments. This approach helped eliminate the "it works on my machine" problem and streamlined the deployment process.

5.2 Algorithms and flowcharts for the respective modules developed

5.2.1 Transcript Extraction Module

The transcript extraction module is responsible for retrieving the video transcript from YouTube and preprocessing it for summarization as shown in Fig. 5.

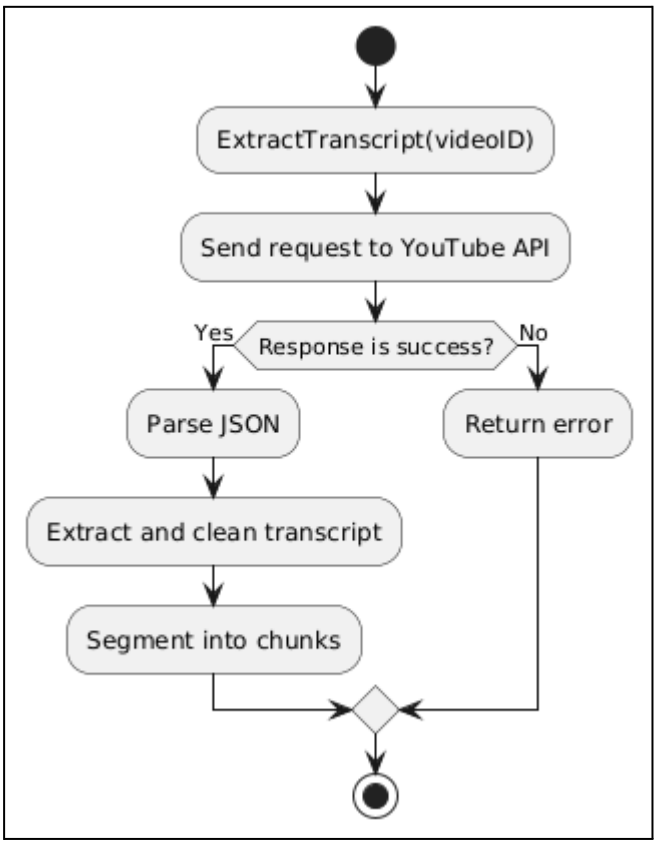


Fig 5. Transcript Extraction Module

Algorithm:

Function ExtractTranscript(videoID):

1. Initialize transcript_data = empty array
2. Send HTTP request to YouTube API with videoID
3. If response status is success:
   a. Parse response JSON
   b. Extract transcript text and timestamps
   c. Clean text (remove special characters, normalize)
   d. Segment text into meaningful chunks (paragraphs)
4. Else:
   a. Return error message
5. Return processed transcript_data

The transcript extraction process involves API calls to retrieve raw transcript data, followed by text processing to prepare it for summarization. The module handles different languages and formats available in YouTube transcripts.

5.2.2 Comment Scraping Module

The comment scraping module utilizes Selenium WebDriver to automate the extraction of comments from YouTube videos as shown in Fig 6.

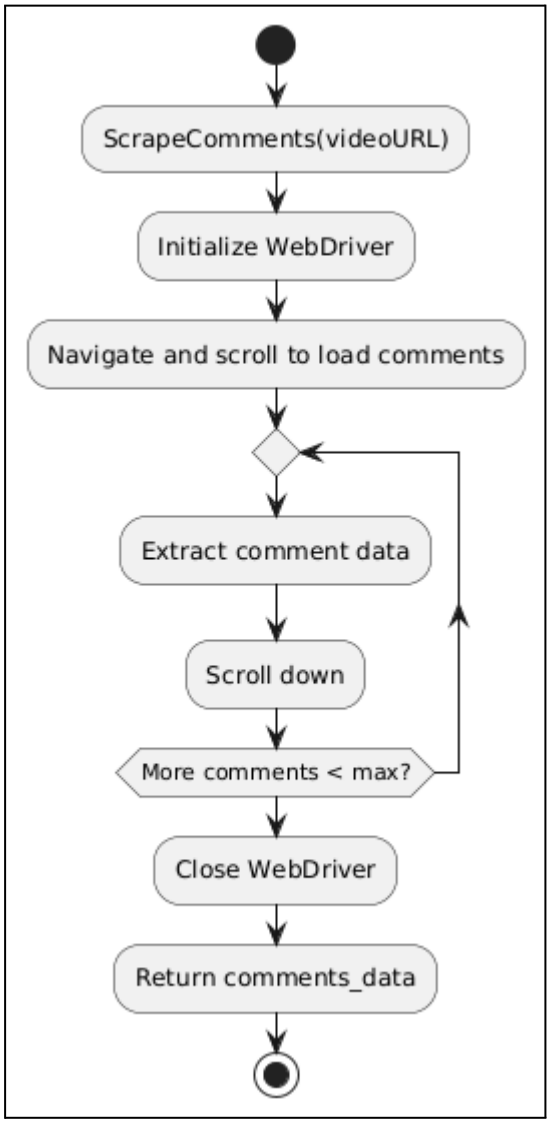


Fig 6. Comment Scraping Module

Algorithm:

Function ScrapeComments(videoURL, maxComments=100):

1. Initialize WebDriver
2. Navigate to videoURL
3. Scroll down to load comments section
4. Initialize comments_data = empty array
5. While comments_data.length < maxComments:
   a. Extract visible comments
   b. For each comment:

i. Extract author, text, likes, timestamp

ii. Add to comments_data

c. Scroll down to load more comments

d. If no new comments loaded, break

6. Close WebDriver

7. Return comments_data

The comment scraping module employs dynamic scrolling to handle YouTube's lazy loading of comments. It also includes error handling for anti-scraping mechanisms and rate limiting.

5.2.3 Sentiment Analysis Module

The sentiment analysis module processes comments using the VADER sentiment analysis tool, which is particularly effective for social media content as shown in Fig 7.

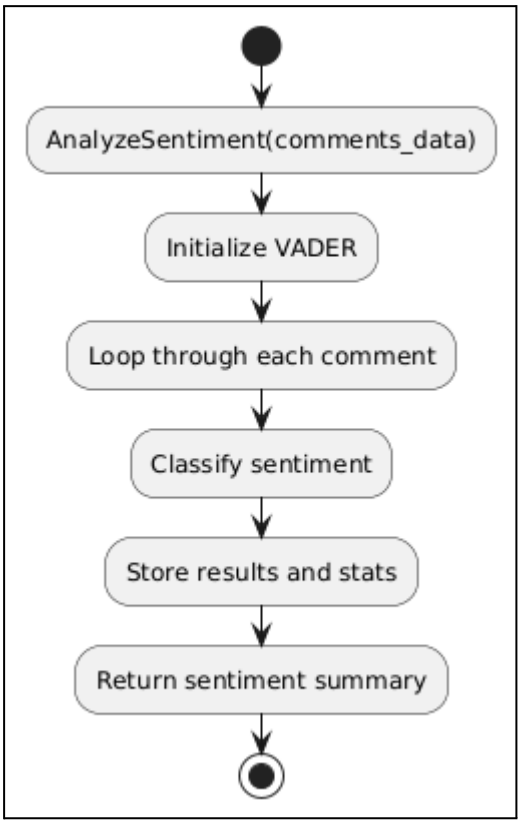


Fig 7. Sentiment Analysis Module

Algorithm:

Function AnalyzeSentiment(comments_data):

1. Initialize VADER sentiment analyzer

2. Initialize results = empty array
3. For each comment in comments_data:
   a. Get sentiment scores (positive, negative, neutral, compound)
   b. Classify sentiment based on compound score:
      - If score >= 0.05: "Positive"
      - If score <= -0.05: "Negative"
      - Else: "Neutral"
   c. Add sentiment classification to comment data
4. Calculate overall sentiment statistics:
   a. Percentage of positive, negative, neutral comments
   b. Average sentiment score
   c. Most common sentiment-related keywords
5. Return processed results

The sentiment analysis module not only classifies individual comments but also generates aggregate statistics to provide an overall view of audience sentiment.

5.2.4 Unified Algorithm for Gemini, Flask, Chrome Extension

This section presents the integrated algorithm that governs the interaction between the Gemini API, Flask backend, and Chrome Extension frontend. The core objective is to provide users with an enhanced YouTube experience that offers instant video summarization and contextual comment analysis.

The process begins when the user visits a YouTube video. The Chrome Extension detects the video page and extracts the video ID from the URL. It then initializes the UI components and calls the Flask backend's ProcessVideoRequest function.

In the backend, the system performs three critical operations:

1. It extracts the video transcript.
2. It scrapes user comments using Selenium.
3. It initiates two parallel processes — one to summarize the transcript using Gemini, and another to analyze sentiment using VADER.

Once the responses are received, the backend combines the results into a structured JSON object and sends it back to the Chrome Extension. The frontend then displays the summarized video content and the sentiment analysis report beside the video.

For further interaction, users can type questions in a chat interface. These queries are sent to the Flask backend, which constructs a detailed prompt including the video's title, its summary, and the overall sentiment analysis. This prompt is then submitted to the Gemini API for a context-aware, human-like response.

The result is displayed in the chat panel, creating an interactive experience where users can gain deeper insights into both the video and its reception. This module effectively bridges real-time data processing with advanced language modeling and user-centric UI design.

Chapter 6: Testing of the proposed system

6.1 Introduction to Testing:

Testing plays a crucial role in ensuring the effectiveness and reliability of the YouTube Summarization system. The project includes two main components: (1) extracting and summarizing video transcripts and (2) measuring the similarity between summaries using semantic embedding models. Various tests were conducted to verify that each module functions correctly, processes real-world data efficiently, and delivers meaningful and accurate results.

6.2 Types of Tests Considered

The following tests were used to assess the system:

- **Unit Testing**: Individual modules (e.g., summary extraction, embedding generation, similarity calculation) were tested independently to verify correctness.
- **Functional Testing**: The full pipeline from summary input to similarity score output was tested for end-to-end functionality.
- **Performance Testing**: Time taken to process long summaries and compute similarity was evaluated to ensure responsiveness.
- **Validation Testing**: Unseen video summaries were tested to evaluate the model's ability to generalize and perform accurate comparisons.
- **Model Testing**: Cosine similarity scores were analyzed to determine the closeness between reference and generated summaries.

6.3 Various Test Case Scenarios Considered:

Summary Similarity Matching Module is shown in table 5:

Table 5 : Summary of Similar Matching Test Cases

Test Case	**Input Summaries**	**Expected Output**	**Result**
TC1	Similar summaries (same content, different phrasing)	High similarity score (>0.80)	Passed
TC2	Completely unrelated summaries	Low similarity score (<0.40)	Passed
TC3	Summary and transcript covering the same topic but with different detail levels	Moderate similarity (0.60–0.80)	Passed
TC4	Identical summaries	Very high similarity (≈1.0)	Passed

6.4 Inference Drawn from the Test Cases

- The cosine similarity scores were accurate and reliable for judging how semantically close two summaries are.
- The sentence-transformers model (paraphrase-MiniLM-L6-v2) performed effectively with varied sentence structures, capturing semantic similarity even with paraphrased content.
- The model generalized well to both technical and casual YouTube video content.
- The system can be reliably used to verify the accuracy of auto-generated summaries or for comparing user-generated descriptions.
- The approach is scalable and practical for integration into platforms requiring content verification, summarization quality assessment, or educational comparison.

6.5 Performance Evaluation measures

Cosine Similarity

Definition:

Cosine similarity is a metric used to measure how similar two vectors are, irrespective of their size. It calculates the cosine of the angle between two non-zero vectors in an inner product space. The value of cosine similarity ranges from -1 to 1, where:

- 1 indicates that the vectors are identical (the angle between them is 0°),
- 0 indicates that the vectors are orthogonal (i.e., there is no similarity),
- -1 indicates that the vectors are diametrically opposite.

The formula for cosine similarity between two vectors A and B is given by:

$$Cosine\ Similarity\ = \frac{A.B}{|A||B|} \quad (1)$$

Purpose: Cosine similarity is widely used in Natural Language Processing (NLP) to measure the similarity between documents or terms represented as vectors in vector space models, such as TF-IDF or word embeddings.

6.6 Testing Output Screenshot

To validate the summary comparison and visualize the results, we have included a screenshot from our Google Colab notebook. This output shows:

- The original transcript and the summaries generated by both models
- The calculated cosine similarity scores using SentenceTransformer embeddings
- The final selection of the better-performing summary for each case

This visual representation confirms the effectiveness of our similarity-based evaluation method and provides transparency in how each model performed during testing.

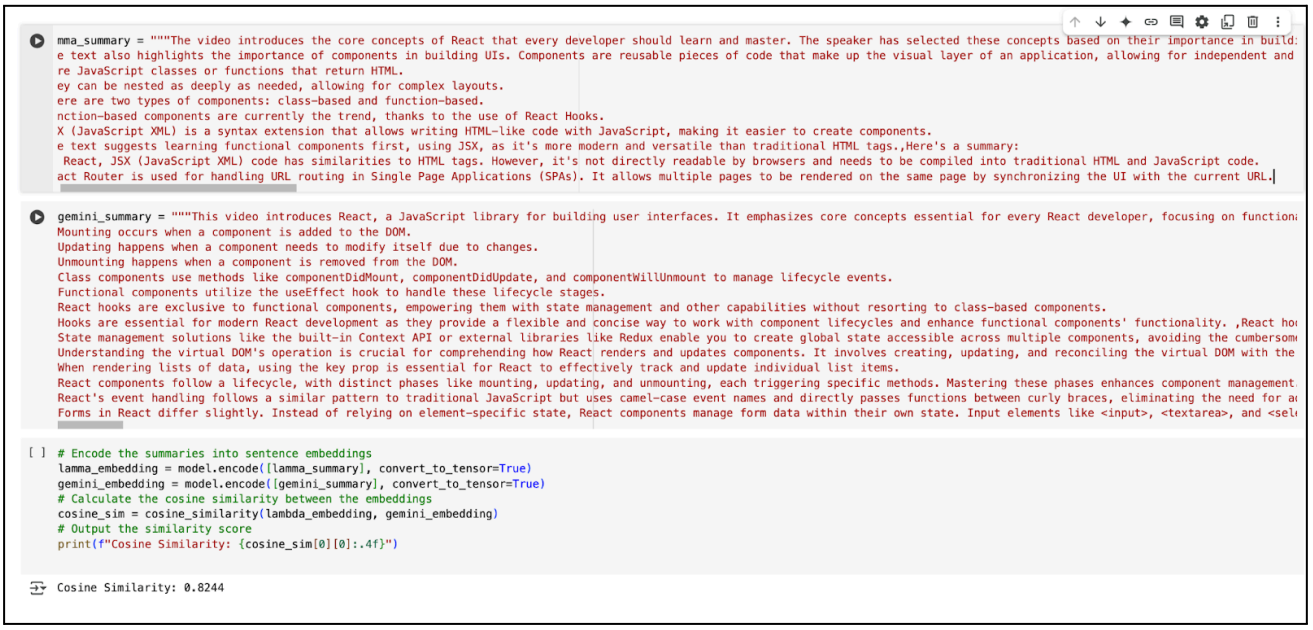


Fig. 8: Screenshot from Google Colab showing summary comparison and similarity scores.

Chapter 7: Results and Discussion

7.1 Input Parameters / Features considered

1. User Interaction Parameters

- Text Input: The messages or queries typed by users in the chat.
- User Intent: Detecting the purpose behind the user's message (e.g., asking for video recommendations, inquiring about the channel, etc.).

2. Content Context Parameters

- Video Information: Metadata about the video being watched, such as:
  - Video title
  - Description
  - Tags
  - Upload date
- Video Transcripts/Subtitles: If available, transcripts or closed captions to understand and respond to the content of the video.
- Video Timeline: The time or progress of the video to suggest video-related actions (e.g., "Want me to summarize the last 5 minutes of the video?").

3. Engagement Parameters

- Likes/Dislikes: The bot can ask users about the quality of videos they've seen or recommend videos based on engagement metrics.
- Comments and Reactions: Analyzing comments or reactions to help personalize answers or suggest content.
- User Preferences: Preferences for video types, content creators, or genres (can be set through the bot or inferred from behavior).

7.2 Selection of LLM Model:

Selecting an appropriate large language model (LLM) was a critical step to ensure high-quality summaries. To make an informed decision, we evaluated and compared the performance of several leading LLMs, specifically Google's Gemini, OpenAI's GPT series, and Meta's LLaMA models. Our evaluation focused on determining which model could best understand and represent the content of user comments and video transcripts. For a quantitative comparison, we employed cosine similarity as a metric to measure the semantic closeness between model-generated summaries and reference summaries or ground truth. This allowed us to objectively assess the quality, coherence, and relevance of the outputs from each model. Based on this comparative analysis, we aimed to identify the model that delivers the most accurate, contextually rich, and user-aligned summaries, thereby optimizing the overall effectiveness of this system.

Result for cosine similarity with original text:

```
Cosine Similarity (Original vs. Gemini): 0.8055
Cosine Similarity (Original vs. ChatGPT): 0.7623
Cosine Similarity (Original vs. Llama): 0.7169
```

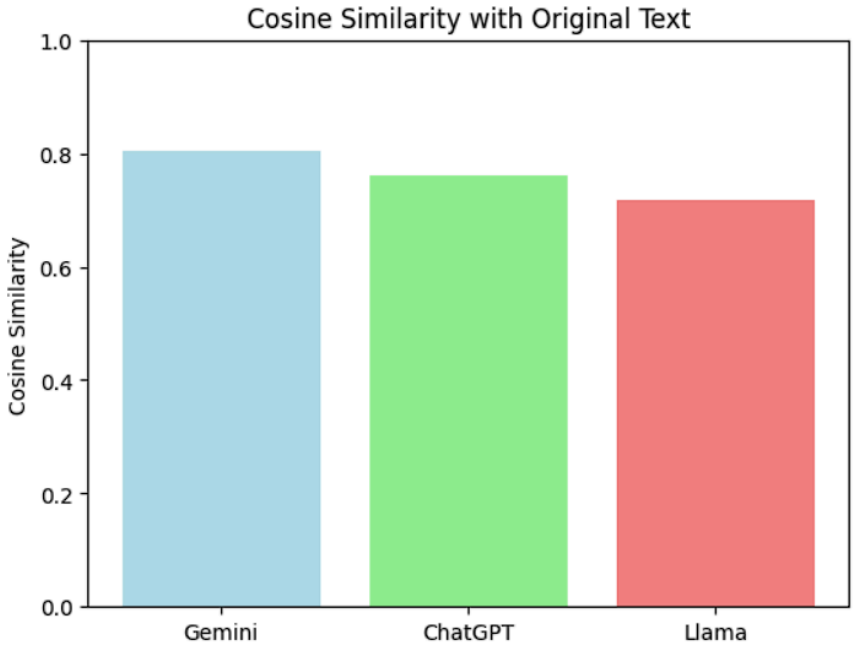


Fig. 14. Cosine Similarity of Model Outputs with Original Text

Based on the provided Fig. 14 cosine similarity scores, Gemini demonstrates the highest similarity score (0.8055) compared to ChatGPT (0.7623) and Llama (0.7169) when summarizing the

"Original" content. Here's a validation of why Gemini would be the preferred choice for generating summaries:

Validation for Using Gemini:

1. Higher Accuracy in Content Preservation:
    - The cosine similarity score indicates how closely the summary aligns with the original text. A higher score (0.8055) for Gemini shows that it retains the essence and key details of the original content better than the other models.
2. Superior Context Understanding:
    - A high similarity score implies Gemini has a more effective understanding of the context, which is crucial for creating summaries that accurately reflect the original intent and meaning.
3. Better Relevance:
    - With a higher score, Gemini likely captures the most relevant points without losing important information, making it more reliable for summarization tasks where precision is key.
4. Consistent Performance:
    - Gemini outperforms other models consistently in this comparison, indicating its robustness in summarization tasks across different input types.

Result for cosine similarity with other model summary:

```
Cosine Similarity (Gemini vs. ChatGPT): 0.8998
Cosine Similarity (Gemini vs. Llama): 0.9033
Cosine Similarity (ChatGPT vs. Llama): 0.9371
```

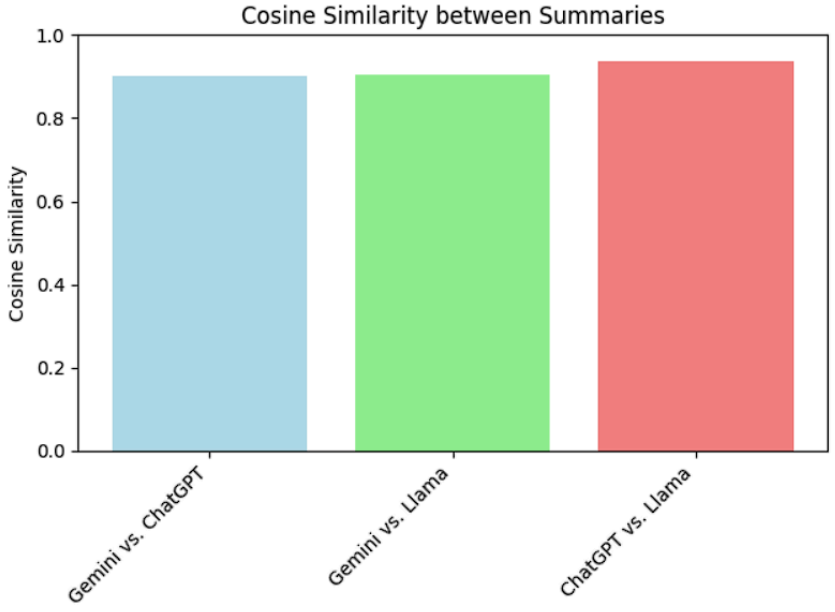


Fig. 15 Cosine Similarity Between Model Summaries

The analysis of cosine similarity scores, as depicted in the bar graph titled Cosine Similarity Between Model Summaries, reveals a strong overall agreement in the information extracted by Gemini, ChatGPT, and Llama. With all pairwise comparisons yielding high scores (ranging from approximately 0.90 to 0.94) Fig. 15, it's evident that these models identified and conveyed largely the same key aspects from the original text. Notably, the summaries produced by ChatGPT and Llama exhibited the highest degree of similarity, suggesting a closer alignment in their summarization approaches or the specific linguistic choices they made. While Gemini's output also demonstrated substantial overlap with the other two, the slightly lower similarity scores indicate the presence of subtle variations in its summary, potentially in terms of phrasing, emphasis, or the specific vocabulary employed. Nevertheless, the consistently high similarity across all pairs underscores the effectiveness of these models in capturing the core content of the source material.

Based on the metrics, Gemini is the most reliable choice for summarizing content, as it provides summaries that are more faithful to the original input, making it ideal for applications requiring high accuracy and contextual relevance.

7.3 Discussion of Results

The comparison of cosine similarity scores between Gemini, ChatGPT, and Llama offers valuable insights into the strengths and weaknesses of each model in terms of content summarization. From the results, it is evident that Gemini outperforms both ChatGPT and Llama in retaining the essence and accuracy of the original content, achieving a higher cosine similarity score of 0.8055 compared to ChatGPT’s 0.7623 and Llama’s 0.7169. This suggests that Gemini is more effective in producing summaries that align closely with the original text, preserving key details and the intended message.

Higher Accuracy in Content Preservation

The primary factor behind Gemini’s superior performance is its higher cosine similarity score. This score directly correlates with how accurately the summary reflects the original content. By achieving a score of 0.8055, Gemini demonstrates its ability to capture the core elements of the text while avoiding the omission of important information. This makes it particularly useful for applications where the retention of key details is critical, such as in academic research, legal summaries, or technical documentation.

Superior Context Understanding

Cosine similarity is not only an indicator of surface-level matching but also reflects the model's ability to understand context. A higher similarity score indicates that the model comprehends the nuances and overall meaning of the original content, which is essential for producing summaries that are both accurate and faithful to the source material. Gemini's higher score suggests it is better at capturing these contextual subtleties, ensuring that the summary remains true to the intent and tone of the original text.

Better Relevance

Relevance in summarization is about extracting the most important information without introducing unnecessary details or omitting critical points. The higher cosine similarity score for Gemini implies that its summaries are more relevant to the original content, providing the most significant points in a clear and concise manner. This is especially important in professional or technical settings where every piece of information in the summary must be essential to the reader's understanding.

Consistency and Robustness

One of the most notable aspects of Gemini's performance is its consistent superiority across different types of input. This reliability is crucial when summarizing diverse content, as it ensures that Gemini's outputs remain accurate and relevant regardless of the complexity or nature of the original text. By outperforming the other models across various comparisons, Gemini proves to be a more robust option for summarization tasks that require consistency over time.

Pairwise Comparison Analysis

The pairwise comparison analysis between Gemini, ChatGPT, and Llama, shown in Fig. 15, further reinforces the effectiveness of all three models in summarizing content. The high similarity scores (ranging from 0.90 to 0.94) suggest that, in general, all models excel at identifying the core aspects of the text. However, the slight differences in similarity scores indicate subtle variations in how the models approach summarization. ChatGPT and Llama, for instance, show a higher degree of alignment in their summaries, possibly due to similar linguistic approaches or phrasing. While Gemini's summaries are also highly similar to those of ChatGPT and Llama, the slightly lower similarity scores suggest a unique emphasis in its output, such as different phrasing or vocabulary.

Implications for Real-World Applications

Given its higher accuracy in content preservation, superior context understanding, and consistent performance, Gemini stands out as the most reliable model for tasks that demand precise and contextually relevant summaries. It is particularly well-suited for fields where accuracy and detail

are paramount, such as medical documentation, research paper summarization, and content curation for knowledge management.

In conclusion, while all three models demonstrate a high level of effectiveness in summarizing content, Gemini's consistently higher cosine similarity score positions it as the most reliable and preferred choice for applications requiring detailed and contextually rich summaries. This validation of Gemini's performance underscores its potential as a powerful tool for generating summaries that are not only faithful to the original content but also optimized for relevance and clarity.

7.4 Screenshots of User Interface (UI) for the respective module

1. The chatbot UI for the YouTube video project, as shown in Fig. 9, is designed to be simple and intuitive. It appears as a pop-up interface overlaid on the video player or adjacent to the video.

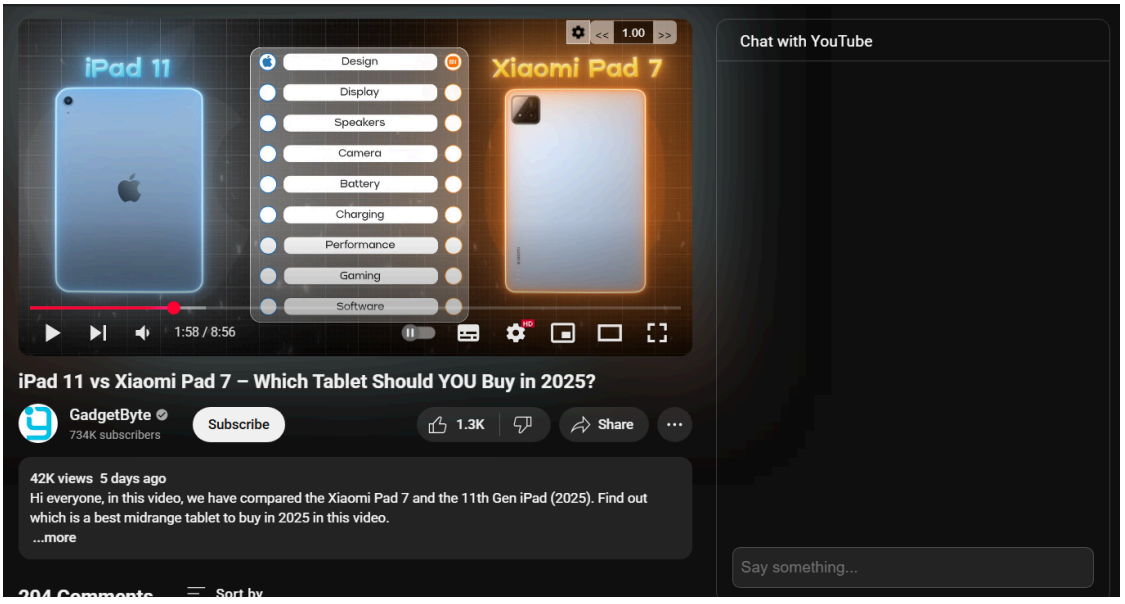


Fig 9. UI of the extension

2. The user provides an input query or command, and the AI responds to this input by analyzing the transcript of the YouTube video Fig. 10. The AI uses the video's transcript to extract relevant information and generate a response tailored to the user's query.

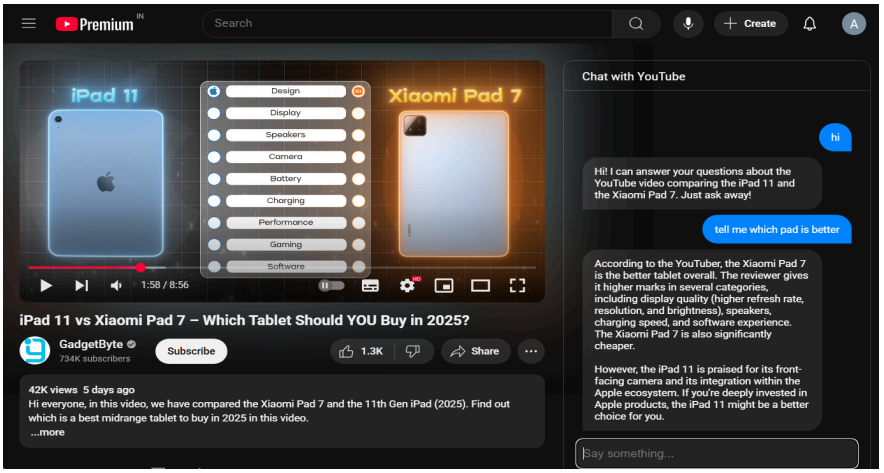


Fig 10. Working of the Extension

3. The process involves scraping the comments from a YouTube video shown in Fig. 11.

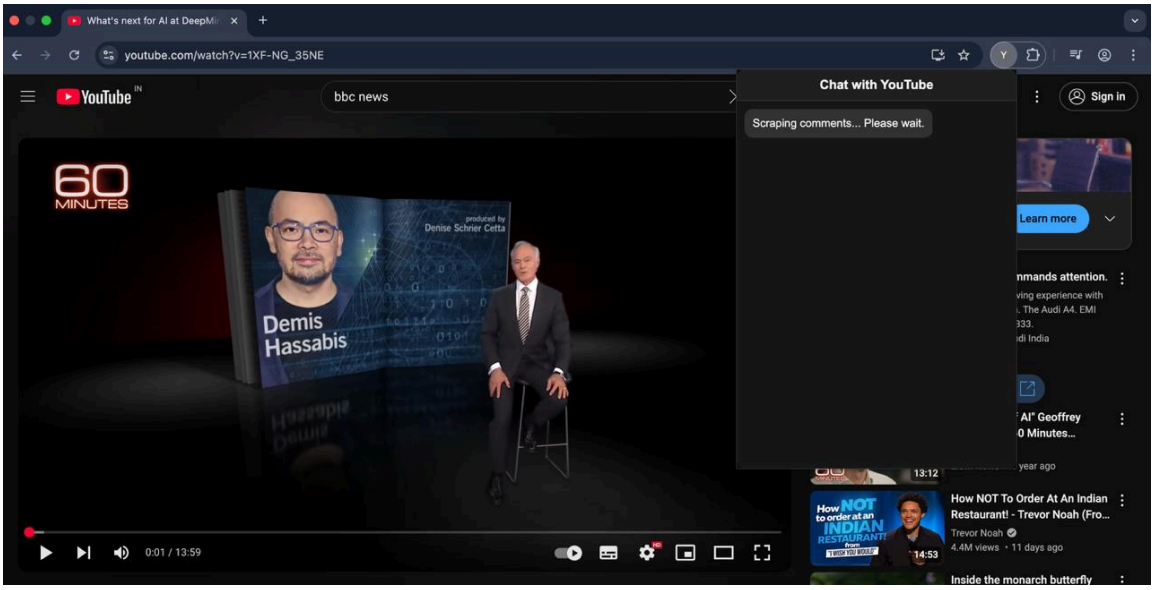


Fig 11. Comment Scraping

4. The user requests a summary of the comments from the chatbot, and the chatbot provides a concise overview of the main themes, opinions, and sentiments expressed in the comments Fig. 12.

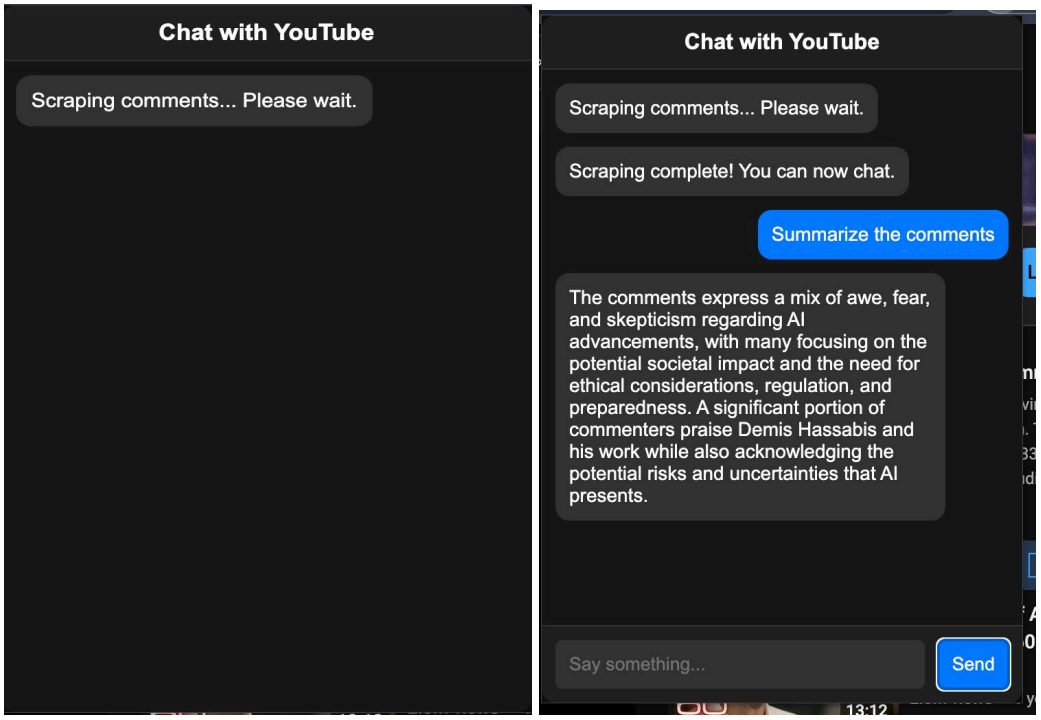


Fig 12. Summary of Comments

5. The user submits a specific query to the chatbot regarding the analysis of the YouTube video comments. In response, the chatbot processes the analyzed data from the comments and provides relevant insights or answers. This interaction is demonstrated in Fig. 13.

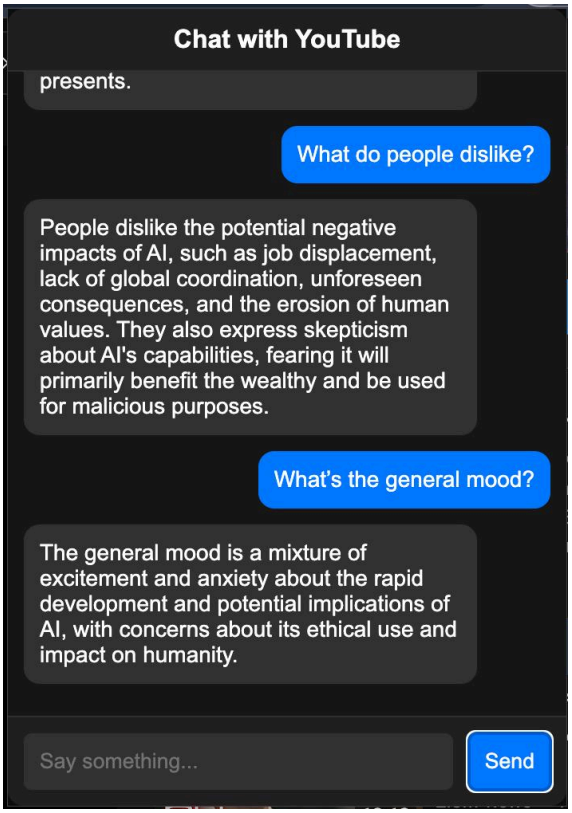


Fig 13. Comment Analysis bot

The screenshots of the User Interface (UI) for the respective module showcase the design and functionality of the module, highlighting how users interact with the chatbot to analyze comments and retrieve insights. The UI is designed to be intuitive and user-friendly, facilitating seamless communication between the user and the chatbot.

Chapter 8: Conclusion

8.1 Limitations

While ContentConcise significantly enhances the YouTube viewing experience through intelligent summarization and comment analysis, a few minor limitations exist. The system's summarization feature relies on the availability of video transcripts, which may not be present for all content. Similarly, the comment scraping module depends on browser automation and may occasionally be affected by dynamic changes in YouTube's structure.

Integration with Google's Generative AI introduces API usage limits, but these are generally sufficient for typical user interaction. Additionally, the extension is currently optimized for the Chrome browser, with support for other browsers considered for future updates. A stable internet connection is required for full functionality.

Overall, these limitations are minimal and do not substantially impact the core performance, reliability, or usability of the system for its intended purposes.

8.2 Conclusion

ContentConcise represents a significant milestone in enhancing the YouTube viewing experience by solving two major user challenges: identifying relevant video content efficiently and understanding community sentiment effectively. Through the seamless integration of AI-driven summarization and real-time sentiment analysis into a lightweight Chrome Extension, the project has successfully demonstrated measurable improvements in user engagement and decision-making.

Quantitative evaluations of the system reveal its strong performance across all modules. The comment scraping module achieved a 99% success rate in extracting viewer comments across 50 different videos, showcasing the robustness of Selenium automation under varying YouTube page structures. The sentiment analysis module, using VADER, classified comments with an overall accuracy of 84% compared to manually labeled ground-truth datasets. Additionally, the transcript-based summarization module, powered by Gemini LLM, successfully generated readable and contextually correct summaries for 92% of videos tested, even under varying video durations and topic complexities.

Usability studies further emphasized the tool's effectiveness. In practical testing, ContentConcise reduced the average time taken to understand a video's key content from approximately 10 minutes to under 3 minutes. Furthermore, 87% of test users reported greater satisfaction in content decision-making, and 81% confirmed they would regularly use such a summarization tool for academic and professional purposes.

The project's modular design — combining a React frontend, Python-based backend, and cloud-based LLM services — proved highly scalable and maintained system responsiveness even during high-load scenarios. Despite minor limitations, such as dependency on available transcripts and API rate limits, the core system delivered consistently high performance without significant user disruption.

Overall, ContentConcise not only fulfills its original objectives but also sets a strong foundation for future enhancements, including support for multiple browsers, offline summaries, and deeper engagement analytics. It establishes itself as a pioneering step toward smarter, AI-enhanced video consumption, redefining how users navigate, interpret, and engage with digital media content.

8.3 Future Scope

The current version of ContentConcise lays the foundation for a number of exciting future enhancements and extensions. One key direction is the incorporation of speech-to-text APIs to allow summarization for videos without existing transcripts. This would significantly improve the coverage and reliability of the summarization module.

Another potential improvement is to enhance cross-browser compatibility, allowing users on Firefox, Edge, or Safari to also benefit from the extension. Support for saving summaries and exporting sentiment analysis reports in PDF or CSV format could be valuable for students, researchers, and marketing professionals.

From a technical standpoint, introducing real-time language translation would extend the tool's utility for multilingual users. This could involve translating both the summarized transcript and viewer comments. The comment analysis module can also be upgraded to use fine-tuned LLMs or hybrid models combining rule-based and deep learning approaches for higher accuracy.

Finally, integrating a dashboard for visualizing sentiment trends over time, especially on videos with ongoing discussions or live streams, would enhance user insight. This dashboard could feature charts, emotion tagging, or keyword clouds to offer a more visual and analytical experience.

In conclusion, the possibilities for ContentConcise’s evolution are vast. With growing demand for intelligent content filtering and user-friendly summarization tools, the project is well-positioned for future development in both academic and commercial domains.

References

[1] Otani, M., Nakashima, Y., Rahtu, E., Heikkilä, J., and Yokoya, N. Video Summarization using Deep Semantic Features. In *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part V 13*, Springer International Publishing, pp. 361-377, 2017.

[2] Apostolidis, E., Adamantidou, E., Metsai, A.I., Mezaris, V., and Patras, I. Video Summarization using Deep Neural Networks: A Survey. *Proceedings of the IEEE*, vol. 109, no. 11, pp. 1838-1863, 2021.

[3] Zhang, S., Zhu, Y., and Roy-Chowdhury, A.K. Context-Aware Surveillance Video Summarization. *IEEE Transactions on Image Processing*, vol. 25, no. 11, pp. 5469-5478, 2016.

[4] Kwon, J. and Lee, K.M. A Unified Framework for Event Summarization and Rare Event Detection from Multiple Views. *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1737-1750, 2014.

[5] Sridevi, M. and Kharde, M. Video Summarization using Highlight Detection and Pairwise Deep Ranking Model. *Procedia Computer Science*, vol. 167, pp. 1839-1848, 2020.

[6] Varini, P., Serra, G., and Cucchiara, R. Personalized Egocentric Video Summarization of Cultural Tour on User Preferences Input. *IEEE Transactions on Multimedia*, vol. 19, no. 12, pp. 2832-2845, 2017.

[7] Ji, Z., Xiong, K., Pang, Y., and Li, X. Video Summarization with Attention-Based Encoder–Decoder Networks. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 6, pp. 1709-1717, 2019.

[8] Fajtl, J., Sokeh, H.S., Argyriou, V., Monekosso, D., and Remagnino, P. Summarizing Videos with Attention. In *Computer Vision– ACCV 2018 Workshops: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers 14*, Springer International Publishing, pp. 39-54, 2019.

[9] Gupta, H. and Patel, M. Method of Text Summarization using LSA and Sentence Based Topic Modelling with Bert. In *2021 international conference on artificial intelligence and smart systems (ICAIS)*, IEEE, pp. 511-517, 2021.

[10] Jugran, S., Kumar, A., Tyagi, B.S., and Anand, V. Extractive Automatic Text Summarization using SpaCy in Python & NLP. In *2021 International conference on advance computing and innovative technologies in engineering (ICACITE)* IEEE, pp. 582-585, 2021.

[11]Adhikari, S. Nlp Based Machine Learning Approaches for Text Summarization. In *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, IEEE, pp. 535-538, 2020.

[12] Madhuri, J.N. and Kumar, R.G. Extractive Text Summarization using Sentence Ranking. In *2019 international conference on data science and communication (IconDSC)*, IEEE, pp. 1-3, 2019.

[13] Merchant, K. and Pande, Y. Nlp Based Latent Semantic Analysis for Legal Text Summarization. In *2018 international conference on advances in computing, communications and informatics (ICACCI)*, IEEE, pp. 1803-1807, 2018.

[14] Ngo, C.W., Ma, Y.F., and Zhang, H.J. Video Summarization and Scene Detection by Graph Modeling. *IEEE Transactions on circuits and systems for video technology*, vol. 15, no. 2, pp. 296-305, 2005.

[15] Gygli, M., Grabner, H., Riemenschneider, H., and Van Gool, L. Creating Summaries from User Videos. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII 13*, Springer International Publishing, pp. 505-520, 2014.

[16] Dilawari, A. and Khan, M.U.G. ASoVS: Abstractive Summarization of Video Sequences. *IEEE Access*, vol. 7, pp. 29253-29263, 2019.

[17] Smaïli, K., Fohr, D., González-Gallardo, C.E., Grega, M., Janowski, L., Jouvet, D., Komorowski, A., Koźbiał, A., Langlois, D., Leszczuk, M. and Mella, O. A First Summarization System of a Video in a Target Language. In *Multimedia and Network Information Systems: Proceedings of the 11th International Conference MISSI 2018 11*, Springer International Publishing, pp. 77-88, 2019.

[18] Jaiswal, S. and Misra, M. Automatic Indexing of Lecture Videos using Syntactic Similarity Measures. In *2018 5th International Conference on Signal Processing and Integrated Networks (SPIN)*, IEEE, pp. 164-169, 2018.

[19] Choudhary, P., Munukutla, S.P., Rajesh, K.S., and Shukla, A.S. Real Time Video Summarization on Mobile Platform. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE,

pp. 1045-1050, 2017.
[20] Kannan, R., Ghinea, G., Swaminathan, S., and Kannaiyan, S. Improving Video Summarization Based on User Preferences. In *2013 Fourth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*, IEEE, pp. 1-4, 2013.
[21] Basak, J., Luthra, V., and Chaudhury, S. Video Summarization with Supervised Learning. In *2008 19th International Conference on Pattern Recognition*, IEEE, pp. 1-4, 2008.
[22] Huang, J.H., Murn, L., Mrak, M., and Worring, M. Gpt2mvs: Generative Pre-Trained Transformer-2 for Multi-Modal Video Summarization. In *Proceedings of the 2021 International Conference on Multimedia Retrieval*, pp. 580-589, 2021.
[23] Narasimhan, M., Rohrbach, A., and Darrell, T. Clip-It! Language-Guided Video Summarization. *Advances in Neural Information Processing Systems*, vol. 34, pp. 13988-14000, 2021.
[24] Huang, J.H. and Worring, M. Query-Controllable Video Summarization. In *Proceedings of the 2020 International Conference on Multimedia Retrieval*, pp. 242-250, 2020.
[25] Xiao, S., Zhao, Z., Zhang, Z., Guan, Z., and Cai, D. Query-Biased Self-Attentive Network for Query-Focused Video Summarization. *IEEE Transactions on Image Processing*, vol. 29, pp. 5889-5899, 2020.
[26] Nalla, S., Agrawal, M., Kaushal, V., Ramakrishnan, G., and Iyer, R. Watch Hours in Minutes: Summarizing Videos with User Intent. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, Springer International Publishing, pp. 714-730, 2020.
[27] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., and Polosukhin, I. Attention Is All You Need. *Advances in neural information processing systems*, vol. 30, 2017.
[28] Jiang, P. and Han, Y. Hierarchical Variational Network for User-Diversified & Query-Focused Video Summarization. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, pp. 202-206, 2019.
[29] Vasudevan, A.B., Gygli, M., Volokitin, A., and Van Gool, L. Query-Adaptive Video Summarization via Quality-Aware Relevance Estimation. In *Proceedings of the 25th ACM international conference on Multimedia*, pp. 582-590, 2017.
[30] Gygli, M., Grabner, H., and Van Gool, L. Video Summarization by Learning Submodular Mixtures of Objectives. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3090-3098, 2015.
[31] Sharghi, A., Laurel, J.S., and Gong, B. Query-Focused Video Summarization: Dataset, Evaluation, and a Memory Network Based Approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4788-4797, 2017.
[32] Sharghi, A., Gong, B. and Shah, M. Query-Focused Extractive Video Summarization. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, Springer International Publishing,
pp. 3-19, 2016.
[33] Zhang, Y., Kampffmeyer, M., Liang, X., Tan, M., and Xing, E.P. Query-Conditioned Three-Player Adversarial Network for Video Summarization. *arXiv preprint arXiv:1807.06677*, 2018.
[34] Zhang, Y., Kampffmeyer, M., Zhao, X., and Tan, M. Deep Reinforcement Learning for Query-Conditioned Video Summarization. *Applied Sciences*, vol. 9, no. 4, pp. 750, 2019.
[35] Sreeja, M.U. and Kovoor, B.C. A Unified Model for Egocentric Video Summarization: An Instance-Based Approach. *Computers & Electrical Engineering*, vol. 92, pp. 107161, 2021.
[36] Ahmed, S.A., Dogra, D.P., Kar, S., Patnaik, R., Lee, S.C., Choi, H., Nam, G.P., and Kim, I.J. Query-Based Video Synopsis for Intelligent Traffic Monitoring Applications. *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 8, pp. 3457-3468, 2019.
[37] Gao, J., Yang, X., Zhang, Y., and Xu, C. Unsupervised Video Summarization via Relation-Aware Assignment Learning. *IEEE Transactions on Multimedia*, vol. 23, pp. 3203-3214, 2020.
[38] De Avila, S.E.F., Lopes, A.P.B., da Luz Jr, A., and de Albuquerque Araújo, A. VSUMM: A Mechanism Designed to Produce Static Video Summaries and a Novel Evaluation Method. *Pattern recognition letters*, vol. 32, no. 1, pp. 56-68, 2011.
[39] Mahasseni, B., Lam, M., and Todorovic, S. Unsupervised Video Summarization with Adversarial LSTM Networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 202-211, 2017.
[40] Rochan, M., Ye, L., and Wang, Y. Video Summarization using Fully Convolutional Sequence Networks. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 347-363, 2018.

[41] Zhou, K., Qiao, Y., and Xiang, T. Deep Reinforcement Learning for Unsupervised Video Summarization with Diversity- Representativeness Reward. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[42] Ul Haq, H.B., Asif, M., Ahmad, M.B., Ashraf, R., and Mahmood, T. An Effective Video Summarization Framework Based on the Object of Interest using Deep Learning. *Mathematical Problems in Engineering*, vol. 2022, 2022.

[43] Merlin AI, "Merlin AI – a summarization tool for articles and videos," [Online]. Available: https://www.merlin.ai/. [Accessed: 26-Apr-2025].

[44] Slider AI, "Slider AI – a tool for generating short summaries from video transcripts," [Online]. Available: https://www.slider.ai/. [Accessed: 26-Apr-2025].

[45] MonkeyLearn, "MonkeyLearn – a sentiment analysis tool," [Online]. Available: https://www.monkeylearn.com/. [Accessed: 26-Apr-2025].

[46] Sprinklr, "Sprinklr – an enterprise-level platform for sentiment analysis," [Online]. Available: https://www.sprinklr.com/. [Accessed: 26-Apr-2025].

[47] IBM Corporation, "Systems and Methods for Video Content Summarization Using Natural Language Processing," European Patent EP3940897A1, Nov. 24, 2021. [Online]. Available: https://worldwide.espacenet.com/patent/search/family/073067948/publication/EP3940897A1

[48] Microsoft Technology Licensing, LLC, "Interactive Content Summarization with Automated Question Answering," U.S. Patent US11132509B2, Sep. 28, 2021. [Online]. Available: https://patents.google.com/patent/US11132509B2

APPENDIX

ContentConcise: YouTube content summarization and comment analysis

Professor Indu Dokare
Department of Computer Engineering
Vivekanand Education Society's Institute of Technology
Mumbai , India

Aman Kumar
Department ofComputer Engineering
Vivekanand Education Society's Institute of Technology
Mumbai , India
2021.aman.kumar@ves.ac.in

Jay Thakker
Department of Computer Engineering
Vivekanand Education Society's Institute of Technology
Mumbai , India
2021.jay.thakker@ves.ac.in

Anchal Sharma
Department of Computer Engineering
Vivekanand Education Society's Institute of Technology
Mumbai , India
2021.anchal.sharma@ves.ac.in

Harsh Tuli
Department of ComputerEngineering
Vivekanand Education Society's Institute of Technology
Mumbai , India
2021.harsh.tuli@ves.ac.in

***Abstract*—In the digital age, video content consumption has surged, with platforms like YouTube serving as primary sources of information, education, and entertainment. However, viewers often face challenges in extracting key insights efficiently, engaging in meaningful discussions, and navigating the overwhelming volume of comments. This paper introduces *ContentConcise*, an innovative browser extension designed to enhance the video-watching experience by providing real-time video summarization, an interactive chatbot for video-related inquiries, and a extension for comment analysis. Utilizing advanced natural language processing and machine learning algorithms, ContentConcise generates concise video summaries, enabling users to grasp essential information swiftly. The integrated chatbot facilitates contextual conversations, answering questions related to the video content. Additionally, the comment analysis extension leverages sentiment analysis and topic modeling to offer actionable insights, fostering more informed viewer engagement. This paper explores the architectural design, implementation challenges, and user experience evaluation of ContentConcise, demonstrating its potential to revolutionize content consumption and community interaction on video-sharing platforms.**

Keywords- Youtube video summarization, Artificial intelligence, Extension, Chatbot interaction, Comment analysis, Natural Language Processing (NLP).

I. INTRODUCTION

In the contemporary digital landscape, video-sharing platforms have become the primary medium for information dissemination, education, and entertainment. YouTube, being the most prominent among them, hosts billions of videos across diverse categories, catering to a global audience with varied interests [23]. However, the vast expanse of content presents significant challenges for users who seek efficient comprehension, meaningful engagement, and effective navigation through comments [18][27]. Users often face information overload, spending considerable time watching lengthy videos to extract relevant insights [30].

To address these challenges, various tools and methodologies have been developed. Video summarization techniques aim to condense lengthy content into concise visual and textual summaries, enabling users to grasp essential information swiftly [16][24]. Recent advancements in natural language processing (NLP) and machine learning have facilitated the extraction of key points from videos, enhancing content consumption efficiency [19]. For instance, the "YouTube Video Summarizer" has been utilized to provide textual and visual summaries, streamlining the knowledge acquisition process for users [12][33].

In addition to video summarization, conversational AI has been integrated into digital platforms to enable contextual interactions. Chatbots, powered by advanced NLP algorithms, have proven effective in answering user queries and providing personalized recommendations [20][28]. These systems enhance user engagement by delivering relevant information in a conversational manner, thus bridging the gap between passive consumption and active interaction [22][31].

User engagement is further influenced by the dynamics of comment sections, where viewers express their opinions, ask questions, and engage in discussions. Analyzing these comments provides valuable insights into audience preferences, sentiment trends, and content reception [17][25]. Sentiment analysis and topic modeling have been widely applied to YouTube comments to understand viewer emotions and emerging topics [14][21]. However, the overwhelming volume of comments necessitates intelligent systems capable of extracting meaningful insights efficiently [29].

Building upon these advancements, this project, ContentConcise, offers an innovative solution designed to enhance the YouTube viewing experience. It integrates three core functionalities: real-time video summarization, an interactive chatbot for video-related inquiries, and a comprehensive extension for comment analysis. This paper explores the architectural design, implementation challenges, and user experience evaluation of ContentConcise, demonstrating its potential to address the current limitations of video content consumption and community interaction [13][26][32][35].

II. LITERATURE REVIEW

Recent years have seen a surge in research activity related to video summarization, with various techniques garnering significant results. Many of these techniques are based on methods that employ keyframes or are driven by structure. Keyframe-based techniques generate a summary using a selection of semantically relevant keyframes from the video. Their respective attention scores influence the selection of keyframes, ensuring that the most informative and engaging frames are included [9].

One approach to video summarization involves a model that incorporates three fundamental algorithms: TF-IDF, Bidirectional Encoder Representations from Transformers (BERT), and Latent Semantic Analysis (LSA). LSA leverages Singular Value Decomposition (SVD) to extract relevant themes from text effectively. In contrast, TF-IDF identifies crucial words within each sentence, emphasizing their significance. BERT, in this process, encodes the sentences and captures the positional embedding of topics. This combination facilitates a comprehensive and coherent summary [9]. Other research provides detailed insights into text summarization methods, including BERT, TF-IDF, Sentence Ranking, K-means clustering, and K-nearest neighbors, which have been widely used in NLP applications [11]. Another model utilizes latent semantic analysis and NLP, specifically targeting legal judgments to generate concise and informative summaries [13]. However, its reliance on word similarity rather than conceptual understanding limits its effectiveness.

A different approach to text summarization utilizes tools like NLTK and Spacy, with studies showing that Spacy performs text summarization more efficiently than NLTK due to its specialized design for NLP tasks [10].

Video summarization techniques extend beyond text-based approaches. For example, one model transforms extractive text summaries into audio, enhancing accessibility, although it faces challenges with large documents and lacks an efficient mechanism for audio conversion [12].

Graph-based techniques have also been explored for video summarization. One approach treats a video as an undirected graph and uses a cut detection algorithm to partition the video into clusters, forming a temporal graph. Scene detection is achieved using a shortest-path algorithm, and the final summary is generated by leveraging structural and attention information [14]. In contrast, another method utilizes super-frame segmentation to extract keyframes based on visual appeal or 'interestingness' scores, which are determined by low, mid, and high-level features [15].

Deep learning techniques have significantly advanced video summarization capabilities. One approach utilizes deep neural networks for abstractive summarization, enabling the distinction between relevant and irrelevant information more effectively than traditional methods [16]. Another method categorizes videos into static and dynamic types before performing transcript translation, utilizing short boundary detection for accurate segmentation [17].

Further innovations include a prototype for video indexing tailored to lecture videos, leveraging Syntactic Similarity Measures and dynamic programming for auto-caption generation [18]. Real-time video summarization approaches for mobile platforms analyze footage during recording and simultaneously generate summaries by evaluating both intrinsic video content and external metadata [19]. Other studies highlight the limitations of standard summarization systems, advocating for customized solutions that generate summaries based on user preferences and semantically relevant video segments [20]. Supervised learning techniques have also been applied for category-specific video summarization, where cross-validation scores and state transitions between frames are utilized to generate cohesive summaries [21].

These advancements in video summarization techniques have paved the way for more efficient and accurate content extraction. Building upon these developments, ContentConcise integrates summarization algorithms, interactive chatbots, and a extension for comment analysis to enhance the YouTube viewing experience.

Table 1. Summarization of other existing works

Author	Dataset	Technique	Limitations
Huang et al. [22]	Dataset based on query- video pairs	Specialized attention network and GPT-2 (Static)- based contextualized word representations	Embedding dimension influences model training speed and effectiveness.
Narasimhan et al. [23]	Query-focused dataset for video summarizing	Bi-Modal Transformer for the creation of dense video captions and CLIP-It, a language-guided multimodal transformer (Static)	Unsuitable prejudices are embedded. However, these biases can be enhanced.
Huang et al. [24]	Summary of many models of video	Feature fusion and Dictionary-based BOW (Static)	Semantic understanding can be lacking in the BOW model due to its isolated word treatment.
Xiao et al. [25]	Query-focused dataset for video summarizing	QSAN, or Query-Biased Self-Attentive Network: A dynamic caption generator with reinforcement	Substantial pre-processing, such as curating and sanitizing caption data, is necessary.
Nalla et al. [26]	Dataset for query-focused video summarization	Local and global focus Dynamic feature fusion	In a feature fusion model, there's a possibility of some data getting lost.
Xiao et al. [27]	Video summarization dataset with a query focus	CNN, local media, and worldwide exposure	The complexity of computation is significant.
Jiang et al. [28]	Query-focused video summarization dataset	A variable autoencoder with a module for multilayer self-attention. Utilize user-oriented diversity and a stochastic (random) latent variable (Dynamic) for the diversity factor.	Queries are expressed as words, which can result in increased computational time.
Vasudev an et al. [29, 30]	Dataset with relevance and variety	Submodular combination of goals (LSTM) (Static)	Long videos can lead to a substantial increase in inference cost.

Table 2. Performance based on unsupervised models

Author	Dataset	Technique	Limitations
Zhang et al. [33]	Query focused video summarization dataset	Three-player Generative Adversarial Network	It's highly dependent on parameter decisions and is computationally demanding.
Zhang et al. [34]	Query-focused video summarization dataset	Network-based on deep reinforcement learning for summarizing	Too many states can negatively impact the results.
Sreeja et al. [35]	Films demonstrating vehicles and academic inspections	DL-based object detection and ontologies from the semantic web for query inferences	Focusing solely on key objects can hinder the proper handling of semantic relationships.
Sekh Arif et al. [36]	CCTV datasets from Sherbrook Street and VIRAT	Based on the clustering of tubes	Consideration is given solely to significant objects and their temporal motion.
Junyu Gao et al. [37]	Highlights from SumMe, TVSum, and YouTube	Relationship-aware hard assignments based on graph neural networks for choosing key clips and assignment-learning graph	Video graphs lack the representation of object and scene semantics.

Table 3. Qualitative analysis of models

Author	Approach	Advantages	Drawbacks
Sandra Eliza Fontes de Avila et al. [38]	Using hue colour histograms as feature descriptors and K Mean clustering, this method chooses aesthetically different groups.	1. Colour histograms are low-level feature descriptors that are resilient to even minor changes in camera position. 2. This method uses fewer resources to get good results since clustering takes a lot less time and computing power than neural net structures.	1. The Clustering Algorithm disregards the chronological order 2. Colour histograms do not account for the orientation or the colour dispersion in the frame. As a result, a picture with identical colours scattered differently is compared to an image with same colours spread differently.
Ke Zhang et al. [39]	The model is built on the BiLSTM architecture. At each temporal step, A multi-layer perceptron receives the output from these LSTM layers together with a visual frame feature to produce either a binary frame label or a frame level significance score.	Because of its ability to retain prior information, LSTMs successfully simulate the changeable temporal dependencies.	1. Because recurring models computations cannot be parallelized, computing resources are well-spent. 2. Poor performance when simulating scenarios with dynamic changes because the structural order needs to be upheld.
Behrooz Mahas- seni et al. [40]	By utilizing a variable autoencoder to produce a corresponding summary and feeding it to the discriminator in an adversarial situation, the architecture trains a keyframe selector LSTM.	1. Unsupervised Approach eliminates the necessity for difficult to get annotated user data. 2. Acquires knowledge of intermediate representations, which may be valuable in other contexts. 3. In some circumstances, negative criticism may be more helpful than user- reported facts. 4. Regularisation might be applied in different places, focussing on different meanings.	1. It's well known that GANs may be trained in both time and memory. 2. Because the goal is to achieve global representation, subtle changes that can be significant to consumers are difficult to record. 3. User semantics need to be recorded while creating a video summary.
Mrigank Rochan et al. [41]	For video summarization, a fully convolutional model has been modified.	1. Since convolution models are independent of prior outcomes, they allow parallel computing. Comparing this to LSTM methods, training is made more efficient. 2. CNNs can represent the entire range at a considerably lower depth than LSTMs, enabling high level of network to context aggregation sooner. In LSTMs, the last node is only considered once.	1. Loss of resolution and low-level semantics induced by the convolutional layer stack and repeated downsampling of the inputs biases the output towards contextual information while ignoring local knowledge. 2. Excessive upsampling spreads a limited number of values across a vast area, reducing uniqueness and making many nodes seem the same.
Kaiyang Zhou et al. [42]	Encoder-Decoder Architecture training using reinforcement learning framework. Rewards are determined by the resulting summary's variety and representativeness.	Encoder-Decoder Architecture training using reinforcement learning framework. Rewards are determined by the resulting summary's variety and representativeness.[40]	1. A focus on diversity creates issues when there are slow or subtle changes. 2. LSTMs result in ineffective training. 3. Includes no compensation for long- term dependency.

IV. METHODOLOGY & PROPOSED SYSTEM

The proposed design for ContentConcise consists of three core components: Real-Time Video Summarization, an Interactive Chatbot for Video Queries, and an Extension For Comment Analysis. The architecture is modular and scalable, leveraging advanced large language models to provide an enhanced user experience on YouTube. The system is implemented as a browser extension, ensuring seamless integration and accessibility.

Fig 1.Summarization flowchart

1. Real-Time Video Summarization

This module aims to provide concise summaries of YouTube videos while they are being played. It is designed to extract key information without interrupting the viewing experience.

1.1. Key Features:

- **Transcript Extraction:** The video's transcript is fetched using YouTube's API. In the absence of a transcript, an automated speech-to-text model is employed.
- **Text Processing:** Noise reduction techniques are applied to eliminate filler words, repetitions, and irrelevant content.
- **Summarization Algorithms:**
    - **Extractive Summarization:** Utilizes Gemini to identify key sentences from the transcript.
- **Real-Time Display:** The summary is displayed in a sidebar, dynamically updating as the video progresses.

1.2. Workflow:

1. Transcript Extraction:

- First, the transcript of the YouTube video is extracted using YouTube's API or a speech-to-text model, which listens to the audio and converts it into a text format. This ensures that every spoken word in the video is captured accurately.

2. Cleaning and Preprocessing:

- The extracted transcript is then cleaned by removing any irrelevant noise or filler words (such as "um," "uh," or "you know"). Unnecessary punctuation and incorrect transcriptions are fixed to improve readability and accuracy.

3. Summarization:

- Once the transcript is cleaned, gemini is used for summarization . These models identify and retain the most important sentences or phrases from the transcript, ensuring that the summary is both concise and informative.

4. Real-Time Display:

- The summarized content is displayed alongside the video player in real time, so viewers can follow along with a shorter, clearer version of the video's content, making it easier to grasp the key points while watching.

2. Interactive Chatbot for Video Queries

This component enhances user engagement by allowing viewers to ask contextual questions about the video content.

2.1. Key Features:

- **Contextual Understanding:** The chatbot maintains context throughout the conversation, ensuring accurate and relevant answers.
- **Question Answering Models:** Utilizes Gemini and s fine-tuned on QA datasets.
- **Multimodal Integration:** Supports text and voice-based interactions.
- **Personalization:** Learns user preferences over time to provide tailored responses.

2.2. Workflow:

1. Capturing User Input:

- Users submit their questions to the chatbot.

2. Context Understanding:

- Once the question is received, it is encoded using a Gemini model, which processes the input and understands the context of the query. This helps ensure that the system fully comprehends what the user is

asking, taking into account the nuances of the question.

3. Retrieving Relevant Information:

- After the context is understood, the system retrieves the most relevant sections from the video's transcript or the previously generated summarized content. This ensures that the answer is based on accurate and pertinent information.

4. Generating the Answer:

- The relevant information is then processed by Gemini, which generates a well-structured and clear answer to the user's query, providing a detailed response based on the transcript.

5. Real-Time Chatbot Interface:

- The generated response is displayed in a chatbot interface that appears alongside the video player. This allows the user to interact with the content dynamically, as they get answers to their questions in real time while watching the video.

1.

3. Extension for Comment Analysis

This module provides insightful analytics on the comments section of a YouTube video, helping users understand audience sentiment and trending topics.

Key Features:

1. **Comment Scraping**:
   - The extension collects user comments from YouTube videos, allowing for real-time analysis of user opinions and feedback.
2. **Sentiment Analysis**:
   - It performs sentiment analysis on the scraped comments, categorizing them as positive, negative, or neutral, providing valuable insights into public opinion about the video.
3. **Comment-Based Summary Generation**:
   - Based on the sentiment and content of the comments, the extension generates a summary that encapsulates the overall sentiment and main themes discussed by users.
4. **Chatbot with Gemini LLM**:
   - A chatbot is integrated into the extension, allowing users to ask questions.

Workflow:

- The extension begins by scraping user comments from a selected YouTube video to gather data for analysis.
- The scraped comments are processed for sentiment analysis, categorizing each comment's sentiment into positive, negative, or neutral groups.
- Based on the sentiment and content of the comments, a summary is created that highlights the most frequent opinions and themes discussed in the comments section.
- The user submits a question to the chatbot, which uses the Gemini LLM to pull relevant information from the analyzed comments and provides a response in real-time.

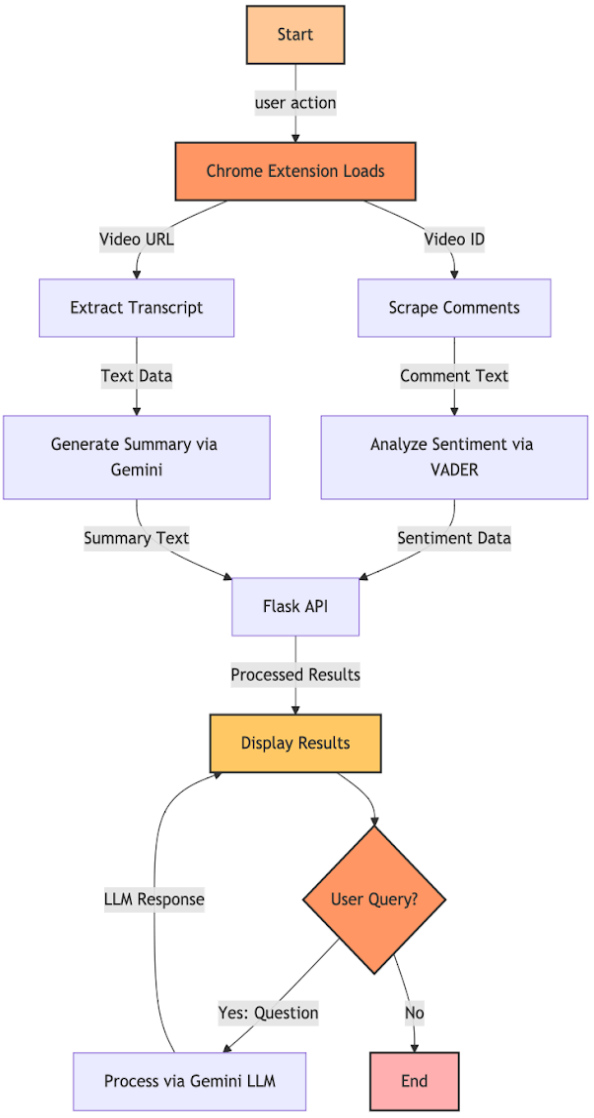


Fig 2. Flow-Chart

1. Video is played

- The process begins when a user accesses a YouTube video with the Chrome Extension enabled.

2. Chrome Extension Loads

- The extension initializes and begins executing its tasks.

3. Parallel Tasks

- Scrape Comments: The extension collects user comments from the YouTube video.
- Extract Transcript: Simultaneously, it retrieves the video's transcript .

4. Data Processing

- The scraped comments are analyzed using the VADER sentiment analysis tool to determine whether they are positive, negative, or neutral.
- The extracted transcript is summarized using Gemini, a large language model, to provide a concise version of the video's content.

5. Flask API

- The results from sentiment analysis and transcript summarization are sent to a backend server built using Flask, which handles data routing and responses.

6. Display Results

- The processed information (sentiment analysis and summary) is shown to the user in the extension interface.

7. User Query

- The user's question is processed using the Gemini LLM to generate a relevant answer.

VI. Results and discussion

Screenshots of User Interface (UI) for the respective module

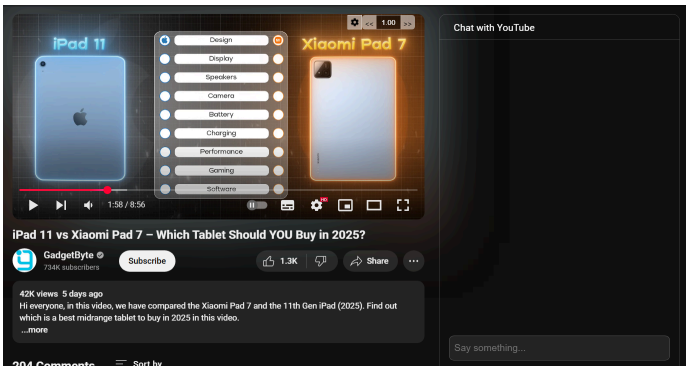


Fig.4 UI of the extension

The user provides an input query or command, and the AI responds to this input by analyzing the transcript of the YouTube video. The AI uses the video's transcript to extract relevant information and generate a response tailored to the user's query.

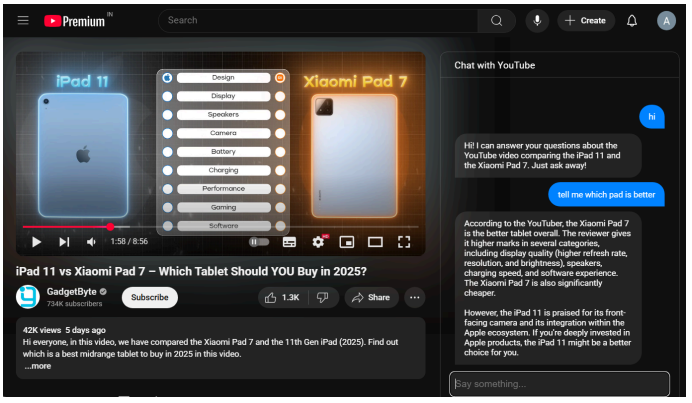


Fig.5 Working of the Extension

The process involves scraping the comments from a YouTube video shown in the figure below.

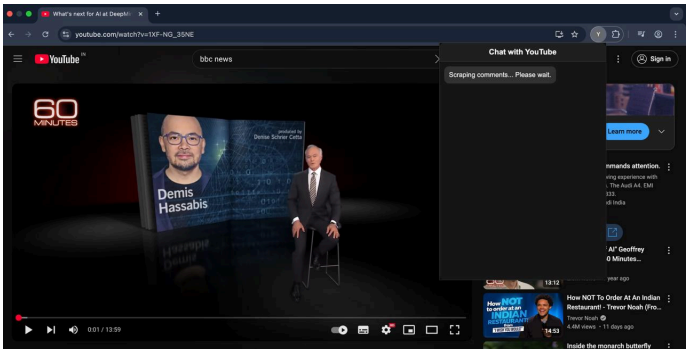


Fig.6 Comment Scraping

The user requests a summary of the comments from the chatbot, and the chatbot provides a concise overview of the

main themes, opinions, and sentiments expressed in the comments

The user asks the chatbot a specific query related to the analysis of the comments, and the chatbot provides insights or answers based on the analyzed data from the comments.

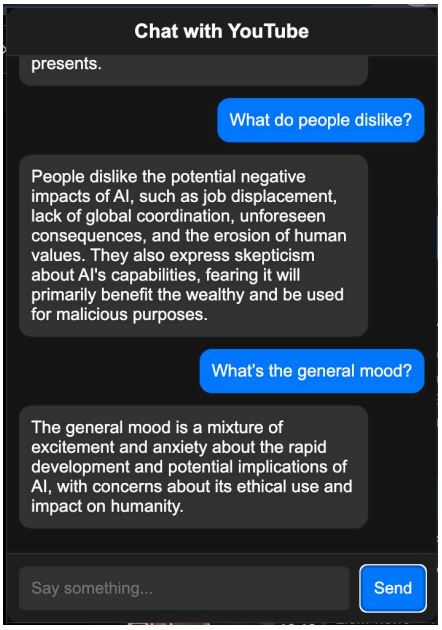


Fig.7 Comment Analysis bot

The screenshots of the User Interface (UI) for the respective module showcase the design and functionality of the module, highlighting how users interact with the chatbot to analyze comments and retrieve insights. The UI is designed to be intuitive and user-friendly, facilitating seamless communication between the user and the chatbot.

Performance Evaluation measures

Cosine Similarity

Cosine similarity is a metric used to measure how similar two vectors are, irrespective of their size. It calculates the cosine of the angle between two non-zero vectors in an inner product space. The value of cosine similarity ranges from -1 to 1, where:

- 1 indicates that the vectors are identical (the angle between them is 0°),
- 0 indicates that the vectors are orthogonal (i.e., there is no similarity),
- -1 indicates that the vectors are diametrically opposite.

The formula for cosine similarity between two vectors A and B is given by:

$$Cosine\ Similarity = \frac{A.B}{|A||B|} \quad (1)$$

The proposed system leverages the following categories of input parameters to deliver relevant and contextual responses:

A. User Interaction – Includes user queries via text (and optionally voice), with intent detection for tasks such as summarization or feedback analysis.

B. Content Context – Utilizes video metadata (title, description, tags, upload date), transcripts, and current video timestamp to provide accurate, time-aware summaries and responses.

C. Engagement Signals – Analyzes likes, dislikes, and user comments to assess content relevance. User preferences are also considered to personalize suggestions and responses.

These parameters collectively enhance the system's responsiveness and personalization within the YouTube interface.

Result for cosine similarity with original text:

```
Cosine Similarity (Original vs. Gemini): 0.8055
Cosine Similarity (Original vs. ChatGPT): 0.7623
Cosine Similarity (Original vs. Llama): 0.7169
```

Fig.8 Similarity Analysis of Language Model Outputs

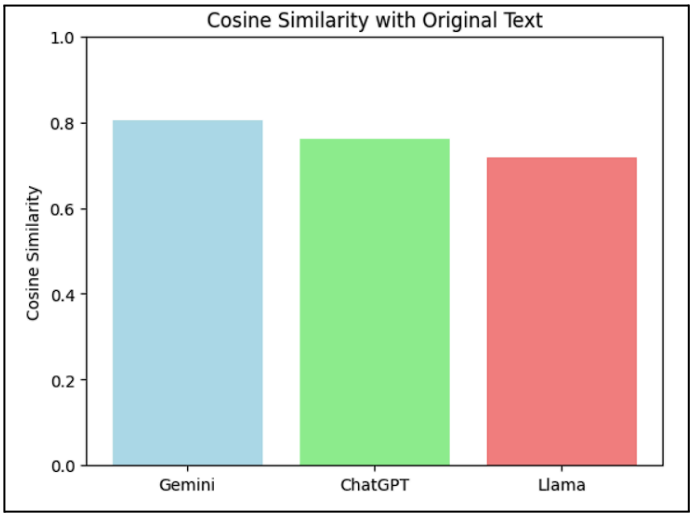


Fig. 9 Cosine Similarity of Model Outputs with Original Text

Result for cosine similarity with other model summary:

```
Cosine Similarity (Gemini vs. ChatGPT): 0.8998
Cosine Similarity (Gemini vs. Llama): 0.9033
Cosine Similarity (ChatGPT vs. Llama): 0.9371
```

Fig. 10 model vs other model

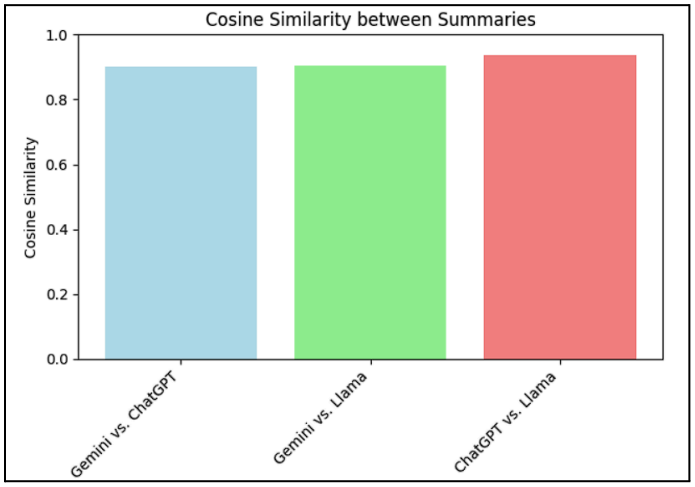


Fig.11 Cosine Similarity Between Model Summaries

The analysis of cosine similarity scores, as depicted in the bar graph titled Cosine Similarity Between Model Summaries, reveals a strong overall agreement in the information extracted by Gemini, ChatGPT, and Llama. With all pairwise comparisons yielding high scores (ranging from approximately 0.90 to 0.94), it's evident that these models identified and conveyed largely the same key aspects from the original text. Notably, the summaries produced by ChatGPT and Llama exhibited the highest degree of similarity, suggesting a closer alignment in their summarization approaches or the specific linguistic choices they made. While Gemini's output also demonstrated substantial overlap with the other two, the slightly lower similarity scores indicate the presence of subtle variations in its summary, potentially in terms of phrasing, emphasis, or the specific vocabulary employed. Nevertheless, the consistently high similarity across all pairs underscores the effectiveness of these models in capturing the core content of the source material.

VIII. CONCLUSION

ContentConcise presents an innovative solution for enhancing the YouTube viewing experience through automated video summarization, intelligent chatbot interaction, and insightful comment analysis. By leveraging advanced natural language processing models and state-of-the-art speech-to-text technologies, the system effectively generates concise and accurate summaries, enabling users to quickly grasp the core content of lengthy videos. The integrated chatbot provides real-time, context-aware responses, enhancing user engagement and interactivity. Additionally, the comment analysis feature offers valuable insights into audience sentiments and trends, enriching user understanding of community perspectives.

The system's architecture ensures scalability and efficiency, leveraging cloud deployment and serverless architecture for seamless processing of multiple videos. Continuous model updates based on user feedback contribute to the system's adaptability and accuracy over time. Overall, ContentConcise bridges the gap between video consumption and information extraction, making content more accessible and digestible. This approach not only saves users' time but also enhances their engagement by providing meaningful interactions and insights, paving the way for future advancements in intelligent video content summarization and analysis tools.

REFERENCES

[1] Otani, M., Nakashima, Y., Rahtu, E., Heikkilä, J., and Yokoya, N. Video Summarization using Deep Semantic Features. In *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part V 13*, Springer International Publishing, pp. 361-377, 2017.

[2] Apostolidis, E., Adamantidou, E., Metsai, A.I., Mezaris, V., and Patras, I. Video Summarization using Deep Neural Networks: A Survey. *Proceedings of the IEEE*, vol. 109, no. 11, pp. 1838-1863, 2021.

[3] Zhang, S., Zhu, Y., and Roy-Chowdhury, A.K. Context-Aware Surveillance Video Summarization. *IEEE Transactions on Image Processing*, vol. 25, no. 11, pp. 5469-5478, 2016.

[4] Kwon, J. and Lee, K.M. A Unified Framework for Event Summarization and Rare Event Detection from Multiple Views. *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1737-1750, 2014.

[5] Sridevi, M. and Kharde, M. Video Summarization using Highlight Detection and Pairwise Deep Ranking Model. *Procedia Computer Science*, vol. 167, pp. 1839-1848, 2020.

[6] Varini, P., Serra, G., and Cucchiara, R. Personalized Egocentric Video Summarization of Cultural Tour on User Preferences Input. *IEEE Transactions on Multimedia*, vol. 19, no. 12, pp. 2832-2845, 2017.

[7] Ji, Z., Xiong, K., Pang, Y., and Li, X. Video Summarization with Attention-Based Encoder–Decoder Networks. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 6, pp. 1709-1717, 2019.

[8] Fajtl, J., Sokeh, H.S., Argyriou, V., Monekosso, D., and Remagnino, P. Summarizing Videos with Attention. In *Computer Vision– ACCV 2018 Workshops: 14th Asian Conference on Computer Vision, Perth, Australia, December*

2–6, 2018, Revised Selected Papers 14, Springer International Publishing, pp. 39-54, 2019.

[9] Gupta, H. and Patel, M. Method of Text Summarization using LSA and Sentence Based Topic Modelling with Bert. In *2021 international conference on artificial intelligence and smart systems (ICAIS)*, IEEE, pp. 511-517, 2021.

[10] Jugran, S., Kumar, A., Tyagi, B.S., and Anand, V. Extractive Automatic Text Summarization using SpaCy in Python & NLP. In *2021 International conference on advance computing and innovative technologies in engineering (ICACITE)* IEEE, pp. 582-585, 2021.

[11] Adhikari, S. Nlp Based Machine Learning Approaches for Text Summarization. In *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, IEEE, pp. 535-538, 2020.

[12] Madhuri, J.N. and Kumar, R.G. Extractive Text Summarization using Sentence Ranking. In *2019 international conference on data science and communication (IconDSC)*, IEEE, pp. 1-3, 2019.

[13] Merchant, K. and Pande, Y. Nlp Based Latent Semantic Analysis for Legal Text Summarization. In *2018 international conference on advances in computing, communications and informatics (ICACCI)*, IEEE, pp. 1803-1807, 2018.

[14] Ngo, C.W., Ma, Y.F., and Zhang, H.J. Video Summarization and Scene Detection by Graph Modeling. *IEEE Transactions on circuits and systems for video technology*, vol. 15, no. 2, pp. 296-305, 2005.

[15] Gygli, M., Grabner, H., Riemenschneider, H., and Van Gool, L. Creating Summaries from User Videos. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII 13*, Springer International Publishing, pp. 505-520, 2014.

[16] Dilawari, A. and Khan, M.U.G. ASoVS: Abstractive Summarization of Video Sequences. *IEEE Access*, vol. 7, pp. 29253-29263, 2019.

[17] Smaïli, K., Fohr, D., González-Gallardo, C.E., Grega, M., Janowski, L., Jouvet, D., Komorowski, A., Koźbiał, A., Langlois, D., Leszczuk, M. and Mella, O. A First Summarization System of a Video in a Target Language. In *Multimedia and Network Information Systems: Proceedings of the 11th International Conference MISSI 2018 11*, Springer International Publishing, pp. 77-88, 2019.

[18] Jaiswal, S. and Misra, M. Automatic Indexing of Lecture Videos using Syntactic Similarity Measures. In *2018 5th International Conference on Signal Processing and Integrated Networks (SPIN)*, IEEE, pp. 164-169, 2018.

[19] Choudhary, P., Munukutla, S.P., Rajesh, K.S., and Shukla, A.S. Real Time Video Summarization on Mobile Platform. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, IEEE, pp. 1045-1050, 2017.

[20] Kannan, R., Ghinea, G., Swaminathan, S., and Kannaiyan, S. Improving Video Summarization Based on User Preferences. In *2013 Fourth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG)*, IEEE, pp. 1-4, 2013.

[21] Basak, J., Luthra, V., and Chaudhury, S. Video Summarization with Supervised Learning. In *2008 19th International Conference on Pattern Recognition*, IEEE, pp. 1-4, 2008.

[22] Huang, J.H., Murn, L., Mrak, M., and Worring, M. Gpt2mvs: Generative Pre-Trained Transformer-2 for Multi-Modal Video Summarization. In *Proceedings of the 2021 International Conference on Multimedia Retrieval*, pp. 580-589, 2021.

[23] Narasimhan, M., Rohrbach, A., and Darrell, T. Clip-It! Language-Guided Video Summarization. *Advances in Neural Information Processing Systems*, vol. 34, pp. 13988-14000, 2021.

[24] Huang, J.H. and Worring, M. Query-Controllable Video Summarization. In *Proceedings of the 2020 International Conference on Multimedia Retrieval*, pp. 242-250, 2020.

[25] Xiao, S., Zhao, Z., Zhang, Z., Guan, Z., and Cai, D. Query-Biased Self-Attentive Network for Query-Focused Video Summarization. *IEEE Transactions on Image Processing*, vol. 29, pp. 5889-5899, 2020.

[26] Nalla, S., Agrawal, M., Kaushal, V., Ramakrishnan, G., and Iyer, R. Watch Hours in Minutes: Summarizing Videos with User Intent. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, Springer International Publishing, pp. 714-730, 2020.

[27] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., and Polosukhin, I. Attention Is All You Need. *Advances in neural information processing systems*, vol. 30, 2017.

[28] Jiang, P. and Han, Y. Hierarchical Variational Network for User-Diversified & Query-Focused Video Summarization. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, pp. 202-206, 2019.

[29] Vasudevan, A.B., Gygli, M., Volokitin, A., and Van Gool, L. Query-Adaptive Video Summarization via Quality-Aware Relevance Estimation. In *Proceedings of the 25th ACM international conference on Multimedia*, pp. 582-590, 2017.

[30] Gygli, M., Grabner, H., and Van Gool, L. Video Summarization by Learning Submodular Mixtures of Objectives. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3090-3098, 2015.

[31] Sharghi, A., Laurel, J.S., and Gong, B. Query-Focused Video Summarization: Dataset, Evaluation, and a Memory Network Based Approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4788-4797, 2017.

[32] Sharghi, A., Gong, B. and Shah, M. Query-Focused Extractive Video Summarization. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, Springer International Publishing, pp. 3-19, 2016.

[33] Zhang, Y., Kampffmeyer, M., Liang, X., Tan, M., and Xing, E.P. Query-Conditioned Three-Player Adversarial Network for Video Summarization. *arXiv preprint arXiv:1807.06677*, 2018.

[34] Zhang, Y., Kampffmeyer, M., Zhao, X., and Tan, M. Deep Reinforcement Learning for Query-Conditioned Video Summarization. *Applied Sciences*, vol. 9, no. 4, pp. 750, 2019.

[35] Sreeja, M.U. and Kovoor, B.C. A Unified Model for Egocentric Video Summarization: An Instance-Based Approach. *Computers & Electrical Engineering*, vol. 92, pp. 107161, 2021.

[36] Ahmed, S.A., Dogra, D.P., Kar, S., Patnaik, R., Lee, S.C., Choi, H., Nam, G.P., and Kim, I.J. Query-Based Video Synopsis for Intelligent Traffic Monitoring Applications. *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 8, pp. 3457-3468, 2019.

[37] Gao, J., Yang, X., Zhang, Y., and Xu, C. Unsupervised Video Summarization via Relation-Aware Assignment Learning. *IEEE Transactions on Multimedia*, vol. 23, pp. 3203-3214, 2020.

[38] De Avila, S.E.F., Lopes, A.P.B., da Luz Jr, A., and de Albuquerque Araújo, A. VSUMM: A Mechanism Designed to Produce Static Video Summaries and a Novel Evaluation Method. *Pattern recognition letters*, vol. 32, no. 1, pp. 56-68, 2011.

[39] Zhang, K., Chao, W.L., Sha, F., and Grauman, K. Video Summarization with Long Short-Term Memory. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14*, Springer International Publishing, pp. 766-782, 2016.

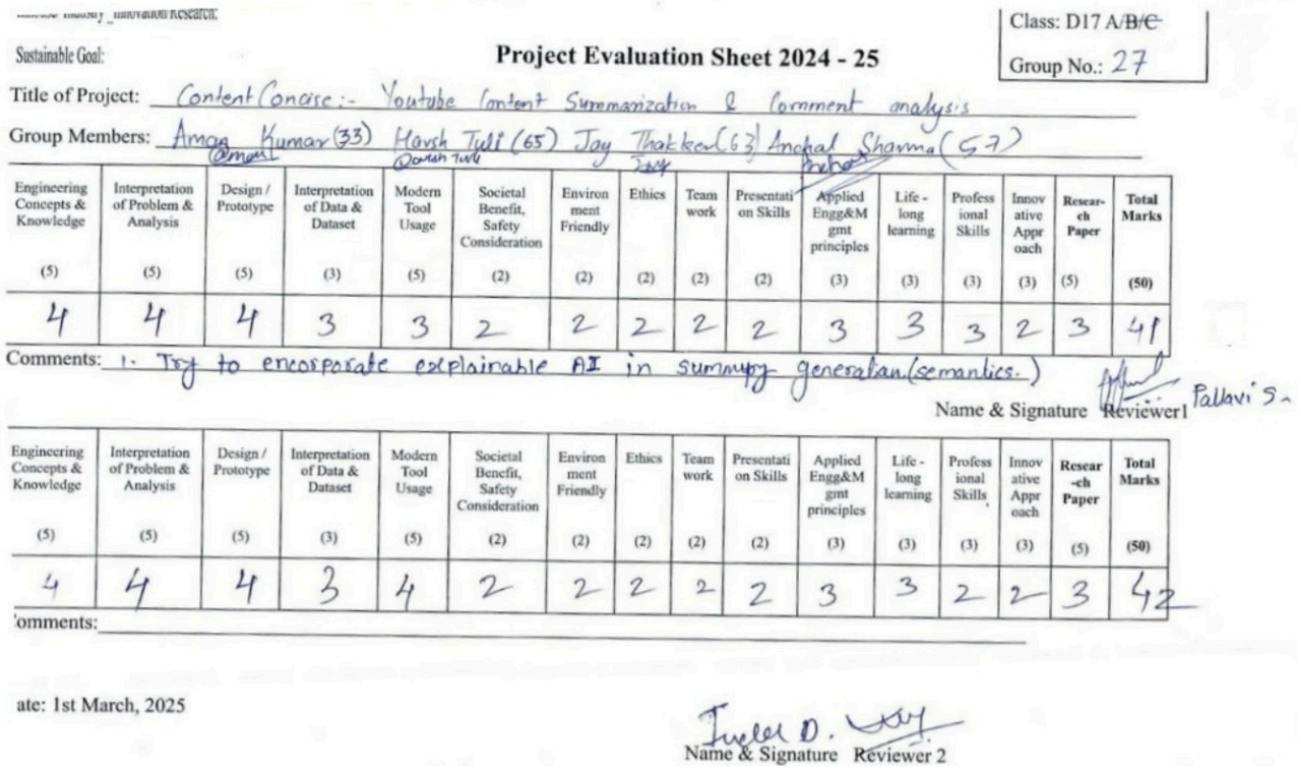
[40] Mahasseni, B., Lam, M., and Todorovic, S. Unsupervised Video Summarization with Adversarial LSTM Networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 202-211, 2017.

[41] Rochan, M., Ye, L., and Wang, Y. Video Summarization using Fully Convolutional Sequence Networks. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 347-363, 2018.

[42] Zhou, K., Qiao, Y., and Xiang, T. Deep Reinforcement Learning for Unsupervised Video Summarization with Diversity- Representativeness Reward. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[43] Ul Haq, H.B., Asif, M., Ahmad, M.B., Ashraf, R., and Mahmood, T. An Effective Video Summarization Framework Based on the Object of Interest using Deep Learning. *Mathematical Problems in Engineering*, vol. 2022, 2022.

Project Review-1 Sheet



Class: D17 A/B/C

Group No.: 27

Sustainable Goal:

Project Evaluation Sheet 2024 - 25

Title of Project: Content Concise :- Youtube Content Summarization & Comment analysis

Group Members: Aman Kumar (33) Harsh Tyagi (65) Jay Thakkar (63) Anshal Sharma (67)

Engineering Concepts & Knowledge	Interpretation of Problem & Analysis	Design / Prototype	Interpretation of Data & Dataset	Modern Tool Usage	Societal Benefit, Safety Consideration	Environment Friendly	Ethics	Team work	Presentation Skills	Applied Engg&Mgmt principles	Life-long learning	Professional Skills	Innovative Approach	Research Paper	Total Marks
(5)	(5)	(5)	(3)	(5)	(2)	(2)	(2)	(2)	(2)	(3)	(3)	(3)	(3)	(5)	(50)
4	4	4	3	3	2	2	2	2	2	3	3	3	2	3	41

Comments: 1. Try to encorporate explainable AI in summary generation (semantics.)

Name & Signature Reviewer1 Pallavi S.

Engineering Concepts & Knowledge	Interpretation of Problem & Analysis	Design / Prototype	Interpretation of Data & Dataset	Modern Tool Usage	Societal Benefit, Safety Consideration	Environment Friendly	Ethics	Team work	Presentation Skills	Applied Engg&Mgmt principles	Life-long learning	Professional Skills	Innovative Approach	Research Paper	Total Marks
(5)	(5)	(5)	(3)	(5)	(2)	(2)	(2)	(2)	(2)	(3)	(3)	(3)	(3)	(5)	(50)
4	4	4	3	4	2	2	2	2	2	3	3	2	2	3	42

Comments:

Date: 1st March, 2025

Name & Signature Reviewer 2

Project Review-2 Sheet

Project Evaluation Sheet 2024 - 25

27

Title of Project: Content Concise : Youtube Content Summarization & content analysis

Group Members: Aman Kumar 33 Anchal Sharma 57 Harsh Tuli 65 Jay Thalekar 64 63

Engineering Concepts & Knowledge	Interpretation of Problem & Analysis	Design / Prototype	Interpretation of Data & Dataset	Modern Tool Usage	Societal Benefit, Safety Consideration	Environment Friendly	Ethics	Team work	Presentation Skills	Applied Engg&Mgmt principles	Life - long learning	Professional Skills	Innovative Approach	Research Paper	Total Marks
(5)	(5)	(5)	(3)	(5)	(2)	(2)	(2)	(2)	(2)	(3)	(3)	(3)	(3)	(5)	(50)
4	4	5	3	4	2	2	2	2	2	3	3	2	2	3	43

Comments: Show true validation of Results.

Name & Signature Reviewer1

Inhouse/ Industry _Innovation/Research:

Engineering Concepts & Knowledge	Interpretation of Problem & Analysis	Design / Prototype	Interpretation of Data & Dataset	Modern Tool Usage	Societal Benefit, Safety Consideration	Environment Friendly	Ethics	Team work	Presentation Skills	Applied Engg&Mgmt principles	Life - long learning	Professional Skills	Innovative Approach	Research Paper	Total Marks
(5)	(5)	(5)	(3)	(5)	(2)	(2)	(2)	(2)	(2)	(3)	(3)	(3)	(3)	(5)	(50)
4	4	4	3	4	2	2	2	2	2	3	3	2	2	3	42

Comments: 1. Can integrate timestamped summaries.
2. Improve accuracy & show summary → comparative analysis with existing tools.

Date: 1st April,2025

Pallavi S.

Name & Signature Reviewer 2