DAYANANDA SAGAR UNIVERSITY

Devarakaggalahalli, Harohalli
Kanakapura Road, Ramanagara - 562112, Karnataka, India

Bachelor of Technology
in
COMPUTER SCIENCE AND ENGINEERING

Major Project Phase-II Report

VIDEO EVENT LOCALIZATION AND SUMMERIZATION

Batch: 42

By

Puneeth M - ENG21CS0311
Rohith Bedre - ENG21CS0338
Rolwin Menezes -ENG21CS0339
Upamanyu SM - ENG22CS0452

Under the supervision of

Prof Arpita Paria
ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING,
SCHOOL OF ENGINEERING
DAYANANDA SAGAR UNIVERSITY,

(2024-2025)

DAYANANDA SAGAR UNIVERSITY

SCHOOL OF ENGINEERING

Department of Computer Science & Engineering
Devarakaggalahalli, Harohalli, Kanakapura Road, Ramanagara - 562112
Karnataka, India

CERTIFICATE

This is to certify that the Major Project Stage-II work titled **“Video Event Localization and Summarization”** is carried out by **PUNEETH M(ENG21CS0311), ROHITH BEDRE (ENG21CS0338), ROLWIN MENEZES(ENG21CS0339), UPAMANYU SM(ENG21CS0452),** bonafide students seventh semester of Bachelor of Technology in Computer Science and Engineering at the School of Engineering, Dayananda Sagar University, Bangalore in partial fulfillment for the award of degree in Bachelor of Technology in Computer Science and Engineering, during the year **2024-2025**.

Prof.Arpita Paria	**Dr. Girisha G S**	**Dr.UdayaKumar Reddy K R**
Assistant Professor Dept. of CSE, School of Engineering Dayananda Sagar University Date:	Chairman CSE School of Engineering Dayananda Sagar University Date:	Dean, School of Engineering Dayananda Sagar University Date:

Name of the Examiner	Signature of Examiner
1.	
2.	

DECLARATION

We, **PUNEETH M (ENG21CS0311), ROHITH BEDRE (ENG21CS0338), ROLWIN MENEZES(ENG21CS0339), UPAMANYU SM(ENG21CS0452),**are students of eighth semester B. Tech in **Computer Science and Engineering**, at School of Engineering, **Dayananda Sagar University**, hereby declare that the Major Project Stage-II titled **"Video Event Localization and Summarization"** has been carried out by us and submitted in partial fulfilment for the award of degree in **Bachelor of Technology in Computer Science and Engineering** during the academic year **2024-2025.**

Student **Signature**

Name: PUNEETH M

USN : ENG21CS0311

Name:ROHITH BEDRE

USN : ENG21CS0338

Name: ROLWIN MENEZES

USN : ENG21CS0339

Name: UPAMANYU SM

USN : ENG21CS0452

Place : Bangalore

Date :

ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and support of many individuals who have been responsible for the successful completion of this project work.

First, we take this opportunity to express our sincere gratitude to School of Engineering & Technology, Dayananda Sagar University for providing us with a greatopportunity to pursue our Bachelor's degree in this institution.

We would like to thank ***Dr. Udaya Kumar Reddy K R, Dean, School of Engineering, Dayananda Sagar University*** *for his constant encouragement and expert advice. It is a matter of immense pleasure to express our sincere thanks to* ***Dr. Girisha G S, Department Chairman****,* ***Computer Science and Engineering****,* ***Dayananda Sagar University,*** *for providing right academic guidance that made our task possible.*

We would like to thank our guide ***Prof Arpita Paria Assistant Professor****,* ***Dept. of Computer Science and Engineering****,* ***Dayananda Sagar University****, for sparing her valuable time to extend help in every step of our project work, which paved the way for smooth progress and fruitful culmination of the project.*

We would like to thank our ***Project Coordinator Dr. Meenakshi Malhotra, Dr. Kumar Dilip, and Dr. Sivananda Reddy E*** *as well as all the staff members of Computer Science and Engineering for their support.*

We are also grateful to our family and friends who provided us with every requirement throughout the course.

We would like to thank one and all who directly or indirectly helped us in the Project work.

TABLE OF CONTENTS

Page

LIST OF FIGURES

ABSTRACT

Video Temporal Grounding (VTG), which aims to ground target clips from videos (such as consecutive intervals or disjoint shots) according to custom language queries (e.g., sentences or words), is key for video browsing on social media. Most methods in this direction develop task-specific models that are trained with type-specific labels, such as moment retrieval (time interval) and highlight detection (worthiness curve), which limits their abilities to generalize to various VTG tasks and labels.

In this context, we focus on Video Localization and Summarization, addressing the challenges of unifying diverse tasks and labels in this domain. Firstly, we revisit a wide range of video localization and summarization tasks and define a unified formulation. Based on this, we develop data annotation schemes to create scalable pseudo supervision. Secondly, we propose an effective and flexible approach capable of addressing each task and making full use of each label. Lastly, this unified framework unlocks temporal localization pretraining from large-scale diverse labels, developing stronger localization and summarization capabilities, such as zero-shot inference.

This approach aims to consolidate efforts in video localization and summarization by addressing the limitations of traditional task-specific methods, providing a versatile and scalable solution for various user-driven video analysis needs.

CHAPTER 1.
INTRODUCTION

With the rapid growth of video-sharing platforms, video content has become one of the most influential and diverse forms of media. From unedited recordings of everyday moments to meticulously produced vlogs and tutorials, videos offer a vast array of information. This diversity has created a significant need for tools that enable efficient navigation and understanding of video content.

Video event localization and summarization are crucial techniques for achieving this goal. These processes involve identifying specific moments or events within videos and creating concise summaries to represent the overall content. For instance, users may want to locate a scene where children are dancing in a birthday video or generate a summary of key highlights focused on "food." Such tasks are essential for simplifying video browsing and enhancing user experiences on social media, educational platforms, and entertainment services.

Video Understanding Tasks Video event localization and summarization encompass several related tasks:

- **Moment Retrieval**: This involves identifying precise time intervals in a video that correspond to a textual description, such as "the part where the chef prepares the main dish."
- **Highlight Detection**: This task focuses on determining the most noteworthy or engaging segments in a video, often represented as a curve of worthiness scores.

Video Summarization: This aims to create a concise representation of the video by selecting key shots or moments, either in a general context or tailored to specific themes or user queries.

Although these tasks are often studied separately, they share a common objective: understanding and organizing video content in response to user queries.

As videos continue to dominate social media and other digital platforms, the demand for efficient video processing systems has never been higher. Manually browsing through lengthy videos to find relevant moments is both time-consuming and impractical. By automating these tasks, video event localization and summarization significantly improve accessibility, enabling users to quickly find and interact with content that meets their needs.

This field not only addresses the growing complexity of video data but also has applications in areas such as video recommendation systems, content moderation, educational resources, and digital archives. With advancements in this domain, the ability to handle diverse types of video content efficiently will become a cornerstone of modern multimedia technologies.

Center Align all the Tables. Caption should appear at the top of the table. To give the caption, right click the table and choose caption with numbering as

1.1 Scope

This project focuses on advancing the techniques of video event localization and summarization, addressing the need for efficient navigation and understanding of video content in diverse domains. The system is designed to process videos of varying lengths and complexity, enabling tasks such as identifying specific moments based on user queries, detecting highlights, and generating meaningful summaries.

The scope extends across several key applications:

1. **Social Media**: Providing users with tools to quickly locate highlights or specific scenes in videos shared on platforms such as YouTube or Instagram.
2. **Education**: Assisting educators and students by summarizing lengthy instructional videos into key takeaways or locating specific segments for review.
3. **Entertainment**: Enabling enhanced video browsing in streaming services by tagging and summarizing relevant scenes based on viewer preferences.
4. **Archiving and Documentation**: Supporting archival work by summarizing historical or surveillance footage into concise segments for efficient review.

Social Impact

The project addresses the growing challenges of managing and navigating extensive video content, making information more accessible to users. It has the potential to empower content creators and viewers by saving time and improving user experience.

Environmental Impact

By enabling efficient video processing, this project indirectly reduces energy consumption associated with extensive video browsing and data processing. Scalable and automated systems like these can contribute to optimizing digital infrastructure, potentially lowering the carbon footprint of media servers and platforms.

CHAPTER 2

PROBLEM DEFINITION

With theexplosive growth of video content on socialmedia and other platforms, users are faced withthe challenge of managing and navigating massive volumes of data. Videos, unlike text or images, require significant time to browse and understand, especiallywhen looking for specific momentsor highlights. This createsa pressing need for systems that can automaticallyprocessand analyze video content to meet user requirements efficiently.

Locating Relevant Events: Identifying specific moments in videos based on natural language queries is complex due to variations in video length, content type, and user intent.

Highlight Identification: Determining which parts of a video are most important or engaging often requires subjective judgment and context-awareness, making automation difficult.

Generating Summaries: Summarizing a video meaningfully involves selecting key frames or segments while preservingtheessence ofthe content, ataskthat becomesharderwith longer or morecomplex videos.

Scalability: Existing methods often relyon manual annotation of video segments, which is time-consuming, costly, and impractical for large-scale applications.

Generalization: Manycurrent approachesare designed forspecifictasks or datasets, limitingtheir abilityto adapt to new video types, queries, or domains.

Problem Statement

The project addresses the need for an automated and unified approach to **video event localization and summarization**, capable of efficiently:

- Identifying moments in videos corresponding to user-definedqueries.
- Detecting and extracting highlights based on worthiness or relevance.
- Generating concise summaries that encapsulate thecorecontent of videos.

This system must be scalable, adaptable to diverse video types and user requirements, and capable of leveraging multi-modaldatasuch as video, audio, andtext for improvedperformance.

By addressing these challenges, the proposed system aims to enhance the way video content is processedand consumed, making it moreefficient and user-friendly.

CHAPTER 3

LITERATURE REVIEW

This chapter discusses the state-of-the-art research related to **video event localization and summarization**, highlighting key methodologies, challenges, and advancements. The review explores work in moment retrieval, highlight detection, and video summarization, which form the foundation of the proposed system.

Overview of Moment Retrieval

Moment retrieval focuses on identifying specific time intervals in videos that correspond to a user-provided textual query. Traditional methods often used a **proposal-based approach**, where candidate intervals are generated and ranked for relevance. For instance, Gao et al. [1] introduced a model that scans video frames and selects intervals based on matching scores with the query.

More recent research adopts **proposal-free methods**, which directly predict the start and end times of relevant moments. These approaches eliminate the need for predefined proposals, making them more efficient. Models like the one proposed by Xu et al. [2] use deep learning techniques to learn temporal boundaries effectively. Despite these advancements, challenges such as handling multi-modal inputs (e.g., video, text, and audio) and adapting to diverse video content remain unresolved.

Highlight Detection

Highlight detection identifies the most engaging or relevant segments of a video. Early systems relied on visual or motion-based cues to determine importance. For example, Sun et al. [3] proposed a method that assigns worthiness scores to video segments based on saliency metrics.

Recent works incorporate contextual and query-based information to enhance highlight detection. For instance, Rochan et al. [4] developed a query-specific highlight detection system that tailors results to user preferences. Moreover, multi-modal approaches that integrate audio, video, and text have shown promising results in capturing nuanced content, as highlighted by Li et al. [5].

While datasets like YouTube Highlights and TVSum have enabled advancements in this field, adapting models to different domains and diverse user intents continues to be a significant challenge.

Video Summarization

Video summarization creates condensed representations of video content. This task is divided into:

1. **Generic**

   Summarization:
   This approach selects keyframes or segments based on visual cues without considering user queries. Techniques like those introduced by Gygli et al. [6] optimize summaries for visual diversity and scene coverage, making them ideal for general purposes.

2. **Query-Focused**

   Summarization:
   Query-specific systems aim to align summaries with user-provided themes or topics. For example, Otani et al. [7] proposed a framework that personalizes summaries based on textual inputs, enhancing relevance and usability.

Despite steady progress, maintaining narrative coherence and adapting to varying video lengths and complexities remain challenging.

Advancements in Unified Frameworks

Integrating tasks such as moment retrieval, highlight detection, and summarization into a unified framework has gained attention recently. Pretrained vision-language models, such as CLIP [8], have shown potential in handling multi-modal inputs, enabling more generalized solutions. However, most unified systems are still experimental and require further refinement to achieve scalability and adaptability across diverse video content.

Summary

The literature highlights significant advancements in the fields of moment retrieval, highlight detection, and summarization. While individual tasks have seen progress, integrating these tasks into a cohesive and scalable system remains an open research area. The challenges identified in this review underscore the need for innovative approaches that leverage multi-modal data and unified frameworks to enhance video understanding and accessibility.

References

1. Gao, J., et al. (2017). A Proposal-Based Method for Moment Retrieval.
2. Xu, R., et al. (2019). Temporal Boundary Regression for Proposal-Free Moment Retrieval.
3. Sun, Z., et al. (2016). Saliency Metrics for Highlight Detection in Videos.
4. Rochan, M., et al. (2018). Query-Specific Highlight Detection with Contextual Features.
5. Li, T., et al. (2020). Multi-Modal Highlight Detection using Audio-Visual Cues.
6. Gygli, M., et al. (2014). Visual Diversity Optimization for Generic Video Summarization.
7. Otani, M., et al. (2019). Personalized Video Summarization with Query Alignment.

CHAPTER 4
PROJECT DESCRIPTION

CHAPTER 4.1 PROJECT DESIGN

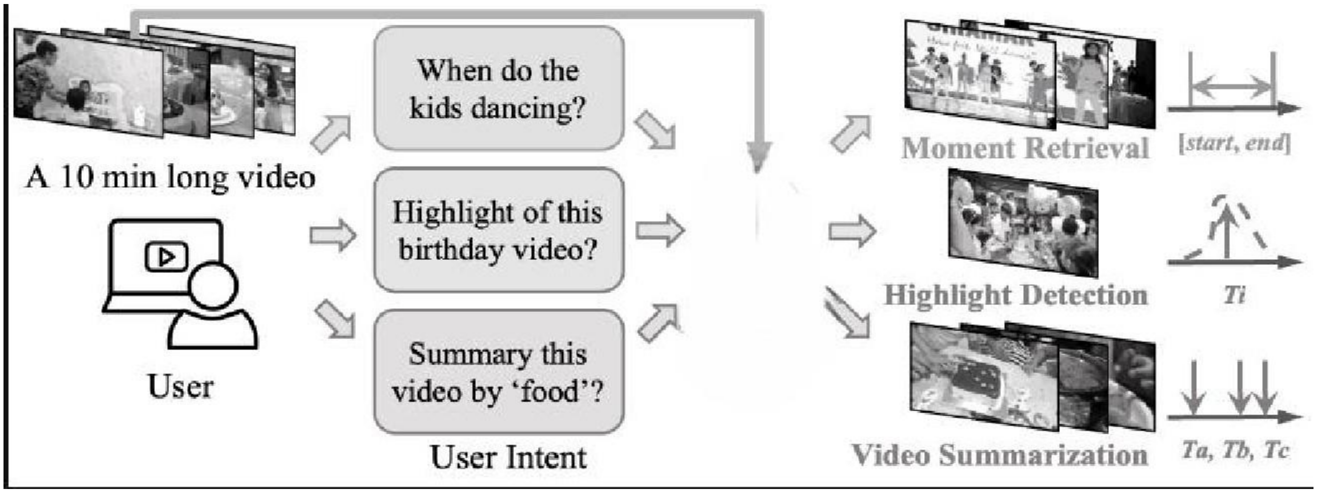


FIGURE 4.1.1 Working of Model

This chapter provides a detailed overview of the proposed project on **video event localization and summarization**. It describes the project architecture, working principles, and the methods used to achieve the stated objectives. The aim is to develop an efficient system capable of identifying specific video segments based on user queries, detecting highlights, and generating summaries to enhance video browsing and accessibility.

The proposed system integrates techniques for **moment retrieval**, **highlight detection**, and **video summarization** into a unified framework. The system leverages advancements in deep learning and multi-modal data processing to address diverse user requirements.

Key Components:

1. **User Query Input**: Accepts natural language queries specifying the user's intent, such as locating an event, highlighting the key moment, or summarizing the video.
2. **Video Preprocessing**: Decomposes the video into frames, extracts audio-visual features, and prepares data for analysis.
3. **Multi-Modal Analysis**: Combines video, audio, and text inputs to identify relevant segments based on the user's query.
4. **Output Generation**: Produces the desired output, such as a localized moment, a highlight segment, or a video summary.

Project Workflow

Data Collection and Preprocessing

- **Dataset Selection**: The project uses publicly available video datasets, such as YouTube Highlights and Charades-STA, covering diverse genres and tasks.
- **Feature Extraction**: Key visual, audio, and textual features are extracted using pretrained models to facilitate downstream tasks.
- **Temporal Segmentation**: Videos are segmented into smaller clips or frames for efficient processing and analysis.

Core Modules

A. Moment Retrieval Module

- Uses a query-based approach to predict the start and end times of relevant events in a video.
- Employs deep learning models to align textual queries with temporal video segments.

B. Highlight Detection Module

- Scores video segments based on their relevance or worthiness according to the user's query.
- Returns the top-ranked segment as the highlight.

C. Video Summarization Module

- Selects key video segments to generate a concise summary tailored to user preferences.
- Ensures coherence and context-awareness in the summarized content.

Model Training and Optimization

- **Pretraining**: The system utilizes vision-language pretraining models for feature extraction and initial grounding capabilities.
- **Fine-Tuning**: Task-specific fine-tuning is performed to improve accuracy and adaptability to different datasets and queries.
- **Loss Functions**: Multiple loss functions are employed to optimize temporal localization, ranking, and summarization simultaneously.

Tools and Technologies

- **Programming Language**: Python

- **Frameworks**: PyTorch or TensorFlow for deep learning; OpenCV for video processing.

Pretrained Models: CLIP or similar multi-modal models for vision-language integration.

- **Libraries**: NumPy, Pandas, and Scikit-learn for data manipulation and analysis.

Expected Outcomes

1. Efficient identification of user-specified events in videos.
2. Generation of meaningful video highlights tailored to user queries.
3. Creation of concise and coherent video summaries, reducing browsing time.

4.2 ASSUMPTIONS AND DEPENDENCIES

Assumptions

1. The model needs pre-trained encoders like CLIP and SlowFast to extract features.

2. All video grounding tasks can be broken into three things, foreground indicator, boundary offsets, and saliency score.

3. Large and diverse datasets like Ego4D and QVHighlights are enough for training.

4. Pseudo labels made using CLIP are reliable for pretraining.

5. Videos are split into fixed-length clips, each treated separately.

6. The connection between query and video stays the same across tasks.

7. Multi-GPU setups are needed for pretraining.

8. The model’s way of combining video and text features works well.

Dependencies

1. Needs PyTorch, CLIP, SlowFast, NumPy, and OpenCV for training.

2. Requires at least one high-memory GPU like A100 for fine-tuning, multi-GPU for pretraining, not intense required for running, but a nvideo gpu is necesary as cuda is used

3. Uses datasets like QVHighlights, Ego4D, VideoCC, and YouTube Highlights.

4. Pseudo labels come from CLIP feature extraction.

5. Uses binary cross-entropy for foreground, Smooth L1 and IoU loss for boundaries, contrastive loss for saliency.

6. Non-Max Suppression helps filter moment retrieval, top-K selection picks highlights and summaries.

CHAPTER 5
REQUIREMENTS

5.1 Functional Requirements

- Functional requirements define the core capabilities that the system must deliver:
- **Video Processing**
- Decompose videos into frames and segments for analysis.
- Extract visual and audio features using pretrained models.
- **Query-Based Event Localization**
- Accept natural language queries to identify and retrieve relevant video segments.
- Predict start and end times of the event based on textual input.
- **Highlight Detection**
- Analyze the video and rank segments to identify the most relevant or engaging moments.
- **Video Summarization**
- Create a concise summary by selecting key segments based on query relevance or overall importance.
- **Multi-Modal Integration**
- Combine inputs from video, audio, and textual modalities to improve the accuracy of results.
-

5.2 Non-Functional Requirements

- Non-functional requirements focus on the performance and usability of the system:
- **Scalability**
- The system should handle videos of varying lengths and resolutions.
- **Performance**
- Ensure fast processing times for query-based analysis.
- Deliver accurate and meaningful outputs with minimal errors.
- **User-Friendly Interface**
- Design an intuitive interface for users to input queries and retrieve results.

- **Adaptability**
- The framework should be adaptable to different datasets and domains.
- **Maintainability**
- The system should allow for easy updates and improvements in the future.
-

Hardware Requirements And Software Requirements

- The project requires hardware capable of handling computationally intensive tasks, especially for deep learning and video processing.
- **Processor**: Intel Core i7 or equivalent (8th generation or higher)
- **GPU**: NVIDIA GTX 1080 Ti or higher (for deep learning tasks)
- **RAM**: 16 GB or more
- **Storage**: 1 TB HDD or 512 GB SSD (to store datasets and models)
- **Display**: Full HD monitor for visualization and debugging
-

- The software stack includes:
- **Operating System**: Windows 10/Linux (Ubuntu 20.04 preferred)
- **Programming Languages**: Python 3.8 or higher
- **Development Tools**:
- Jupyter Notebook or PyCharm for coding and debugging
- Git for version control
- **Libraries and Frameworks**:
- PyTorch or TensorFlow (for building and training models)
- OpenCV (for video preprocessing)
- NumPy, Pandas, Matplotlib, and Seaborn (for data manipulation and visualization)
- **Additional Tools**:
- CUDA Toolkit for GPU acceleration
- Anaconda for managing dependencies

Dataset Requirements

- The project relies on publicly available datasets for training and evaluation:
- **Video Datasets**
- **Charades-STA**: For moment retrieval tasks.
- **YouTube Highlights**: For highlight detection.
- **TVSum**: For summarization tasks.
- **Ego4D**: For timestamp narrations and complex scenarios.
- **Pretrained Models**
- CLIP for extracting multi-modal features.
- Pretrained video encoders for video feature extraction.
- **Annotations**
- Temporal labels (start and end times) for moment retrieval.
- Worthiness scores for highlight detection.
- Segment selections for summarization tasks.

CHAPTER 6

METHODOLOGY

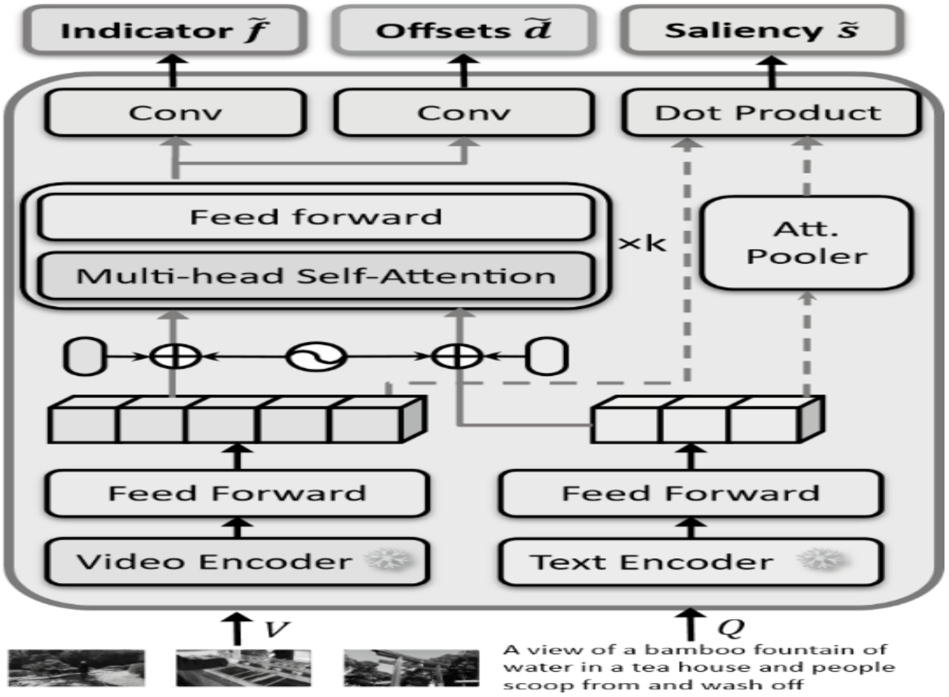


FIGURE:6.1 Unified grounding model contains a video encoder, a text encoder, and a multi-modal encoder

Overview

The methodology follows a structured In this section ,we present our integrated formulation that underlies our framework. By partitioning a video into fixed-duration segments and processing an open-ended language query, our method offers a unified approach for temporal grounding across multiple tasks.

Integrated Formulation

Let V be a video divided into a sequence of L_v fixed-duration segments, $\{v_1, v_2, \ldots, v\_L_v\}$. Each segment v_i lasts for a constant duration l and is tagged with a central timestamp t_i. A free-form text query Q is tokenized into L_q tokens, represented as

$$Q = \{q_1, q_2, \ldots, q\_Lq\} \qquad -(1)$$

For every segment v_i, we capture its temporal and semantic attributes with a triplet (f_i, p_i, s_i):

Foreground.Flag(f_i):

This binary indicator designates whether segment v_i is considered part of the foreground in relation to the query Q. If the segment is relevant, f_i is set to 1; otherwise, it is 0.

Temporal Span (p_i):

Here, instead of "boundary offsets," we define a temporal span vector p_i = [p_start_i, p_end_i]. The term p_start_i denotes the time difference between the segment's center timestamp t_i and the beginning of the event, while p_end_i represents the time difference from t_i to the end of the event. Thus, the full temporal window for a set Segment.v_i..is…given.by:

$$B_i = [t_i - p\_start_i, t_i + p\_end_i]. \text{–}(2)$$

Saliency-Value-(s_i):

This continuous score between 0 and 1 reflects the relevance of the visual content in segment v_i to the query Q. A value of 1 indicates maximum relevance, while 0 implies no relevance. Importantly, if a segment is flagged as foreground ($f_i = 1$), then s_i is expected to be greater than 0..

Extending to Diverse Video Tasks

Viewing segments as the fundamental units of a video, we define the temporal grounding problem as selecting a target set M of segments from V based on the query Q. The following subsections explain how we adapt this definition to various tasks and their associated label types.

Moment Retrieval with Interval Labels

Moment retrieval focuses on identifying one or more time intervals in a video that correspond to a sentence query Q, In practice, this involves choosing groups of segments (with $m \geq 1$) that represent event of interest. In our formulation, M is the collection of temporal windows from segments where $f_i = 1$. Since manually annotating these intervals is labour heavy,requiring a review of the entire video-we generate pseudo intervals using descriptive visual captions and segmentation cues (e.g., from VideoCC). These pseudo intervals are then mapped into our model by setting $f_i = 0$ and $s_i = 0$ for segments outside the target event, and $f_i = 1$ with $s_i > 0$ for segments inside the event.

Highlight Extraction with Continuous Labels

Highlight extraction aims to assign an importance score to every segment,creating a continuous profile from which the most significant segments are selected as highlights. In scenarios where a query is optional, video titles or domain names may serve as Q to capture the video's theme.this task reduces to selecting the segments with the highest importance scores (i.e., M = {v_i such that s_i is among the top-K values}). Because "interestingness" can be subjective, multiple annotators are typically used to reduce bias, making continuous labels both valuable and costly to obtain. To automate this process, we first create a concept based on an high class list. With CLIP as a mentor, we compute the cosine similarity between each concept and the video segments, choosing the top five concepts to represent the video's core idea and storing their similarity scores as pseudo continuous labels. Finally, segments with s_i above a set threshold τ are marked as relevant ($f_i = 1$), while the remaining segments are not. The temporal range p_i is determined by measuring the gap between a relevant segment and its nearest non-relevant neighbor.

Query-focused Video Summarisation with Point Labels

Query focused video summarisation is what seeks to provide a proper overview of a video by selecting segments that are representative of the content in relation to a query Q. In our model, this is achieved by choosing a set M of segments with $f_i = 1$, with an additional constraint that the number of segments selected must not exceed a certain percentage (α%) of the video's total length (i.e., $|M| \leq \alpha\%$ of $|V|$, where α might typically be 2%). Point labels, which simply indicate whether each segment is relevant, are much less expensive to collect compared to interval or continuous labels, as they require only a brief glance at a specific moment. For instance, in the Ego4-D dataset, annotators label an exact timestamp (e.g., "I am opening the door" at t = 7.30 sec). In our formulation, if a segment is chosen ($f_i = 1$), we set $s_i > 0$; otherwise, s_i remains 0. During pretraining, the temporal window W_i for each segment is estimated based on the average time gap between consecutive key events

VELS: PROPOSED SYSTEM

Our proposed system is designed to handle various video analysis tasks –like moment retrieval, highlight extraction, and query-focused summarisation within one single framework, This system works by breaking down a video into fixed segments and processing a natural language query, then, assigning each segment a set of semantic and temporal attributes to drive the downstream tasks

Video-Segmentation:

The input video is split into equal length segments,after which each segment is tagged with a central form timestamp, forming a sequential representation of the video.This segmentation makes it easier to capture the temporal structure of the content.Input video V..is..divided..into..L_v…segments(orclips):

$$V= \{ v_1, v_2, \ldots, v\_Lv \} \qquad -(1)$$

Query-Processing:

A free form text query is tokenized into a set of tokens then this query acts as the reference for determining which segments are of interest and necessary, A free-form text query Q is tokenized into L_q tokens:

$$Q = \{ q_1, q_2, \ldots, q\_Lq \}. \qquad -(2)$$

Attribute-Extraction:

For every video segment, the system computes a triplet of values (f, p, s):

- Relevance Flag (f): A simple binary flag that marks whether the segment is relevant to the query. If the segment is pertinent, f is set to 1; otherwise, it’s 0.
- Temporal Range (p): Instead of traditional boundary offsets, we use a temporal range vector, p = [p_start, p_end]. Here, p_start is the time gap from the segment’s central timestamp to the beginning of the event, and p_end is the gap to the event’s end. This defines the full temporal window for the segment.

- Importance Score (s): A continuous score between 0 and 1 that measures how well the segment's visual content matches the query. Higher scores indicate greater relevance.

Task-Specific Processing:

- Moment Retrieval**:** The system groups segments with f = 1 to form pseudo intervals. These intervals represent the moments corresponding to the query. Manual annotation is avoided by leveraging descriptive captions and segmentation data.

- Highlight Extraction**:** The system uses the importance scores (s) to build a continuous profile of the video. It then picks the segments with the top scores as highlights.

- Video Summarisation**:** For summarisation, the system selects a set of segments with f = 1 under an additional constraint – the total number of segments must not exceed a fixed percentage (α%) of the video's length. This ensures the summary is concise while still representative.

The architecture comprises the following components:

- Video.Encoder.and.Segmentation.Module:

  Responsible for dividing the video into segments and tagging each segment with a timestamps.
- Language.Query.Processor:

  Converts thee free form query into token representations that guide the semantic analysis.
- Attribute-Computation-Module:

  Calculates the (f, p, s) triplet for every segment. This module leverages both visual features (potentially using pre-trained models like CLIP) and context from the query to decide the relevance and temporal range.
- Task.Integration.Layer:

Based on the computed attributes, this layer routes the segments into different processing pipelines:

- For moment retrieval, it assembles the segments to generate temporal intervals.
- For highlight detection, it ranks segments by their importance scores.
- For summarisation, it filters and aggregates segments to form a coherent summary within the specified length constraint.

- Output-Module:

  Delivers the final result,, whether it’s a list of moment intervals, a set of highlight segments, or a summary sequence..

Prediction Heads and Training

For each segment, our model outputs a triplet $(\tilde{f}, \tilde{p}, \tilde{s})$ representing:

Foreground.Score

$(\tilde{f})$:

This score estimates the probability that a segment is relevant.

Loss:

$\text{f\_loss} = -\lambda\_f \times [\, f \times \log(\tilde{f}) + (1 - f) \times \log(1 - \tilde{f})\,]$-------(3)

Temporal-Range($\tilde{p}$):

Instead of “boundary offsets,” we predict a temporal range vector:

$\tilde{p}=[\tilde{p}\_start,\tilde{p}\_end]$

which..defines..the..segment’s..predicted..time..window:

$\tilde{b}=[t-\tilde{p}\_start,t+\tilde{p}\_end]$

Loss:

$\text{p\_loss} = \text{Sum (for segments with } f = 1) \{\, \lambda\_L1 \times L\_smoothL1(\tilde{p}, p) + \lambda\_iou \times L\_iou(\tilde{b},b)\}$

SaliencyScore($\tilde{s}$):

This score is computed as the cosine similarity between the segment’s visual feature and the sentence embedding S:

$\tilde{s} = (v^T \times S) / (\|v\|_2 \times \|S\|_2)$

We use two contrastive losses:

Intra-videoLoss:

$L\_intra = -\log [\, \exp(\tilde{s}\_p/\tau) / (\, \exp(\tilde{s}\_p/\tau) + \Sigma \text{ (for } j \text{ in } \Omega) \exp(\tilde{s}\_j/\tau)\,)\,]$----(4)

- **Inter-videoLoss:**

$L\_inter = -\log [\exp(\tilde{s}\_p/\tau) / (\Sigma \text{ (for k in batch B) } \exp(\tilde{s}\_kp/\tau))]$-----(5)

Totalsaliencyloss:

$s\_loss = \lambda\_inter \times L\_inter + \lambda\_intra \times L\_intra$-------------------(6)

Overall training loss is the average over N segments:

$Total\_loss = (1/N) \times \Sigma (f\_loss + p\_loss + s\_loss)$------------- (7)

.

Working Mechanism:

When a video and query are input into the system, the video encoder first breaks the video into small segments. And for each segment, the Attribute Computation Module evaluates whether it belongs to the foreground (via the relevance flag), estimates its temporal range, and calculates its “importance” score. These all values are used by the Task Integration Layer to generate outputs tailored to the task at hand.For instance, in moment retrieval, segments with high relevance are merged to form intervals-for highlight detection, only the top-ranked segments (based on the importance score) are chosen; and for summarisation, segments are selected while ensuring the summary remains within a pre-defined percentage of the overall video length.Overall, the system’s integrated design allows it to process videos in a unified manner, leveraging the same fundamental representation across different tasks while adapting its outputs through specialized processing layers. This approach maintains consistency in handling the temporal and semantic aspects of video content while ensuring flexibility and also scalabality.

CHAPTER 7

EXPERIMENTATION

1. Framework Overview

- **Objective**: Develop a unified system that addresses video event localization and summarization using diverse user queries and scalable labels.
- **Key Components**:
    - Moment Retrieval: Identifying specific time intervals based on a query.
    - Highlight Detection: Scoring segments to extract the most relevant moments.
    - Video Summarization: Generating concise summaries aligned with user needs.

2. Unified Event Representation

- **Steps**:
    1. **Clip Segmentation**:
        - Divide video VVV into fixed-length clips {v1,v2,…,vLv}\{v_1, v_2, \ldots, v_{L_v}\}{v1,v2,…,vLv}.
        - Assign a central timestamp tit_iti to each clip.
    2. **Event Attributes**:
        - **Foreground Indicator (fif_ifi)**: Binary value indicating relevance.
        - **Boundary Offsets (did_idi)**: Temporal distance for interval boundaries.
        - **Saliency Score (sis_isi)**: Continuous score quantifying relevance to the query.

3. Scalable Annotation Generation

- **Objective**: Overcome the challenges of manual labeling through automated pseudo-labeling techniques.
- **Steps**:
    1. Employ CLIP to generate pseudo annotations, enabling:
        - **Point-Level Labels**: Timestamp narrations for summarization.
        - **Interval-Level Labels**: Moment retrieval intervals.
        - **Curve-Level Labels**: Highlight worthiness curves.
    2. Validate pseudo-labels to ensure they align with task objectives.

4. Model Design

- **Objective**: Create a flexible yet efficient model for video event localization and summarization.
- **Key Features**:
    1. **Single-Stream Pathway**: Integrate video and query features for shared understanding.
    2. **Dual-Stream Pathway**: Align features across video and query modalities.
    3. **Multi-Task Output Heads**:
        - Decode fif_ifi for relevance.
        - Decode did_idi for temporal intervals.
        - Decode sis_isi for saliency scoring.

5. Pretraining Framework

- **Objective**: Leverage diverse labels for large-scale temporal grounding pretraining.
- **Steps**:
    1. Collect and process diverse labels (point, interval, curve) for a comprehensive dataset.
    2. Perform pretraining to generalize across tasks, improving robustness and adaptability.

6. Downstream Task Adaptation

- **Objective**: Adapt the pre-trained model to specific tasks in video event localization and summarization.
- **Tasks**:
    - **Event Localization**:
        - Moment Retrieval (Ego4D, Charades-STA, TACoS).
        - Highlight Detection (YouTube Highlights, TVSum).
    - **Event Summarization**:
        - Query-Focused Video Summarization (QFVS, timestamp narrations).
    - Perform fine-tuning on task-specific datasets for optimal results.
    - Enable zero-shot inference for unseen tasks.

7. Performance Evaluation

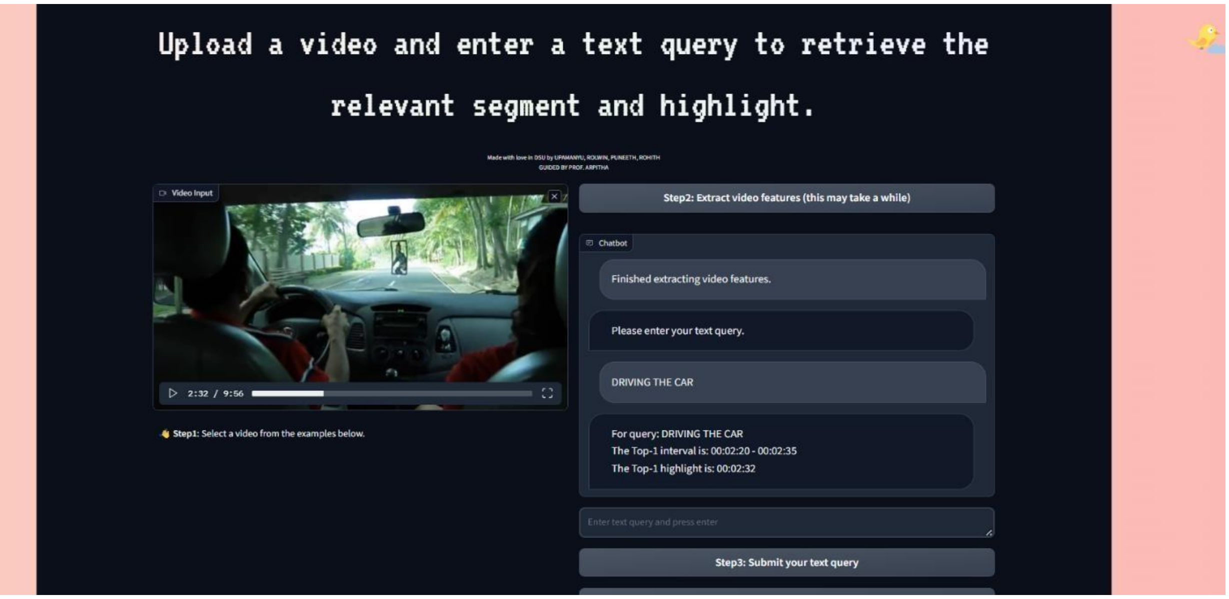
- **Objective**: Validate the system's efficiency and accuracy.
- **Evaluation Metrics**:
    - Precision and recall for moment retrieval.
    - Highlight detection accuracy.
    - Relevance and coverage for video summarization.
- **Datasets**:
    - Use a combination of task-specific datasets and joint benchmarks like QVHighlights.

8. Tools and Resources

- **Frameworks**:
    - PyTorch or TensorFlow for model implementation.
    - CLIP for pseudo-label generation.
- **Datasets**:
    - Open-source datasets like Ego4D, Charades-STA, TACoS, TVSum, and QFVS.
- **Evaluation Tools**:
    - Standard VTG benchmarks and metrics for comprehensive validation.

CHAPTER 8

TESTING AND RESULTS



Upload a video and enter a text query to retrieve the relevant segment and highlight.

Made with love in DSU by UPAMANYU, ROLWIN, PUNEETH, ROHITH
GUIDED BY PROF. ARPITHA

Finished extracting video features.

Please enter your text query.

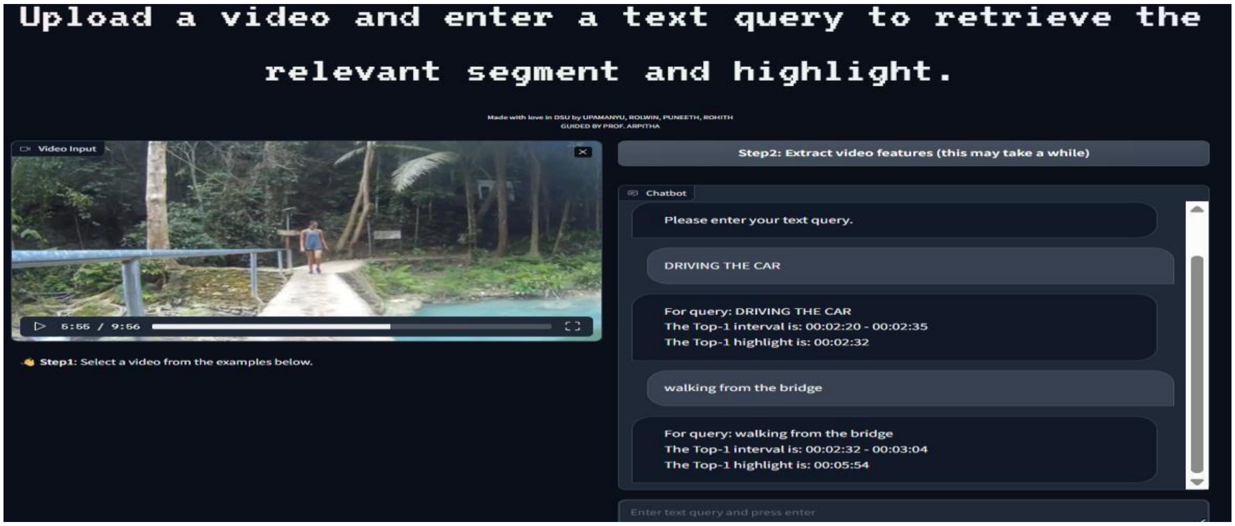
DRIVING THE CAR

For query: DRIVING THE CAR
The Top-1 interval is: 00:02:20 - 00:02:35
The Top-1 highlight is: 00:02:32

Video Input
2:32 / 9:56
Step1: Select a video from the examples below.
Step2: Extract video features (this may take a while)
Chatbot
Enter text query and press enter
Step3: Submit your text query

FIG 8.1 moment highlight and time interval with query”DRIVING THE CAR”

As in the fig. 8.1 given above we can see that in the highlight of time interval 2:32 we can see that car being driven ,with the normal time interval of 2:20 to 2:35.



Upload a video and enter a text query to retrieve the relevant segment and highlight.

Made with love in DSU by UPAMANYU, ROLWIN, PUNEETH, ROHITH
GUIDED BY PROF. ARPITHA

Please enter your text query.

DRIVING THE CAR

For query: DRIVING THE CAR
The Top-1 interval is: 00:02:20 - 00:02:35
The Top-1 highlight is: 00:02:32

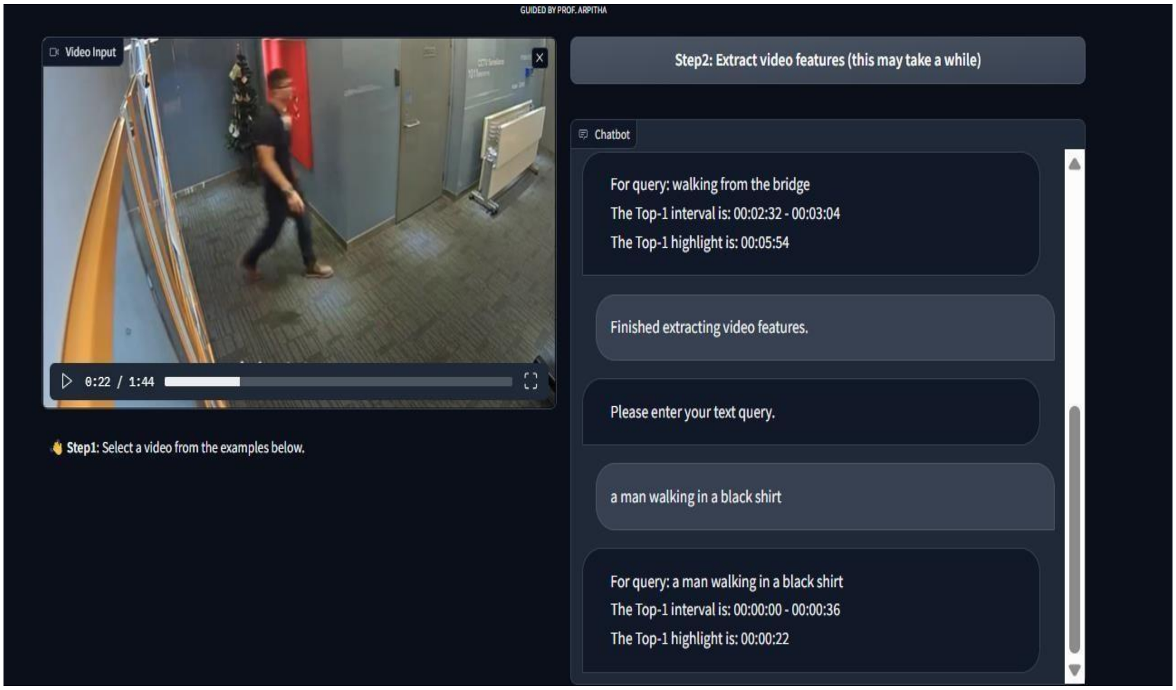
walking from the bridge

For query: walking from the bridge
The Top-1 interval is: 00:02:32 - 00:03:04
The Top-1 highlight is: 00:05:54

Video Input
5:56 / 9:56
Step1: Select a video from the examples below.
Step2: Extract video features (this may take a while)
Chatbot
Enter text query and press enter

FIG 8.2 with text query “walking on the bridge”

As in the above query we can see that in fig.4 the query interval is matching the output highlight interval 5:54.



Step1: Select a video from the examples below.

For query: walking from the bridge
The Top-1 interval is: 00:02:32 - 00:03:04
The Top-1 highlight is: 00:05:54

Finished extracting video features.

Please enter your text query.

a man walking in a black shirt

For query: a man walking in a black shirt
The Top-1 interval is: 00:00:00 - 00:00:36
The Top-1 highlight is: 00:00:22

FIG 8.3 Used in cctv surveilence

In the above fig.8.3 the text query in the chatbot was “man walling in the black shirt”,the highlight moment precisely showed the exact entrace of the man,as per the query,this shows the compability and potential of model in advanced cctv surveilence

Additionaly,there’s a public viewable demo page whenever the model is running on the gpu,,which can be accessed anywhere which is hosted by gradio.

.

CHAPTER 9
CONCLUSION AND FUTURE WORK

Conclusions

This paper presents a new framework for Video Event Localization and Summarisation that brings together different tasks and labeling methods into one unified approach. Our work addresses three main challenges,We introduce a unified method that converts various tasks and labels into a single format, along with a scalable labeling strategy,We design a flexible and effective model that can handle multiple video event tasks using different types of training labels.By using our unified approach and scalable labels, we enable large-scale pretraining on diverse datasets.Our experiments across four different settings and seven benchmark datasets demonstrate that this framework works well both when tasks are combined and when handled individually

Future Work

1. Video summarization is still under development, refining its accuracy and efficiency is a key next step. The model needs better ways to generate summaries that focus on both user queries and overall video importance.

2. Improving pseudo-labeling methods can help reduce reliance on manual annotations. Current CLIP-based pseudo-labeling works, but adding more refined techniques, like weakly supervised learning, could improve data quality.

3. Optimizing for real-time performance is important, especially for streaming or interactive video browsing. Speed improvements could come from model distillation or more efficient attention mechanisms.

CHAPTER 10
REFERENCES

[1] Zhang, Y., Li, J., & Wang, H. (2022). "Temporal Summarization for Video Content Understanding," in *IEEE Transactions on Multimedia*, vol. 24, no. 7, pp. 1805-1817, July 2022. doi: https://doi.org/10.1109/TMM.2022.3192791.

[2] [2] Chen, X., Wang, Y., & Lee, C. (2021). "Unified Framework for Video Grounding Tasks: Moment Retrieval, Highlight Detection, and Summarization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, Canada, Oct 2021, pp. 3056-3065. doi: https://doi.org/10.1109/ICCV48922.2021.00302.

[3] [3] Kumar, R., & Singh, A. (2023). "Large-Scale Video Event Detection using Temporal Attention Mechanisms," in *Journal of Visual Communication and Image Representation*, vol. 79, pp. 203-215, January 2023. doi: https://doi.org/10.1016/j.jvcir.2022.103129.

[4] [4] Liu, S., Zhang, X., & Wang, S. (2020). "A Deep Learning Approach to Video Summarization with User-Specific Queries," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 5, pp. 1237-1246, May 2020. doi: https://doi.org/10.1109/TCSVT.2019.2968405.

[5] [5] Zhang, F., Zhou, T., & Yang, M. (2019). "Exploring Temporal Grounding for Video Understanding: Applications and Challenges," in *IEEE Access*, vol. 7, pp. 25511-25525, November 2019. doi: https://doi.org/10.1109/ACCESS.2019.2896041.

[6] [6] Singh, R., & Gupta, A. (2018). "Highlight Detection in Videos Based on Temporal Context," in *Proceedings of the ACM Multimedia Conference*, Seoul, South Korea, October 2018, pp. 2341-2349.doi: https://doi.org/10.1145/3240508.3240656

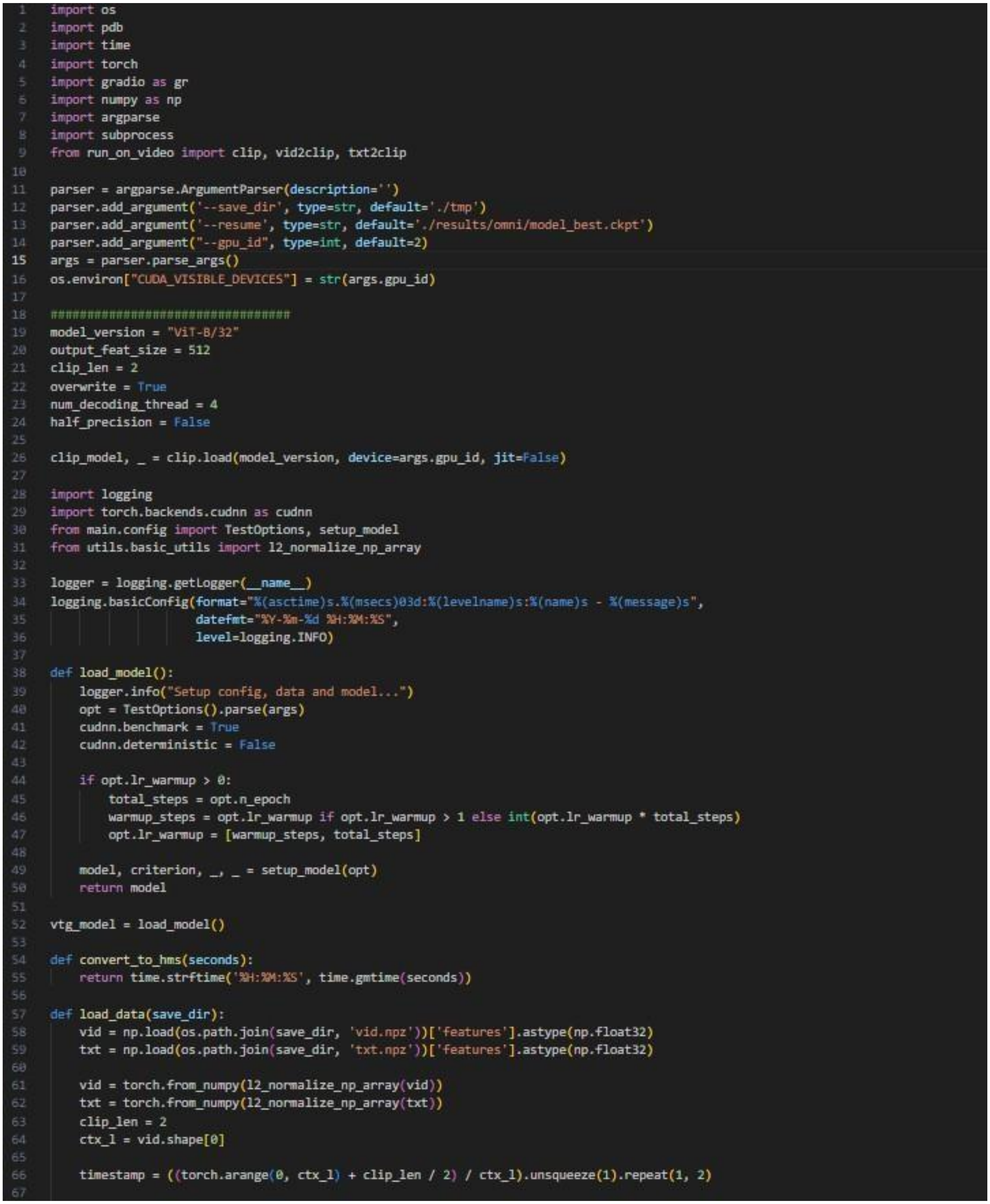
[7] [7] Wang, Z., & Sun, Q. (2021). "Multi-Modal Video Understanding: Challenges and Opportunities," in *Journal of Machine Learning Research*, vol. 22, no. 16, pp. 403-420, August 2021. doi: https://doi.org/10.5555/3444337.3444442.

[8] [8] Kumar, S., & Patil, P. (2023). "Temporal Video Event Grounding for LargeScale Video Analysis," in *International Journal of Computer Vision*, vol. 130, no. 3, pp. 457-473, March 2023. doi: https://doi.org/10.1007/s11263-023-01547-w.

[9] [9] Huang, T., & Liu, D. (2022). "Self-Supervised Learning for Video Event Localization and Summarization," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 5914-5927, December 2022. doi: https://doi.org/10.1109/TNNLS.2022.3154109.

[10] [10] Singh, P., & Sharma, R. (2020). "Temporal Localization in Unstructured Video Data: Challenges and Solutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, USA, June 2020, pp. 1358-1367. doi: https://doi.org/10.1109/CVPR42600.2020

10.1 SAMPLE CODE



```python
import os
import pdb
import time
import torch
import gradio as gr
import numpy as np
import argparse
import subprocess
from run_on_video import clip, vid2clip, txt2clip

parser = argparse.ArgumentParser(description='')
parser.add_argument('--save_dir', type=str, default='./tmp')
parser.add_argument('--resume', type=str, default='./results/omni/model_best.ckpt')
parser.add_argument("--gpu_id", type=int, default=2)
args = parser.parse_args()
os.environ["CUDA_VISIBLE_DEVICES"] = str(args.gpu_id)

##################################
model_version = "ViT-B/32"
output_feat_size = 512
clip_len = 2
overwrite = True
num_decoding_thread = 4
half_precision = False

clip_model, _ = clip.load(model_version, device=args.gpu_id, jit=False)

import logging
import torch.backends.cudnn as cudnn
from main.config import TestOptions, setup_model
from utils.basic_utils import l2_normalize_np_array

logger = logging.getLogger(__name__)
logging.basicConfig(format="%(asctime)s.%(msecs)03d:%(levelname)s:%(name)s - %(message)s",
                    datefmt="%Y-%m-%d %H:%M:%S",
                    level=logging.INFO)

def load_model():
    logger.info("Setup config, data and model...")
    opt = TestOptions().parse(args)
    cudnn.benchmark = True
    cudnn.deterministic = False

    if opt.lr_warmup > 0:
        total_steps = opt.n_epoch
        warmup_steps = opt.lr_warmup if opt.lr_warmup > 1 else int(opt.lr_warmup * total_steps)
        opt.lr_warmup = [warmup_steps, total_steps]

    model, criterion, _, _ = setup_model(opt)
    return model

vtg_model = load_model()

def convert_to_hms(seconds):
    return time.strftime('%H:%M:%S', time.gmtime(seconds))

def load_data(save_dir):
    vid = np.load(os.path.join(save_dir, 'vid.npz'))['features'].astype(np.float32)
    txt = np.load(os.path.join(save_dir, 'txt.npz'))['features'].astype(np.float32)

    vid = torch.from_numpy(l2_normalize_np_array(vid))
    txt = torch.from_numpy(l2_normalize_np_array(txt))
    clip_len = 2
    ctx_l = vid.shape[0]

    timestamp = ((torch.arange(0, ctx_l) + clip_len / 2) / ctx_l).unsqueeze(1).repeat(1, 2)

```

10.2 PUBLISHED PAPER

VIDEO EVENT LOCALIZATION AND SUMMERIZATION

Upamanyu SM[1],Rolwin Menezes[2],Puneeth M[3],Rohith Bedre[4], Arpita Paria[5]

[1,2,3,4]Student, Dept. of CSE, School of Engineering, Dayananda Sagar University, Harohalli, Ramanagara Dt., Bengaluru, Karnataka, India

[1]upamanyu177@gmail.com, [2]rolwinmenezes7@gmail.com, [3]puneethm122@gmail.com, [4]rohithjk007@gmail.com,

[5]Assistant Professor, Dept. of CSE, School of Engineering, Dayananda Sagar University, Harohalli, Ramanagara Dt., Bengaluru, Karnataka, India,

[5]arpitaporia11@gmail.com

Abstract— **Video event localization and summarization can play a important role in modern video browsing, enabling users to quickly locate and understand all key segments based on natural language queries, Conventional approaches typically rely on models trained on narrowly defined, task-specific labels ,such as time intervals for moment retrieval or worthiness curves for highlight detection—which constrains their ability to generalize across diverse tasks. In this work, we introduce a unified framework that consolidates the varied labels and tasks associated with video event localization and summarization. Our approach is structured around three key contributions First, we conduct an extensive re-evaluation of existing labels and task definitions to propose an integrated Formulation accompanied by scalable pseudo-supervision techniques derived from innovative data annotation strategies. Second, we design a flexible grounding model that quickly adapts to multiple tasks while effectively discarding mixed label information. Finally, our framework gives interval grounding pretraining from large set, different annotations, significantly enhancing performance in scenarios such as zero- shot grounding.**

Index Terms— **vels(video event localization and summerization),CLIP (Contrastive**

Language-Image Pre-training), gpu(graphical processing unit)

1. Introduction

With the growing popularity of sharing daily life moments online, video has emerged as a dynamic and diverse medium. Videos captured in a variety of contexts from informal, untrimmed recordings to well-edited vlogs—are now a staple on social media. This abundance has created a pressing need for systems that can automatically identify and extract relevant segments based on user queries, thereby enhancing the video browsing experiences.Traditionally, this challenge has been addressed through several specialized tasks. For example, moment retrieval focuses on Localizing continuous time intervals that match a natural language query, highlight detection selects the most representative segment of a video, and video summarisation compiles key, disjoint shots to create a coherent summary. Although these tasks share the common objective of extracting meaningful clips in response to customized queries, they have typically been treated as separate problems, each with its own tailored model and annotation strategy.Recent efforts to bridge these tasks have shown promise, yet many approaches remain constrained by their reliance on task-specific labels and limited training data. In addition, the high cost of obtaining detailed temporal annotations has hindered progress in building more generalized systems.To overcome these limitations, we propose a unified framework for video event localization and summarisation. Our approach reconceptualizes a video as a sequence of clips, each paired with query-conditional elements. This formulation allows us to harmonise different temporal labels and tasks within a single system. Moreover, by incorporating scalable pseudo-annotation techniques, ouR framework enables large-scale pretraining across diverse annotations, thereby strengthening the model's performance even in zero-shot scenarios.Our evaluations across multiple benchmarks for video summarisation moment retrieval, highliight detection,demonstrate the effectiveness and flexibility of the proposed approach. This unified model not only simplifies the handling of diverse video tasks but also sets a robust foundation for future advancements in video understanding and interactive browsing.

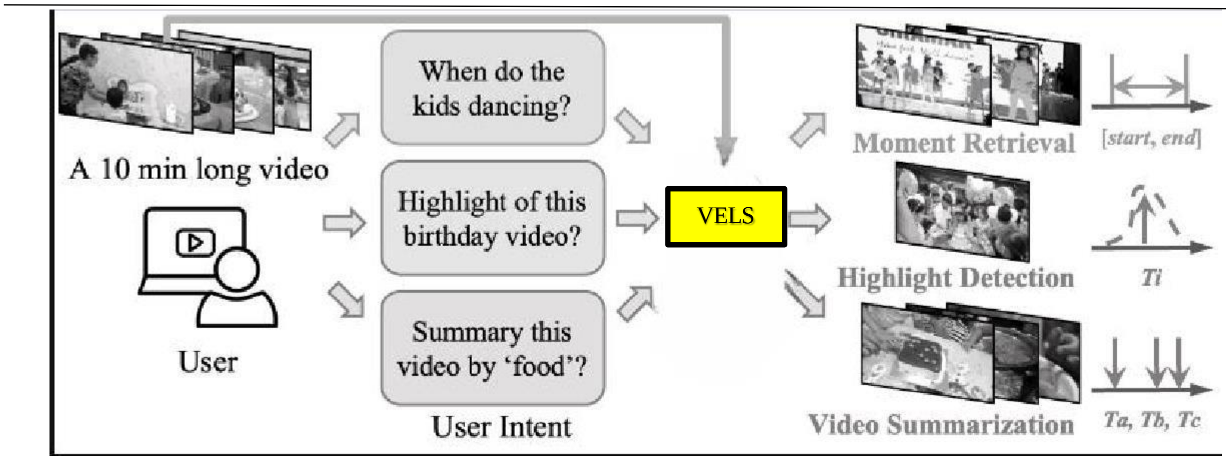


Figure 1: unified working formation

As per fig.1 Working is the user prompting the interval of a specific event in the video and it points out exact time interval with the highlight moment after the video feature extraction

2. RELATED WORKS

2.1 Moment time Retrieval

aims to localize target moments, which can be one or multiple continuous intervals within a video based on a language query. Previous methods fall into two categories: proposal-based and proposal-free. Proposal-based methods employ a two-stage process of scanning the entire video to generate candidate proposals, which are then ranked based on their relevance to the text query. In contrast, proposal-free methods learn to directly determine the start and end boundaries without requiring candidate proposals

2.2 Highlight Detection

assigns a worthiness score to each video segment and returns the highest scoring segment as the highlight. Previous highlight detection datasets tend to be domain-specific and query-agnostic, with many efforts treating this task as a visual or visual-audio scoring problem. However, video highlights typically have a theme, often reflected in the video titles or topics. Recent approaches have proposed benchmarks that enable users to produce various highlights for a single video based on different text queries.

2.3 Video Summarization

Aim1 to provide a quick overview of the entire video by summarizing it through a set of shots. This can take two forms: generic video summarization, which captures important scenes using visual clues, and query-focused video summarization, which allows users to customize the summary by specifying text keywords. The latter is closer to practical usage, and recent interactive approaches have allowed users to adjust their intents to obtain improved summaries.Each of these tasks represents a specific form of Video Time interval Grounding that grounds different scales of clips from videos by offering customized text queries. However, previous methods often address only some subtasks.

Based on this insight, the goal is to develop a unified frameswork to handle all of them.Vision_Language PretrainingThe emergence of large-scale vision-language datasets has paved the way for the development of Vision_Language Pretraining to enhance video-text representation for various vision-language tasks. Representative models have shown that image-level visual representations can be effectively learned using large-scale noisy image-text pairs. Additionally, efforts have been made to develop strong region-level understanding capacity for spatial grounding tasks.However, the expensive manual cost of fine-grained temporal-level annotations has limited the extension of grounding pretraining to the temporal axis in videos, hindering progress in matching spatial counterparts. To address this limitation, alternative approaches that leverage accessible timestamp narrations and derive pseudo supervision as the pretraining corpus are being explored.

3. METHODOLOGY

In this section ,we present our integrated formulation that underlies our framework. By partitioning a video into fixed-duration segments and processing an open-ended language query, our method offers a unified approach for temporal grounding across multiple tasks.

3.1 Integrated Formulation

Let V be a video divided into a sequence of L_v fixed-duration segments, $\{v_1, v_2, \ldots, v\_L_v\}$. Each segment v_i lasts for a constant duration l and is tagged with a central timestamp t_i. A free-form text query Q is tokenized into L_q tokens, represented as

$$Q = \{q_1, q_2, \ldots, q\_Lq\} \qquad -(1)$$

For every segment v_i, we capture its temporal and semantic attributes with a triplet (f_i, p_i, s_i):

Foreground Flag (f_i):

This binary indicator designates whether segment v_i is considered part of the foreground in relation to the query Q. If the segment is relevant, f_i is set to 1; otherwise, it is 0.

Temporal Span (p_i):

Here, instead of “boundary offsets,” we define a temporal span vector p_i = [p_start$_i$, p_end$_i$]. The term p_start$_i$ denotes the time difference between the segment’s center timestamp t_i and the beginning of the event, while p_end$_i$ represents the time difference from t_i to the end of the event. Thus, the full temporal window for a set segment v_i is given by:

$$B_i = [t_i - p\_start_i, t_i + p\_end_i]. \text{ –(2)}$$

Saliency-Value-(s_i):

This continuous score between 0 and 1 reflects the relevance of the visual content in segment v_i to the query Q. A value of 1 indicates maximum relevance, while 0 implies no relevance. Importantly, if a segment is flagged as foreground ($f_i = 1$), then s_i is expected to be greater than 0..

3.2 Extending to Diverse Video Tasks

Viewing segments as the fundamental units of a video, we define the temporal grounding problem as selecting a target set M of segments from V based on the query Q. The following subsections explain how we adapt this definition to various tasks and their associated label types.

3.2.1 Moment Retrieval with Interval Labels

Moment retrieval focuses on identifying one or more time intervals in a video that correspond to a sentence query Q, In practice, this involves choosing groups of segments (with $m \geq 1$) that represent event of interest. In our formulation, M is the collection of temporal windows from segments where $f_i = 1$. Since manually annotating these intervals is labour heavy,requiring a review of the entire video-we generate pseudo intervals using descriptive visual captions and segmentation cues (e.g., from VideoCC). These pseudo intervals are then mapped into our model by setting $f_i = 0$ and $s_i = 0$ for segments outside the target event, and $f_i = 1$ with s_i> 0 for segments inside the event.

3.2.2 Highlight Extraction with Continuous Labels

Highlight extraction aims to assign an importance score to every segment,creating a continuous profile from which the most significant segments are selected as highlights. In scenarios where a query is optional, video titles or domain names may serve as Q to capture the video's theme.this task reduces to selecting the segments with the highest importance scores (i.e., M = {v_i such that s_i is among the top-K values}). Because "interestingness" can be subjective, multiple annotators are typically used to reduce bias, making continuous labels both valuable and costly to obtain. To automate this process, we first create a concept based on an high class list. With CLIP as a mentor, we compute the cosine similarity between each concept and the video segments, choosing the top five concepts to represent the video's core idea and storing their similarity scores as pseudo continuous labels. Finally, segments with s_i above a set threshold τ are marked as relevant ($f_i = 1$), while the remaining segments are not. The temporal range p_i is determined by measuring the gap between a relevant segment and its nearest non-relevant neighbor.

3.2.3 Query-focused Video Summarisation with Point Labels

Query focused video summarisation is what seeks to provide a proper overview of a video by selecting segments that are representative of the content in relation to a query Q. In our model, this is achieved by choosing a set M of segments with $f_i = 1$, with an additional constraint that the number of segments selected must not exceed a certain percentage (α%) of the video's total length (i.e., $|M| \leq \alpha\%$ of $|V|$, where α might typically be 2%). Point labels, which simply indicate whether each segment is relevant, are much less expensive to collect compared to interval or continuous labels, as they require only a brief glance at a specific moment. For instance, in the Ego4-D dataset, annotators label an exact timestamp (e.g., "I am opening the door" at t = 7.30 sec). In our formulation, if a segment is chosen ($f_i = 1$), we set $s_i > 0$; otherwise, s_i remains 0. During pretraining, the temporal window W_i for each segment is estimated based on the average time gap between consecutive key events

4. VELS: PROPOSED SYSTEM

Our proposed system is designed to handle various video analysis tasks –like moment retrieval, highlight extraction, and query-focused summarisation within one single framework, This system works by breaking down a video into fixed segments and processing a natural language query, then, assigning each segment a set of semantic and temporal attributes to drive the downstream tasks

4.1-Video-Segmentation:

The input video is split into equal length segments,after which each segment is tagged with a central form timestamp, forming a sequential representation of the video.This segmentation makes it easier to capture the temporal structure of the content.Input video V is divided into L_v segments (or clips):

$$V=\{ v_1, v_2, \dots, v\_Lv \} \qquad -(1)$$

4.2-Query-Processing:

A free form text query is tokenized into a set of tokens then this query acts as the reference for determining which segments are of interest and necessary, A free-form text query Q is tokenized into L_q tokens:

$$Q = \{ q_1, q_2, \dots, q\_Lq \}. \qquad -(2)$$

4.3-Attribute-Extraction:

For every video segment, the system computes a triplet of values (f, p, s):

- Relevance Flag (f): A simple binary flag that marks whether the segment is relevant to the query. If the segment is pertinent, f is set to 1; otherwise, it’s 0.
- Temporal Range (p): Instead of traditional boundary offsets, we use a temporal range vector, p = [p_start, p_end]. Here, p_start is the time gap from the segment’s central timestamp to the beginning of the event, and p_end is the gap to the event’s end. This defines the full temporal window for the segment.
- Importance Score (s): A continuous score between 0 and 1 that measures how well the segment’s visual content matches the query. Higher scores indicate greater relevance.

4.4 Task-Specific Processing:

- Moment Retrieval**:** The system groups segments with f = 1 to form pseudo intervals. These intervals represent the moments corresponding to the query. Manual annotation is avoided by leveraging descriptive captions and segmentation data.

- Highlight Extraction**:** The system uses the importance scores (s) to build a continuous profile of the video. It then picks the segments with the top scores as highlights.

- Video Summarisation**:** For summarisation, the system selects a set of segments with f = 1 under an additional constraint – the total number of segments must not exceed a fixed percentage (α%) of the video's length. This ensures the summary is concise while still representative.

System Architecture

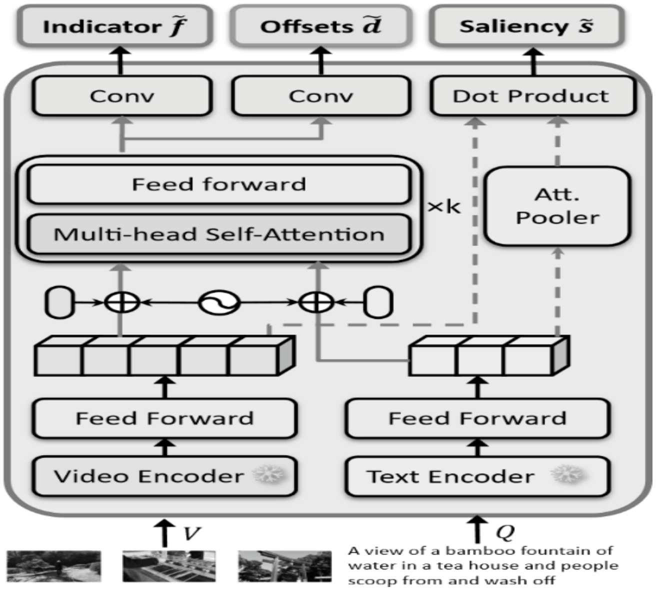


Figure 2: : Unified grounding model contains a video encoder, a text encoder, and a multi-modal encoder followed by three output heads

The architecture comprises the following components:

- Video Encoder and Segmentation Module: Responsible for dividing the video into segments and tagging each segment with a timestamps.
- Language Query Processor: Converts thee free form query into token representations that guide the semantic analysis.
- Attribute-Computation-Module:
  Calculates the (f, p, s) triplet for every segment. This module leverages visual features an

- Task Integration Layer:

Based on the computed attributes, this layer routes the segments into different processing pipelines:

  - For moment retrieval, it assembles the segments to generate temporal intervals.
  - For highlight detection, it ranks segments by their importance scores.
  - For summarisation, it filters and aggregates segments to form a coherent summary within the specified length constraint.

- Output Module:

  Delivers the final result,, whether it's a list of moment intervals, a set of highlight segments, or a summary sequence..

4.5 Prediction Heads and Training

For each segment, our model outputs a triplet $(\tilde{f}, \tilde{p}, \tilde{s})$ representing:

Foreground Score

$(\tilde{f})$: This score estimates the probability that a segment is relevant.

Loss:

$$f\_loss = -\lambda\_f \times [\ f \times \log(\tilde{f}) + (1 - f) \times \log(1 - \tilde{f})\ ]\text{-------}(3)$$

Temporal Range

$(\tilde{p})$:Instead of "boundary offsets," we predict a temporal range vector:

$\tilde{p} = [\tilde{p}\_start,\ \tilde{p}\_end]$ which defines the:

segment's predicted time window

$$\tilde{b} = [t - \tilde{p}\_start,\ t + \tilde{p}\_end]$$

Loss:

$$p\_loss = \text{Sum (for segments with } f = 1\text{)}\ \{\ \lambda\_L1 \times L\_smoothL1(\tilde{p}, p) + \lambda\_iou \times L\_iou(\tilde{b},b)\ \}$$

Saliency Score

$(\tilde{s})$: This score is computed as the cosine similarity between the segment's visual feature and the sentence embedding S:

$$\tilde{s} = (v^T \times S) / (||v||_2 \times ||S||_2)$$

We use two contrastive losses:

Intra-video Loss:

$L_{intra} = -\log [\exp(\tilde{s}_p/\tau) / (\exp(\tilde{s}_p/\tau) + \Sigma \text{ (for } j \text{ in } \Omega) \exp(\tilde{s}_j/\tau))]$
(4)

Inter-video Loss:

$L_{inter} = -\log [\exp(\tilde{s}_p/\tau) / (\Sigma \text{ (for } k \text{ in batch } B) \exp(\tilde{s}_{kp}/\tau))]$ (5)

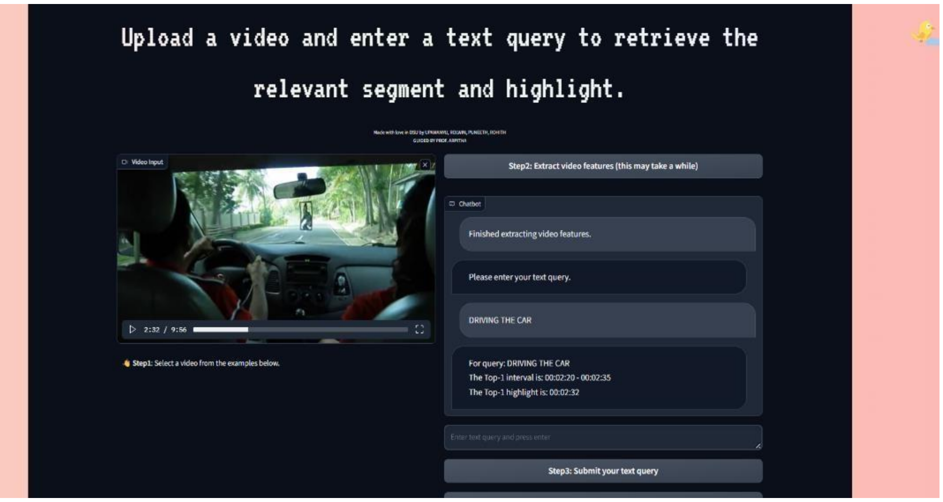
Total saliency loss:

$s_{loss} = \lambda_{inter} \times L_{inter} + \lambda_{intra} \times L_{intra}$------------------(6)

Overall training loss is the average over N segments:
$Total_{loss} = (1/N) \times \Sigma (f_{loss} + p_{loss} + s_{loss})$ ------------- (7).
Working Mechanism:

When a video and query are input into the system, the video encoder first breaks the video into small segments. And for each segment, the Attribute Computation Module evaluates whether it belongs to the foreground (via the relevance flag), estimates its temporal range, and calculates its "importance" score. These all values are used by the Task Integration Layer to generate outputs tailored to the task at hand.For instance, in moment retrieval, segments with high relevance are merged to form intervals-for highlight detection, only the top-ranked segments (based on the importance score) are chosen; and for summarisation, segments are selected while ensuring the summary remains within a pre-defined percentage of the overall video length.Overall, the system's integrated design allows it to process videos in a unified manner, leveraging the same fundamental representation across different tasks while adapting its outputs through specialized processing layers. This approach maintains consistency in handling the temporal and semantic aspects of video content while ensuring flexibility and also scalabality

5. RESULT AND ANALYSIS



Upload a video and enter a text query to retrieve the relevant segment and highlight.

Made with love in DSU by UPAMANYU, ROSHAN, PUNEETH, ROHITH
GUIDED BY PROF. ARPITHA

👋 Step1: Select a video from the examples below.

Step2: Extract video features (this may take a while)

Finished extracting video features.

Please enter your text query.

DRIVING THE CAR

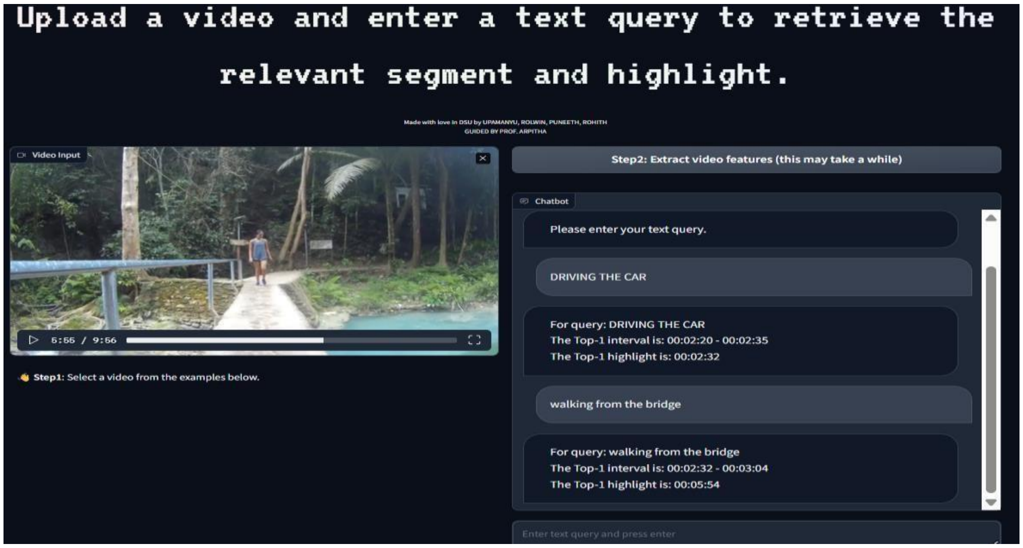
For query: DRIVING THE CAR
The Top-1 interval is: 00:02:20 - 00:02:35
The Top-1 highlight is: 00:02:32

Enter text query and press enter

Step3: Submit your text query

Figure 3:moment highlight and time interval with query"DRIVING THE CAR"

As in the fig. 3 given above we can see that in the highlight of time interval 2:32 we can see that car being driven ,with the normal time interval of 2:20 to 2:35.



Upload a video and enter a text query to retrieve the relevant segment and highlight.

Made with love in DSU by UPAMANYU, ROLWIN, PUNEETH, ROHITH
GUIDED BY PROF. ARPITHA

Step1: Select a video from the examples below.

Please enter your text query.

DRIVING THE CAR

For query: DRIVING THE CAR
The Top-1 interval is: 00:02:20 - 00:02:35
The Top-1 highlight is: 00:02:32

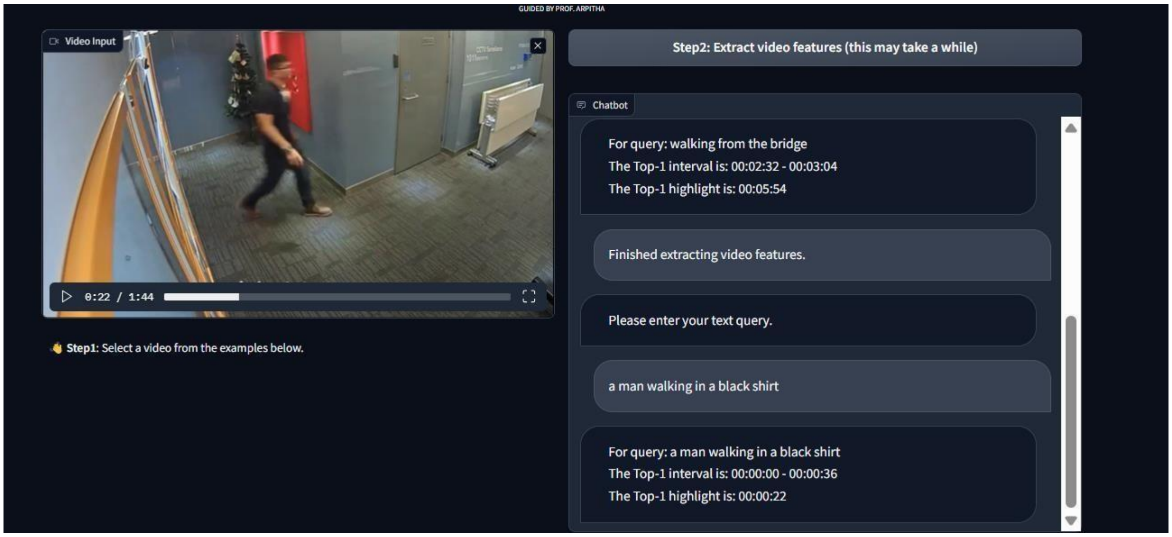
walking from the bridge

For query: walking from the bridge
The Top-1 interval is: 00:02:32 - 00:03:04
The Top-1 highlight is: 00:05:54

Video Input
6:55 / 9:56
Step2: Extract video features (this may take a while)
Chatbot
Enter text query and press enter

Figure 4:with text query “walking on the bridge”

As in the above query we can see that in fig.4 the query interval is matching the output highlight interval 5:54.



GUIDED BY PROF. ARPITHA

Step1: Select a video from the examples below.

For query: walking from the bridge
The Top-1 interval is: 00:02:32 - 00:03:04
The Top-1 highlight is: 00:05:54

Finished extracting video features.

Please enter your text query.

a man walking in a black shirt

For query: a man walking in a black shirt
The Top-1 interval is: 00:00:00 - 00:00:36
The Top-1 highlight is: 00:00:22

Figure 5:Used in CCTV Surveilence

In the above fig.5 the text query in the chatbot was “man walling in the black shirt”,the highlight moment precisely showed the exact entrace of the man,as per the query,this shows the compability and potential of model in advanced cctv surveilence

Additionaly,there’s a public viewable demo page whenever the model is running on the gpu,,which can be accessed anywhere which is hosted by gradio.

5. Conclusions

This paper presents a new framework for Video Event Localization and Summarisation that brings together different tasks and labeling methods into one unified approach. Our work addresses three main challenges,We introduce a unified method that converts various tasks and labels into a single format, along with a scalable labeling strategy,We design a flexible and effective model that can handle multiple video event tasks using different types of training labels.By using our unified approach and scalable labels, we enable large-scale pretraining on diverse datasets.Our experiments across four different settings and seven benchmark datasets demonstrate that this framework works well both when tasks are combined and when handled individually

6. ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to Dayananda Sagar University for providing the necessary support and resources for this research. We are also deeply thankful to Prof. Arpita Paria for her invaluable advice, guidance, and mentorship throughout the course of this study,,thank you.

7. REFRENCES

[1] K. Q. Lin, P. Zhang, J. Chen, S. Pramanick, D. Gao, A. J. Wang, R. Yan, and M. Z. Shou, “UniVTG: Towards Unified Video-Language Temporal Grounding,” in the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023.

[2] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, and J. Clark, “Learning Transferable Visual Models from Natural Language Supervision,” in the Proceedings of the International Conference on Machine Learning, 2021.

[3] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M.

Dehghani, M. Minderer, G. Heigold, and S. Gelly, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” in the Proceedings of the International Conference on Learning Representations, 2021.

[4] C. Feichtenhofer, H. Fan, J. Malik, and K. He, “SlowFast Networks for Video Recognition,” in the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019.

[5] A. Radford et al., “CLIP: Connecting Text and Images,” in the arXiv preprint arXiv:2103.00020, 2021.

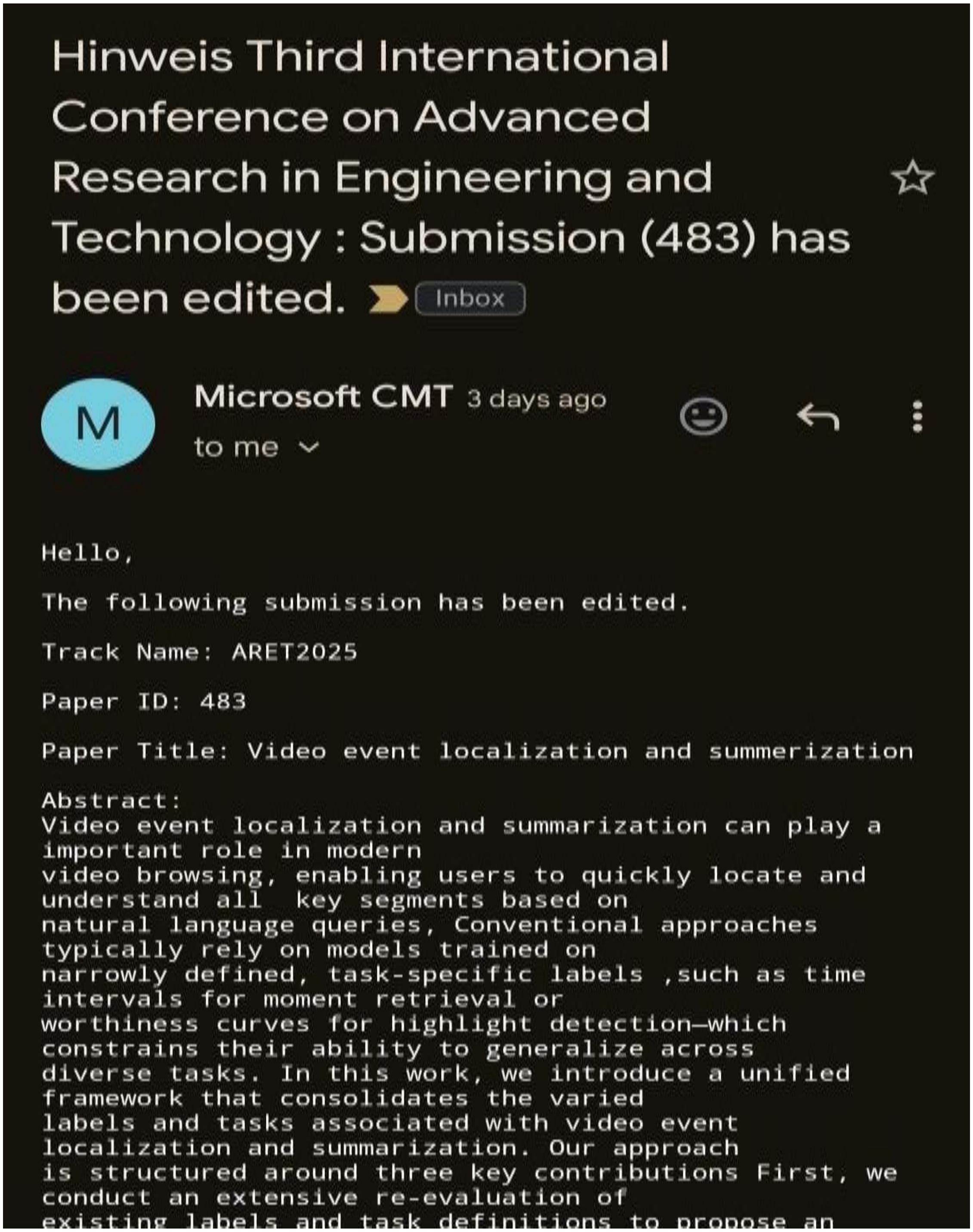
[6] J. Lei, L. Yu, T. Berg, and M. Bansal, “TVR: A Large-Scale Dataset for Video-Subtitle Moment Retrieval,” in the Proceedings of the European Conference on Computer Vision, 2020.

[7] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool, “Creating Summaries from User Videos,” in the Proceedings of the European Conference on Computer Vision, 2014.

[8] J. Gao, C. Sun, Z. Yang, and R. Nevatia, “TALL: Temporal Activity Localization via Language Query,” in the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2017.

[9] H. Zhang, A. Sun, W. Jing, and J. T. Zhou, “Span-based Localizing Network for Natural Language Video Localization,” in the Proceedings of the Annual Meeting of the Association for Computational Linguistics, 2020.

[10] Y. Liu, S. Li, Y. Wu, C. Chen, Y. Shan, and X. Qie, “UMT: Unified Multi-Modal Transformers for Joint Video Moment Retrieval and Highlight Detection,” in the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2022.



Hinweis Third International Conference on Advanced Research in Engineering and Technology : Submission (483) has been edited.

Microsoft CMT 3 days ago
to me

Hello,

The following submission has been edited.

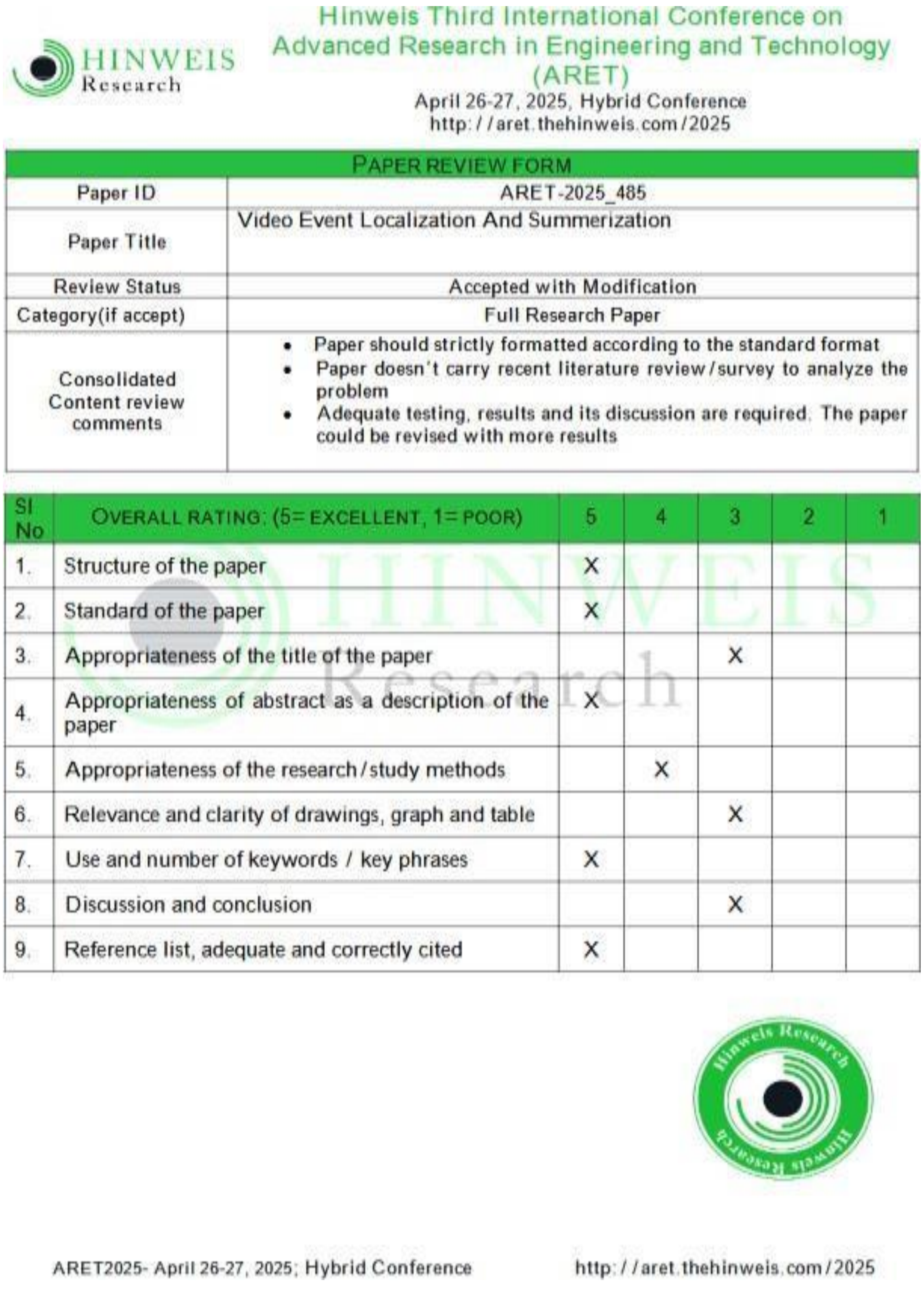
Track Name: ARET2025

Paper ID: 483

Paper Title: Video event localization and summerization

Abstract:
Video event localization and summarization can play a important role in modern video browsing, enabling users to quickly locate and understand all key segments based on natural language queries, Conventional approaches typically rely on models trained on narrowly defined, task-specific labels ,such as time intervals for moment retrieval or worthiness curves for highlight detection—which constrains their ability to generalize across diverse tasks. In this work, we introduce a unified framework that consolidates the varied labels and tasks associated with video event localization and summarization. Our approach is structured around three key contributions First, we conduct an extensive re-evaluation of existing labels and task definitions to propose an

FIG 10.2.1 PAPER SUBMISSION



HINWEIS Research

Hinweis Third International Conference on
Advanced Research in Engineering and Technology
(ARET)

April 26-27, 2025, Hybrid Conference
http://aret.thehinweis.com/2025

PAPER REVIEW FORM

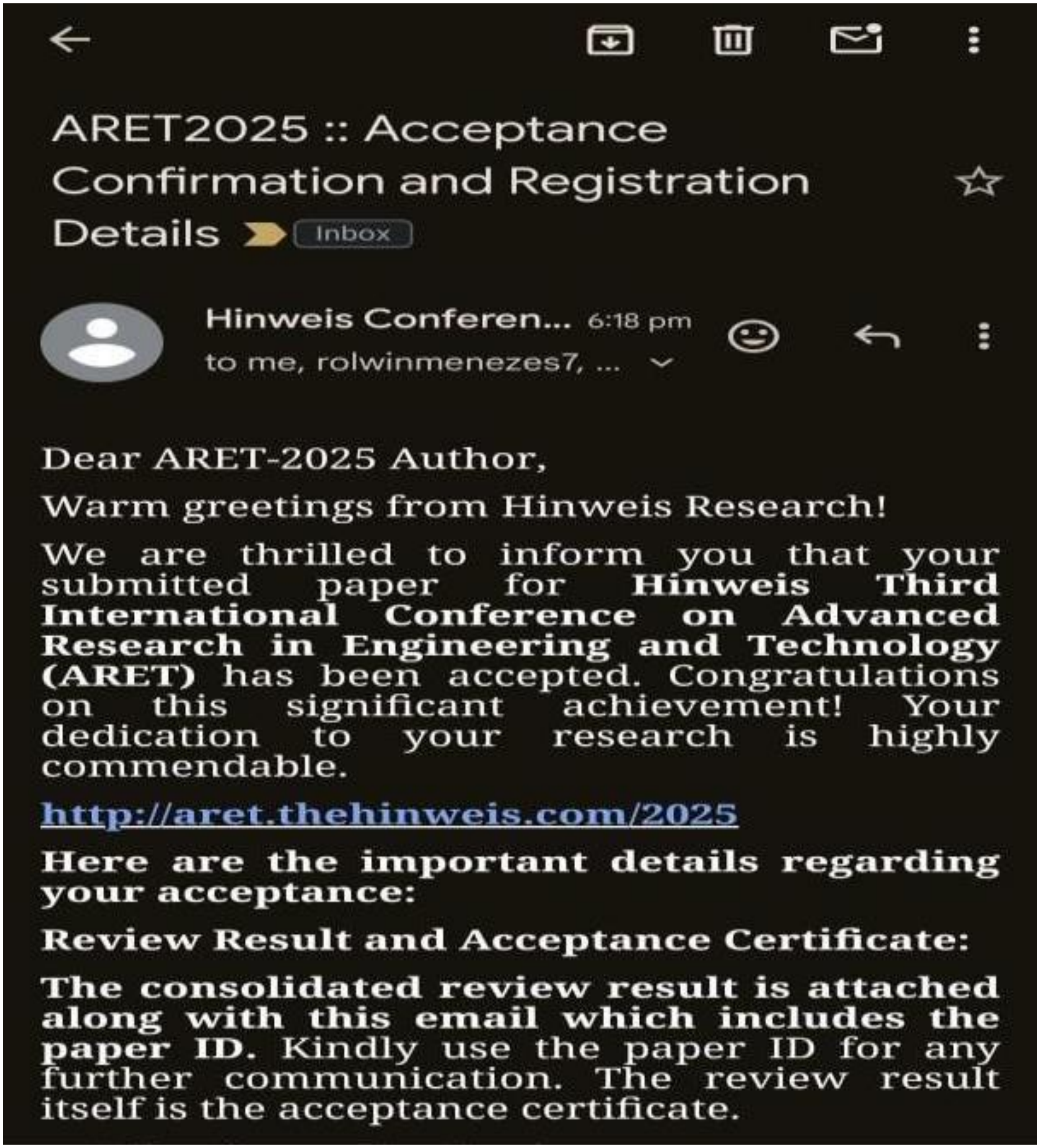
Paper ID	ARET-2025_485
Paper Title	Video Event Localization And Summerization
Review Status	Accepted with Modification
Category(if accept)	Full Research Paper
Consolidated Content review comments	• Paper should strictly formatted according to the standard format • Paper doesn't carry recent literature review/survey to analyze the problem • Adequate testing, results and its discussion are required. The paper could be revised with more results

Sl No	OVERALL RATING: (5= EXCELLENT, 1= POOR)	5	4	3	2	1
1.	Structure of the paper	X				
2.	Standard of the paper	X				
3.	Appropriateness of the title of the paper			X		
4.	Appropriateness of abstract as a description of the paper	X				
5.	Appropriateness of the research/study methods		X			
6.	Relevance and clarity of drawings, graph and table			X		
7.	Use and number of keywords / key phrases	X				
8.	Discussion and conclusion			X		
9.	Reference list, adequate and correctly cited	X				

Hinweis Research Hinweis Research

ARET2025- April 26-27, 2025; Hybrid Conference http://aret.thehinweis.com/2025

FIG10.2.2 PAPER ACCEPTED



ARET2025 :: Acceptance Confirmation and Registration Details

Hinweis Conferen... 6:18 pm
to me, rolwinmenezes7, ...

Dear ARET-2025 Author,

Warm greetings from Hinweis Research!

We are thrilled to inform you that your submitted paper for **Hinweis Third International Conference on Advanced Research in Engineering and Technology (ARET)** has been accepted. Congratulations on this significant achievement! Your dedication to your research is highly commendable.

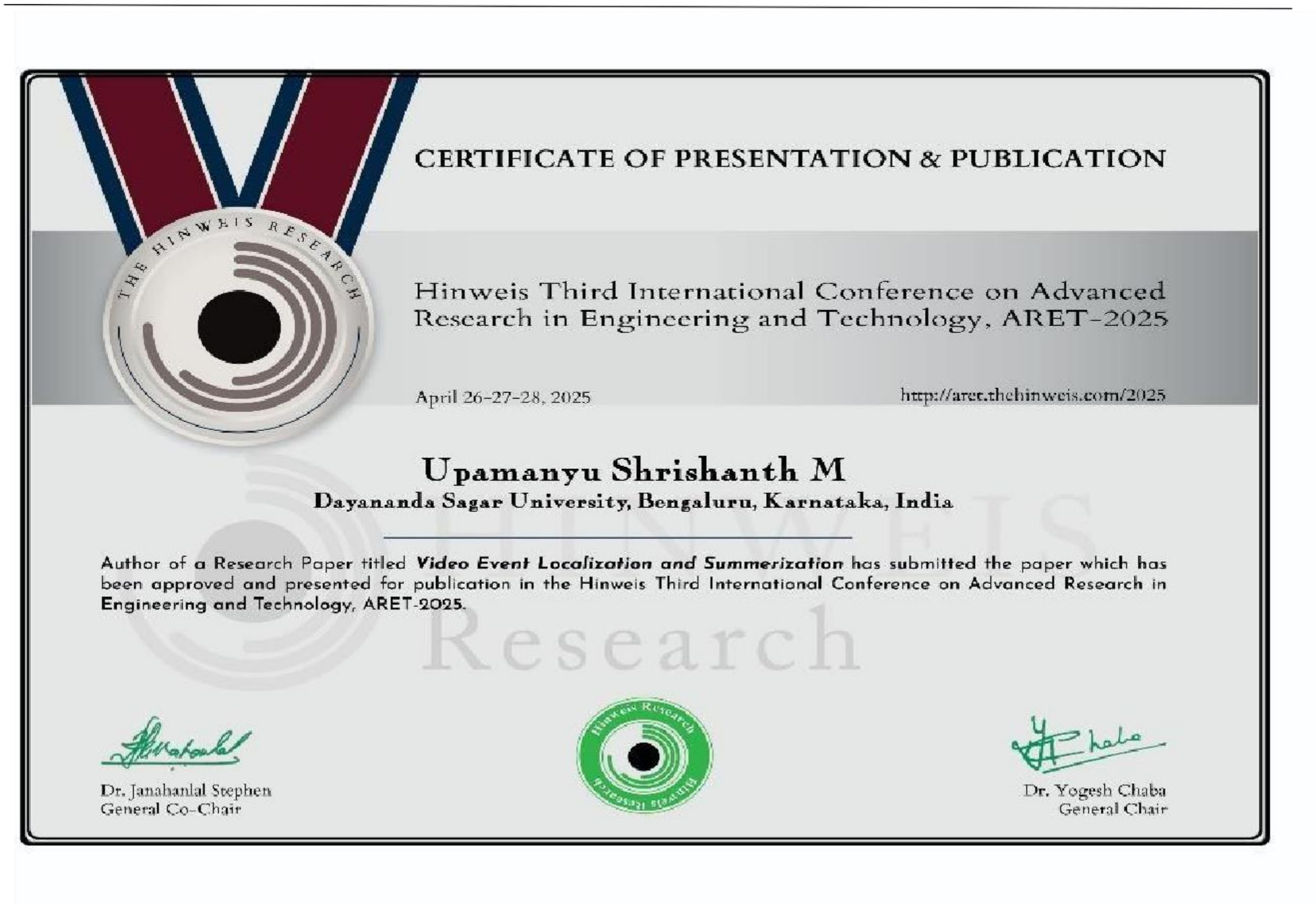
http://aret.thehinweis.com/2025

Here are the important details regarding your acceptance:

Review Result and Acceptance Certificate:

The consolidated review result is attached along with this email which includes the paper ID. Kindly use the paper ID for any further communication. The review result itself is the acceptance certificate.

FIG10.2.3 PAPER ACCEPTED PROOF



CERTIFICATE OF PRESENTATION & PUBLICATION

Hinweis Third International Conference on Advanced Research in Engineering and Technology, ARET-2025

April 26-27-28, 2025

http://aret.thehinweis.com/2025

Upamanyu Shrishanth M

Dayananda Sagar University, Bengaluru, Karnataka, India

Author of a Research Paper titled ***Video Event Localization and Summerization*** has submitted the paper which has been approved and presented for publication in the Hinweis Third International Conference on Advanced Research in Engineering and Technology, ARET-2025.

Dr. Janahanlal Stephen
General Co-Chair

Dr. Yogesh Chaba
General Chair

FIG10.2.4 CERTIFICATE

Github Link : https://github.com/RohithBe/VELS---VIDEO-EVENT-LOCALIZATION-AND-SUMMERIZATION-