Smart AI Video Summarizer with Text and Video Highlights

A Project Report
submitted in partial fulfillment of the
requirements for the award of the degree of

Bachelor of Technology
in
CSE (Artificial Intelligence & Machine Learning)

by

Morem Charith
23951A6662

Department of
CSE (Artificial Intelligence & Machine Learning)

INSTITUTE OF AERONAUTICAL ENGINEERING
(Autonomous)
Dundigal, Hyderabad – 500 043, Telangana

January, 2026

DECLARATION

I certify that

a. The work contained in this report is original and has been done by me under the guidance of my supervisor(s).

b. The work has not been submitted to any other Institute for any degree or diploma.

c. I have followed the guidelines provided by the Institute for preparing the report.

d. I have confirmed the norms and guidelines given in the Ethical Code of Conduct of the Institute.

e. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the report and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Place: **Signature of the Student**
Date: **Roll No.**

CERTIFICATE

This is to certify that the project report entitled **Smart AI Video Summarizer with Text and Video Highlights** submitted by **Mr. Morem Charith** of Institute of Aeronautical Engineering, Hyderabad in partial fulfillment of the requirements for the award of the Degree Bachelor of Technology in **CSE (Artificial Intelligence & Machine Learning)** is a Bonafide record of work carried out by him/her under my guidance and supervision. The contents of this report, in full or in parts, have not been submitted to any other Institute for the award of any Degree.

Supervisor **Head of the Department**

Date:

APPROVAL SHEET

This project report entitled **Smart AI Video Summarizer with Text and Video Highlights** by **Morem Charith** is approved for the award of the Degree Bachelor of Technology in **CSE (Artificial Intelligence & Machine Learning).**

Examiners **Supervisor**

Principal

Date:

Place:

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without acknowledging the people who made it possible and whose constant guidance and encouragement crowned all efforts with success.

I express my sincere gratitude and profound indebtedness to my project guide, **Dr. D Khalandar Basha,** Associate Professor, CSE (Artificial Intelligence & Machine Learning), Department of Computer Science and Engineering (Artificial Intelligence & Machine Learning), for his valuable guidance, constant support, and constructive suggestions throughout the course of this project work.

I express my sincere gratitude to **Mr. B Mohan,** Assistant Professor and Project Coordinator, for his effective coordination, constant encouragement, and timely guidance throughout the course of this project. His valuable suggestions, systematic monitoring, and readiness to clarify doubts played a crucial role in the smooth progress and successful completion of this work.

I am grateful to **Dr. M Purushotham Reddy**, Professor and Head of the Department, Department of Computer Science and Engineering (Artificial Intelligence & Machine Learning), for his encouragement and support in carrying out this project successfully.

I express my sincere thanks to **Dr. L V Narasimha Prasad**, Professor and Principal, for being a great source of inspiration and for providing the necessary facilities to accomplish this work.

I also thank the college management and **Sri. M Rajashekar Reddy**, Chairman of the Institute of Aeronautical Engineering (IARE), Dundigal, for providing the necessary infrastructure and resources to complete this project.

Finally, I take this opportunity to express my heartfelt gratitude to all those who directly or indirectly supported and helped me in bringing this project to its present form.

ABSTRACT

The rapid growth of digital content in the form of text documents, PDFs, and online videos has made it increasingly difficult for users to extract relevant information quickly. Long articles, research papers, and lengthy videos often lead to information overload, making manual summarization time-consuming and inefficient. To address this challenge, this project proposes a Smart AI Multimodal Summarization System that automatically generates concise summaries and extracts key highlights from multiple content formats within a single unified platform.

The main problem tackled by the system is the absence of an integrated solution capable of processing diverse input types while maintaining accuracy, efficiency, and scalability. Existing tools are typically limited to a single modality or require significant manual effort to identify important video segments. The proposed system overcomes these limitations through a hybrid architecture that combines local and cloud-based artificial intelligence models. Text and PDF inputs are processed locally using the Gemma-3 12B model to ensure faster response times and improved data privacy, while video content is analyzed using the Gemini Cloud API with the Gemma-3 27B Vision model to achieve accurate visual understanding and timestamp extraction.

The system accepts text, PDF, YouTube video URLs, and highlight requests as inputs. The backend server intelligently routes each request to the appropriate processing module, performs semantic analysis, and generates summaries in paragraph or bullet format. For highlight extraction, the system identifies key moments and compiles them into a merged video that users can download and share. The results are stored for future access, enabling users to review their summarization history.

The implementation demonstrates that the proposed system significantly reduces content consumption time while preserving essential information and contextual relevance. The hybrid AI approach improves processing efficiency and scalability, making the system suitable for students, researchers, and professionals who require quick access to critical insights from large volumes of multimedia content.

Keywords: Smart Video Summarizer, Gemma 3,Text Summarization, PDF Processing, Highlight Extraction, Hybrid AI Architecture, Fast API, React, TF-IDF.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS	DEFINITION
NLP	Natural Language Processing
LLM	Large Language Model
API	Application Programming Interface
TF-IDF	Term Frequency – Inverse Document Frequency
GPU	Graphics Processing Unit
DFD	Data Flow Diagram
REST	Representational State Transfer
SQL	Structured Query Language
FFmpeg	Fast Forward Moving Picture Experts Group
SDK	Software Development Kit
URL	Uniform Resource Locator
OSS	Open Source Software
LMS	Learning Management System

CHAPTER 1
INTRODUCTION

1.1 Introduction

In the digital era, the exponential growth of multimedia content has created a significant challenge known as information overload, where users must invest substantial time to extract meaningful insights from lengthy videos, documents, and textual resources. Educational lectures, technical tutorials, research documents, and online media often contain valuable information; however, manually reviewing such content is time-consuming and inefficient.

To address this challenge, the Smart AI Video Summarizer with Text, PDF, and Highlights is proposed as an intelligent, hybrid multimodal system capable of automatically generating concise summaries and extracting important moments from different types of input data. The system supports four primary input modes: YouTube video URLs, plain text, PDF documents, and highlight extraction, enabling users to obtain structured knowledge in significantly reduced time.

Unlike conventional summarization tools that rely solely on cloud-based services or simple extractive methods, the proposed solution adopts a Hybrid Artificial Intelligence Architecture that combines both local inference and cloud-based multimodal intelligence. Textual and document-based summarization tasks are processed using a locally hosted Gemma 3 (12B) model through Ollama, ensuring reduced latency, improved privacy, and cost efficiency. For video content, which requires multimodal understanding of transcripts and visual context, the system integrates the Gemma 3 (27B IT) model via the Google Gemini Cloud API.

The application is designed using a Client–Server Architecture, where a React-based frontend provides a responsive user interface with Google Authentication, and a FastAPI backend orchestrates AI processing, data extraction, and media handling. The system employs Natural Language Processing (NLP) techniques such as tokenization, keyword scoring, and extractive filtering before generating refined abstractive summaries. Additionally, the highlights module identifies important timestamps within videos and enables users to export selected segments as a single compiled highlight reel.

By combining multimodal analysis, hybrid AI deployment, and automated highlight generation, the proposed system aims to enhance productivity, improve knowledge accessibility, and provide users with a scalable solution for intelligent content understanding.

1.2 Objectives

The objectives of the Smart AI Video Summarizer with Text, PDF, and Highlights project focus on developing a comprehensive intelligent system that improves the efficiency of content consumption by automating the extraction of key information from multiple data sources. The system is designed not only to generate summaries but also to enhance user productivity through accurate highlight detection and export capabilities.

Primary Objective

The primary objective of the proposed system is to design and implement a hybrid multimodal summarization platform capable of analyzing YouTube videos, textual data, and PDF documents to automatically generate concise, meaningful summaries and identify important segments with minimal user effort.

Functional Objectives

- Support Multimodal Inputs: Enable users to submit YouTube video URLs, plain text, and PDF documents within a single platform for unified processing.
- Automated Content Processing & Summarization: Extract transcripts or textual content and generate concise, high-quality summaries with customizable length and format options.
- Highlight Detection & Timestamp Mapping: Identify key moments in videos by detecting important sentences and linking them to accurate timestamps.
- Export & Sharing Features: Allow users to select highlights and export them as a merged summarized video for easy access and sharing.

Technical Objectives

- Hybrid AI Architecture: Design a system that combines local and cloud-based LLMs to balance performance, cost, and privacy.
- Local & Cloud Model Integration: Use Gemma 3 (12B) via Ollama for text/PDF summarization and Gemini API with Gemma 3 (27B IT) for video analysis.
- Advanced NLP & Media Processing: Apply NLP techniques and tools like yt-dlp, OpenCV, and FFmpeg for content extraction, analysis, and clip generation.
- Scalable Backend Implementation: Develop a FastAPI-based backend to manage data flow, AI processing, and multimedia operations efficiently.

1.3 Feasibility

The feasibility of the Smart AI Video Summarizer with Text, PDF, and Highlights project is evaluated to determine whether the system can be successfully developed, implemented, and utilized using the available resources, technologies, and time constraints.

The proposed system is technically feasible due to the availability of reliable and widely supported development tools, frameworks, and artificial intelligence models. The system uses a hybrid architecture that combines local and cloud-based Large Language Models to efficiently handle different types of content. Text and PDF summarization tasks are processed locally using the Gemma 3 (12B) model through Ollama, ensuring low latency, privacy, and reduced dependency on internet connectivity. For video processing, which requires multimodal understanding, the system integrates the Gemma 3 (27B IT) model through the Google Gemini Cloud API. The backend is developed using FastAPI, which supports asynchronous operations and efficient request handling, while tools such as yt-dlp, OpenCV, and FFmpeg enable video metadata extraction, frame analysis, and highlight clip generation. Since all required technologies are accessible and compatible with standard development environments, the system can be implemented without major technical challenges.

From an economic perspective, the project is cost-effective because it primarily relies

on open-source tools and libraries. The use of local AI inference for text and document processing significantly reduces the need for continuous cloud resource usage, thereby lowering operational costs. Although the system utilizes cloud-based APIs for video summarization, this usage is limited to specific tasks that require advanced multimodal processing. Additionally, no specialized hardware investment is required beyond a standard computer capable of running the development environment, making the overall cost of development and deployment affordable.

Operationally, the system is feasible as it is designed with a user-friendly interface that supports multiple input formats, including YouTube video URLs, text, and PDF documents. The application provides clear workflow steps, allowing users to easily generate summaries and extract highlights without requiring technical expertise. The integration of Google Authentication ensures secure and convenient access, while the history management feature allows users to revisit previously generated outputs. These design choices ensure that the system can be effectively used in practical environments such as education, research, and professional content analysis.

In terms of scheduling, the modular architecture of the system allows different components—such as text summarization, PDF processing, video analysis, and highlight generation—to be developed and tested independently. The use of established frameworks like React and FastAPI accelerates development and simplifies integration. This structured approach ensures that the project can be completed within a typical academic development timeline while allowing future enhancements without major redesign.

1.4 Existing Methodologies

Before the development of intelligent multimodal summarization systems, several traditional and modern approaches have been used to extract information from long videos, documents, and textual content. These methodologies primarily focus on either extractive summarization techniques or single-modal processing.

One of the most widely used approaches is manual content review, where users watch entire videos or read lengthy documents to identify key information. Although this method ensures high accuracy, it is extremely time-consuming and inefficient, especially when dealing with large volumes of content.

Another commonly used methodology is traditional extractive text summarization, which relies on statistical Natural Language Processing techniques such as keyword frequency analysis and TF-IDF scoring. These methods select important sentences directly from the original content without rewriting or improving readability. While extractive summarization is computationally efficient, it often produces summaries that lack coherence and natural flow.

Several existing tools also depend entirely on cloud-based AI summarization services. These systems send all data to remote servers for processing using Large Language Models. Although such methods can generate high-quality summaries, they introduce concerns related to privacy, latency, and operational cost, especially when processing large documents or frequent requests.

In the context of video content, traditional approaches include basic transcript-based summarization, where only subtitles or captions are analyzed without considering visual context. If transcripts are unavailable or inaccurate, the summarization quality significantly decreases. Some platforms provide simple timestamp detection using keyword matching, but these methods often fail to identify the true importance of segments.

Furthermore, many existing applications are designed for single input type processing, meaning separate tools must be used for text, PDF, and video summarization. This fragmentation reduces usability and efficiency for users who frequently work with multiple content formats.

Demerits of Existing Systems:

- Manual analysis requires significant time and effort, making it impractical for long-duration videos or large documents.
- Traditional extractive summarization methods often produce fragmented and less readable outputs since they do not generate new, context-aware sentences.
- Systems relying entirely on cloud-based processing introduce privacy risks because user data must be transmitted externally.
- Continuous cloud API usage increases operational cost and may result in higher latency.

- Transcript-only video summarization fails when captions are unavailable or inaccurate.
- Existing methods rarely incorporate multimodal understanding, ignoring visual and contextual information present in videos.
- Many available tools support only a single content type, forcing users to switch between multiple platforms.
- Basic keyword-based highlight detection often misses important segments or produces redundant timestamps.
- Lack of integrated highlight export functionality prevents efficient extraction of key video moments.

1.5 System Requirements

The successful implementation and execution of the Smart AI Video Summarizer with Text, PDF, and Highlights system requires a combination of appropriate hardware resources and supporting software technologies. These requirements ensure efficient processing of multimedia content, AI model inference, and smooth user interaction.

Software Requirements

- **Programming Languages:**

  The application is developed using Python for backend processing and JavaScript for frontend development.

- **Frontend Technologies:**

  React.js is used to build the user interface, Vite is used for fast development and bundling, and Tailwind CSS is used for styling and responsive design.

- **Backend Framework:**

  FastAPI is used for building the backend API, handling user requests, managing data processing workflows, and integrating AI services.

- **Database Management System:**

MySQL is used for storing user history, generated summaries, highlight data, and system statistics.

- **Local AI Runtime:**

  Ollama is used to run the local Gemma 3 (12B) language model for text, PDF, and highlight processing.

- **Artificial Intelligence Models:**

  The system uses Gemma 3 (12B) for local summarization tasks and Gemma 3 (27B IT) through the Google Gemini API for video-based multimodal analysis.

- **Natural Language Processing Libraries:**

  NLTK and Scikit-learn are used for text preprocessing, tokenization, and sentence scoring.

- **Video and Media Processing Tools:**

  yt-dlp is used for extracting video metadata and transcripts, OpenCV is used for frame-related processing, and FFmpeg is used for generating and merging highlight clips.

- **Authentication Service:**

  Google Authentication (Firebase) is implemented to provide secure login and user management.

- **Development and Version Control Tools:**

  Visual Studio Code is used as the primary development environment, and Git is used for version control and project management.

- **Web Browser Support:**

  The application is compatible with modern browsers such as Google Chrome, Microsoft Edge, and Mozilla Firefox.

CHAPTER 2
LITERATURE REVIEW

Hang Hua et al. (2024) [1] – V2Xum-LLM: Cross-Modal Video Summarization with Temporal Prompt Instruction Tuning

Methodology:

This paper proposes a unified framework called V2Xum-LLM for multimodal video summarization that leverages large language models equipped with *temporal prompt instruction tuning*. The authors introduce a new large dataset with aligned video and text summaries to support effective training. Their model integrates visual features and temporal prompts, enabling the system to understand both the visual content and its temporal structure. Unlike traditional methods that separate tasks by modality, this approach uses a single language model for multiple summarization tasks, improving semantic coherence across video-to-text and other summarization formats.

Drawbacks:

The model requires extensive training data that may not exist in all domains, making generalization difficult. Temporal conditioning increases computational complexity, and current evaluation metrics for multimodal summarization are still limited, posing challenges for consistent performance comparison.

Bo He et al. (2023) [2] – Align and Attend: Multimodal Summarization with Dual Contrastive Losses

Methodology:

The study introduces A2Summ, a unified transformer-based model for multimodal summarization that effectively aligns and attends to inputs from different modalities such as video frames and text transcripts. Unlike traditional approaches that treat modalities independently, A2Summ uses an *alignment-guided self-attention mechanism* to incorporate temporal correspondence across modalities, improving cross-modal integration. The model also introduces two novel *contrastive loss functions*—inter-sample and intra-sample—that help the model learn both global and fine-grained relationships between video and text pairs. Experiments on standard video summarization datasets (TVSum, SumMe) and multimodal summarization benchmarks (Daily Mail, CNN) show that this method improves summarization quality by capturing meaningful semantic signals from both video and text.

Drawbacks:

The model's complexity increases due to additional loss components and alignment mechanisms, requiring more computational resources and careful training. Performance gains also depend on the availability of well-aligned multimodal datasets, which may not be available for all domains.

Aman Khullar & Udit Arora (2020) [3] – MAST: Multimodal Abstractive Summarization with Trimodal Hierarchical Attention

Methodology:

Khullar and Arora present MAST, a multimodal abstractive summarization model that combines text, audio, and video modalities using a *trimodal hierarchical attention* mechanism. Prior approaches mainly integrated text and video features, but this model also incorporates audio information to better capture comprehensive multimodal context. The architecture uses modality encoders for each data type and applies hierarchical attention to fuse paired modality context vectors, ultimately enhancing the representation used to generate the summary. Experiments on the How2 dataset demonstrate that MAST achieves better F1 and ROUGE-L scores compared to previous bimodal methods, validating the benefit of including audio modality and hierarchical attention for multimodal summarization.

Drawbacks:

Despite improved performance, the model increases computational complexity due to the trimodal hierarchical attention layers and requires well-aligned text, audio, and video data for effective learning. Audio features may also introduce noise if not well correlated with important textual content, limiting universal applicability.

Parul Saini et al. (2023) [4] – Video summarization using deep learning techniques: A detailed analysis and investigation

Methodology:

This paper provides a comprehensive analysis of video summarization approaches that use deep learning methods. The authors review a variety of techniques such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), long short-term memory (LSTM) networks, and other deep architectures used to identify and extract important events and frames from videos. The paper categorizes existing methods based on how they handle feature extraction, event detection, keyframe selection, and summarization strategies. It also discusses the application-specific

performance of these techniques and highlights major trends in deep learning-based video summarization research, including supervised, unsupervised, and weakly supervised models.

Drawbacks:

Despite thorough coverage, the paper notes that many deep learning approaches struggle with long-duration videos due to computational complexity, and may not generalize well across different video domains. There are also challenges related to dataset availability, inconsistent evaluation metrics, and difficulties in real-time deployment for large video streams.

Anubhav Jangra et al. (2021) [5] – A Survey on Multi-modal Summarization

Methodology:

This paper presents a comprehensive survey of research in the area of multi-modal summarization, covering techniques that integrate information from multiple sources such as text, audio, image, and video to generate concise summaries. The authors review existing methods for extracting salient information across modalities and discuss various *fusion strategies* that combine heterogeneous data types, as well as commonly used *datasets* and *evaluation metrics*. The survey highlights how multimodal approaches extend traditional text summarization frameworks to handle richer information by aligning features from different modalities and addresses the challenges in multimodal learning and representation. The work also outlines current research trends and open problems to guide future investigations.

Drawbacks:

The paper largely provides a high-level overview without proposing a specific model or quantitative comparisons between techniques. It identifies gaps in standard benchmarking datasets and notes that many multimodal methods have inconsistent evaluation criteria, which makes direct performance comparisons difficult

Min Jung Lee, Dayoung Gong & Minsu Cho (2025) [6] – Video Summarization with Large Language Models

Methodology:

This paper proposes a new framework called LLMVS (LLM-based Video Summarization) that leverages the semantic understanding capabilities of large language models (LLMs) to improve video summarization quality. The approach first

generates textual captions for each video frame using a multimodal LLM, transforming visual information into a text-based representation. Based on these captions, an LLM assesses the importance of each frame within its local context and then refines those importance scores using global context attention over the entire caption sequence. This hybrid strategy enables the model to preserve both detailed scene information and overall narrative flow, resulting in more coherent and semantically meaningful summaries compared to traditional frame-based methods. Experimental results on standard benchmarks demonstrate that the LLM-centric method outperforms existing approaches by better capturing semantic relevance.

Drawbacks:

The reliance on large pretrained LLMs increases computational overhead and inference time, particularly for long videos. The method also depends on the quality of initial caption generation, meaning that poor captions can negatively affect overall summary accuracy. Additionally, the approach may perform inconsistently when high-quality multimodal captioning models are unavailable.

Toqa Alaa et al. (2024) [7] – Video Summarization Techniques: A Comprehensive Review

Methodology:

This paper provides a detailed survey of state-of-the-art video summarization techniques, covering both *extractive* and *abstractive* approaches used to condense video data while retaining essential information. The authors review a wide range of methods, including keyframe and keyshot extraction, clustering, attention mechanisms, generative adversarial networks, and multimodal learning strategies. The survey also discusses commonly used benchmark datasets and highlights the differences between traditional and deep learning-oriented approaches. By classifying video summarization methods and examining their strengths and weaknesses, the paper gives insight into present research trends and practical challenges in efficiently generating meaningful summaries from long video content.

Drawbacks:

Because it is a survey, the paper does not propose a new model or empirical results; rather, it synthesizes existing work, so it lacks direct performance evaluations. It also highlights that current evaluation metrics and dataset variety are limited, making fair comparison of summarization techniques difficult.

Deepali Vora et al. (2025) [8] – AI-Driven Video Summarization for Optimizing Content Retrieval and Management

Methodology:

This study focuses on developing an AI-driven video summarization system that enhances content retrieval and management by combining deep learning techniques such as CNNs (Convolutional Neural Networks), LSTM (Long Short-Term Memory), and ResNet50 feature extraction to summarize video content efficiently. The approach extracts frame-level and temporal features to identify representative segments and optimize search within large multimedia archives. By integrating multiple deep learning models, the system improves semantic understanding of video scenes and produces summaries that highlight the most informative sections. The research emphasizes practical application for managing extensive video collections where rapid browsing and retrieval of key content is crucial.

Drawbacks:

The reliance on complex deep learning architectures increases computational overhead and may require significant hardware resources, especially for long videos. Additionally, performance can vary depending on the quality and diversity of training data, and the model does not explicitly address multimodal features such as audio or text transcripts.

Yogendra Singh et al. (2023) [9] – YouTube Video Summarizer using NLP: A Review

Methodology:

This review paper examines the use of Natural Language Processing (NLP) techniques specifically for summarizing YouTube video content. It begins by explaining the need for automated summarization due to the explosive growth of online video and the difficulty users face in accessing core information efficiently. The study systematically investigates how textual data associated with videos — including transcripts, captions, metadata, and even comments — can be leveraged to extract meaningful summaries. Multiple NLP techniques such as traditional text summarization methods, sentiment analysis, topic modeling, and deep learning-based models are discussed, highlighting how each approach contributes to capturing essential information from video transcripts. The paper also outlines common evaluation metrics and datasets used in this emerging area of research.

Drawbacks:

Since this is a survey, the paper does not propose a new algorithm or perform quantitative evaluations itself, which limits empirical comparisons. Moreover, many research works discussed face challenges due to inconsistent evaluation standards and the variability in transcript quality from YouTube videos, making performance comparisons difficult

Elmin Marevac et al. (2025) [10] – Multimodal Video Summarization Using Machine Learning

Methodology:

This study presents a multimodal video summarization framework that combines audio, visual, and fused features to classify video segments as *informative* or *non-informative*, aiming to generate meaningful summaries from user-generated content. Extensive feature extraction is performed using audio processing libraries (e.g., pyAudioAnalysis) and visual descriptors (e.g., color histograms, optical flow, object detection) to produce enriched representations for each video segment. Six supervised classifiers—including Random Forest, Logistic Regression, KNN, and XGBoost—are evaluated on a large multimodal dataset spanning around 60 hours of diverse video categories. Temporal coherence is enhanced using post-processing techniques like median filtering, and the best classifier results are deployed as part of a practical web service. Experimental results demonstrate that multimodal fusion significantly improves classification performance and highlights the potential of combining ensemble learning with feature fusion for real-world video summarization applications.

Drawbacks:

Although combining multiple modalities enhances performance, this methodology still relies on handcrafted feature extraction, which may not scale well to high-resolution or very long videos. The supervised learning approach also depends on extensive manual annotation, which can be resource-intensive. Additionally, while the system shows good classifier performance, it does not integrate deep learning architectures like transformers or large language models, which may limit semantic understanding compared to newer AI-based summarization techniques.

Libin Lan et al. (2025) [11] – FullTransNet: Full Transformer with Local-Global Attention for Video Summarization

Methodology:

This paper proposes FullTransNet, a novel transformer-based architecture for video summarization that uses a full encoder-decoder transformer, rather than the common encoder-only setups. The approach formulates video summarization as a sequence-to-sequence learning task where both video understanding and summary generation are performed by the same architecture. To handle long video sequences efficiently, the model replaces standard full attention with a local-global sparse attention mechanism at the encoder side, enabling it to capture long-range dependencies while significantly reducing computational costs. Experiments on benchmark datasets such as *SumMe* and *TVSum* demonstrate that FullTransNet achieves higher F-scores than traditional methods while requiring relatively lower compute and memory resources, validating its effectiveness for summarization.

Drawbacks:

The reliance on transformer architectures still involves relatively high computational overhead compared to traditional shallow models. Additionally, although sparse attention reduces complexity, it may still face scalability challenges with very long videos or high frame rates. The method also requires supervised training with annotated summaries, which can limit applicability when labeled data are scarce.

Turan G. Altundogan et al. (2025) [12] – QUBVIS: Query-Based Multimodal Summarization System Using CLIP-Based Transformer and Vision-Language Models

Methodology:

This study proposes QUBVIS, a query-based multimodal video summarization system that enables users to generate summaries based on *natural language queries* related to the content they want emphasized. The system integrates a CLIP-based transformer architecture along with vision-language models to extract semantic features from both visual frames and associated text or transcript data. By incorporating user queries into the summarization pipeline, QUBVIS can focus on segments that are most relevant to the user's specified intent, producing summaries that reflect query semantics rather than general importance. Experimental results demonstrate that query-conditioned summarization offers improved relevance and personalization compared with generic summarization baselines, especially for long and diverse content where generic

summaries may overlook query-specific details.

Drawbacks:

The system's reliance on natural language queries can degrade performance if the query is poorly formulated or ambiguous, which may lead to less accurate content relevance. Additionally, integrating CLIP and vision-language models increases computational overhead relative to simpler extractive methods, potentially impacting scalability for real-time or large-scale deployment. Finally, while tailored summaries improve personalization, they may neglect broader contextual information outside the query scope.

Yubo Zhu et al. (2023) [13] – Topic-Aware Video Summarization Using Multimodal Transformer

Methodology:

This study introduces the concept of topic-aware video summarization, in which multiple video summaries are generated from the same video based on different topics of interest rather than a single general summary. To support research in this area, the authors build a new benchmark dataset called TopicSum, containing annotated topic labels and frame-level importance scores for diverse movie clips. The proposed method uses a multimodal transformer to simultaneously predict topic labels and generate topic-specific summary segments by adaptively fusing multimodal features (visual, audio, and textual) extracted from the video. By capturing both semantic topic information and multimodal content interactions, the model produces summaries that can cater to varying user interests and preferences. Experimental results demonstrate that the approach effectively generates topic-related summaries that better reflect user-specific interests compared to traditional summarization techniques.

Drawbacks:

The requirement for topic annotations and frame-level importance labels increases dataset creation complexity and limits applicability where such rich annotations are not available. The reliance on transformer-based models also adds computational overhead during training and inference. In addition, generating multiple topic-specific summaries may increase processing time compared to single-summary methods.

Guangli Wu, Miaomiao Wang & Ning Ma (2025) [14] – Multimodal Video Summarization Based on Graph Contrastive Learning and Fine-Grained Graph Interaction

Methodology:

This paper proposes a novel multimodal video summarization model that leverages graph contrastive learning and fine-grained graph interaction to improve semantic understanding and reduce noise during multimodal fusion. The video and text data are each represented as spatial–temporal graphs, where nodes correspond to visual frames and textual segments. A spatial–temporal graph network jointly models dependencies within and across modalities, while graph contrastive learning optimizes node representations by suppressing redundancy and noise. In addition, multiple semantic perspectives are used in cross-modal graph matching to capture fine-grained interactions between video and textual features. Experiments on benchmark datasets such as SumMe and TVSum demonstrate that the proposed method outperforms existing state-of-the-art techniques.

Drawbacks:

Although this approach improves cross-modal interaction and reduces redundant representations, it introduces additional computational complexity due to graph construction and contrastive training. Optimizing and aligning graph node features may require significant resources, making it less efficient for real-time video summarization tasks. Moreover, accurately constructing multimodal graphs can be challenging if transcripts or text descriptions are noisy or incomplete.

Jungin Park, Jiyoung Lee & Kwanghoon Sohn (2025) [15] – Language-Guided Recursive Spatiotemporal Graph Modeling for Video Summarization

Methodology:

This paper proposes VideoGraph, a novel video summarization framework treating video summarization as a language-guided spatiotemporal graph modeling problem. Instead of relying solely on frame similarity, the authors construct recursive spatiotemporal graphs where video objects and frames are modeled as nodes, and edges capture semantic relationships in space and time. Language guidance derived from the video's narrative is incorporated into node features to enrich semantic context, allowing the model to better understand which frames are crucial. A recursive strategy refines the initial graph and classifies nodes as keyframes. Experimental

results demonstrate that VideoGraph achieves state-of-the-art performance on benchmarks for both generic and query-focused summarization scenarios.

Drawbacks:

The model's graph construction and refinement process increases computational complexity compared to standard CNN- or transformer-based methods, which may impact scalability for long videos. Additionally, the reliance on language guidance means the approach performance depends on high-quality linguistic input, limiting effectiveness when text narratives are noisy or absent

Mario Barbara & Alaa Maalouf (2025) [16] – Prompts to Summaries: Zero-Shot Language-Guided Video Summarization

Methodology:

This paper proposes a novel zero-shot video summarization framework that enables natural language-guided summarization without requiring any labeled training data. The system, called *Prompts-to-Summaries*, first segments raw video into coherent scenes and generates rich scene descriptions using a video-language model prompting scheme that scales efficiently to long videos. A large language model (LLM) then judges the relative importance of each scene based on carefully designed prompts, and two new metrics—*consistency* (temporal coherence) and *uniqueness* (novelty)—are applied to derive detailed frame importance scores. These scores are propagated to summarize content into a concise multimodal video summary. Experimental results show that this approach, despite using no supervised training data, surpasses many unsupervised and even supervised methods on standard benchmarks like SumMe and TVSum, and performs competitively on query-focused summarization tasks.

Drawbacks:

Because the method heavily relies on off-the-shelf video-language models and LLM prompting, its effectiveness depends on the quality of pretrained models and prompt design, which may vary across domains. Additionally, while zero-shot approaches reduce the need for labeled data, they may not achieve the same semantic precision as finely-tuned supervised frameworks when domain-specific nuances are important. Computational costs can also be high due to repeated LLM inference and scene description generation.

Shuo Wang & Jihao Zhang (2025) [17] – MF2Summ: Multimodal Fusion for Video Summarization with Temporal Alignment

Methodology:

This paper introduces MF2Summ, a novel approach to video summarization that focuses on multimodal feature fusion and temporal alignment to better capture both visual and audio information over time. The method includes five main stages: visual feature extraction using pre-trained models, auditory feature extraction, cross-modal attention interaction, fusion of multimodal features, and prediction of segment importance. A key component of the approach is the alignment-guided self-attention transformer, which models inter-modal relationships and temporal correspondences across video segments. Important shots are then selected using techniques like Non-Maximum Suppression (NMS) and Kernel Temporal Segmentation (KTS). Evaluation on standard benchmarks such as SumMe and TVSum shows that MF2Summ achieves competitive performance with improved F1-scores compared to several existing models.

Drawbacks:

Despite its strong performance, MF2Summ's reliance on complex transformer architectures and multi-stage processing increases computational overhead, potentially limiting scalability for real-time applications or long-duration videos. Additionally, its performance depends on the quality of pre-trained feature extractors, which might not generalize well across all video types or domains

Sicheng Liu et al. (2024) [18] – SITransformer: Shared Information-Guided Transformer for Extreme Multimodal Summarization

Methodology:

This paper introduces SITransformer, a transformer-based architecture designed for extreme multimodal summarization, where the goal is to generate highly concise yet informative summaries from complex multimodal inputs such as video, text, and audio simultaneously. Unlike traditional multimodal summarization methods that treat each modality separately, SITransformer extracts shared semantic information across all modalities using a *shared information extractor* followed by a *cross-modal interaction module*. This design enables the model to focus on the most salient content shared across the modalities, which improves the overall quality of short summaries. The system uses differentiable top-k selection and gating mechanisms to filter out

irrelevant features before generating the final summary through cross-modal attention. Experimental results show that SITransformer significantly enhances multimodal summary performance on benchmark datasets by improving coherence and relevance of the generated summaries.

Drawbacks:

While the shared information guidance improves summarization quality, the model's reliance on multiple modality processing increases computational complexity compared with simpler models. The need to align and fuse information from diverse modalities also makes training more resource-intensive, and cooperation between modalities may be sensitive to inconsistent or noisy input from any single modality. Additionally, the approach may require careful design of gating and filtering mechanisms to ensure that only truly shared information is preserved without excluding important unique modality details.

Galann Pennec et al (2025) [19] – Integrating Video and Text: A Balanced Approach to Multimodal Summary Generation and Evaluation

Methodology:

This paper proposes a zero-shot multimodal video-to-text summarization approach that builds a unified screenplay–like representation of long-form video content by integrating visual scenes, dialogue, and character information. Instead of relying on supervised training or extensive labeled data, the method uses *vision-language models (VLMs)* to automatically generate descriptive captions and frame metadata from video and audio modalities. These are combined into a structured "screenplay" format, which an LLM then summarizes into a concise textual narrative. The paper also introduces MFactSum, a novel evaluation metric designed to assess *both text and visual factual coverage* in multimodal summaries — addressing limitations of traditional text-only scores like ROUGE or METEOR. Experiments on the SummScreen3D dataset demonstrate that this approach can generate summaries with more relevant visual information while using significantly less input video content.

Drawbacks:

Since the method relies on off-the-shelf pretrained vision-language and large language models, its performance depends heavily on the quality of those underlying models and prompt design. Zero-shot techniques may struggle with domain-specific semantics when compared with supervised learning approaches optimized for specific

datasets. The screenplay representation step also introduces complexity, which may require careful engineering for diverse or noisy video content.

Jingyang Lin et al. (2023) [20] – VideoXum: Cross-modal Visual and Textual Summarization of Videos

Methodology:

This paper introduces VideoXum, a novel framework for cross-modal video summarization that jointly generates both abridged video clips and aligned textual summaries from long videos. The authors identify that traditional methods usually handle video and text summarization separately, resulting in inconsistent semantic coherence between modalities. To address this, they establish a large-scale human-annotated benchmark dataset based on ActivityNet Captions containing over 14 K videos with paired visual and textual summary annotations. They propose an end-to-end model called VTSUM-BLIP that uses a vision-language pretrained (VLP) encoder-decoder architecture to jointly optimize both video and text summary generation, while a variant of CLIPScore termed VT-CLIPScore evaluates semantic alignment between generated summaries. Experimental results demonstrate that the cross-modal model achieves promising performance on VideoXum and other video summarization datasets, establishing strong baselines for future research in joint video–text summarization.

Drawbacks:

Because the joint generation framework relies heavily on a large paired dataset, it may be less effective in domains where such aligned data are scarce or unavailable. The requirement for large human-annotated summary pairs also increases annotation cost and time. In addition, the model's complexity comes with higher computational demands for training and inference compared to unimodal summarization approaches.

CHAPTER 3

METHODOLOGY

The methodology of the Smart AI Video Summarizer with Text, PDF, and Highlights system defines the structured approach followed to process different types of input data and generate concise summaries and highlight segments. The proposed system adopts a hybrid multimodal processing methodology that combines local and cloud-based Artificial Intelligence models to achieve accurate, efficient, and scalable results.

The system is designed as a multi-stage pipeline that includes data acquisition, preprocessing, extractive analysis, abstractive summarization, and highlight generation. This approach ensures that only the most relevant information is processed by the AI models, improving performance and summary quality.

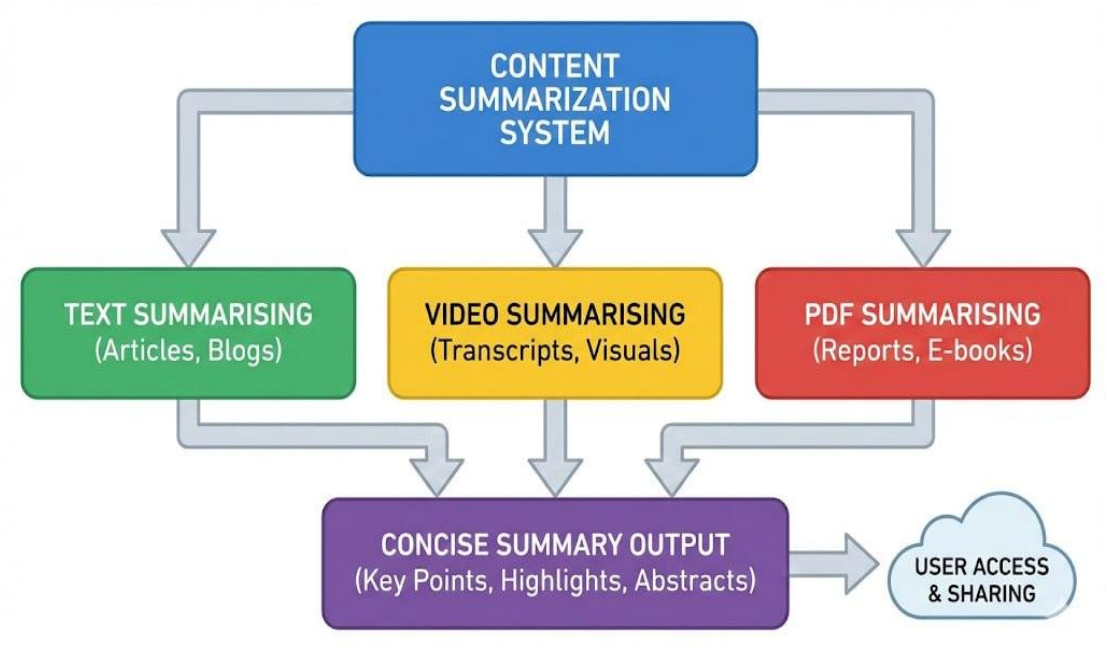

Figure 3.1 Functional Overview of the Smart AI Summarization System

Figure 3.1 illustrates the functional workflow of the proposed content summarization system. The system accepts multiple input formats, including text, video, and PDF documents, and processes them through their respective summarization modules. Each module extracts key information from the input content and generates a concise summary consisting of important points, highlights, or abstracts. The final summarized output is then provided to the user for access and sharing, demonstrating the system's ability to handle multimodal inputs and produce unified summarized results.

3.1 Overall Methodology

3.1.1 Data Acquisition and Input Processing

In this stage, the system collects input data from the user.

- Text input is directly received from the user interface.
- PDF files are uploaded and processed using document parsing techniques to extract textual content.
- YouTube video URLs are processed using yt-dlp to retrieve video metadata and transcripts.
- Highlight extraction requests utilize the transcript data as the primary source.

All collected data is converted into structured textual format to enable further processing.

3.1.2 Preprocessing and Extractive Analysis

After data acquisition, the content undergoes preprocessing to remove noise and improve relevance.

- Tokenization is performed to break text into sentences and words.
- Stopwords and unnecessary characters are removed.
- Text is divided into manageable chunks to preserve context.
- Sentence scoring is applied using statistical measures such as TF-IDF, keyword density, and positional weighting.

The highest-scoring sentences are selected to form a "content skeleton," which represents the most informative portion of the input.

3.1.3 Abstractive Summary Generation

The refined content is passed to the generative AI model for producing a coherent summary.

- For text and PDF inputs, the local Gemma 3 (12B) model is used through Ollama.
- For video inputs, the cloud-based Gemma 3 (27B IT) model is used for multimodal reasoning.

The AI model rewrites extracted content into a concise, readable summary while preserving contextual meaning.

3.1.4 Highlight Extraction Method

The highlights module identifies important segments within video content.

- Key sentences or quotes are detected from the transcript.
- A timestamp mapping algorithm locates the exact occurrence of these segments.
- Overlapping timestamps are merged to avoid redundancy.
- Selected segments are processed using FFmpeg to generate and combine highlight clips.

This process allows users to quickly access the most important moments of the video.

3.1.5 Output Generation and Storage

After processing is complete:

- The generated summary is returned to the frontend.
- Highlight timestamps and exported clips are made available for download.
- Results are stored in the database for future reference.

This ensures that users can access previous outputs without repeating the processing steps.

3.2 Smart AI Video Summarizer System Architecture

The Smart AI Video Summarizer with Text, PDF, and Highlights system is designed using a Client–Server architecture combined with a hybrid Artificial Intelligence processing layer. This architecture enables efficient communication between the user interface, backend services, AI models, and media processing tools. The system is structured to process different types of inputs such as YouTube video URLs, text content, and PDF documents while generating summaries and highlight outputs.

The architecture is divided into three major layers: Frontend Layer, Backend Layer, and AI Intelligence Layer, each responsible for specific functionalities that collectively ensure smooth system operation.

3.2.1 Frontend Layer

The frontend is developed using React.js and provides an interactive interface through which users can access all system functionalities. It acts as the entry point where users submit their inputs and view the generated results.

The main functions of the frontend include:

- Accepting inputs such as YouTube video URLs, text content, and PDF uploads
- Allowing users to select summary length and output format
- Displaying generated summaries and highlight timestamps
- Providing options to export generated highlight clips
- Handling user authentication through Google Sign-In

The frontend communicates with the backend through API requests and displays results dynamically.

3.2.2 Backend Layer

The backend is implemented using FastAPI, which acts as the central processing unit of the system. It manages data flow, coordinates processing tasks, and connects the frontend with AI services and databases.

The backend performs several key operations:

- Receiving and validating user inputs
- Extracting video metadata and transcripts using yt-dlp
- Processing uploaded text and PDF documents
- Applying preprocessing techniques such as tokenization and filtering
- Routing tasks to the appropriate AI model
- Managing highlight clip generation using FFmpeg
- Storing generated summaries and history in the SQLite database

By centralizing these operations, the backend ensures efficient and organized execution of all system functions.

3.2.3 AI Intelligence Layer

The AI Intelligence Layer provides the core functionality of the system by generating summaries and identifying highlights. The system uses a hybrid AI approach to optimize performance and resource utilization.

Local AI Processing (Gemma 3 – 12B):

- Used for text and PDF summarization
- Used for highlight detection
- Provides low latency and improved privacy

Cloud AI Processing (Gemma 3 – 27B IT via Google Gemini API):

- Used for video content understanding
- Handles multimodal reasoning
- Processes large context data

The backend dynamically selects between local and cloud models depending on the input type.

3.2.4 Media Processing Components

To support video analysis and export functionality, the system integrates several multimedia processing tools.

- yt-dlp is used to retrieve video metadata and transcripts from YouTube.
- OpenCV is used for frame extraction when required.
- FFmpeg is used to generate and merge highlight clips into a single video.

These tools enable efficient handling of multimedia data within the system.

3.2.5 Data Flow Overview

The overall workflow begins when the user submits input through the frontend interface. The request is sent to the FastAPI backend, where data extraction and preprocessing are performed. The backend then selects the appropriate AI model to generate summaries or highlights. Finally, the processed output is returned to the frontend and stored in the database for future access.

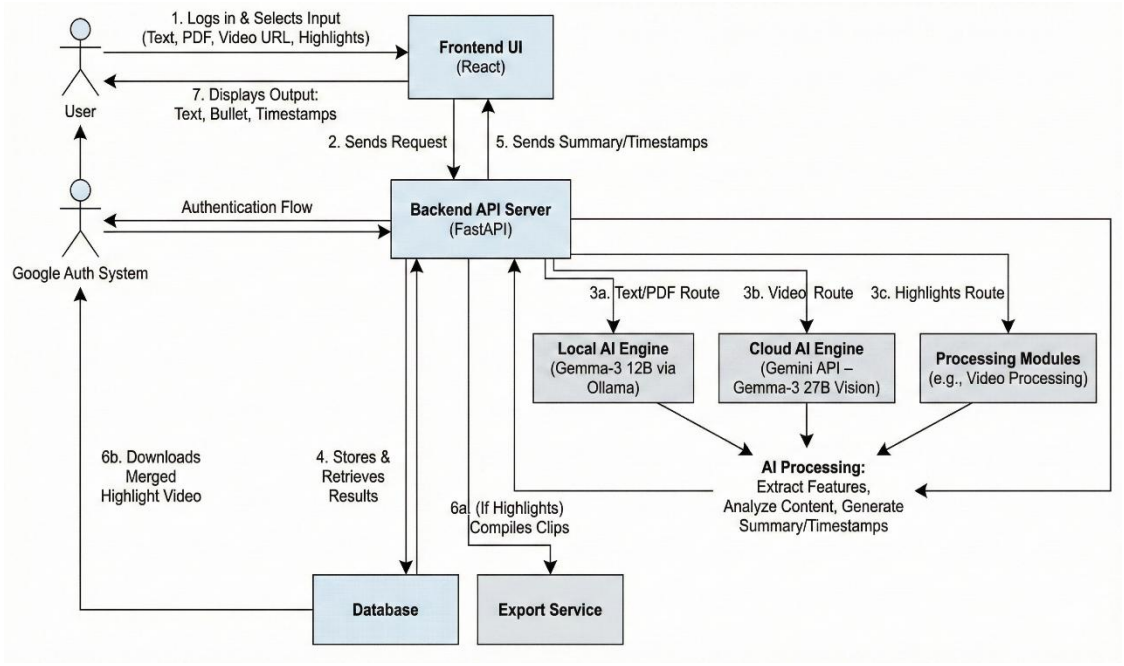

Figure 3.2 Smart AI Multimodal Summarization System Architecture

Figure 3.2 illustrates the overall architecture of the proposed system and the flow of data between its components. The user interacts with the frontend interface and submits input such as text, PDF, video URL, or highlights, which is processed through the backend server. Based on input type, the system routes requests to either the local AI model, cloud AI model, or processing modules for analysis. The generated summaries or timestamps are stored in the database and returned to the user interface for display, while highlight results can be compiled and downloaded as a merged video file.

3.3 Algorithm Design

The Algorithm Design of the Smart AI Video Summarizer with Text, PDF, and Highlights describes the logical sequence of operations used to process different types of input data and generate summaries and highlight segments. The system employs a hybrid approach that combines extractive processing and abstractive Artificial Intelligence–based summarization.

The core algorithm used in the system is referred to as the User-Aware Multimodal Summarization Algorithm (UAMSA), which dynamically selects processing steps based on the type of input provided by the user.

Overview of the Algorithm

The algorithm is designed to efficiently handle multimodal inputs such as YouTube videos, textual data, and PDF documents. It processes the input through stages including data acquisition, preprocessing, content filtering, AI-based synthesis, and output generation. This

ensures that only relevant information is used for final summary generation.

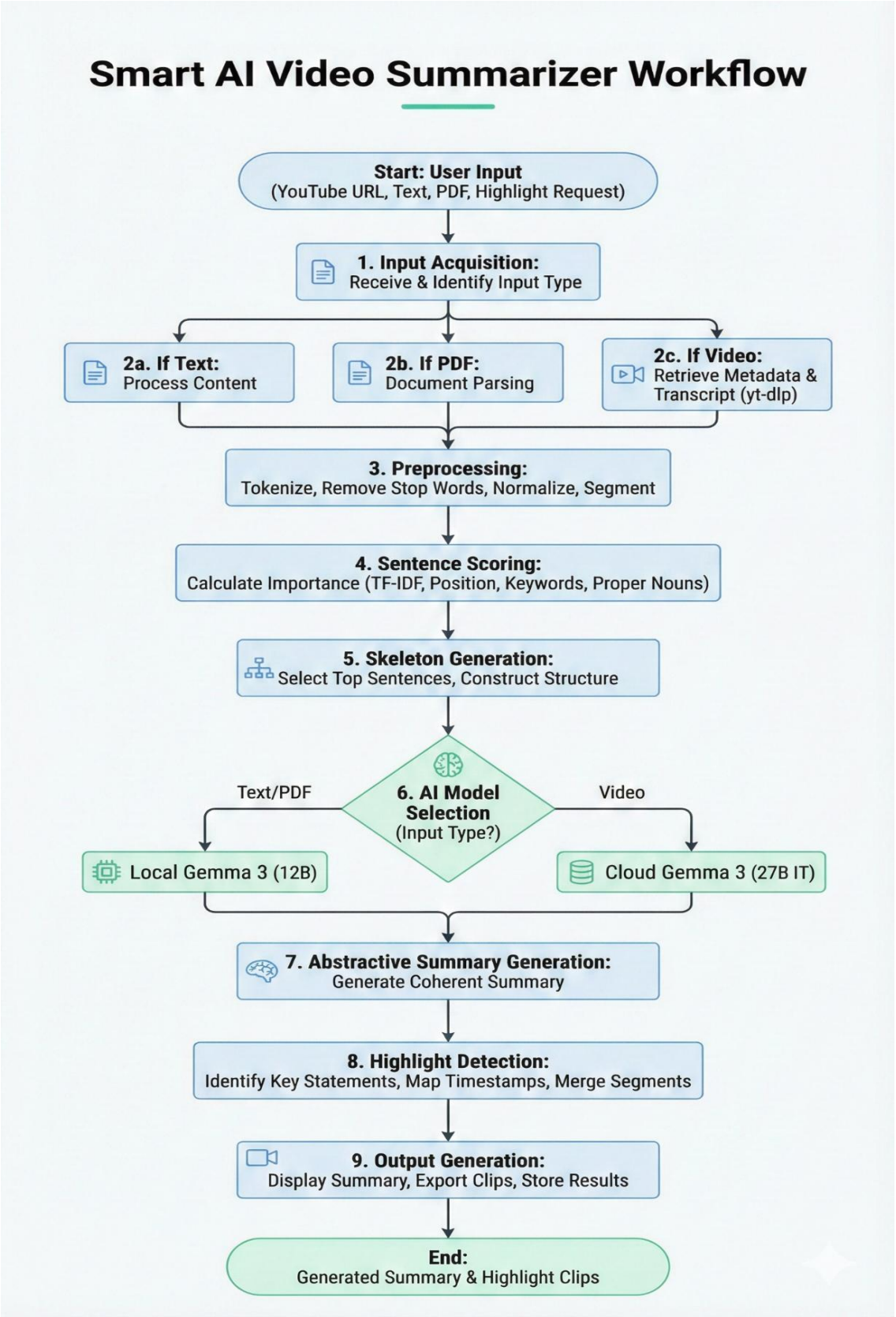

Figure 3.3 Architectural Workflow of the Smart AI Video Summarizer

Figure 3.3 illustrates the end-to-end data processing pipeline for the Smart AI Video Summarizer. The system utilizes a multi-modal ingestion layer (Step 1-2) capable of handling raw text, PDF documents, and YouTube URLs. The pipeline employs a hybrid extractive-abstractive approach: it first performs extractive preprocessing using TF-IDF and sentence scoring (Steps 3-5) to establish a content "skeleton," followed by abstractive generation (Step 7). The architecture features a dynamic AI Model Selection logic (Step 6) that routes data based on computational requirements—utilizing a Local Gemma 3 (12B) model for standard text/PDF tasks and a Cloud-based Gemma 3 (27B IT) model for complex video data processing. The final stages involve temporal mapping for highlight detection and the concurrent generation of text summaries and timestamped video clips.

3.3.1 Step-by-Step Algorithm Procedure

Input: YouTubeURL / Text / PDF / Highlight Request

Output: Generated Summary and Highlight Clips

Step 1: Input Acquisition

- Receive input from the user interface. The input type may be Video, Text, PDF, or Highlights.

Step 2:

- If Text → Directly process the provided content.
- If PDF → Extract text using document parsing.
- If Video → Retrieve metadata and transcript using yt-dlp.

Step 3: Preprocessing

- Tokenize the extracted text.
- Remove stop words and unnecessary symbols.
- Normalize and segment the content into sentences.

Step 4: Sentence Scoring (Extractive Phase)

- Calculate importance of each sentence using statistical features:
    - TF-IDF weighting

- Sentence position
- Keyword density
- Proper noun frequency

Step 5: Skeleton Generation

- Select top-ranked sentences.
- Construct an intermediate structured representation of important content.

Step 6: AI Model Selection

- If input is Text or PDF → Use local Gemma 3 (12B).
- If input is Video → Use cloud Gemma 3 (27B IT).

Step 7: Abstractive Summary Generation

- Provide extracted content to AI model.
- Generate coherent and concise summary.

Step 8: Highlight Detection

- Identify key statements.
- Map statements to timestamps.
- Merge overlapping segments.

Step 9: Output Generation

- Display summary.
- Export highlight clips if requested.
- Store results in database.

Mathematical Sentence Scoring Formula

During the extractive phase, sentence importance is calculated using a weighted scoring model:

$$Score(S) = \alpha(TF\text{-}IDF) + \beta(Position) + \gamma(ProperNounWeight) + \delta(KeywordDensity)$$

Where:

- $\alpha, \beta, \gamma, \delta$ are adjustable weighting parameters.
- TF-IDF measures term importance.
- Position prioritizes introductory and concluding sentences.
- Proper noun weight emphasizes entities.
- Keyword density reflects contextual relevance.

This scoring mechanism ensures selection of the most informative content before AI summarization.

3.3.2 Advantages of the Proposed Algorithm

- Combines extractive and abstractive summarization.
- Supports multimodal content processing.
- Improves summary coherence and readability.
- Reduces noise before AI inference.
- Enables accurate highlight detection.
- Optimizes resource usage through hybrid AI selection.

CHAPTER 4

RESULTS AND DISCUSSION

The Smart AI Video Summarizer with Text, PDF, and Highlights system was tested using multiple input types including YouTube videos, textual content, and PDF documents to evaluate its performance, accuracy, and usability. The results demonstrate that the system successfully generates concise summaries and identifies key moments efficiently.

The system produced effective summaries across all supported input formats. For text and PDF inputs, the locally deployed Gemma 3 (12B) model generated coherent and contextually relevant summaries with reduced latency. For video inputs, the cloud-based Gemma 3 (27B IT) model successfully processed YouTube transcripts and generated meaningful summaries that retained the original content's key ideas.

The highlights module accurately identified important timestamps from video transcripts and generated corresponding video segments. Overlapping segments were merged correctly, and the exported highlight clips were produced without loss of synchronization.

The user interface provided real-time output display and allowed users to select summary length and format. Generated results were stored in the database, enabling users to access previously processed summaries.

Overall, the system achieved the following outcomes:

- Successful summarization of long textual and document content
- Accurate extraction of video highlights with timestamp mapping
- Efficient processing using hybrid AI architecture
- Reduced processing time for text and PDF inputs due to local inference
- Smooth export of combined highlight videos

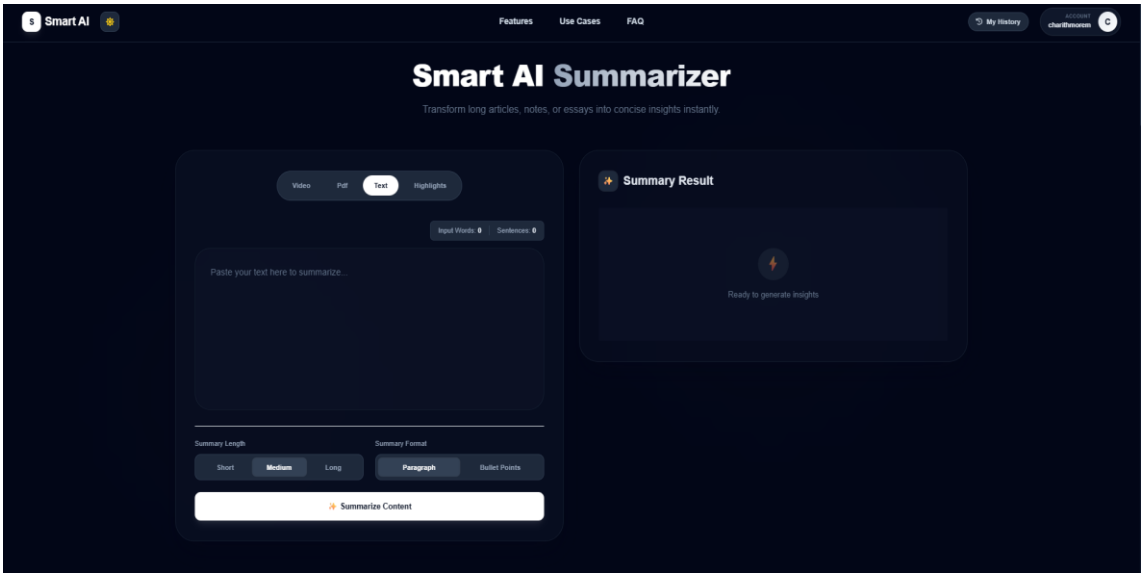

Figure 4.1 Text Input Module Interface for Content Summarization

Figure 4.1 illustrates the Text Summarization Interface of the Smart AI Summarizer web application. The module allows users to input textual content directly into the provided text area for processing. Users can select the desired summary length (Short, Medium, or Long) and choose the output format as either paragraph or bullet points. Upon clicking the "Summarize Content" button, the system processes the input using the local AI model and generates a concise summary, which is displayed in the Summary Result panel on the right side of the interface. The module also displays input statistics such as word count and sentence count to provide additional feedback to the user.

Table 4.1 Features of Text Summarization Module

Component	Description
Text Input Area	Allows users to paste or type the content that needs to be summarized.
Input Statistics	Displays the number of words and sentences in the entered text.
Summary Length Selection	Enables users to choose the desired summary size (Short, Medium, Long).
Summary Format Option	Allows selection between Paragraph format or Bullet Points output.
Summarize Content Button	Initiates the summarization process using the local AI model.
Summary Result Panel	Displays the generated summarized output after processing.
Navigation Tabs	Allows switching between Video, PDF, Text, and Highlights modules.

Table 4.1 describes the key components and functionalities of the Text Summarization Module in the Smart AI Summarizer web application. It outlines the purpose of each interface element, explaining how users interact with the text input area, configure summarization preferences, and generate summarized output. The table highlights how the module supports customizable summary length and format while providing real-time feedback and a dedicated output panel for displaying results.

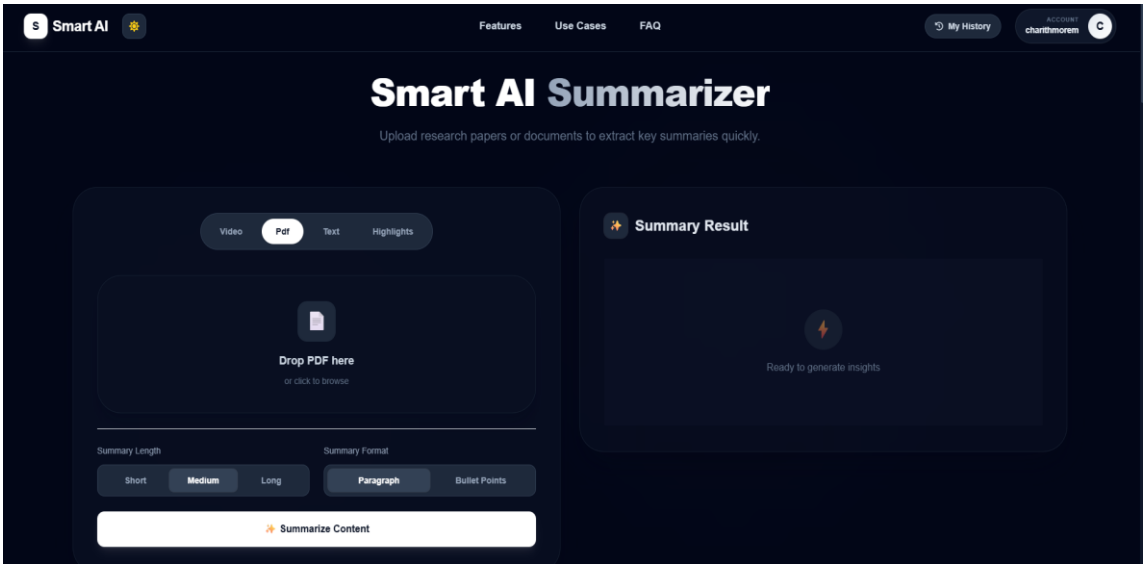

Figure 4.2 PDF Upload and Summarization Module Interface

Figure 4.2 shows the PDF Summarization Module of the Smart AI Summarizer web application. The interface allows users to upload research papers or documents in PDF format using a drag-and-drop area or file browser. After uploading the PDF, users can configure the summary length (Short, Medium, or Long) and select the output format as paragraph or bullet points. Once the summarization process is initiated, the system extracts textual content from the PDF, processes it using the local AI model, and displays the generated summary in the Summary Result panel. This module enables efficient extraction of key insights from lengthy documents.

Table 4.2 Description: Video Summarization Output Features

Feature	Description
YouTube URL Processing	Accepts YouTube video links as input.
Transcript-Based Analysis	Uses extracted transcripts for summarization.
Summary Output Panel	Displays generated video summary in text format.
Format Selection	Supports paragraph and bullet-point summaries.
Cloud AI Processing	Uses multimodal AI for deeper video understanding.
Output Storage	Stores generated summaries for future reference.

Table 4.2 explains the output features of the Video Summarization module, detailing

how YouTube video content is analyzed using transcript-based processing and multimodal AI models. It highlights the generation of readable summaries, format selection options, and the storage of results. The table illustrates the system's ability to efficiently convert long videos into concise textual summaries.

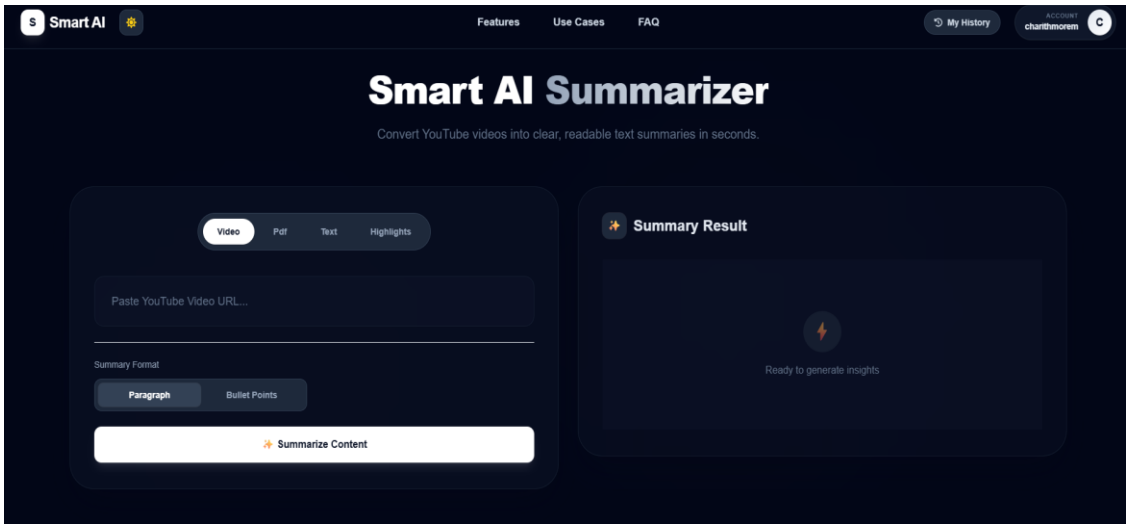

Figure 4.3 YouTube Video Summarization Module Interface

Figure 4.3 illustrates the Video Summarization Module of the Smart AI Summarizer web application. The interface allows users to input a YouTube video URL for automated summarization. Once the URL is provided, the system retrieves video metadata and transcript information for processing. Users can select the preferred summary format, such as paragraph or bullet points, before initiating the summarization process. The generated summary is displayed in the Summary Result panel on the right side of the interface. This module leverages cloud-based multimodal AI processing to convert long YouTube videos into concise, readable summaries efficiently.

Table 4.3 Highlights Output and Download Features

Feature	Description
Key Moments List	Displays extracted important timestamps and descriptions.
Highlight Reel Player	Allows preview of summarized video clips.
Auto-Skip Option	Automatically jumps between key moments.
Clip Merging	Combines multiple highlights into a single video.
Download Option	Enables downloading the compiled highlight reel.
Resolution Selection	Allows selection of video quality before export.

Table 4.3 outlines the features of the Highlights Output module, focusing on the extraction of key moments and timestamps from video content. It describes the system's capability to generate a highlights reel, preview important segments, merge clips, and download the compiled video. The table highlights how this module enhances user efficiency by enabling quick access to significant video segments.

Table 4.4 User History (My Activity) Module Features

Feature	Description
Saved Summaries	Stores previously generated text, PDF, and video summaries.
Highlight Records	Maintains extracted highlight reels and timestamps.
Search History	Allows users to search through saved records.
Filter Options	Filters results by input type (Text, PDF, Video, Highlights).
View & Reuse	Enables reopening and reusing past summaries.
History Management	Allows deletion or organization of stored outputs.

Table 4.4 describes the functionalities of the User History (My Activity) module, which allows users to manage previously generated summaries and highlights. It explains features such as search, filtering, and reuse of saved content. The table demonstrates how the module supports efficient organization, retrieval, and long-term usability of generated insights.

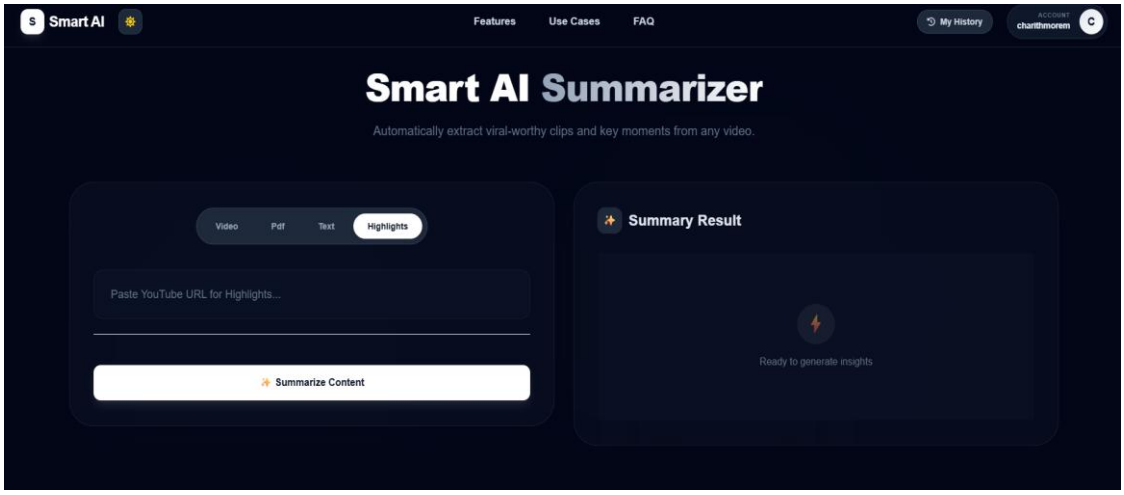

Figure 4.4 Video Highlights Extraction Module Interface

Figure 4.4 represents the Highlights Extraction Module of the Smart AI Summarizer web application. The module allows users to input a YouTube video URL for automatically identifying key moments and important segments within the video. Upon submission, the system analyzes the video transcript to detect significant statements and maps them to their corresponding timestamps. The extracted highlights are processed and prepared for further actions such as previewing key moments or exporting selected clips. The Summary Result panel displays the generated insights, enabling users to efficiently access the most relevant portions of long video content.

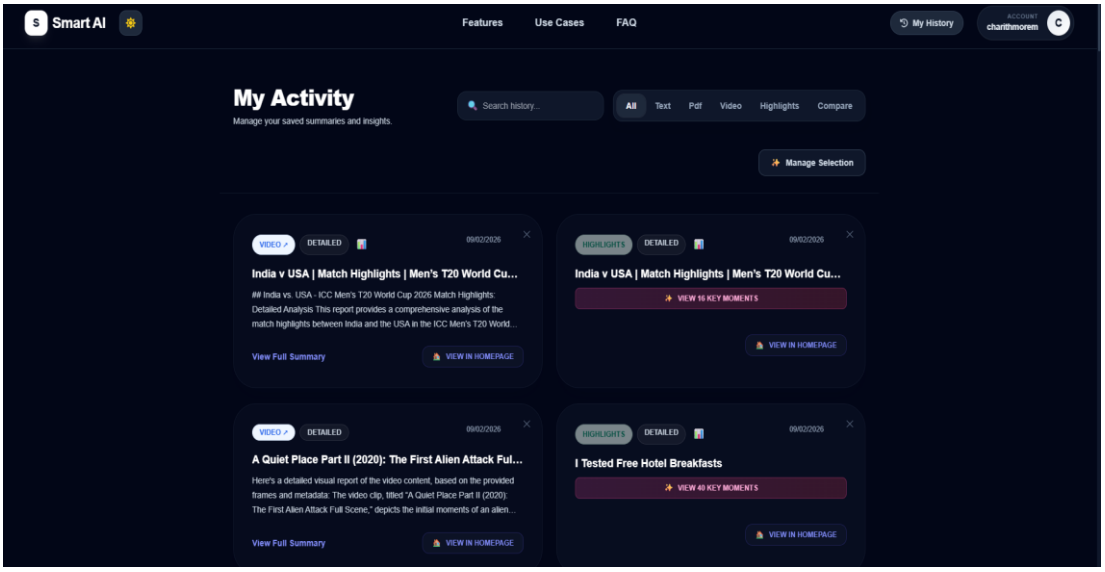

Figure 4.5 User History and Summary Management Interface

Figure 4.5 illustrates the My Activity (History) Module of the Smart AI Summarizer web application. The interface allows users to view and manage their previously generated summaries and highlights. Users can search their history, filter records based on content type such as text, PDF, video, or highlights, and access detailed summaries generated earlier. Each entry displays relevant metadata including the content type, title, and date of creation. The module also provides options to revisit summaries, view extracted key moments, and manage saved outputs, enabling efficient tracking and reuse of previously analyzed content.

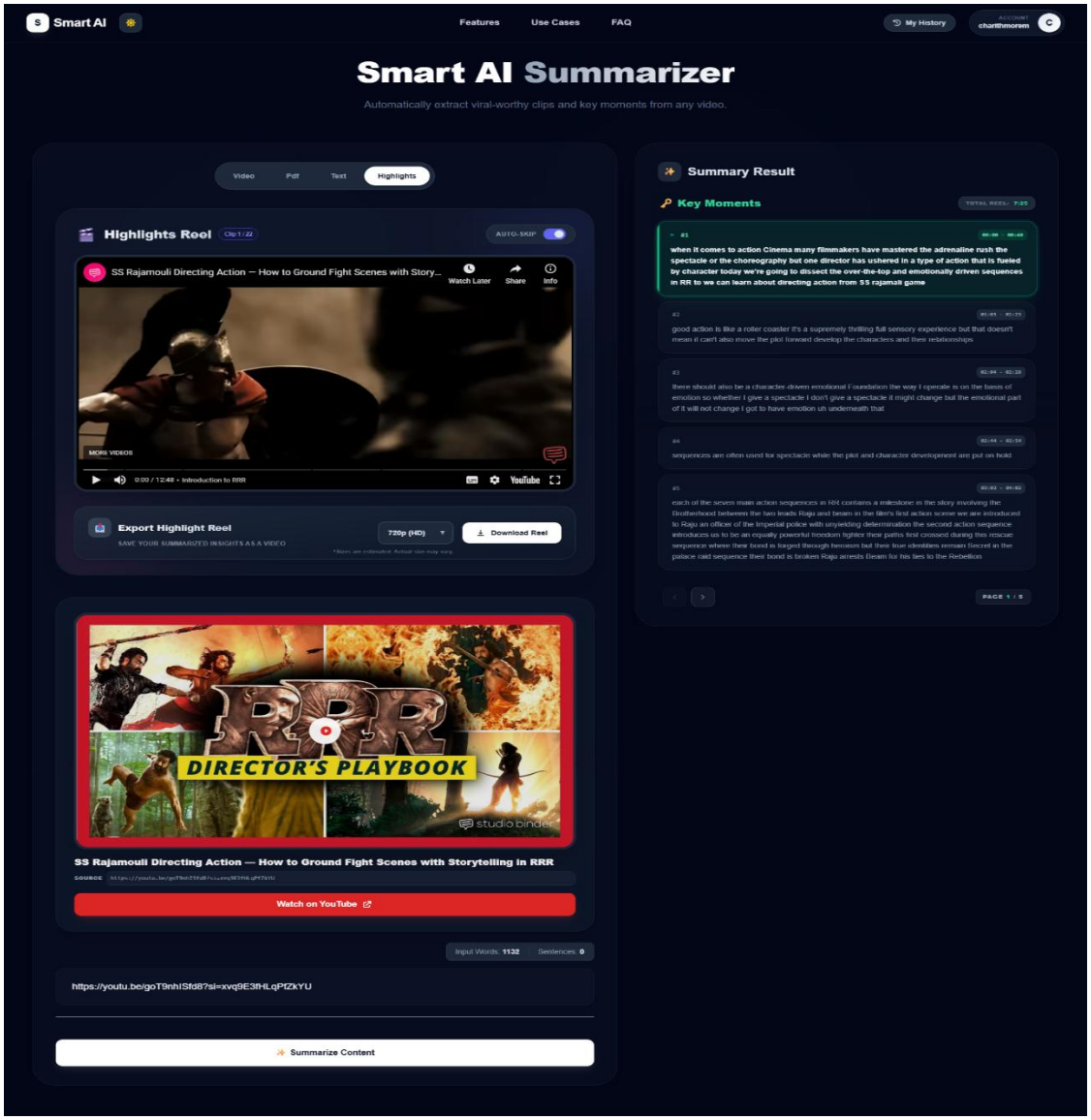

Figure 4.6 Highlight Extraction Output and Video Download Interface

Figure 4.6 illustrates the output interface of the Highlights Module in the Smart AI Summarizer web application. The interface displays the automatically generated highlights reel along with a list of detected key moments extracted from the YouTube video. Each key moment is mapped to its corresponding timestamp, allowing users to quickly understand the context of important segments. The system provides an option to export the summarized highlights as a single compiled video, with selectable resolution and a direct download feature. This module enables users to efficiently review, navigate, and download the most significant portions of long video content, enhancing usability for educational and content analysis purposes.

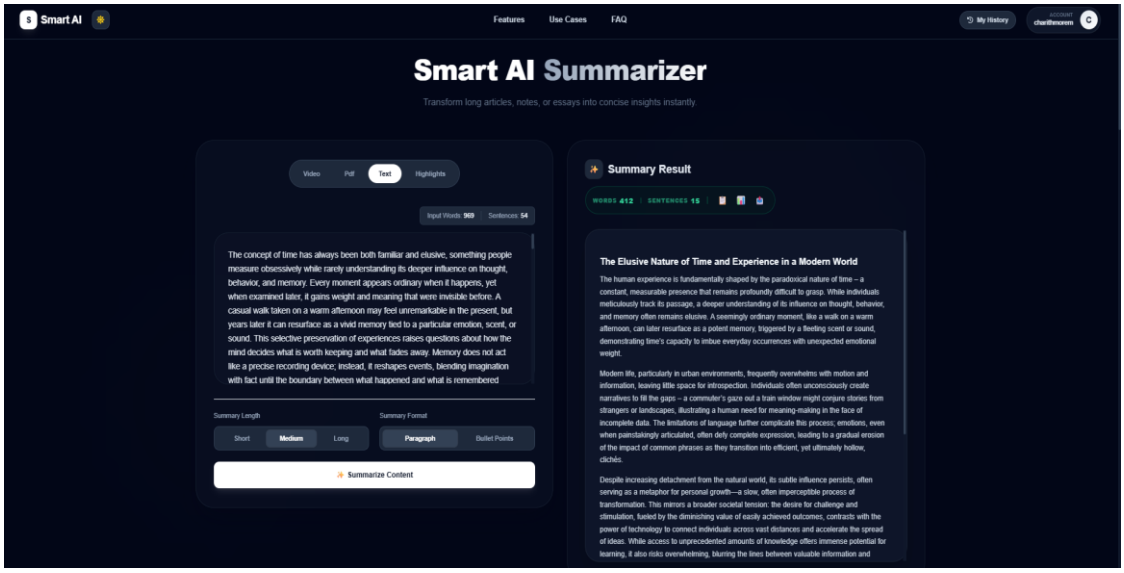

Figure 4.7 Text Summarization Output Interface

Figure 4.7 shows the output interface of the Text Summarization Module in the Smart AI Summarizer web application. The interface displays the original input text on the left panel and the generated summarized content on the right panel. The system provides statistical details such as word count and sentence count for the generated summary. Users can choose the summary length and output format prior to generation, and the resulting summary is presented in a clear, readable layout. This output interface enables users to easily compare the original content with the summarized version and efficiently extract key insights.

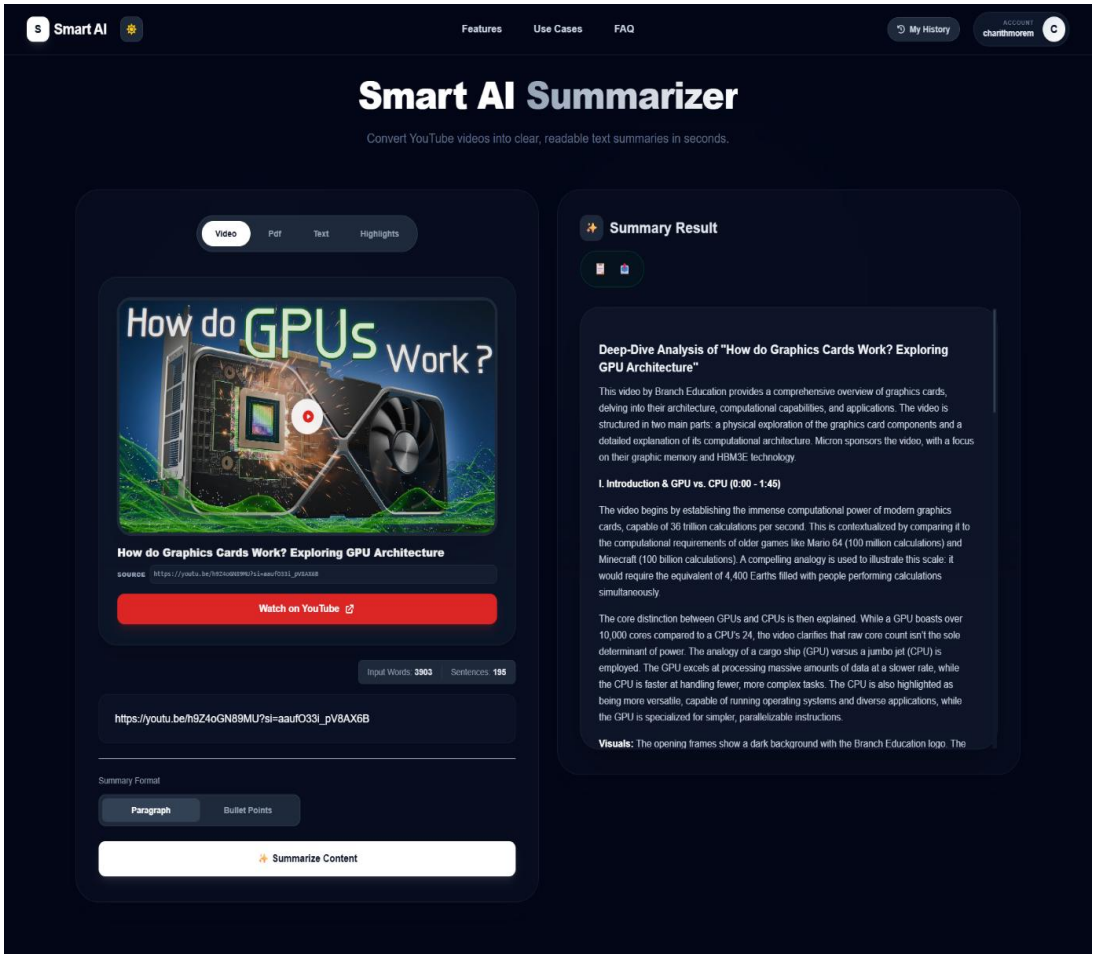

Figure 4.8 Video Summarization Output Interface

Figure 4.8 depicts the output interface of the Video Summarization Module in the Smart AI Summarizer web application. The interface displays the selected YouTube video along with its metadata on the left panel, while the generated textual summary is shown on the right panel. The summary is produced by analyzing the video transcript and visual context using a cloud-based multimodal AI model. Input statistics such as word count and sentence count are provided to indicate the scale of the original content. Users can choose the summary format before generation, and the resulting output presents a structured, readable overview of the video's key concepts, enabling quick understanding of long video content.

Observations

- The system demonstrated improved performance when processing text-based inputs locally, reducing dependency on cloud resources. Video processing required more computational time due to transcript extraction and multimodal analysis, but produced higher-quality summaries.
- The hybrid approach helped balance cost and efficiency by using local resources whenever possible. Real-time streaming of generated summaries improved user experience.
- The results indicate that combining extractive preprocessing with abstractive AI generation significantly improves the quality and readability of summaries. The use of TF-IDF scoring ensured that only important sentences were processed by the AI model, reducing noise.
- The integration of local and cloud-based models enhanced system flexibility. Local inference ensured privacy and reduced operational costs, while cloud-based multimodal processing provided advanced video understanding capabilities.
- The highlights module added practical value by enabling users to quickly navigate important segments of lengthy videos. This feature is particularly beneficial for educational and technical content analysis.
- However, the system's performance depends on the availability and accuracy of video transcripts. When transcripts are incomplete, summary quality may slightly decrease. Additionally, cloud API processing introduces network latency.
- Despite these limitations, the system successfully demonstrates a scalable and efficient approach to automated multimodal summarization.

CHAPTER 5

CONCLUSIONS AND FUTURE SCOPE

5.1 Conclusions

The Smart AI Video Summarizer with Text, PDF, and Highlights project presents a comprehensive and scalable solution to the growing challenge of information overload in the digital age. With the rapid increase in multimedia content, users often struggle to extract meaningful insights from lengthy videos, research documents, and textual data. This system addresses that challenge by integrating multimodal Artificial Intelligence techniques into a unified platform capable of processing diverse content formats efficiently.

The proposed system successfully combines extractive Natural Language Processing techniques with advanced Large Language Models to generate coherent, concise, and context-aware summaries. By incorporating a hybrid AI architecture, the system optimizes performance and resource utilization. Text and PDF summarization tasks are processed locally using the Gemma 3 (12B) model through Ollama, ensuring reduced latency, improved privacy, and lower operational cost. For video content, which requires multimodal understanding and contextual reasoning, the system utilizes the Gemma 3 (27B IT) model through the Google Gemini Cloud API. This selective use of cloud resources enhances scalability while maintaining cost efficiency.

The implementation demonstrates effective data preprocessing, including tokenization, sentence scoring, TF-IDF-based filtering, and structured content refinement before abstractive summary generation. This layered approach improves summary coherence and reduces irrelevant noise in the final output. The system's ability to dynamically select AI models based on input type highlights its modular and adaptable architecture.

A significant contribution of the project is the Highlights Extraction Module, which identifies key moments within YouTube videos and maps them to precise timestamps. The system further allows merging these segments into a single downloadable highlight reel. This functionality extends beyond traditional summarization by providing actionable multimedia outputs, enhancing practical usability for

educational, analytical, and content creation purposes.

The user interface, developed using React, ensures smooth interaction and accessibility. Features such as configurable summary length, format selection, Google Authentication, and history management enhance user experience and usability. The backend, implemented with FastAPI, effectively manages asynchronous processing, AI orchestration, media handling, and database storage.

- Overall, the system meets its core objectives by delivering:
- Efficient multimodal summarization
- Hybrid local-cloud AI integration
- Automated highlight extraction and export
- Secure authentication and history tracking
- Scalable and modular architecture

The project successfully demonstrates how modern AI technologies can be applied to real-world multimedia processing challenges. It provides a strong foundation for future research and commercial development in intelligent content analysis systems.

5.2 Future Scope

While the current system performs effectively across supported input types, there are several opportunities for future enhancement and expansion that can significantly increase its capability, scalability, and real-world applicability.

One potential improvement is the integration of direct video file upload support, enabling users to summarize locally stored videos rather than limiting input to YouTube URLs. This would broaden usability across enterprise and academic settings where private video content needs to be processed.

Another major enhancement would be the implementation of multilingual summarization capabilities. By incorporating multilingual language models and translation pipelines, the system could process and summarize content in multiple languages, expanding its accessibility to a global audience.

Further research can be conducted to improve semantic highlight ranking mechanisms. Currently, highlights are identified using transcript-based analysis; future versions could integrate deeper contextual understanding, emotion detection, or engagement scoring to enhance highlight relevance.

Performance optimization can also be achieved through GPU acceleration for local inference, enabling faster summarization of large documents. Additionally, adopting distributed or microservices-based cloud deployment could enhance scalability for large-scale enterprise usage.

Another valuable enhancement would involve user-personalized summarization preferences, allowing customization based on tone (formal, academic, conversational), depth of detail, or domain-specific focus. Integration with external tools such as Learning Management Systems (LMS), research databases, or content management platforms could further increase practical applicability.

Future versions may also incorporate:

- Advanced speech recognition for improved transcript accuracy
- AI-powered summarization comparison tools
- Sentiment analysis and topic clustering
- Automated metadata tagging
- Mobile application deployment
- Enterprise-level dashboard analytics
- Offline summarization mode for private environments

In the long term, this system can evolve into a full-fledged AI-driven multimedia intelligence platform, capable of serving industries such as education, journalism, research, digital marketing, and enterprise knowledge management.

REFERENCES

[1] Hua H., Tang Y., Xu C., Luo J., "V2Xum-LLM: Cross-Modal Video Summarization with Temporal Prompt Instruction Tuning", in *Proc. AAAI Conf. Artificial Intelligence*, vol. 39, no. 4, pp. 3599-3607, 2025. DOI: 10.1609/aaai.v39i4.32374.

[2] He B., Wang J., Qiu J., Bui T., Shrivastava A., Wang Z. Align and Attend: Multimodal Summarization with Dual Contrastive Losses. Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), pp. 14867-14878, 2023. DOI: 10.1109/CVPR52729.2023.01428.

[3] Khullar A., Arora U. MAST: Multimodal Abstractive Summarization with Trimodal Hierarchical Attention. Proc. First Workshop on Natural Language Processing Beyond Text (NLPBT), Association for Computational Linguistics, pp. 60-69, 2020. DOI: 10.18653/v1/2020.nlpbt-1.7.

[4] Saini P., Kumar K., Kashid S., Saini A., Negi A. Video summarization using deep learning techniques: a detailed analysis and investigation. Artificial Intelligence Review, vol. 56, no. 11, pp. 12347–12385, 2023. DOI: 10.1007/s10462-023-10444-0.

[5] Jangra A., Mukherjee S., Jatowt A., Saha S., Hasanuzzaman M. A Survey on Multi-modal Summarization. ACM Computing Surveys, vol. 55, no. 13s, Article 296, pp. 1–36, 2023. DOI: 10.1145/3584700.

[6] Lee M. J., Gong D., Cho M. Video Summarization with Large Language Models. Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), pp. 18981–18991, 2025. DOI: 10.48550/arXiv.2504.11199..

[7] Alaa T., Mongy A., Bakr A., Diab M., Gomaa W. Video Summarization Techniques: A Comprehensive Review. Proc. 21st International Conference on Informatics in Control, Automation and Robotics (ICINCO), vol. 2, pp. 141–148, 2024. DOI: 10.5220/0012936400003822

[8] Vora D., Kadam P., Mohite D. D., Kumar N., Kumar N., Radhakrishnan P., Bhagwat S. AI-driven video summarization for optimizing content retrieval and management through deep learning techniques. Scientific Reports, vol. 15, Article 4058, 2025. DOI: 10.1038/s41598-025-87824-9.

[9] Singh Y., Kumar R., Kabdal S., Upadhyay P. YouTube Video Summarizer using NLP: A Review. International Journal of Performability Engineering, vol. 19, no. 12, pp. 817–823, 2023. DOI: 10.23940/ijpe.23.12.p6.817823.

[10] Marevac E., Kadušić E., Živić N., Buzađija N., Tabak E., Velić S. Multimodal Video Summarization Using Machine Learning: A Comprehensive Benchmark of Feature Selection and Classifier Performance. Algorithms, vol. 18, no. 9, p. 572, 2025. DOI: 10.3390/a18090572.

[11] Lan L., Jiang L., Yu T., Liu X., He Z. FullTransNet: Full Transformer with Local-Global Attention for Video Summarization. arXiv preprint arXiv:2501.00882, 2025.

[12] Altundogan T. G., Karakose M. QUBVIS: Query-Based Multi-Modal Summarization System Using CLIP-Based Transformer and Vision-Language Models. SoftwareX, vol. 31, Article 102303, 2025. DOI: 10.1016/j.softx.2025.102303.

[13] Zhu Y., Zhao W., Hua R., Wu X. Topic-aware video summarization using multimodal transformer. Pattern Recognition, vol. 140, Article 109578, Aug. 2023. DOI: 10.1016/j.patcog.2023.109578.

[14] Wu G., Wang M., Ma N. Multimodal video summarization based on graph contrastive learning with fine-grained graph interaction. Signal Processing, vol. 239, p. 110250, 2026. DOI: 10.1016/j.sigpro.2025.110250.

.

[15] Park J., Lee J., Sohn K. Language-guided Recursive Spatiotemporal Graph Modeling for Video Summarization. International Journal of Computer Vision, vol. 133, no. 12, pp. 8617–8641, 2025. DOI: 10.1007/s11263-025-02577-2.

[16] Barbara M., Maalouf A. Prompts to Summaries: Zero-Shot Language-Guided Video Summarization. arXiv preprint arXiv:2506.10807, 2025. DOI: 10.48550/arXiv.2506.10807.

[17] Wang S., Zhang J. MF2Summ: Multimodal Fusion for Video Summarization with Temporal Alignment. arXiv preprint arXiv:2506.10430, 2025.

[18] Liu S., Wang L., Zhu X., Lu X., Wang Z., Hu K. SITransformer: Shared Information-Guided Transformer for Extreme Multimodal Summarization. Proc. 6th ACM International Conference on Multimedia in Asia (MMAsia), pp. 1–7, 2024. DOI: 10.1145/3696409.3700234.

[19] Pennec G., Liu Z., Asher N., Muller P., Chen N. F. Integrating Video and Text: A Balanced Approach to Multimodal Summary Generation and Evaluation. Proc. 14th International Joint Conference on Natural Language Processing (IJCNLP) and 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pp. 2403–2426, 2025.

[20] Lin J., Hua H., Chen M., Li Y., Hsiao J., Ho C., Luo J. VideoXum: Cross-modal Visual and Textural Summarization of Videos. IEEE Transactions on Multimedia, 2023. DOI: 10.48550/arXiv.2303.12060.