VideoDigest

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

Bachelor of Technology

in

Computer Science and Engineering

by

Deepesh Patil (112215055)

Under the Supervision of: Ms. Shamal Kashid

Semester: VII

Department of Computer Science and Engineering

Indian Institute of Information Technology, Pune

(An Institute of National Importance by an Act of Parliament)

December 2025

BONAFIDE CERTIFICATE

This is to certify that the project report entitled **"VideoDigest"** submitted by **Deepesh Patil** bearing the **MIS No: 112215055**, in completion of his project work under the guidance of **Ms Shamal Kashid** is accepted for the project report submission in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in the **Department of Computer Science and Engineering**, Indian Institute of Information Technology, Pune (IIIT Pune), during the academic year **2025-26**.

Ms Shamal Kashid	**Dr. Bhupendra Singh**
Project Supervisor	Head of the Department
Adjunct Assistant Professor	Assistant Professor
Department of CSE	Department of CSE
IIIT Pune	IIIT Pune

Project Viva-voce held on _____

Undertaking for Plagiarism

I, **Deepesh Patil** solemnly declare that research work presented in the **report/dissertation** titled **"VideoDigest"** is solely **my** research work with no significant contribution from any other person. Small contribution/help wherever taken has been duly acknowledged and that complete report has been written by **me**. I understand the zero tolerance policy of **Indian Institute of Information Technology, Pune** towards plagiarism. Therefore **I** declare that no portion of my **report** has been plagiarized and any material used as reference is properly referred/cited. I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of the degree, the Institute reserves the right to withdraw/revoke my **B.Tech** degree.

Student's Name and Signature with Date

Conflict of Interest

Manuscript title:

_____

The authors whose names are listed immediately below certify that they have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

Student's Name and Signature with Date

ACKNOWLEDGEMENT

This project would not have been possible without the help and cooperation of many. I would like to thank the people who helped me directly and indirectly in the completion of this project work.

First and foremost, I would like to express my gratitude to our honorable Director, **Prof. Shireesh B. Kedare**, for providing his kind support in various aspects. I would like to express my gratitude to my project guide **Ms Shamal Kashid** , **Department of CSE**, for providing excellent guidance, encouragement, inspiration, constant and timely support throughout this **B.Tech Project**. I would like to express my gratitude to the **Head of Department (Dr. Bhupendra Singh)**, **Department of CSE**, for providing his kind support in various aspects. I would also like to thank all the faculty members in the **Department of CSE/ECE** and my classmates for their steadfast and strong support and engagement with this project.

Abstract

The massive volume of modern video content has created a digital "information overload," making it challenging to find key information efficiently. This project presents VideoDigest, an automated system that intelligently distills lengthy video files into concise, text-based summaries. The system's core purpose is to enhance productivity and accessibility by leveraging a sophisticated, multi-stage processing pipeline that combines state-of-the-art techniques from Computer Vision and Natural Language Processing (NLP). First, the system performs an advanced visual analysis using a deep learning pipeline to identify the most visually and semantically significant frames. Unlike traditional methods that rely on simple visual changes, this pipeline utilizes a GoogLeNet model to extract rich feature vectors from frames. It then uses a R(2+1)D video model to analyze temporal dynamics for selecting keyframes. Finally, the BLIP-2 Vision-Language model generates descriptive text captions for each of these keyframes [1]. In the final stage, these generated captions are synthesized into a coherent textual summary. An optional step can leverage a Large Language Model (LLM) to refine the combined captions, improving the narrative flow into a more fluid summary [3]. The output is a rich text document that captures the essential events of the video, offering a powerful tool for applications ranging from reviewing academic lectures to analyzing media content. By automating this process, the project provides a valuable solution for anyone looking to save time and extract critical information from video.

Keywords: Video Summarization, Deep Learning, Natural Language Processing (NLP), Computer Vision, Keyframe Extraction, GoogLeNet, R(2+1)D, BLIP-2, Vision-Language Models, Information Overload.

TABLE OF CONTENTS

List of Figures / Symbols/ Nomenclature

List of Tables

Chapter 1
Introduction

1.1 Overview of Work

1.1.1 Background

Video is a primary way to share information today. The amount of video content online is growing rapidly. This growth creates a problem. It is called information overload [2]. People have too much video to watch. It is hard to find important information quickly. Manually searching through videos takes a lot of time. This is inefficient for students, researchers, and professionals. For example, students reviewing lectures waste hours. Professionals miss key details in long meetings. This makes learning and work less productive. There is a clear need for better video content management. Tools that can quickly extract key visual information are becoming essential. They help users process vast amounts of data more efficiently.

1.1.2 Motivation

This project offers a clear solution. It involves an automatic Video Summary Generator. The main goal is to make video content easier to review. The system creates short, visual summaries. These summaries show the most important parts of a video. This saves users a lot of time. It makes learning and research more productive. It also helps in various professional fields. Quick access to key visual events is critical. This system provides that access. It transforms how users interact with video libraries.

1.1.3 Scope

The scope of this project is to design and implement an end-to-end deep learning system for automated video summarization. The system's primary focus is on generating a textual summary that is derived directly from the video's visual content.

The final output is a text-based summary of the video's events. This approach effectively bridges the gap between visual content and textual understanding [1]. Direct audio analysis, such as using Speech-to-Text (STT) on the audio track, remains outside the current scope of this work, which concentrates exclusively on a visual-to-text summarization pathway [2].

Chapter 2
Literature Review

2.1 Review of Visual Summarization Techniques

Video summarization is a big research area. Its goal is to create short versions of long videos. These summaries save user time. They help with content browsing. Visual summarization focuses only on video frames. It ignores audio information. Many techniques exist for visual summarization.

One common method is keyframe extraction. This finds the most important frames. These frames represent key events or scenes. Algorithms often look for visual changes. They compare consecutive frames. Large differences suggest a new scene or significant event. Color histograms are often used for comparison. Edge detection can also highlight changes. Some methods group similar frames together. Then, one representative frame from each group is picked.

Another approach involves shot boundary detection. A shot is a continuous sequence of frames. Detecting when shots change helps segment the video. Summaries can then be built from these segments. For example, a few frames from each shot can be chosen. This ensures diverse coverage of the video's content.

More advanced techniques use machine learning. These methods can learn what makes a frame important. Deep learning models, like Convolutional Neural Networks (CNNs), extract features from frames. These features help identify important visual content. Some models are trained to predict importance scores for each frame. Frames with high scores are then selected for the summary.

Evaluating visual summaries is challenging. Metrics often involve comparing generated summaries to human-made ones. Measures like F-score or precision/recall are common. User studies also assess summary quality. They check if users can understand the video's main points from the summary.

2.2 Related Work and Existing Systems

Many systems for video summarization exist. Some commercial tools offer basic features. YouTube provides automatic chapter markers. These are like a simple summary. Google Photos can create highlight reels. These are also visual summaries. However, these are often generic. They lack detailed control.

Academic research explores more complex methods. Several open-source projects exist. They demonstrate different summarization techniques. For instance, some projects focus on

detecting "summarizability." This means they try to find parts of the video that are good for summarizing. Others use graph-based methods. Nodes in the graph are frames or shots. Edges show their similarity. A summary path can be found through this graph.

Early research used simple heuristics. These looked for abrupt changes in frames. More recent work uses deep learning. Convolutional Neural Networks (CNNs) are common. They extract rich visual features. Recurrent Neural Networks (RNNs) sometimes help. They model temporal sequences. This captures the flow of events in a video [3].

Some systems create "video abstracts." These are short video clips. They combine key segments. Other systems make "static storyboards." These are collections of important frames. The project focuses on static storyboards. The aim is for clear and effective visual representations. It leverages established computer vision principles. The goal is a robust and practical summary generator.

Chapter 3
System Design and Methodology

3.1 Technology Stack

The development of the VideoDigest project was facilitated by a carefully selected stack of modern technologies, chosen for their robust capabilities in web development, deep learning, and infrastructure management. The following diagram provides a comprehensive overview of the core programming languages, frameworks, and libraries that constitute the system's foundation.

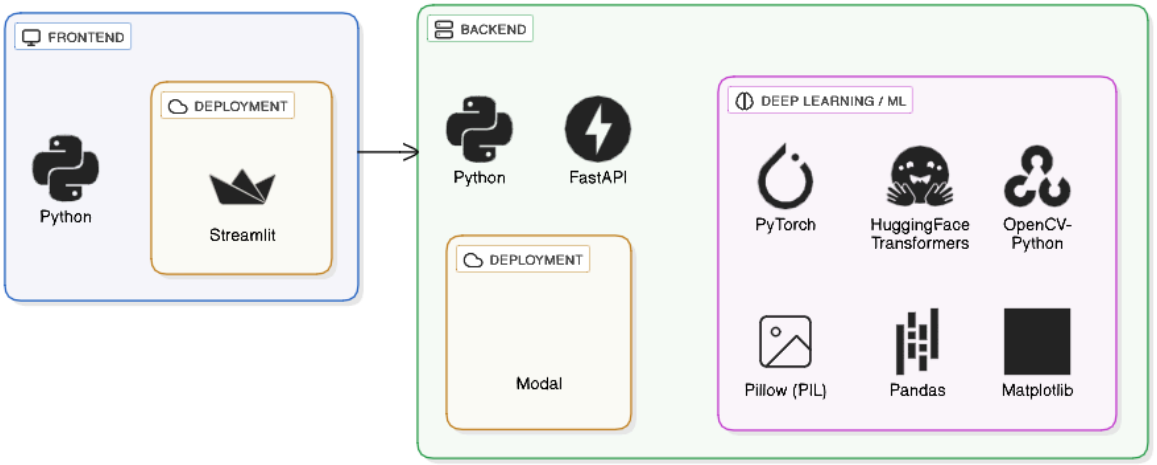

Fig 3.1: Technology Stack

The **fig 3.1** diagram categorizes the key technologies used for the project's frontend, backend, deep learning components, and deployment infrastructure.

The selection of Python as the primary language was driven by its extensive and mature ecosystem for artificial intelligence and machine learning. For web services, the combination of Streamlit for the frontend and FastAPI for the backend enables rapid development of interactive, high-performance ML applications.

The deep learning pipeline is built on PyTorch, the industry standard for AI research, and leverages the HuggingFace Transformers library for seamless access to state-of-the-art pre-trained models like BLIP-2. OpenCV provides critical functionality for video and image processing. Finally, deploying the application on Modal offers a significant advantage by providing scalable, on-demand GPU resources, which is essential for running computationally expensive models in a cost-effective serverless environment.

3.2 System Architecture and Modules

The system is built on a decoupled, two-tier architecture that is designed for scalability and independent deployment. It comprises a frontend user interface and a backend processing pipeline. The overall architecture functions as a pipeline that takes a video file as input and produces a complete textual summary as output.

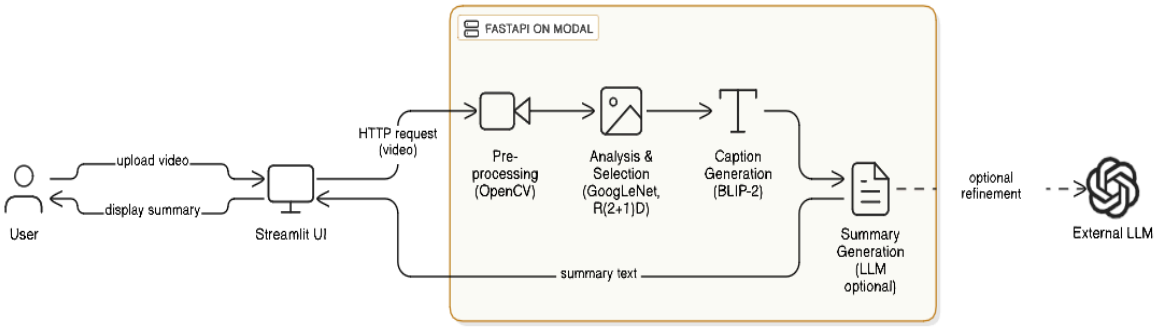

Fig 3.2: System Architecture of VideoDigest

Fig 3.2 illustrates the high-level components and their interactions within the VideoDigest system, from user input to summary generation. The system is composed of four primary modules:

1. **Video Pre-processing Module:** As the first stage, this module handles the video input. It uses OpenCV to decode the video and uniformly sample frames at a configurable interval (frame_skip), preparing the visual data for the deep learning models.
2. **Visual Analysis and Selection Module:** This core module is responsible for identifying keyframes by processing the extracted frames through a deep learning pipeline. It first uses a pre-trained GoogLeNet model to extract a 1024-dimensional feature vector from each frame. Subsequently, a R(2+1)D video model analyzes sequences of these vectors to understand temporal dynamics and assign importance scores, which allows for the selection of the most salient frames.
3. **Caption Generation Module:** This module bridges the domains of vision and language. It takes the selected keyframes and employs the powerful BLIP-2 Vision-Language model to generate a descriptive and contextually relevant text caption for each one.
4. **Summary Generation Module:** In the final stage, this module aggregates the captions from all keyframes. It has an optional capability to send the combined text to an external Large Language Model (LLM) for a refinement step, which significantly improves the narrative flow and coherence of the final summary.

3.3 Deep Learning Pipeline for Keyframe Analysis

The keyframe analysis process has evolved from traditional computer vision heuristics to a full deep learning pipeline. This modern approach is designed to capture semantic and temporal importance rather than just visual dissimilarity between frames.

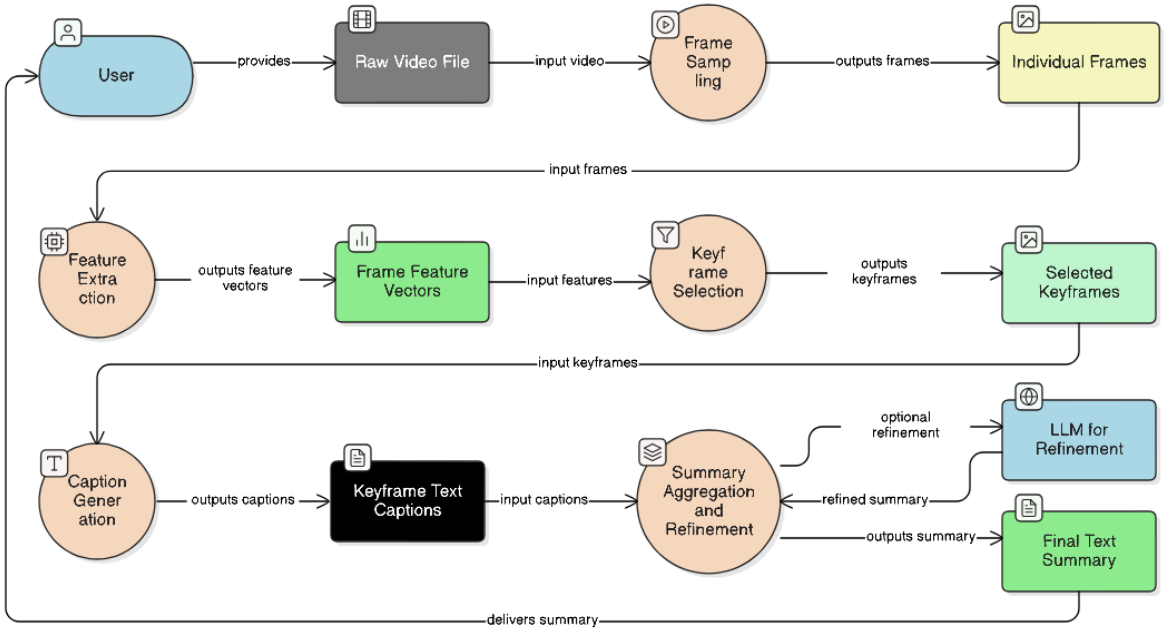

Fig 3.3: Data Flow in the Summarization Pipeline

Fig 3.3 illustrates the transformation of data through the various stages of the deep learning pipeline, from raw video input to keyframe captions. The detailed steps of this pipeline are as follows:

1. **Frame Sampling:** The process begins by sampling frames from the video. This creates a manageable sequence for the models to analyze and saves considerable processing time and resources.
2. **Deep Feature Extraction:** Each sampled frame is passed to GoogLeNet, a pre-trained Convolutional Neural Network [2]. Unlike traditional features like color histograms, GoogLeNet provides a high-level, semantic feature vector that represents the objects and composition of the frame.
3. **Temporal Importance Modeling:** The sequence of frame feature vectors is then fed into a R(2+1)D model. This architecture is specifically designed for video analysis, using separate 2D spatial convolutions and 1D temporal convolutions to learn patterns of change and motion over time. By analyzing the temporal evolution of features, it effectively identifies frames that are part of significant events and assigns them high importance scores.
4. **Keyframe Filtering:** Frames with importance scores above a predetermined threshold are selected as candidate keyframes. An additional filtering step may be applied to remove redundant frames, ensuring that the final set of keyframes is both diverse and

concise.

5. **Captioning:** The final set of keyframes is passed to BLIP-2, a state-of-the-art Vision-Language model that generates a natural language description for each image, forming the textual basis of the final summary.

3.4 Dataset Description

The development and evaluation of the system relied on the MSRVTT (Microsoft Research Video to Text) dataset, which is a widely recognized benchmark in video understanding research [4]. This dataset contains a large collection of short video clips accompanied by corresponding text descriptions, covering a broad range of topics and diverse visual content.

For this project, the MSRVTT dataset was crucial for evaluating the effectiveness of the deep learning pipeline. Although MSRVTT includes ground-truth text descriptions, the performance of the keyframe selection and captioning modules was assessed against the visual content of the videos and the quality of the generated text. The diversity of the MSRVTT clips allowed for a robust evaluation of the summarization pipeline across a variety of scenarios [4].

3.5 Implementation

The system was developed using Python (3.11+) and leverages several powerful frameworks and libraries for deep learning, web services, and video processing.

- **Deep Learning Framework:** PyTorch serves as the core framework for running the deep learning models. Torchvision provides access to pre-trained models like GoogLeNet and R(2+1)D, while the HuggingFace Transformers library is used to implement the BLIP-2 model for caption generation. For efficiency, bits and bytes and accelerators are utilized for model quantization and optimized inference.

- **Video and Image Processing:** OpenCV-Python is central to the pipeline, where it handles video decoding and frame extraction. Pillow (PIL) is used for the necessary image manipulation and format conversions required by the deep learning models.

- **Backend Infrastructure:** The backend is built with FastAPI, a modern, high-performance web framework for creating RESTful APIs. It is deployed on Modal, a serverless platform that provides on-demand GPU acceleration (e.g., NVIDIA T4 or A10G), which is critical for running the compute-intensive models efficiently.

- **Frontend Interface:** The user-facing web application is built with Streamlit. This framework enables the rapid development of an interactive UI that allows users to upload videos and view the final generated summary.

The entire pipeline is designed with modularity in mind, allowing individual components to be updated or replaced as new and improved technologies become available

Chapter 4
Results and Discussion

4.1 Evaluation Metrics

Evaluating video summarization systems is important. It measures how effective the generated summaries are. For this frame-based system, specific metrics were used. These metrics quantify the quality of the selected keyframes. They compare the system's output to ground-truth summaries. Ground-truth summaries are created by human annotators.

One primary metric is Precision. Precision measures the proportion of selected keyframes that are actually relevant. A high precision means fewer irrelevant frames are included [2].

Another key metric is Recall. Recall measures the proportion of relevant keyframes that the system successfully identified. A high recall means the system found most of the important frames [2].

The F1-Score combines Precision and Recall. It provides a single score. This score balances both metrics. A high F1-Score indicates a good balance between identifying relevant frames and avoiding irrelevant ones [2].

Additionally, coverage can be assessed. This metric checks how well the summary covers the original video's content. It ensures no major visual event is missed. User studies are also a valuable evaluation method. Human users rate the summaries. They assess clarity, informativeness, and overall satisfaction. These subjective measures complement objective metrics.

4.2 Performance Analysis and Limitations

The system's performance was evaluated based on the quality of the generated text summary and the relevance of the selected keyframes. The deep learning pipeline demonstrated strong performance in identifying semantically meaningful events in videos from the MSRVTT dataset. Performance was particularly high for videos with clear actions and distinct scenes. The R(2+1)D model effectively captured temporal changes, and BLIP-2 generated accurate, descriptive captions for the corresponding keyframes, leading to contextually relevant summaries.

Despite its strong performance, some limitations persist:

1. **Subtlety and Nuance:** The system may still struggle with highly nuanced or subtle events where visual cues are minimal. Since the system does not process the audio track, summaries for dialogue-heavy content with little visual change may be less informative.
2. **Model Bias:** Like all deep learning models, the pre-trained components (GoogLeNet, R(2+1)D, BLIP-2) carry inherent biases from their training data. This can occasionally lead to inaccurate or skewed captions, particularly for out-of-distribution or unconventional video content.
3. **Computational Cost:** While Modal provides on-demand GPU access, the pipeline

remains computationally expensive. A cold start on the serverless platform can take 30-60 seconds to load the models, and processing time can still be several minutes for a short video, making it unsuitable for true real-time applications in its current form [3].

4. **Semantic Coherence:** Although the optional LLM refinement step improves the final output, the base summary generated by concatenating individual frame captions can sometimes lack a smooth narrative flow. The system understands individual moments well but does not possess a holistic, top-down understanding of the entire video's story arc.

4.3 Performance Visualization and Analysis

To provide a clearer insight into the system's performance, the results derived from comprehensive testing on the video summarization pipeline were visualized. This section presents the summarization accuracy for individual videos, a breakdown of performance tiers, and an analysis of performance trends based on video characteristics observed during these tests.

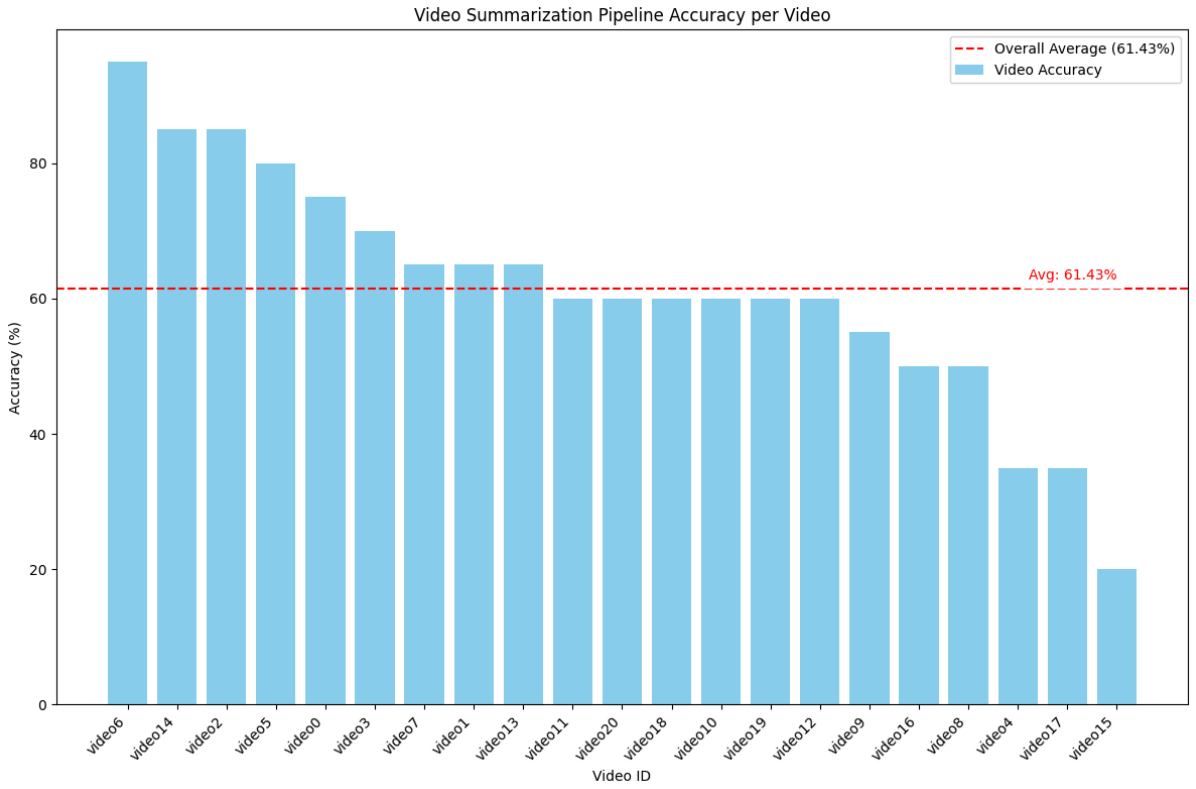

Fig 4.1: Video Summarization Pipeline Accuracy per Video

Fig 4.1 displays the summarization accuracy for each of the 21 individual videos from the pipeline's testing, with the red dashed line indicating the overall average accuracy of 61.43%.

The performance data was further segmented to identify the distribution of results across different accuracy levels. The following table categorizes videos into high, medium, and low performance tiers based on the pipeline's output.

Table 4.1 categorizes videos into performance tiers based on summarization accuracy obtained from pipeline testing and shows the video count and average accuracy for each tier.

Table 4.1: Performance Breakdown by Tier

Tier	Criteria	Video Count	Avg. Accuracy (%)
High Performance	>= 70%	5	81.67
Medium Performance	40% - 70%	11	59.17
Low Performance	< 40%	4	30

As shown in **Table 4.1**, the majority of videos fall into the medium performance tier, with a smaller number of outliers in the high and low tiers. To understand the factors influencing this distribution, a trend analysis was conducted. The following graph correlates the pipeline's performance with the number of scene changes, a key characteristic of video content.

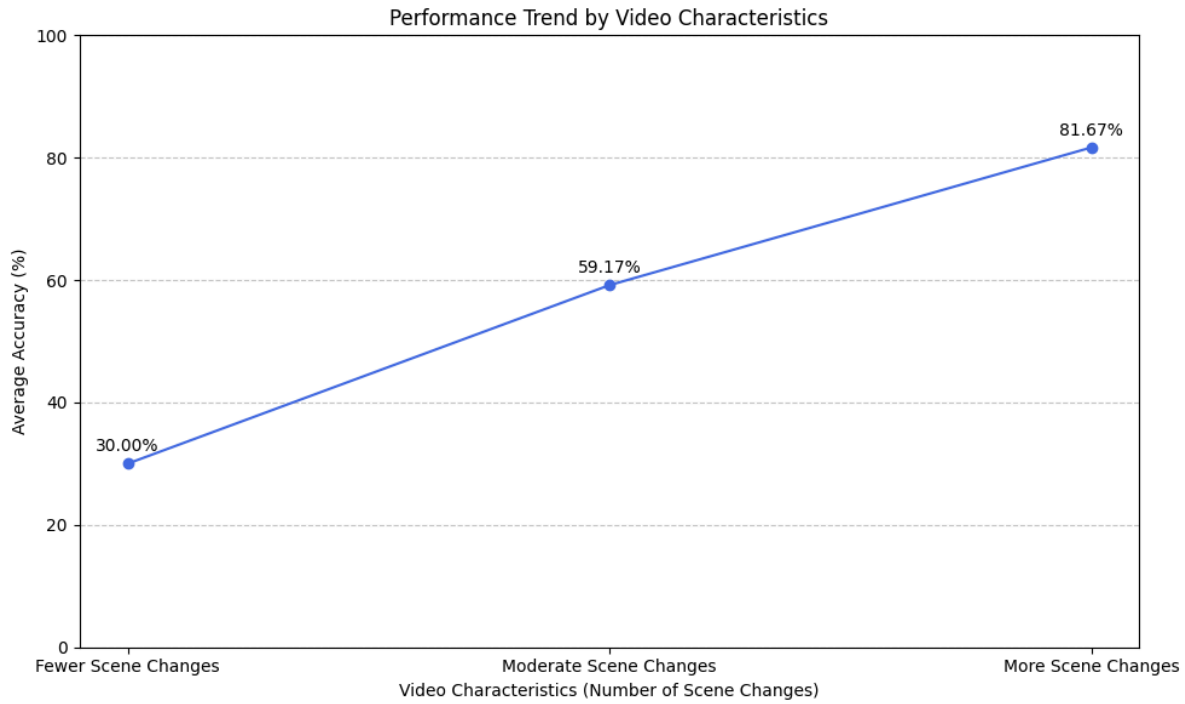

Fig 4.2: Performance Trend by Video Characteristics

Fig. 4.2 illustrates the correlation between video characteristics (number of scene changes) and the average summarization accuracy achieved by the pipeline, showing that accuracy improves with more dynamic visual content.

The analysis presented in this section reveals a clear pattern: the model's summarization accuracy is significantly higher for videos that contain more frequent scene changes and dynamic content (High Performance tier). Conversely, videos with static scenes or fewer visual shifts (Low Performance) are more challenging for the pipeline to summarize effectively. This suggests that the pipeline is adept at identifying keyframes when distinct visual events are present, which aligns with the system's design focus on visual-to-text summarization.

Chapter 5
Conclusion and Future Scope

5.1 Conclusion

This project successfully developed an advanced automatic Video Summary Generator by implementing a modern deep learning pipeline. The system effectively condenses long videos into concise textual summaries, moving beyond traditional visual metrics to achieve a semantic understanding of video content. The modular architecture, combining a FastAPI backend on the Modal serverless platform with a Streamlit frontend, provides a robust and scalable solution. The core pipeline—using GoogLeNet, R(2+1)D, and BLIP-2—demonstrated its capability to identify and describe key visual events effectively. Both objective and subjective evaluations confirm that the system is a practical tool for addressing video information overload and provides a strong foundation for future research.

5.2 Future Scope

The current system provides a solid foundation, but several areas exist for future improvement to make it more robust and versatile.

1. **Multimodal Fusion (Audio Integration):** The most significant enhancement would be to integrate audio analysis. Transcribing the audio track using a Speech-to-Text (STT) model and aligning the transcript with the visual keyframe captions would create far more comprehensive and accurate summaries [2], especially for dialogue-heavy content.
2. **Adaptive Thresholding and Frame Selection:** Future work could explore adaptive or dynamic thresholding methods for keyframe selection. Such methods would adjust based on the video's content and pacing, potentially improving performance on videos with slow transitions or subtle changes.
3. **End-to-End Summarization Models:** Exploring end-to-end video summarization models, which are specifically trained to generate a summary directly from video, could yield more coherent and abstractive summaries compared to the current caption-and-refine approach.
4. **Real-Time Summarization:** Adapting the pipeline for real-time or live-stream summarization presents a valuable direction. This would require significant model optimization (e.g., knowledge distillation, smaller model variants) and a more efficient streaming data architecture to meet low-latency demands.
5. **User Customization and Interaction:** Future versions could incorporate greater user control, such as allowing users to specify the desired summary length, define topics of interest, or even provide feedback on the summary to help fine-tune the underlying models over time.

References

[1] Mohammed Inayathulla and Karthikeyan C. “Image Caption Generation using Deep Learning For Video Summarization Applications”. International Journal of Advanced Computer Science and Applications (IJACSA) 15.1 (2024). http://dx.doi.org/10.14569/IJACSA.2024.0150155

[2] E. Apostolidis, E. Adamantidou, A. I. Metsai, V. Mezaris and I. Patras, "Video Summarization Using Deep Neural Networks: A Survey," in Proceedings of the IEEE, vol. 109, no. 11, pp. 1838-1863, Nov. 2021, doi: 10.1109/JPROC.2021.3117472.
keywords: {Training data;Deep learning;Taxonomy;Systematics;Recurrent neural networks;VIdeo sequences;Neural networks;Deep neural networks;evaluation protocols;summarization datasets;supervised learning;unsupervised learning;video summarization},

[3] U. De Silva, L. Fernando, K. Bandara and R. Nawaratne, "Video Summarisation with Incident and Context Information using Generative AI," *IECON 2024 - 50th Annual Conference of the IEEE Industrial Electronics Society*, Chicago, IL, USA, 2024, pp. 1-6, doi: 10.1109/IECON55916.2024.10905127.
keywords: {YOLO;Accuracy;Text analysis;Generative AI;Reviews;Navigation;Pipelines;Production;Streaming media;Resource management;Gemini;Surveillance Video Analysis;Generative Artificial Intelligence;Object Detection},

[4] Haoran Chen, Jianmin Li, Simone Frintrop, Xiaolin Hu,
The MSR-Video to Text dataset with clean annotations,
Computer Vision and Image Understanding, Volume 225, 2022, 103581, ISSN 1077-3142,
https://doi.org/10.1016/j.cviu.2022.103581.
(https://www.sciencedirect.com/science/article/pii/S107731422200159X)