AI Video Analytics System using Vision-Language Model for Precise Frame Identification in YouTube Videos

NG Kin Pak

1155143402@link.cuhk.edu.hk

LI Yinxi

1155160255@link.cuhk.edu.hk

GitHub Repository: https://github.com/PeterNg2333/CSCI4140-CLIP-Frame-Identifier

February 24, 2024

1. Background and Problem Statement

Efficiently locating and pinpointing specific moments within a sea of online video content or stock Footage is one of the significant challenges in today's digital age. Traditional solutions rely on human identification of video content or various video analysis tools attempt to categorize and summarize video content based on visual and audio features. Nevertheless, both approaches, which predefine the summarized description of the content, possess constraints. The use of Vision-Language Models, such as Contrastive Language-Image Pre-training (CLIP) model, could improve the searchability and accessibility of video content, making it easier for users to find exactly what they're looking for without relying solely on metadata. Therefore, the objective of this project is to

develop a video searching system capable of identifying specific frames or moments within YouTube videos that are relevant to a given textual query to offer efficient navigation and interaction between users and video content.

2. Project objectives

This project aims to develop a novel video search system that surpasses existing YouTube-like platforms by leveraging the capabilities of video analytics system using vision-language models (VLMs), specifically CLIP, to achieve the following:

i. **Enhanced Searchability and Accessibility:** Utilize CLIP's ability to understand visual information to enable users to search for specific moments within videos using natural language queries. This goes beyond traditional keyword-based search, allowing users to find content based on detailed descriptions, actions, or objects within the video.

ii. **To optimize system performance**: This involves implementing techniques to accelerate the response time of the system to user queries, such as employing sampling mechanisms to reduce the computational load and K-Means clustering algorithm to generate top-k results.

iii. **To create an intuitive user interface**: Design a user-friendly and intuitive web application that guides users through the search process effectively. The interface should clearly display search results, allowing users to navigate to the desired video frames or segments with ease and minimal cognitive load.

3. Novelty

The novelty of this project lies in its unique approach to video searching and frame identification. Unlike traditional methods that rely on predefined metadata or manual identification, our system leverages the power of Vision-Language Models to match video frames with textual descriptions in a real-time manner. Also, we will focus on enhancing the user experience of Vision-Language Models for precise frame identification in videos content.

4. Methodology

The project will focus on the following key areas:

i. Integration of Vision-Language Model (VLM): Leveraging the pre-trained model such as CLIP, the system will be designed to process text queries and retrieve the most relevant video frames. The integration will focus on ensuring that the model's inference capabilities are finely tuned for accuracy in matching frames with text descriptions.

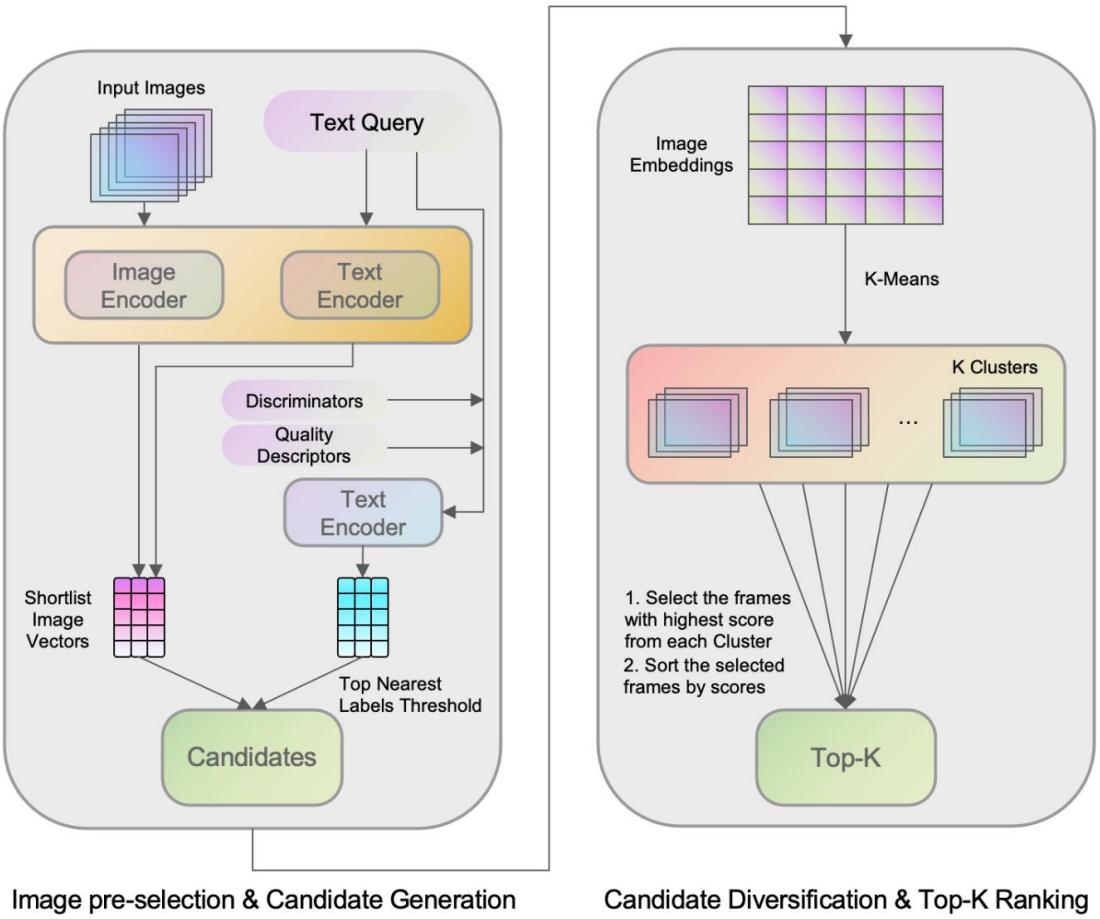


Figure 1: Vision-language Model Video Analytics System Overview

ii. System Development (Figure 1): Building the core system architecture that allows users to input text queries and receive corresponding top-k video frame outputs. This will involve developing an intuitive and responsive user interface (UI) and backend services that handle query processing, model interaction, and video handling efficiently.

iii. Performance Optimization: Implementing techniques to accelerate the response time of the system, such as optimizing the model inference pipeline, employing efficient indexing and retrieval mechanisms for video content and utilizing hardware acceleration where possible. The goal is to minimize latency to provide real-time or near-real-time

user experience.

iv. Usability Enhancements: Ensuring that the system is not only fast but also easy to use. This involves designing the UI to guide users effectively through the search process and present results in a way that makes it simple to navigate to the desired video frame.

5. Technical Challenge

The rapid expansion of the digital era has heralded a surge in video content, necessitating the evolution of sophisticated video analytics systems. These systems are essential for interpreting the vast arrays of visual data generated daily, with applications spanning from urban development to wildlife monitoring. Systems like VIVA[1], NoScope[2], BlazeIt[3], and others[4][5] allow users to query videos for objects, characters, scenes, or complex analysis by specifying a list of predicates that

---

[1] Daniel Kang, Francisco Romero, Peter Bailis, Christos Kozyrakis, and Matei Zaharia, "VIVA: An End-to-End System for Interactive Video Analytics," *12th Annual Conference on Innovative Data Systems Research (CIDR '22)*, January 9–12, 2022.

[2] Daniel Kang, John Emmons, Firas Abuzaid, Peter Bailis, and Matei Zaharia, "NoScope: optimizing neural network queries over video at scale," *Proceedings of the VLDB Endowment*, vol. 10, no. 11, pp. 1586-1597, Aug. 2017.

[3] Daniel Kang, Peter Bailis, and Matei Zaharia, "BlazeIt: Optimizing Declarative Aggregation and Limit Queries for Neural Network-Based Video Analytics," *Proceedings of the VLDB Endowment*, vol. 13, no. 4, pp. 533-546, Dec. 2019.

[4] Jiashen Cao, Karan Sarkar, Ramyad Hadidi, Joy Arulraj, and Hyesoon Kim, "FiGO: Fine-Grained Query Optimization in Video Analytics," *SIGMOD '22: Proceedings of the 2022 International Conference on Management of Data*, pp. 559-572, Jun. 2022.

[5] Zhuangdi Xu, Gaurav Kakker, Joy Arulraj, and Umakishore Ramachandran, "EVA: A Symbolic Approach to Accelerating Exploratory Video Analytics with Materialized Views," *SIGMOD '22: Proceedings of the 2022 International Conference on Management of Data*, pp. 602-616, Jun. 2022.

apply deep convolutional neural network models on video frames. As vision-language models (VLMs) such as CLIP[6] have made a series of breakthroughs on various recognition tasks using standard zero-shot classification procedures, existing frameworks, such as Zelda[7], have showcased the potential of using VLMs to leverage natural language for fast and accurate video analysis, but it also highlights the key limitations due to the architectural design in terms of processing speed and versatility when the system processes top-k results which are often of most interest to users.

Since vision-language models could compute very fast on processing videos, the limitations of current vision-language model systems are primarily due to their computational complexity to generate top-k results, which restricts the ability to handle queries swiftly and accurately. For instance, the Zelda system, despite its innovative use of the FAISS library developed by Meta[8] for calculating cosine similarities, experiences significant delays due to its algorithm design for diversifying results. These constraints are exacerbated in high-demand environments such as urban surveillance and wildlife conservation, where timely and precise data analysis is crucial.

6. Technical Impact

---

[6] Radford Alec et al., "Learning Transferable Visual Models from Natural Language Supervision," *International Conference on Machine Learning*, 2021.

[7] Francisco Romero , Caleb Winston, Johann Hauswal, Matei Zaharia, and Christos Kozyrakis, "Zelda: Video Analytics using Vision-Language Models," *arXiv preprint arXiv: 2305.03785 (2023)*. [Online]. Available: https://doi.org/10.48550/arXiv.2305.03785.

[8] Jeff Johnson, Matthijs Douze, and Hervé Jégou, "Billion-scale similarity search with gpus," *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2019.

This project's successful implementation is expected to have a significant impact on video content management and retrieval. By showcasing the effectiveness of VLMs in real-world applications, we aim to:

- Illustrate the Versatility of VLMs: Demonstrate the wide array of practical applications for VLMs, encouraging developers to adopt and adapt these models for innovative uses.
- Diversifying Results Technique Implementation: Enhance VLMs with a sophisticated clustering algorithm to improve diversity and relevance in top-k results, ensuring accurate and varied outcomes tailored to user intent in dynamic content scenarios.

- Propel Software Development: Provide a blueprint for integrating VLMs into software solutions, influencing future tools and platforms to leverage the full potential of vision and language understanding.

- Stimulate Research and Development: Inspire further research into optimization techniques for VLMs, potentially leading to breakthroughs that could benefit various domains such as online education, digital libraries, and content creation.

Through this project, we aim to not only advance the state of technology in video frame retrieval but also to provide a valuable reference for future research and commercial ventures in the domain of artificial intelligence and multimedia content processing.

7. Social Impact

By improving the searchability of video content, this technology can impact how information is consumed and shared with societies. There are several possible changes.

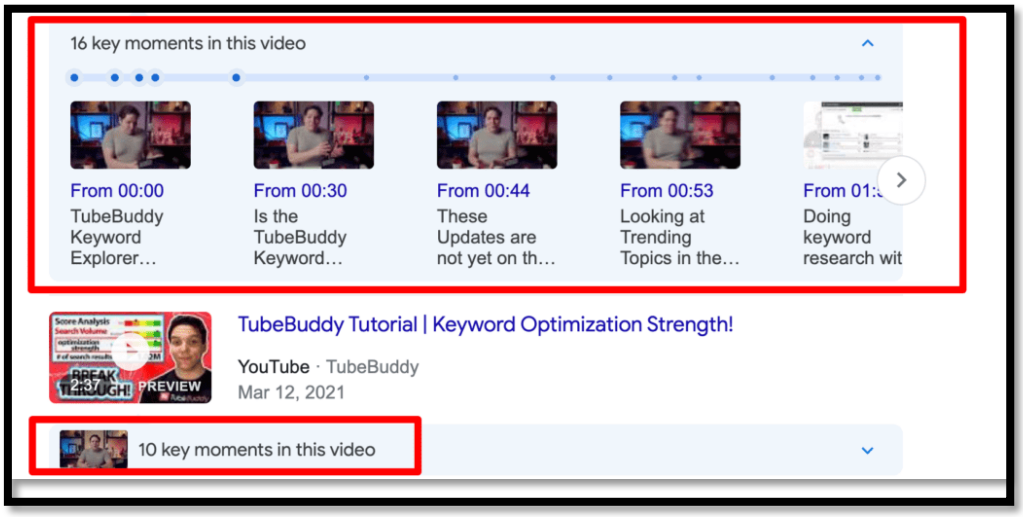
i. Increased efficiency and accuracy in accessing video-based information.
ii. Enhanced work productivity in research or jobs related to video.
iii. Facilitation of information sharing and dissemination.
iv. Crime investigation

8. Evaluation Plan

The evaluation of this project will be carried out in several phrase:

i. Video Collection and Pre-processing: We will check the Youtube link and check the format of the returned video file to see if the system retrieved the correct video.
ii. Latency Measurement: We will measure the response time from when a query is submitted to when the results are displayed to the user. Our objective is to ensure that the system achieves a response as fast as it can for a seamless user experience.
iii. Diversity and Coverage Testing: We will test whether the user gets the desired number of top-k results and whether the variance of these results is as expected.
iv. User-Specified Time Range Functionality: To provide a tailored user experience, we will imple- ment a feature allowing users to specify a time range for the video analysis. This feature will be particularly useful for processing videos beyond the identified maximum size threshold. The eval- uation will test the system's capability to accurately process queries within these user-defined time ranges, ensuring that performance is not compromised when users interact with this functionality.
v. Python server using Flask: We will develop a Python server using Flask to handle user queries and interact with the model.
vi. Userinterfaceaswebapplication:Ourwebapplicationwillemulatethematuredanduser-friendly interface of well-known platforms like YouTube. The search results will be displayed as

thumb- nails for each matching frame or video clip, with timestamps and relevance scores. Users will be able to click on a thumbnail to view the corresponding frame or clip, providing a user-centric and engaging interface. (For each query, the model could return continuous video clips that match the description.



a. Demo search results, retrieved from https://www.tubebuddy.com/wp-content/uploads/2022/06/video-chapter-snippet-1024x674.png

9. Project Solution

Component 1: Video Collection and Pre-processing

- **Target**: This component focuses on gathering a diverse set of YouTube videos and downloading them to the local storage.
- **Activities**:
    i. Uploading a YouTube Video link and using the Pytube to download the video with the user designed name into local storage for the next stage processing.
    ii. Once the downloading is finished, the system will automatically send the

video to the CLIP encoder.

- **Tools and Technologies: Pytube**

Component 2: Vision-Language Model (VLM) Development

- **Target**: Develop a system using VLM that accurately identifies top-k relevant video frames or clips matching user text queries, considering the possibility of multiple matches at different times.
- **Activities**:
    iii. Design and Architecture Setup: Establish the foundational structure of the VLM to handle complex video queries effectively. This involves setting up the initial frameworks and integrating key technologies like Meta FAISS, which aids in efficient similarity computations, and K-Means clustering to group results based on likeness.
    iv. Development of Query Processing Algorithms: Create algorithms that utilize prompt engineering to interpret user text queries and generate appropriate search parameters that guide the VLM in identifying relevant video clips or frames.
    v. Implementation of Ranking Mechanisms: Develop mechanisms within the VLM to rank the results based on their relevance to the user's query, ensuring that the top-k results are the most pertinent.
    vi. Testing and Optimization: Continuously test the VLM with various queries to ensure accuracy and efficiency. Optimize the system based on test results to handle diverse scenarios and improve the speed of result retrieval.

vii. Integration with Other Components: Ensure the VLM can interface effectively with other system components, particularly for data retrieval and user interaction management.

- **Tools and Technologies: Decord, Meta FAISS, prompt engineering, K-Means clustering**

Component 3: Python Server

- **Target**: Develop a Python server using Flask to handle user interactions, communicate with the VLM, and generate dynamic web pages.
- **Activities**:

viii. Implement a Flask API to receive user text queries through web forms or API requests. Process user queries and extract relevant information from API and Server File System

ix. Integrate the VLM with the Flask server through an API call. Send user queries to the VLM and retrieve the corresponding video segments or frames.

x. Utilize Flask's render_template function to generate HTML pages with search results. Display retrieved video segments or frames as thumbnails with timestamps and relevance scores.

- **Tools and Technologies: Flask framework**

Component 4: User Interface as Web Application

- **Target**: Develop a user-friendly web application with intuitive interface and functionalities

for video search and exploration.

- **Activities**:
    i. Main Page: Design a visually appealing and informative main page. Display recommended videos based on user preferences or trending content. Include a prominent search bar for users to enter their text queries.
    ii. Search result: Upon user query submission, retrieve search results from the server API. Display video thumbnails with timestamps and relevance scores for each matching segment or frame.
    iii. Video Player: Implement a customized video player with functionalities like play/pause, volume control, and a progress bar. Integrate an "in-video search" feature that allows users to search for specific frames within the currently playing video.
    iv. Video Editor: Provide a basic video editing functionality using JavaScript and the search API. Allow users to identify specific video frames through search and combine them into a new video clip.
- **Tools and Technologies:**
    i. HTML and CSS
    ii. Python (Flask) for Integration and modularized design (render template and include function)
    iii. Front-end framework (Bootstrap, Font Awesome)
    iv. JavaScript libraries (e.g., for video playback, video editing)

10. Challenges Encountered

- The time complexity of the previous diversification method for top-k result generation

proposed by “Zelda: Video analytics using vision-language models” was $O(n^2d)$ for candidate frames embedded into n vectors each of d-dimensional space.

- Our K-Means clustering algorithm for diversified top-k results generation was is dependent on the number of clusters k, the number of iterations t, the number of data points n, and the dimension of each data point d. For a small number of clusters and iterations, the time complexity can be approximated to $O(tknd)$. As k and t are generally relatively small or fixed in practical applications, the complexity can be further simplified to $O(nd)$.

11. Future works

- Expanding CLIP Use Cases and Scenarios:
  i. **Video Editing:** Develop features like "blooper detection" by identifying inconsistencies or unexpected events within video segments.
  ii. **Content Creation:** Integrate with video chat platforms to analyze video content in real-time and generate contextual responses or reactions.

- Multimodal Integration for Enhanced Searchability:
  i. **Combining CLIP with other AI Models**: Develop the ability to understand the overall context of the video, including the genre, storyline, and speaker intentions.
  ii. **Personalization:** Personalize search results based on user preferences and viewing history to provide a more tailored experience.

12. Sources

- **OpenAI CLIP: https://openai.com/clip/**
- **Faiss: https://github.com/facebookresearch/faiss**

- **HuggingFace: https://huggingface.co/**
- **Decord: https://github.com/dmlc/decord**
- **OpenCV: https://opencv.org/**
- **pytube: https://github.com/pytube/pytube**
- **Flask: https://flask.palletsprojects.com/en/2.0.x/**
- **React: https://reactjs.org/**

Project Contribution

Component	Task	Person in Charge	Share (%)
Frontend (Client-side)	UI Design & Implementation	Ng Kin Pak	25%
	Webpage Interaction (JavaScript Development)	Ng Kin Pak	10%
	Frontend & Backend Integration	Ng Kin Pak	15%
Backend (Server-side)	API Design & Implementation	Li Yinxi	15%
	Vision-Language Model System Development and Analytics System Performance Optimization	Li Yinxi	25%
	Video Collection and Pre-processing	Li Yinxi	10%
Total			*100%*