VClipper

Exploiting CLIP Zero-shot capabilities for moment retrieval in video recordings.

Universitat Oberta
de Catalunya

Oriol Caravaca Müller

U. Master in Data Science
Computer vision

Project supervisor
Ismael Benito Altamirano

Coordinating professor
Ismael Benito Altamirano

Date of submission
09/01/2024

Universitat Oberta de Catalunya
uoc.edu

SUMMARY OF THE FINAL PROJECT

Title of the project:	VClipper:Exploiting CLIP Zero-shot capabilities for moment retrieval in video recordings.
Author name:	Oriol Caravaca Müller
Project supervisor:	Ismael Benito Altamirano
Coordinating professor:	Ismael Benito Altamirano
Date of submission (MM/YYYY):	*09/01/2024*
Name of the degree:	U. Master in Data Science
Topic of the final project:	Computer vision
Language:	English
Keywords:	Video analysis, Moment retrieval, CLIP

Abstract

This research explores the integration of CLIP, a pretrained model, into video content analysis. In a landscape inundated with multimedia data, pinpointing specific moments within videos is a persistent challenge. By leveraging CLIP's semantic and visual search capabilities, this study endeavors to refine content retrieval methods. Emphasizing efficiency and applicability, this study aims to make this process more precise and practical. With this research we also reviewed the state-of-the-art methods and produced empirical analysis on the effects of postprocessing on the similarity vectors obtained from CLIP encoders. Finally, we developed two distinct methods aimed at moment retrieval tasks in audiovisual data, obtaining a model that is able to outperform previous works in Zero-shot moment revival, reaching 57.3 at R@1 IoU=0.5 and 51.6 at mAP@0.5.

Aquesta investigació explora la integració de CLIP, un model preentrenat, en l'anàlisi de contingut de vídeo. En un paisatge inundat de dades multimèdia, identificar moments específics dels vídeos és un repte persistent. Aprofitant les capacitats de cerca semàntica i visual de CLIP, aquest estudi intenta perfeccionar els mètodes de recuperació de contingut. Subratllant l'eficiència i l'aplicabilitat, fent aquest procés més precís i pràctic. En aquesta investigació també s'ha revisat l'estat de l'art i s'ha produït un anàlisis empíric sobre els efectes del postprocessament sobre els vectors de semblança obtinguts a partir dels codificadors de CLIP. Finalment s'han desenvolupat dos mètodes diferents dirigits a tasques de recuperació de moments en dades audiovisuals, obtenint un model que és capaç de superar els treballs anteriors en Zero-shot moment revival, arribant a 57,3 a R@1 IoU=0,5 i 51,6 a mAP@0,5.

Index

List of figures

1. Introduction

1.1. Context and motivation

With the increase in access to cameras and streaming services by the global population, there is increasingly more production, editing, and consumption of audiovisual formats. In this context, there is a growing demand for specific tools for the detection of those moments existing in a video that may be relevant to the user and that, in many cases, are hidden among the tide of information composed of the "superfluous" frames present in the video.

Although there are already some tools that can help in selecting relevant moments, normally the user, whether due to a technological or economic barrier, ends up spending considerable time manually scrolling through the video until the desired moment is detected. To help carry out this task and improve the characteristics of the existing tools, in this project we explore the potential of CLIP(Radford et al., 2021) with the objective of producing a model capable of performing both semantic and visual searches.

1.2. Goals

The primary objective of this project is to obtain a model that is capable of helping the users detect a moment in a video that fits the best description used as an input search. To do so, we aim to replicate, modify, and enhance the research presented in the paper by David Lin(D. C.-E. Lin et al., 2022), which utilizes CLIP as a base model to identify frames within a video that most closely resemble the input images. This project will explore different comparison methods to test the similarity between images, improving upon the original approach, in a similar manner, we will add a semantic search feature to allow users to search for frames not only using images but also using textual input.

In short, our goals with this project are:

- Primary goal: Implement a model based on CLIP that is able to detect highlight moments in a video based on an image or textual input search.

- Secondary goal: Improve on previous research by experiment with alternative methods to improve the precision and the performance of the implemented model

1.3. Sustainability, diversity, and ethical/social challenges

Since our primary goal is to aid professional and amateur video editors into finding the moments in a video, we acknowledge that the product of this project can be used to find individuals present on the video by some of their characteristics, like racial traits, gender etc. At the same time, the product of this project can also be used to detect certain behaviors performed by those individuals, but in any case, can't be used to identify any subject other than the very well-known public figures.

In terms of sustainability this project or its output should not pose a specific threat to nature or future generations, neither in terms of social, economic or environmental aspects. Although this work was developed with computational efficiency in mind, in general terms, exist some energetics concerns derived from high computational resources needed for video processing that may contribute to a large carbon footprint.

Given the zero-shot nature of this project most ethical concerns regarding diversity like using representative samples for the training, avoiding biases and taking into account cultural/racial sensitivity are directly linked to CLIP development, and to the best of our knowledge, have already been tackle by OpenAI development team. It is important to notice that for the development of this project a non-multilingual version of CLIP has been used, consequently the output of this project caters to an English-only audience and in turn the linguistic diversity has not been contemplated.

Finally, due to the nature of VLP's models, some concerns linked to ethical and social challenges might be raised, noticeably, copyright infringements and fair use, manipulation of the image of 3rd party entities like deepfakes and privacy concerns. Thankfully, the first two challenges listed above are associated with generative models and should not be a byproduct of the use of any outcome of this project. On the other hand, as stated in the before, privacy is the major concern that we can foresee for the output of this project, this is because the search feature of the model can aid malicious entities to identify individuals within the video.

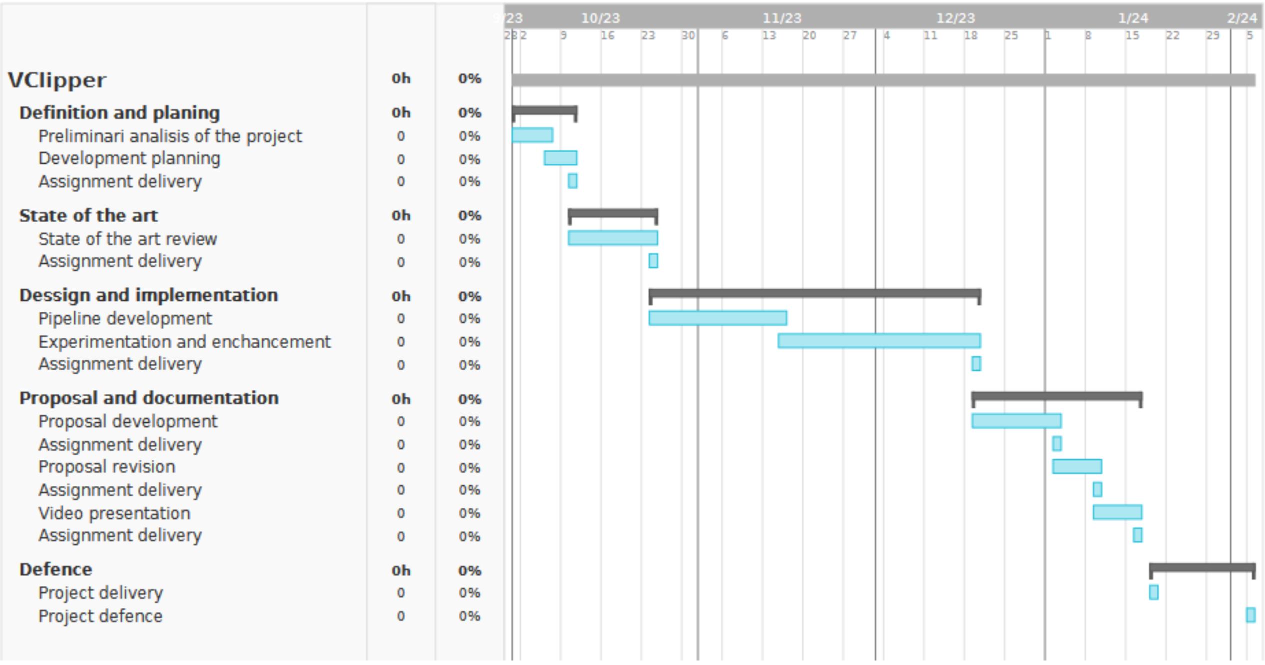
1.4. Approach and methodology

For the development this project, we aim to acquire a model that is able to extract and compare global features of an image, this can be archive either by training a model from scratch or by extracting those features from a pretrained model that already has proven to perform at a high level in the real-world domain. Choosing a pre-trained model over training from scratch can be advantageous for tasks with huge amounts of data and resources. Pre-trained models, having learned from vast datasets, capture intricate patterns and features. Utilizing these learned features as a foundation often leads to quicker, resource-efficient, and effective solutions, especially when time and data are constraints. They serve as a robust starting point, allowing customization for specific tasks without the need to build complex architectures from the ground up.

Considering all the advantages that pretrained models offer, we choose to exploit the capabilities of CLIP, which allows us to obtain the features of an image in semantic manner. Not only allowing us to make a comparison between the features extracted from an image, but also, the comparison between those features and a textual input.

1.5. Schedule

In this project, a series of tasks will be carried out, focusing mostly at the development of the pipeline and multiple experiments, but also at the development of this proposal and the review of the state of the art, as is shown in the schedule of the Gannt diagram. (see Figure 1)



VClipper	0h	0%
Definition and planing	0h	0%
Preliminari analisis of the project	0	0%
Development planning	0	0%
Assignment delivery	0	0%
State of the art	0h	0%
State of the art review	0	0%
Assignment delivery	0	0%
Dessign and implementation	0h	0%
Pipeline development	0	0%
Experimentation and enchancement	0	0%
Assignment delivery	0	0%
Proposal and documentation	0h	0%
Proposal development	0	0%
Assignment delivery	0	0%
Proposal revision	0	0%
Assignment delivery	0	0%
Video presentation	0	0%
Assignment delivery	0	0%
Defence	0h	0%
Project delivery	0	0%
Project defence	0	0%

Fig 1. Gannt schedule

1.6. Summary of the outputs of the project

The outputs obtained from this project will be the proposal of the project and a repository with the code and model resulting from the experimentation. The proposal must include the documentation of the theoretical foundation on which the experiment is based; the description of the implementation; and finally, the comparison of the results and conclusions of the experiments. At the same time, the repository must include at least a complete code, any dataset that might be produced as a result of our work and, finally, the result of the evaluations of the different experiments carried during the development.

1.7. Brief description of the remaining chapters of the report

The report is organized in several chapters. Starting with the Chapter 2: State of the art, where the current state of the field related to video content analysis is reviewed. It provides an overview of existing methodologies, technologies, and challenges in the domain. Understanding the state of the art is crucial as it lays the foundation for the methods employed in the study.

The following chapter. Chapter 3: Methods and resources and Chapter 4: Results. Detail the methodologies, resources, and techniques used and explain how the model was implemented bridging theory and practice. Also, on Chapter 4 the outcomes of the research efforts are presented. Including the findings from applying CLIP in video content analysis, showcasing the efficiency and precision achieved. That chapter provides detailed analysis and interpretation of the results obtained, demonstrating the practical applicability of the proposed methods. The results discussed here are essential for drawing conclusions in the next chapter.

Finally, in Chapter 5, the study concludes with a summary of research outcomes, a discussion on the implications, significance, and potential future research directions. This chapter aims at providing a comprehensive understanding of the study's impact and a ground base for future research.

2. State of the art

2.1. The problem

2.1.1 Description

The process of determining highlights in videos involves identifying distinct segments of the video that are the most significant or interesting elements of the content. Although this may appear straightforward, numerous obstacles exist associated with the classification of highlights in videos.

Firstly, events deserving of attention can happen on a range of temporal scales and the models need to be able to recognize and differentiate between highlights, which can be anything from brief actions to lengthy sequences. Determining the appropriate temporal scale is crucial. It involves understanding the granularity of the content, recognizing different types of events, and deciding what duration qualifies as a highlight.

Secondly, the significance of a particular segment can depend heavily on the context of the video. A specific scene might be a highlight in one context but not in another. Thus, increasing the importance of the semantic interpretation and understanding of the different moments in the video.

Image classification and semantic description of images using AI models has had a big breakthrough in the last decade with the introduction of first the convolutional networks (Krizhevsky et al., 2017; Lecun et al., 1998) and more recently with the use of transformers (Radford et al., 2021; Vaswani et al., 2023). Many models excel nowadays at tasks like action and object recognition, but most of these models are designed to work with still images. Only recently, advancements that include methods that focus on aligning textual descriptions not only with temporal segments but also with specific visual entities or objects within the video frames have been made(Long et al., 2015; Zhao et al., 2017). Specifically in temporal grounding, where models localize actions or events within a few frames accurately, has been a challenging but crucial area of research(Lei et al., 2020; Yang et al., 2022).

2.1.2 Overlapping tasks

Having into account the complexities of highlight detection exposed in the previous chapter. A few Overlapping task have been identified that may lead into a successful highlight moment selection.

- Video grounding: Involves associating natural language descriptions with specific temporal segments or objects in a video. Video grounding is crucial because it helps establish a connection between language descriptions and specific moments in the video. Accurate grounding ensures that textual descriptions align precisely with the visual events.

- Moment retrieval: Focuses on identifying and retrieving specific temporal segments within a video that correspond to particular actions, events, or activities. Moment retrieval is foundational to highlight detection. By accurately retrieving moments that contain significant events, highlight detection algorithms can then assess the importance of these moments, potentially classifying them as highlights based on various criteria.

- Highlight detection: Is the primary task at hand. It encompasses the entire process, incorporating insights from video grounding and moment retrieval. Effective highlight detection algorithms leverage information from these tasks to make informed decisions about what qualifies as a highlight based on temporal, semantic, and contextual clues.

2.2 Challenges

2.2.1 Image to textual description and semantic annotation

Bridging the gap between visual data and natural language, enabling machines to comprehend and interpret images in a manner similar to humans is probably the main challenge that researchers have to overcome when trying to successfully solve the task of video grounding. Thankfully nowadays, big pretrained transformer models exist (Feichtenhofer et al., 2018; Radford et al., 2021) that are able to perform this transcription and offer a translation in form of embeddings. Likewise, multiple databases have been created that offer annotations and training data both in image and video format(Gao et al., 2017; Lei et al., 2021).

2.2.2 Audio treatment

Even though we have been focusing on the visual aspects of the video, audio is a crucial component in videos that holds as much information as the images themselves, using audio for further treatment, analysis and incorporation in a multimodal model is as much as a challenge as is an advantage. Addressing this challenge successfully requires models capable of advanced audio processing techniques, including noise reduction, speech recognition, and emotion analysis (Hannun et al., 2014; Mehrish et al., 2023). Achieving comprehensive audio understanding can be an imperative for holistic multimedia analysis, and in turn for multimodal and fusion models (Liu et al., 2022).

2.2.3 Scene temporal window identification

Many techniques exist to select and capture the frames that compose a specific shot or scene. Some of the most common methods are periodic, event-based and keyframe-based. Identifying and selecting the best approach can not only help improve the accuracy of the model, but also boost the general performance of the implementation.

2.2.4 Resolution and Size of the data

High-resolution images and large video files require substantial computational resources for processing and analyzing the content. Balancing the need for detailed analysis with computational efficiency is a significant concern. Efficient algorithms and scalable methodologies are indispensable to handle large volumes of data without compromising the depth and accuracy of the analysis.

2.3 Pretrained models

The public release of large visual-language pretrained models like CLIP(Radford et al., 2021) or SlowFast(Feichtenhofer et al., 2018) has significantly contributed to the development of new video-language processing (VLP) models. Indeed, in all reviewed works of literature targeting tasks such as video grounding(Yang et al., 2022, 2022), moment retrieval(Lei et al., 2021; Liu et al., 2022), or highlight detection(D. C.-E. Lin et al., 2022), a pretrained model has been utilized. Often, these pretrained models are employed primarily for basic video preprocessing to extract a set of embeddings or features. These are then used as inputs for the VLP models, even though the usage of these pretrained models is sometimes limited to this initial processing stage.

2.3.1 Backbones

In visual models, particularly in the context of VLPs, Backbones refer to the base network architectures that are used primarily for feature extraction. Common Backbone architectures that have been widely used and researched are Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs).

These two architecturally distinct approaches come with their own strengths and weaknesses. In one hand, CNNs are built with convolutional layers and excel at capturing local features through spatial correlations and tend to be more parameter-efficient, making them suitable for tasks with limited data and computational resources. Also, some of the key strengths of the CNNs is their interpretability, a long history incremental improvement and a wide number of successful implementations. On the other hand, ViTs, based on the transformer architecture, process images by dividing them into patches and apply self-attention mechanisms to capture global dependencies across the entire image. Although they tend to require more data and computational power, ViTs have shown better performance in large-scale tasks, surpassing CNNs in some scenarios.

Some of the Backbones relevant to pretrained models used in VLPs are:

- VGG: Developed by Karen Simonyan and Andrew Zisserman(Simonyan & Zisserman, 2015), the VGG architecture highlighted that increasing the depth of CNNs is a straightforward way to improve their performance. VGG utilizes 13 convolutional layers and 3 fully connected layers, surpassing its predecessor, AlexNet(Krizhevsky et al., 2017), in depth and using smaller 3x3 convolutional filters. These smaller filters enable finer feature extraction, while the network's greater depth allows for learning more complex patterns. The VGG models, marked a significant advancement in demonstrating the impact of depth in CNNs and have been extensively applied in various computer vision tasks.

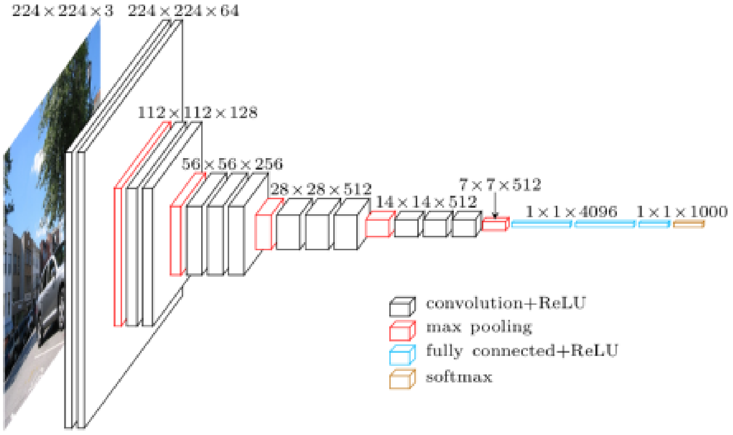


Fig 2. VGG architecture.(source VGG paper)

- ResNet: Developed by Kaiming He et al. (He et al., 2015), the ResNet architecture introduces shortcut or residual connections to tackle the gradient vanishing problem common in deep CNNs. These connections allow for the effective updating of weights and biases across layers by preserving update information and enabling direct gradient flow. This innovation enables the construction of much deeper networks than traditional CNNs, as evidenced by variants like ResNet-50, ResNet-101, and ResNet-152. These versions have significantly advanced image classification and other tasks by facilitating the efficient training of deeper models.

•

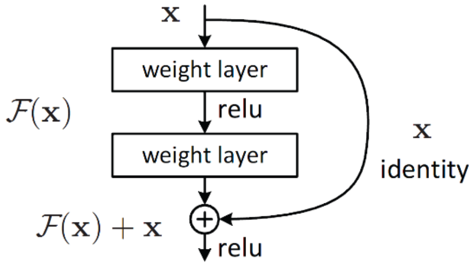

x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Fig 3. Residual connection skip.(Source ResNet paper)

- ViT: Developed at Google Brain by researchers leaded by Alexey Dosovitskiy (Dosovitskiy et al., 2021). This model represents a significant shift in image processing approaches for deep learning tasks. In contrast to traditional convolutional neural networks, ViT treats an image as a sequence of fixed-size patches (32x32 pixels for ViT-32), similar to the way words in a sentence are processed in natural language processing. Each image patch is embedded and then processed through a series of transformer blocks employing self-attention mechanisms, allowing the model to capture complex, dependencies across different parts of the image. This global processing approach marks a major advancement over the local processing typical of CNNs.

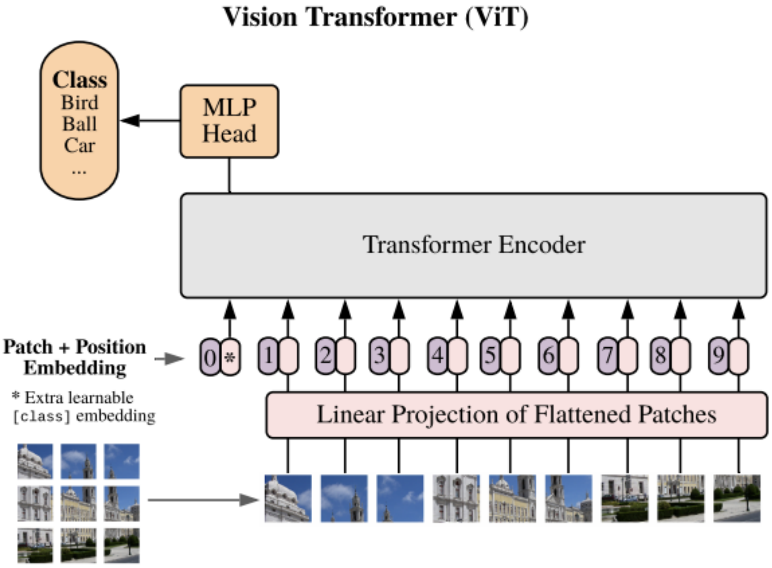


Fig 4. ViT architecture.(Source ViT paper)

2.3.2. CLIP

CLIP(Contrastive Language–Image Pretraining) developed by OpenAI and released in 2021, has become a cornerstone in the realm of visual-language tasks (Radford et al., 2021). CLIP is built upon two primary Backbones, a text encoder, typically a Transformer-based model similar to those used in natural language processing tasks, which processes text converting words and phrases into a high-dimensional vector space, and an image encoder that performs a similar function for visual inputs. This dual-encoder architecture allows CLIP to handle and interpret both textual and visual information within a unified framework, facilitating understanding and a pairing of both modalities.

Thanks to this dual-encoder architecture, CLIP's training uses a contrastive learning approach, which involves teaching the model to correctly match images with their corresponding textual descriptions. During training, the model is presented with batches of image-text pairs and, for each image, the correct textual description is paired alongside incorrect(negative) textual descriptions. Once the text and images are encoded into vectors and paired together, CLIP uses cosine similarity to measure the closeness of these vectors in the embedding space. Finally, the training objective is to adjust the parameters of the encoders so that the distance between the vectors of correctly paired image-text inputs is minimized, while the distance between the vectors of incorrectly paired inputs is maximized.

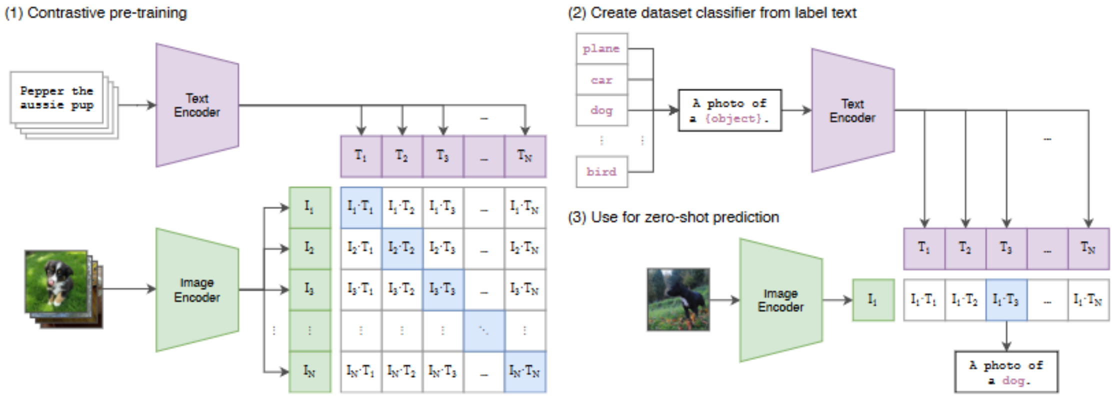

(1) Contrastive pre-training
Pepper the aussie pup
Text Encoder
Image Encoder
T₁ T₂ T₃ ... T_N
I₁ I₂ I₃ ... I_N
I₁·T₁ I₁·T₂ I₁·T₃ ... I₁·T_N
I₂·T₁ I₂·T₂ I₂·T₃ ... I₂·T_N
I₃·T₁ I₃·T₂ I₃·T₃ ... I₃·T_N
I_N·T₁ I_N·T₂ I_N·T₃ ... I_N·T_N
(2) Create dataset classifier from label text
plane
car
dog
bird
A photo of a {object}.
Text Encoder
(3) Use for zero-shot prediction
Image Encoder
I₁
T₁ T₂ T₃ ... T_N
I₁·T₁ I₁·T₂ I₁·T₃ ... I₁·T_N
A photo of a dog.

Fig 5. Ilustration of the CLIP network. (Source CLIP paper)

2.3.3 Slowfast

Slowfast, a network developed by Facebook and introduced in 2019 (Feichtenhofer et al., 2018) , is built on top of two ResNet convolutional networks that act as two parallel pathways that analyze the videos at different framerates. The architecture is based on the idea that different frames in a video sequence can have varying speeds of motion, some events occurring in a video might have slow and smooth motion, while other parts can have fast and rapid motion. Using this architecture have proven extremely effective on capturing motion driven events, and the information offer by the pretrained model of this network is often used in VLP models.

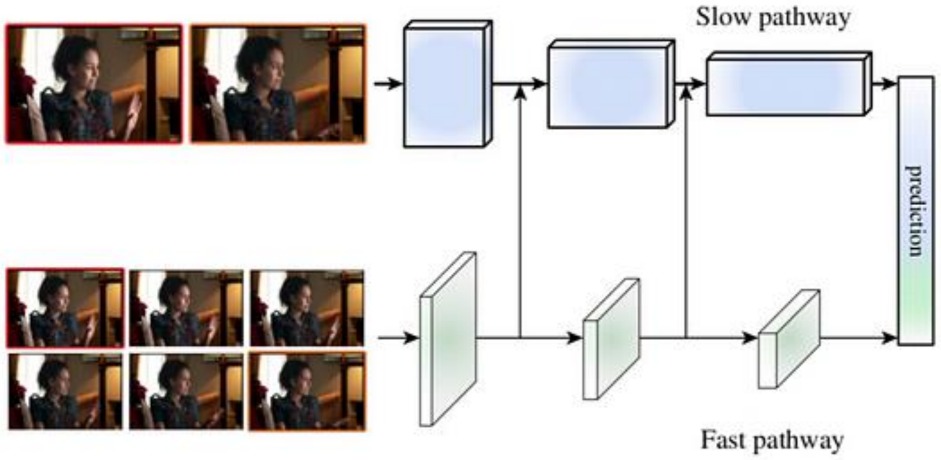


Fig 6. Ilustration of the SlowFast network. (Source SlowFast paper)

2.4. Metrics

A wide range of metrics are used in VLP models, both, to asses the similarity between images and texts, and to evaluate the models. Notably, cosine similarity and Intersection over Union (IoU) play a pivotal role in the development and training of VLP models. While other metrics. like recall at one(R@1) and mean Average Precision(mAP) mark the standard for comparisons between models that perform moment retrieval and highlight detection tasks.

- **Cosine similarity**: Used to measure the distance between vectors defined in an inner product space. Typically used in VLPs to assess the similar between the vectors produced by the text and images encoders. Given two n-dimensional vectors of attributes, *A* and *B*, the cosine similarity, \cos{\left(A, B\right)}, is represented using a dot product and magnitude as

$$\cos(\mathrm{A},\mathrm{B}) = \frac{\mathrm{A}\cdot\mathrm{B}}{|\mathrm{A}||\mathrm{B}|} = \frac{\sum_{i=1}^{n}\mathrm{A}_i\mathrm{B}_i}{\sqrt{\sum_{i=1}^{n}(\mathrm{A}_i)^2}\sqrt{\sum_{i=1}^{n}(\mathrm{B}_i)^2}} \qquad (1)$$

- **Intersection over Union(IoU):** Commonly used in object detection**,** measures de interlap between two boundaries. In VLPs destinated to moment retrieval tasks, the boundaries are defined by the start and end of the temporal window that defines the shot or scene. Given 2 objects A and *B* the *IoU(A,B)* is defined as:

$$IoU(A,B) = \frac{|A\cap B|}{|A\cup B|} = \frac{|A\cap B|}{|A|+|B|-|A\cap B|} \qquad (2)$$

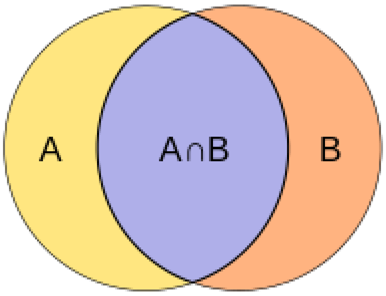


Fig 7. IoU overlap

- **Recall@K (R@k):** Recall at k is the proportion of relevant items found in the top-k recommendations. Where in this case, moment retrieval tasks, a relevant moment recommendation is considered relevant if the window recommended surpasses a given IoU threshold. For example, R@1 IoU=0.5, determines the proportion of times that the first recommendation for a query has an IoU above 0.5.

- **Mean Average Precision (mAP)**: For a set of queries, mAP is calculated as the mean of the average precision scores for each query. Where the average precision is the area under the curve of precision and recall for retrieved windows within the query. This is, with *Q* number of queries and *k* windows retrieved:

$$AveP = \sum_{k=1}^{n} P(k)\Delta r(k) \tag{3}$$

$$MAP = \frac{\sum_{q=1}^{Q} AveP(q)}{Q} \tag{4}$$

2.5. Previous work

To do a comparison of the State of the art we focus on those models that archived a high performance in both moment revival and Highlight detection. Most of these models are encoder-decoder based and exploit CLIP and Slowfast to create the encoders while in most cases implement slightly different techniques to obtain the window offset and saliency score for the highlight.

- Videogenic: Is an Image based, CLIP Zero shot, moment retrieval model. Uses cosine similarity between embedding to perform the comparison between images. (D. C.-E. Lin et al., 2022)

- Zero-shot Video Moment Retrieval: Is one of the few papers that focus on Zero-shot detection and, on their work, exploit ShotDetect[] for scene selection. Noticeably, to the best of our knowledge, is the only Zero-shot work that offers a baseline for comparison(Diwan et al., 2022).

- Moment-Deter: First of its kind on moment retrieval, Moment-DETR, a transformer encoder-decoder, uses CLIP and Slowfast as the video and text feature extractors. In its architecture each decoder layer consists of a multi-head self-attention layer and a cross-attention layer. Another of its main contributions is the development of QvHiglights data set. Moment-DETR offers two baselines, one for the pretrained encoder and one for the untrained encoder (Lei et al., 2021).

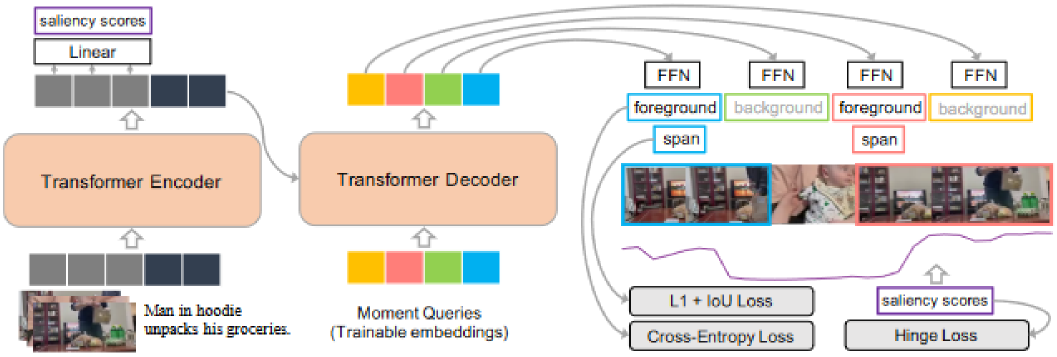


Fig 8. Architecture proposal for Moment-Deter.(Source Moment-Deter paper)

- Bam-Deter:Boundary-Aligned Moment Detection Transformer, follows the transformer encoder-decoder architecture proposed by Moment-deter, and introduces a Dual-pathway decoding layer that aims at refining the anchor and boundaries of the windows detected(Lee & Byun, 2023).

- QD-Deter:Query-Dependent Video Representation, follows the transformer encoder-decoder architecture proposed by Moment-deter, but focuses on Negative Relationship between examples on the training phase (Moon, Hyun, Park, et al., 2023).

- CG-Deter:Correlation-guided Query-Dependency Calibration in Video Representation, follows the Deter architecture, but introduces a Dummy Encoder mechanism to refine the embeddings of video and query before they are feed to the decoder (Moon, Hyun, Lee, et al., 2023).

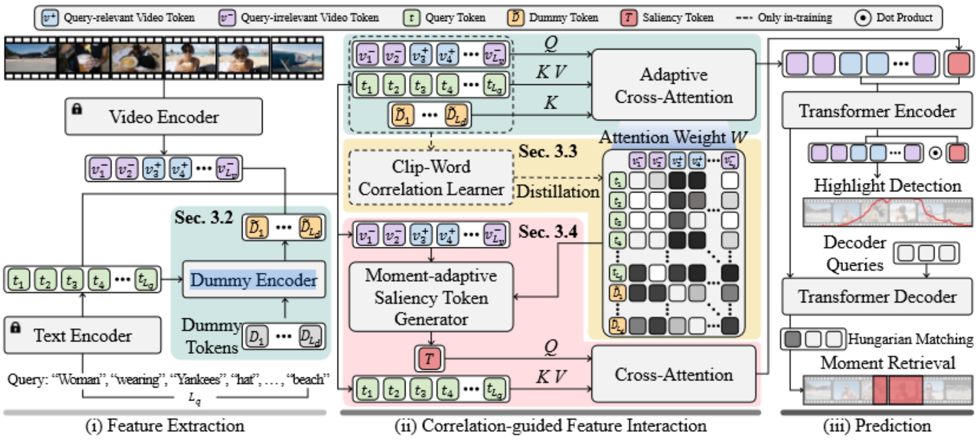


Fig 9. CG-Deter architecture.(Source CG-Deter paper)

- TALL: Is a scene based, Cross-modal Temporal Regression Localizer (CTRL) architecture. It uses encoders to extract both sentence embeddings and text video features that after combined are used as inputs in a temporal regression network (Gao et al., 2017).

- UniVtg: Is a scene based crossmodal autoencoder. In this case inputs to the UMT decoder are clip-aligned text-guided queries instead of positional encodings that uses CLIP and Slowfast as embeddings and features (K. Q. Lin et al., 2023).

Fig 10. Previous work baseline.

		R@1 IoU=0.5	R@1 IoU=0.7	mAP@0.5	mAP@0.75	mAP
Multimodal Vídeo+ audio	Moment-DETR(w/PT)	59.78	40.33	60.51	35.36	36.14
	BAM-DETR	**64.07**	**48.12**	**65.61**	**47.51**	**46.91**
	QD-DETR	63.06	45.1	63.04	40.1	40.19
Video Only	UniVTG(w/PT)	65.43	**50.06**	64.06	**45.02**	**43.63**
	CG-DETR	**65.43**	48.38	**64.51**	42.77	42.86
	QD-DETR	62.4	44.98	62.52	39.88	39.86
	UniVTG	58.86	40.86	57.6	35.59	35.47
Zero-Shot	SD+C+SW	40.24	25.94	41.74	24.11	24.82
	SD+C+SW (w/PT)	42.12	27.89	43	24.68	25.5

3. Methods and resources

3.1 Experimental Setup

For the experimental setup, all the code was written in Python, and executed on a Google Colab account that provided access to a remote machine equipped with a T4 GPU and 50GB of RAM. These resources were chosen to address the computational demands of the project. The T4 GPU's processing capabilities were employed for faster algorithm execution, while the substantial RAM allocation prevented memory-related issues during complex computations.

One TB data space was contracted to manage the storage requirements, accommodating the big video dataset, the processed arrays and embeddings and the python notebooks used to execute the experiments. Google Colab cloud-based environment facilitated remote accessibility, eliminating the need for extensive local hardware. This approach proved cost-effective and accessible, allowing us to focus on experimentation without hardware maintenance concerns.

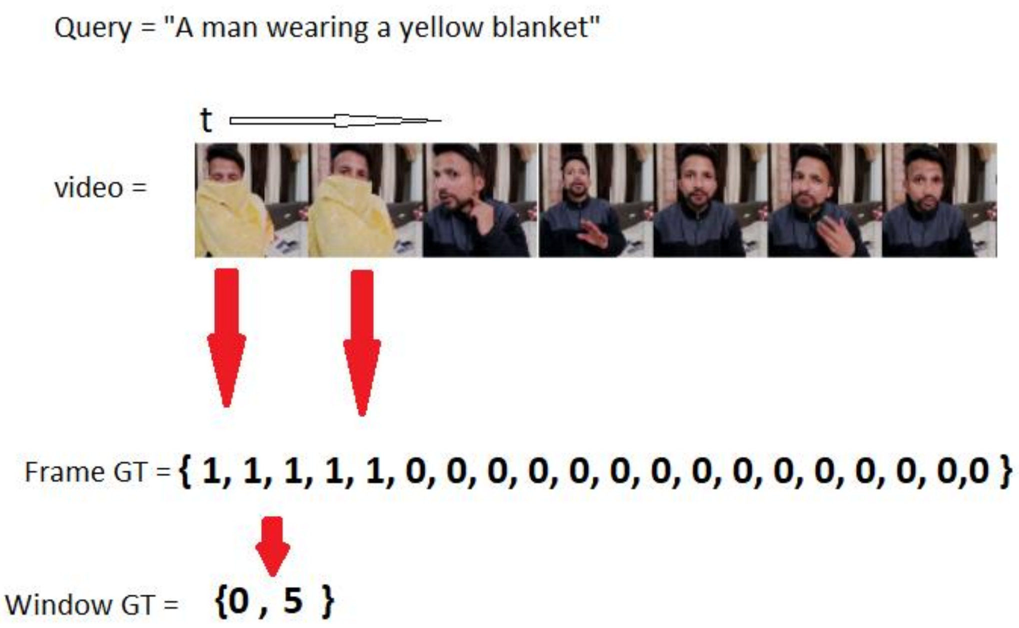
3.2 Dataset

Our experimentation centers around the QvHighlights dataset (Lei et al., 2021), a repository designed for video analysis tasks. Comprising 150-second video clips each paired with a textual query, this dataset serves as a benchmark for evaluating the efficacy of video retrieval systems. Notably, the dataset provides two distinct ground truths: one delineating the relevant frames within the video and another specifying the temporal window associated with the selected segments.

The volume of raw data within the QvHighlights dataset is substantial, totaling 130 gigabytes. The unprocessed dataset was extracted from Youtube segments and annotated to represent the moment expected to be retrieved from the video.

To facilitate a systematic evaluation, the QvHighlights dataset has been partitioned into train and evaluation splits. The training set comprises 7218 observations, while the evaluation split consists of 1550 observations. This partitioning ensures a balance between model training and performance assessment.

Fig 11. Example of an observation in the dataset.


Query = "A man wearing a yellow blanket"
t
video =
Frame GT = { 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0 }
Window GT = {0 , 5 }

3.4 Approach

In our experiments, we use CLIP for zero-shot moment retrieval instead of coupling another model on top of CLIP encoder. We avoid this approach due to its usual limitations, often rigid, not easily scalable, and restrictive in terms of input length.

Skipping the additional model simplifies the process, making it more adaptable and scalable. CLIP's embeddings offer flexibility, allowing for a versatile solution to moment retrieval using embedding similarity. This choice is made to better handle diverse video content and different contextual requirements.

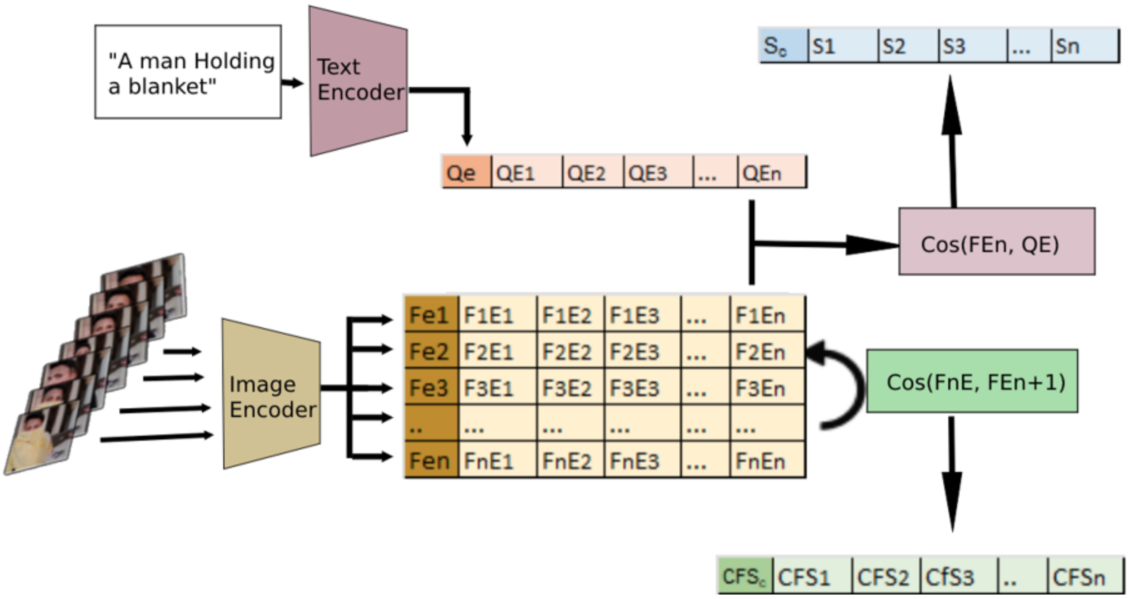
The decision to bypass supplementary models considers their tendency to limit input length. Constraints on input length can hinder the model's ability to process longer video sequences. By relying solely on CLIP's embeddings similarity, we aim to maintain the natural temporal context in unmodified video data on a real-world video context. This approach aligns with our experimental objectives and emphasizes exploring CLIP's inherent capabilities without unnecessary modifications.

3.5 Video Preprocessing

Given that CLIP Zero-shot requires images or texts as inputs, to represent a full video, our preprocessing step involves extracting frames from the videos to properly represent the content. To ensure ease of processing and consistency, we adopt a periodic frame extraction approach. The chosen framerate is set to 1/2, aligning with the dataset's ground truth, which is calculated at this framerate.

The outcome of our selected approach is an array of 75 images for each observation $V=\{F_1,F_2 ... F_n\}$, this array forms the basis for calculating the similarity array. An alternative approach could involve selecting keyframes that mark scene changes and extracting a predefined number of frames to represent each scene.

Upon obtaining the array of frames for each observation, a processing pipeline is implemented, leveraging CLIP image encoder to derive embeddings for each frame and CLIP text encoder to derive embeddings for the query associated with the observation, obtaining $V_E=\{F_{e1},F_{e2} ... F_{en}\}$ and Q_E. Finally using the embeddings, a cosine similarity array is computed, $S_c(V_E,Q_E)= \{ s_1, s_2 .. s_n\}$. This gives us an array that captures the similarity between the embeddings of the frame array and the query embedding, explaining how each frame relates to the query. Additionally, a continuous frame similarity array is also computed reflecting the similarities between each frame and the subsequent one, $CFS_c = \{S_c(F_{e1},F_{e2}), S_c(F_{e2},F_{e3}) ... S_c(F_{en},F_{en+1}) \} = \{ cfs_1, cfs_2 .. cfs_n\}$. These similarity arrays will serve as base components for the empirical evaluation of diverse methodologies aimed at identifying relevant windows.



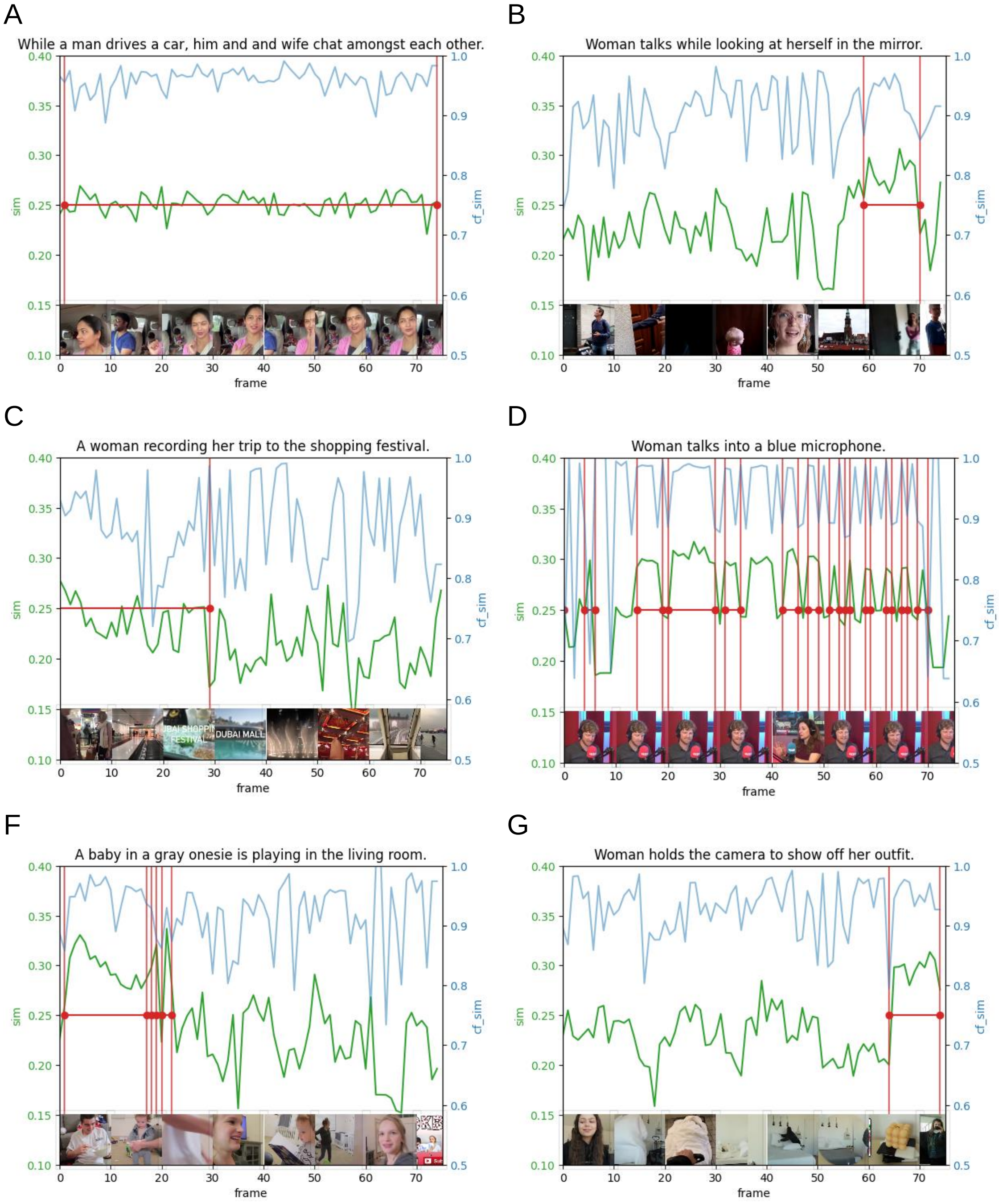
Fe1	F1E1	F1E2	F1E3	...	F1En
Fe2	F2E1	F2E2	F2E3	...	F2En
Fe3	F3E1	F3E2	F3E3	...	F3En
..	...	...	...	...	...
Fen	FnE1	FnE2	FnE3	...	FnEn

Fig 12. Video preprocessing and similarity vectors extraction.

3.7 Data and similarity array Exploration

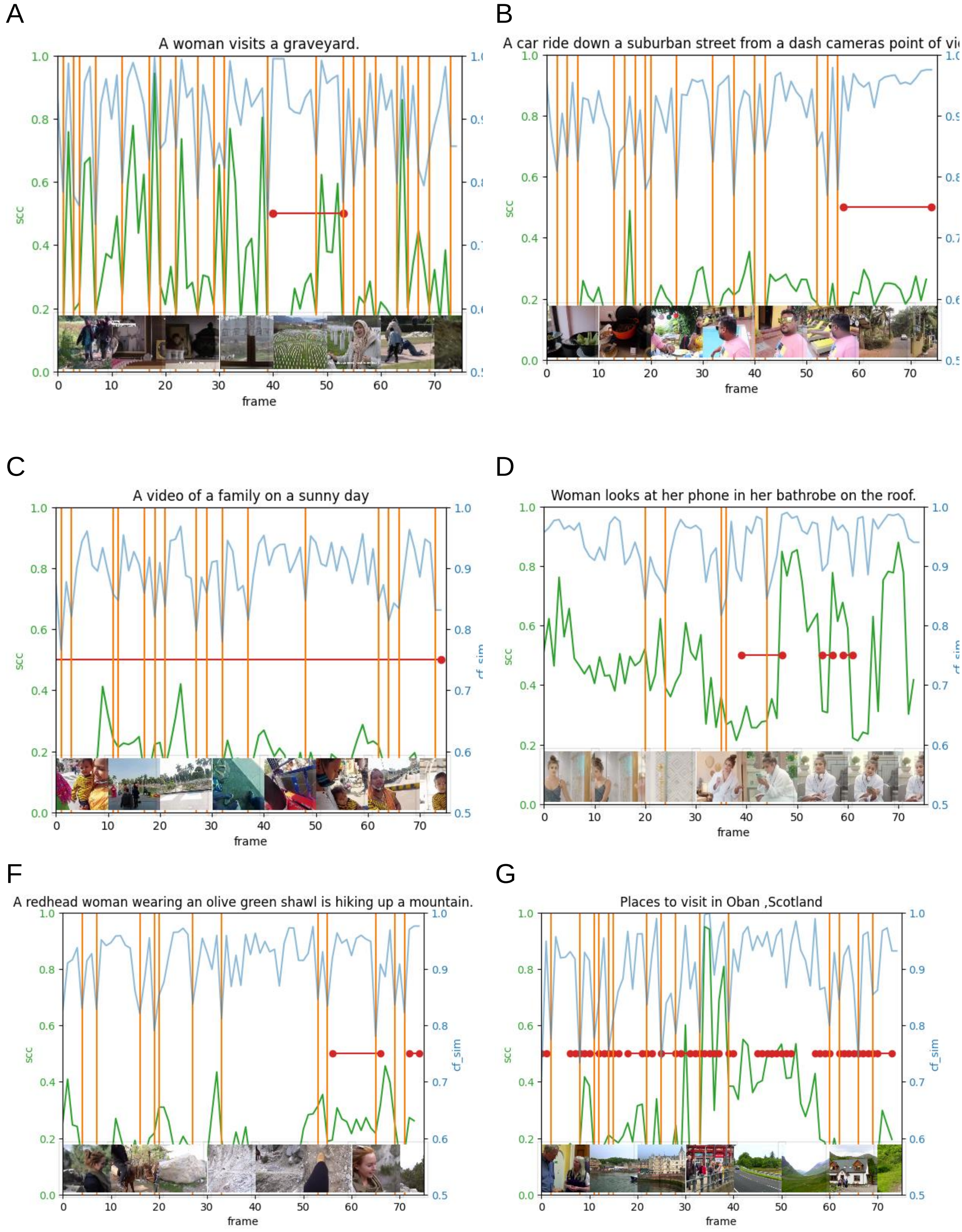
Once the similarity arrays are obtained, a qualitative exploration of a subset of observations was made, this exploration seeks to assess the depth of information encapsulated within the similarity arrays and to scrutinize their accuracy in relation to potential keyframes and pivotal moments present in the video.

Fig 13. Similarity array Exploration. Video observations paired with both similarity vectors, Sc and CFSc, we also display the ground truth for the windows. A, B, D and G display clear patterns for both vectors. F displays a ground truth scene cut that are not clearly reflected on the vectors. C displays a more erratic signal specially on the CFSc.



We also explored the CFS_c alongside a continuous SSIM index(Wang et al., 2004) based array and the window ground truth. The objective was to do a qualitative assessment on how well the CFS_c performs against an Image-to-Image comparison metric. We can observe that the CFS_c follows a similar pattern to the continuous SSIM, although the range on the signal produced seems to be lower, where similarity distances between the frames rarely fall under the 0.5 threshold. When we compare the cuts predicted by $\text{CFS}_c < 0.85$ with the window ground truth, we can observer that the continuous frame similarity tends to fit the boundaries of the windows.

Fig 14. Continuous similarity exploration. Video observations paired with continuous frame similarity and continuous frame SCC, we also display the ground truth for the windows. A, B, C cut detections using the CFSc approach really well the ground truth. D fails to properly detect the cuts.



3.8 Selection methods

3.8.1 VC1: Similarity array comparision with treshold λ

This is the most basic selection method, we defined a threshold λ, and for every similarity s_n in the similarity array $S_{c,}$ we evaluate if the value of S_n is over the threshold λ, if this is the case, we consider the F_n a relevant frame. obtaining a vector of selected frames Sf={0,1 … 1},

To obtain windows from the array of selected frames, we first create a window for every group of continuous selected frames in the Sf vector. Then we use those windows and the similarity array S_c to calculate the confidence of the window. where the $w_{confidence}$ is the mean of the similarities within the window, Finaly, we introduce a second threshold β and select every window with $w_{confidence}$ > β.

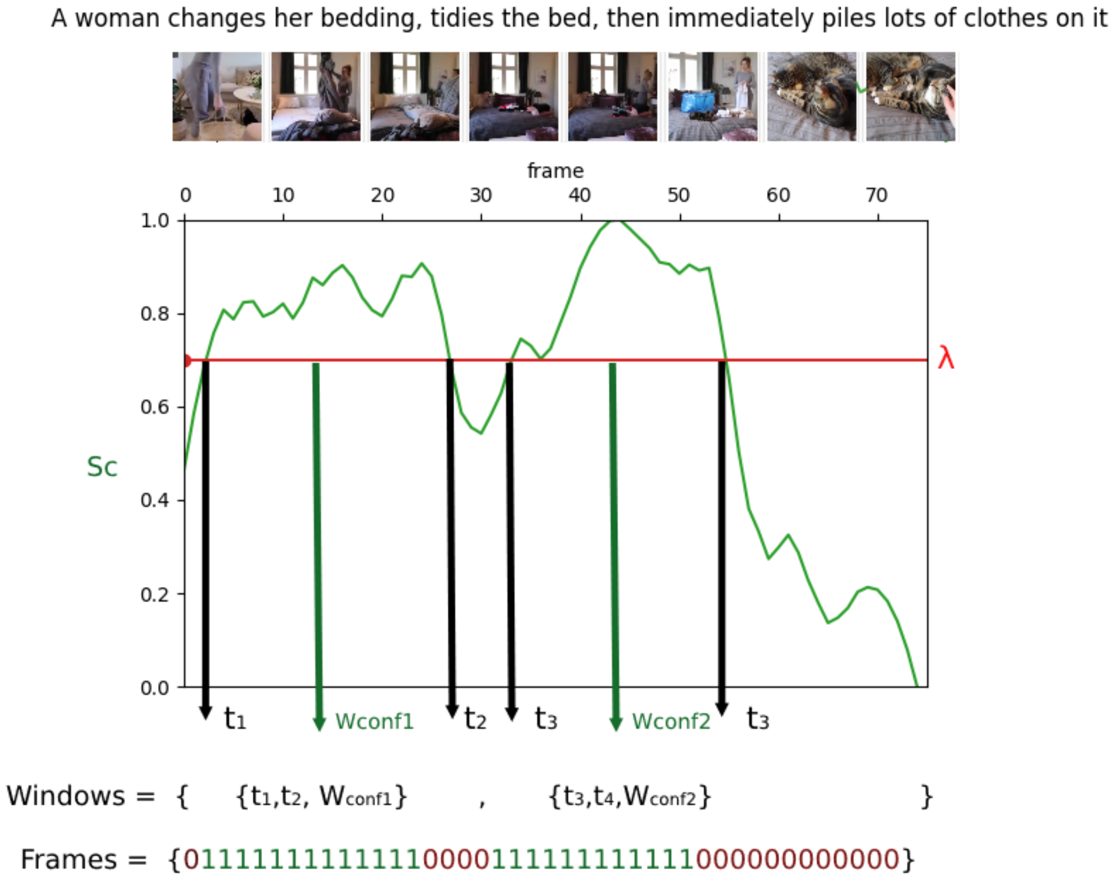


Fig 15. VC1 selection method.

3.8.2 VC2: Continuous frame similarity-based window method

This method leverages the continuous frame similarity array $\mathrm{CFS_c}$ to detect scenes changes within the video. Given a threshold α for each $\mathrm{cfs_n}$ in the $\mathrm{CFS_c}$ array, if $\mathrm{cfs_n}$ falls below the threshold α we evaluate if any of the contiguous frames $\mathrm{cfs_{n-1}}$ and $\mathrm{cfs_{n+1}}$ are above the threshold α, if this is the case, we consider that a change of scene has occurred. Afterwards we introduce a factor *p* and a second threshold λ, and for each scene we use the similarity array S_c and *p* to evaluate the confidence score within the boundaries of the window, if the confidence score of the scene is above the threshold λ the scene is considered relevant and is selected. Finally, if multiple contiguous scenes are selected those scenes are fused together.

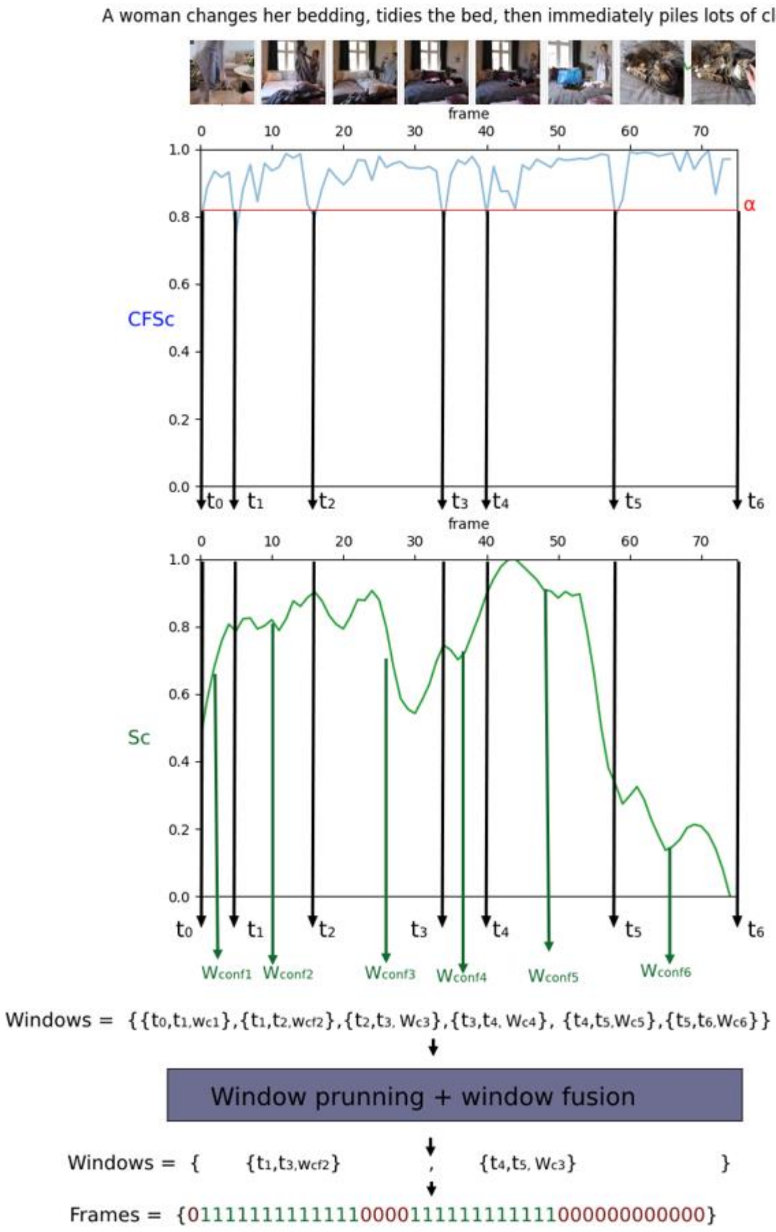

A woman changes her bedding, tidies the bed, then immediately piles lots of clothes on it
frame
0 10 20 30 40 50 60 70
1.0 0.8 0.6 0.4 0.2 0.0
CFSc
α
t0 t1 t2 t3 t4 t5 t6
frame
0 10 20 30 40 50 60 70
1.0 0.8 0.6 0.4 0.2 0.0
Sc
t0 t1 t2 t3 t4 t5 t6
Wconf1 Wconf2 Wconf3 Wconf4 Wconf5 Wconf6
Windows = {{t0,t1,wc1},{t1,t2,wcf2},{t2,t3, Wc3},{t3,t4, Wc4}, {t4,t5,Wc5},{t5,t6,Wc6}}
Window prunning + window fusion
Windows = { {t1,t3,wcf2} , {t4,t5, Wc3} }
Frames = {011111111111110000111111111111000000000000}

Fig 16. VC2 Selection method.

3.9 Similarity array postprocessing

Parallel to the selection methods described above, we defined three postprocessing methods for the similarity array $S_{c.,}$ The objective of this postprocessing is to normalize and further enhance the differentiation between relevant and non-relevant frames within the $S_{c.}$ The three methods selected are the following:

1- Min-max normalization per S_c basis:

$$Sc' = \frac{Sc - min(Sc)}{(max(Sc) - min(Sc))} \quad (5)$$

2- Min-max normalization within the boundaries of all the S_c obtained from all the videos in the dataset, where TscMax = max({S_{c1}, $S_{c2\,...}$ S_{cn} }) and TscMin.= min({S_{c1}, $S_{c2\,...}$ S_{cn} })

$$Sc' = \frac{Sc - TscMin}{(TscMax - TscMin)} \quad (6)$$

3- Obtaining the logarithmic values of the S_c to further enhance the differentiation between similarities within the Sc, and then, smoothing the S_c to get rid of standalone high values. To do the smoothing we choosed the Savitzky–Golay filter(Savitzky & Golay, 1964) with sliding window *M* and polynomial order 2, in general terms the Savitzky–Golay filter tends to preserve local maxima and minima compared to other smoothing techniques. Finally, we normalize the smoothed $S_{c.}$

This is:

Extract logarithmic values:

$$S_c = log(Sc) \quad (7)$$

Given the Savitzky–Golay function, where s_j is an observed value within the S_c', and the values within the S_c' are treated as a set of m convolution coefficients, C_i, according to the expression:

$$S_j = \sum_{i=\frac{1-m2}{2}}^{\frac{m-12}{2}} C_i \; s_{j+i}, \qquad \frac{m+1}{2} \leq j \leq n - \frac{m-1}{2} \quad (8)$$

Apply the filter and normalize:

$$Sc'' = Savitzky\text{–}Golay(Sc\,, M\,, 2) \tag{9}$$

$$Sc''' = \frac{Sc - min(Sc'')}{\left(max(Sc'') - min(Sc'')\right)}$$

3.10 Assessment of pretrained models.

In the evaluation of various pretrained models, we employed the Similarity Array Comparison with Normalization method across a spectrum of thresholds (λ=[0.5,0.55,0.6,0.65,0.7 ,0.75]). This involves testing the Average IoU to ensure the model and pretrained weights selected are the ones that offer the best differentiation over a wide range of thresholds.

This selection criterion focuses on the models offered by the OpenCLIP (Cherti et al., 2022) project. We narrow our selection to models designed for 3*224*224 input images, facilitating a comprehensive understanding of their performance in the context of similarity array analysis.

3.11 Optimization and evaluation

To test the effects of the postprocessing and selection methods first we need to select the optimal values for the hyperparameters λ ,α, *p* and *M* defined in sections 3.8 and 3.9. To do so we use the training split and the Bayesian optimization algorithm offered by Optuna (Akiba et al., 2019), the metrics selected for the optimization process are AP at IoU(0.5) for frame selection and R@1 at IoU(0.5) for window selection. In both cases we seek to maximize the correct detections with IoU > 0.5.

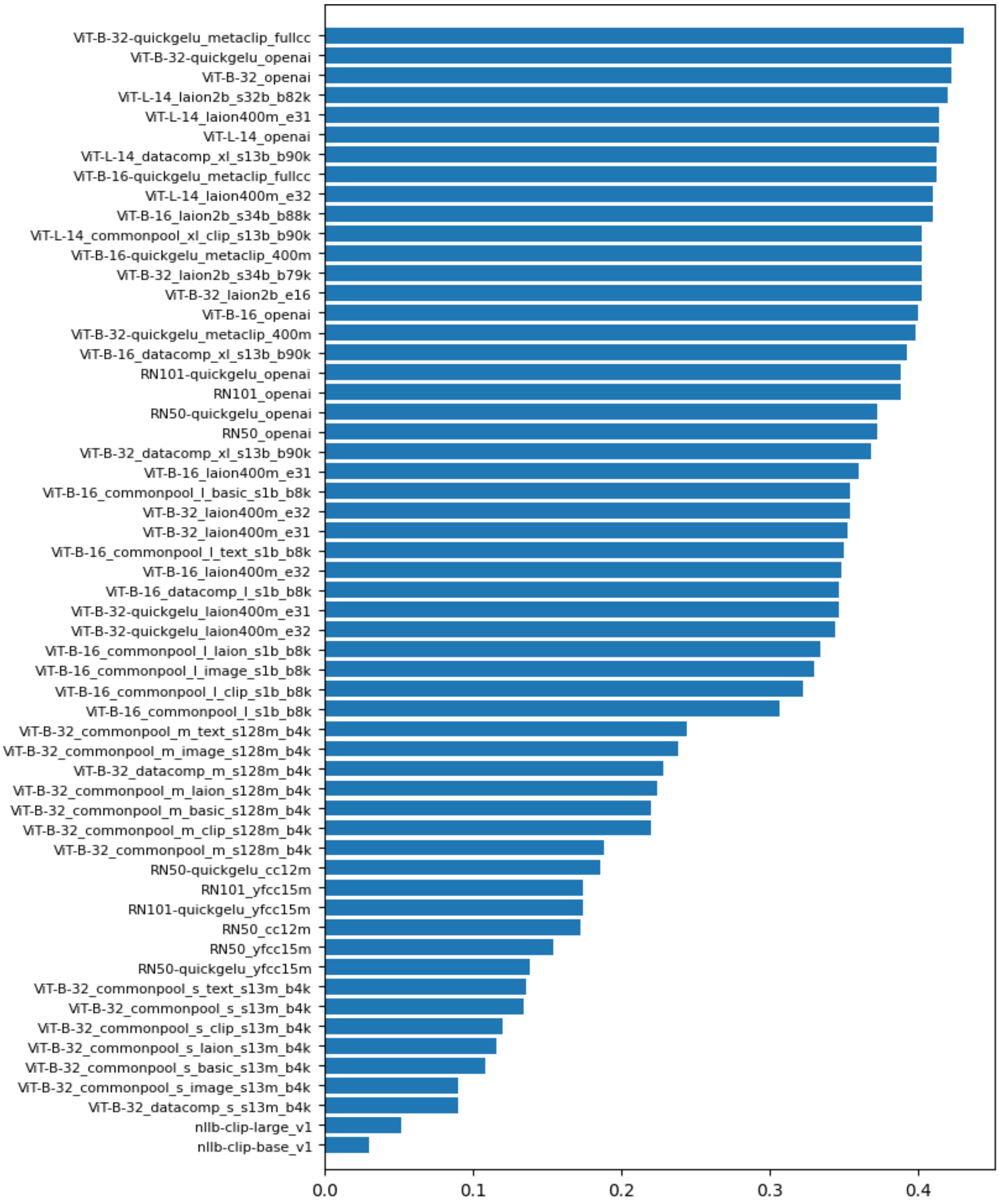
Once the optimal values are chosen for each combination of selection method, postprocessing and selection type, we validate the results over the validation split. The metrics evaluated and displayed are the ones defined in the section 3.3 of this document.

4. Results

4.2 Assessment of pretrained models.

The results indicated that for input images at size 3*224*224, Vit-B-32 backbones with openai or laion2k weights tend to generalize better, and offer a highest IoU overall. This means that the distance between good and bad similarity of frame embeddings and query embeddings tends to be greater for those models and, that for any given threshold, we are able split better between good and bad examples.

Fig 17. Pretrained models assessment on IoU >0.5



4.2 Validation and assessment of proposed methods

4.2.1 Frame per video Ground truth:

After the optimization process, we evaluated the vector of selected frames against the ground truth, and as we analyzed the results, we didn’t see a major improvement for any combination of selection method and preprocessing of the similarity array. Even though the smoothing tends to have a slightly higher avg IoU for both selection methods, the VC1 and the VC2.

method	Sc postprocess	accuracy	precision	recall	F1	IoU	AP 05	AP 07
VC1		0.757	0.723	0.615	0.585	0.460	0.429	0.210
VC1	norm	0.781	0.723	0.629	0.625	0.490	0.473	0.211
VC1	adjust_norm	0.776	0.759	0.631	0.635	0.505	0.484	0.249
VC1	log_smooth_norm	0.798	0.700	0.693	0.641	0.517	0.538	0.292
VC2		0.772	0.703	0.653	0.596	0.481	0.483	0.277
VC2	norm	0.800	0.666	0.696	0.615	0.501	0.531	0.301
VC2	adjust_norm	0.776	0.728	0.666	0.625	0.507	0.516	0.292
VC2	log_smooth_norm	0.804	0.711	0.691	0.647	0.527	0.544	0.314

Fig 18. Frame per video Ground truth:

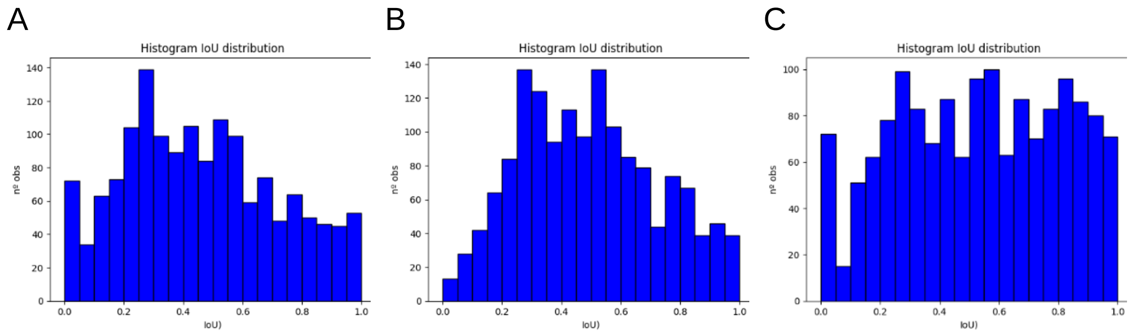


Fig 19. Distribution of observations by IoU. A:Non normalized VC1. B Normalized VC1. C:Smoothed VC2.

4.2.2 Window per video ground truth:

In the evaluation of the selected windows with the ground truth, for any combination of selection method and preprocessing, we could see that the smoothing of the S_c had a significant effect on the VC1 selection method. Improving by 20 points or more on the major metrics compared with the unprocessed similarity array, reaching 57.38 at R@1 and 51.62 mAP at Iou > 0.5.

After looking at the results of the VC2 selection method we detected that the effects of the processing of the S_c has a lower impact in the results compared with the VC1 selection method. This results were expected given that the windows detected using the $CFS_{c.}$ remains fairly constant regardless of the postprocessing of the S_c. This is not the case with the VC1, where windows are determined directly using the S_c.

method	Sc postprocess	R1@0.5	R1@0.7	mAP	mAP@0.5	mAP@0.75	mIoU
VC1	--	35.34	23.16	16.97	28.53	16.61	34.42
VC1	norm	37.26	25.61	21.97	38.95	21.72	34.59
VC1	adjust_norm	45.14	30.05	26.32	44.46	25.77	45.23
VC1	log_smooth_norm	57.38	36.14	28.48	51.62	27.68	51.38
VC2	--	46.86	29.52	22.53	38.08	21.82	46.17
VC2	norm	47.32	29.98	24.22	41.83	23.62	45.37
VC2	adjust_norm	41.03	25.55	24.99	43.53	24.21	41.99
VC2	log_smooth_norm	50.69	33.36	25.19	42.93	24.76	48.03

Reviewing the mAP results by windows lenght. We observed that for any given method, and processing of the S_c, the detection of shorter windows performs significantly worse than long and middle range windows. Wich can be an indication that a lower framerate is needed in the preprocessing of the vídeo.

method	Sc postprocess	MR-long-mAP	MR-middle-mAP	MR-short-mAP
VC1	--	20.65	17.17	2.82
VC1	norm	19.6	26.19	4.95
VC1	adjust_norm	36.76	23.45	3.64
VC1	log_smooth_norm	33.36	30.91	1.93
VC2	--	33.08	19.88	2.46
VC2	norm	33.22	22.75	3.12
VC2	adjust_norm	35.97	22.29	3.93
VC2	log_smooth_norm	33.64	25.22	2.47

Overall, the results from the window per video ground truth systematic evaluation are quite promising, specially on the VC1 selection method with Smoothing.

4.4 Baseline comparison

Looking at the baseline. We can observe that the VC1 selection method with smoothing performs better when compared to other Zero-shot approaches. Although still performs significantly worse than other autoencoder models that use CLIP and Slowfast as base encoders.

		R@1 IoU=0.5	**R@1 IoU=0.7**	**mAP@0.5**	**mAP@0.75**	mAP
multimodal	Moment-DETR(w/PT)	59.78	40.33	60.51	35.36	36.14
	BAM-DETR	**64.07**	**48.12**	**65.61**	**47.51**	**46.91**
	QD-DETR	63.06	45.1	63.04	40.1	40.19
Video Only	UniVTG(w/PT)	65.43	**50.06**	64.06	**45.0**2	**43.63**
	CG-DETR	**65.43**	48.38	**64.51**	42.77	42.86
	QD-DETR	62.4	44.98	62.52	39.88	39.86
	UniVTG	58.86	40.86	57.6	35.59	35.47
Zero-Shot	SD+C+SW	40.24	25.94	41.74	24.11	24.82
	SD+C+SW (w/PT)	42.12	27.89	43	24.68	25.5
	VC1+LSN (Ours)	**57.38**	**36.14**	**51.62**	**27.68**	**28.48**
	VC2+LSN(Ours)	50.69	33.36	42.93	27.68	25.19

5. Conclusions and future work

As a result of this project, we confirmed that by using the similarity on the frame embeddings produced by clip its possible to obtain a vector that carries the information with a fairly high degree of accuracy of the scene changes within the video. We also confirmed that is possible to improve the moment retrieval methods developed using CLIP Zero-Shot explored in the state of the art, while avoiding external tools to detect scene changes, and instead applying a preprocessing to the similarity vector.

We expected a better result on the method that exploits the continuous frame similarity array to determine the windows, on the other hand we observed that the effects of applying smoothing on the similarity array had a greater effect overall, especially on the window detection of the similarity comparison method, which surpassed our expectations.

Although we archived our initial goal to leverage CLIP Zero-Shot capabilities to obtain a model that its able to detect relevant moments within a video and improve the developed methods by doing a systematic assessment of the effect of preprocessing of the continuous similarity array. The results of the developed method still fall short when compared with the state-of-the-art models that couple a decoder on top of CLIP encoders.

On the positive side by applying our systematic assessment methodology, we develop a comprehensive study on the effect of different preprocessing techniques upon the similarity array, that to our knowledge, hasn’t been done before. Even though, the development of the pipeline for that systematic reproducible assessment has taken a longer than expected, robbing us from a precious time to explore some machine learning oriented techniques. By following our methodology and avoiding the development of a machine learning approach, we can confidently say that in general terms the ethical-social challenges are restricted, and directly linked, to the ones faced in the development of the CLIP.

In short, as output of the project, we produced a model that is capable of proposing segments of a video given a textual query, allowing the users to perform a search over the total length of the video, helping them to detect relevant moments in within. We also performed a comprehensive comparison of preprocessing techniques to better enhance Zero-shot methods that use similarity arrays as input features. In future works, we can expand of the preprocessing techniques, explore the effects of increasing the framerate on the proposed methods and implement a more machine learning oriented approach.

6. Glossary

CLIP: Contrastive Language–Image Pretraining
VLP: Video Lange Processing.
CNNs: Convolutional Neural Networks
ViT: Visual transformers
Cs : Similarity Vector
CFS$_{c}$: Continous frame Similarity vector

7. Bibliography

1. Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A Next-generation Hyperparameter Optimization Framework. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2623–2631. https://doi.org/10.1145/3292500.3330701

2. Cherti, M., Beaumont, R., Wightman, R., Wortsman, M., Ilharco, G., Gordon, C., Schuhmann, C., Schmidt, L., & Jitsev, J. (2022). *Reproducible scaling laws for contrastive language-image learning*. https://doi.org/10.48550/ARXIV.2212.07143

3. Diwan, A., Peng, P., & Mooney, R. J. (2022). *Zero-shot Video Moment Retrieval With Off-the-Shelf Models* (arXiv:2211.02178). arXiv. http://arxiv.org/abs/2211.02178

4. Feichtenhofer, C., Fan, H., Malik, J., & He, K. (2018). *SlowFast Networks for Video Recognition*. https://doi.org/10.48550/ARXIV.1812.03982

5. Gao, J., Sun, C., Yang, Z., & Nevatia, R. (2017). *TALL: Temporal Activity Localization via Language Query* (arXiv:1705.02101). arXiv. http://arxiv.org/abs/1705.02101

6. Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., & Ng, A. Y. (2014). *Deep Speech: Scaling up end-to-end speech recognition* (arXiv:1412.5567). arXiv. http://arxiv.org/abs/1412.5567

7. He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep Residual Learning for Image Recognition* (arXiv:1512.03385). arXiv. http://arxiv.org/abs/1512.03385

8. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90. https://doi.org/10.1145/3065386

9. Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324. https://doi.org/10.1109/5.726791

10. Lee, P., & Byun, H. (2023). *BAM-DETR: Boundary-Aligned Moment Detection Transformer for Temporal Sentence Grounding in Videos* (arXiv:2312.00083). arXiv. http://arxiv.org/abs/2312.00083

11. Lei, J., Berg, T. L., & Bansal, M. (2021). *QVHighlights: Detecting Moments and Highlights in Videos via Natural Language Queries* (arXiv:2107.09609). arXiv. http://arxiv.org/abs/2107.09609

12. Lei, J., Yu, L., Berg, T. L., & Bansal, M. (2020). *TVQA+: Spatio-Temporal Grounding for Video Question Answering* (arXiv:1904.11574). arXiv. http://arxiv.org/abs/1904.11574

13. Lin, D. C.-E., Heilbron, F. C., Lee, J.-Y., Wang, O., & Martelaro, N. (2022). *Videogenic: Video Highlights via Photogenic Moments* (arXiv:2211.12493). arXiv. http://arxiv.org/abs/2211.12493

14. Lin, K. Q., Zhang, P., Chen, J., Pramanick, S., Gao, D., Wang, A. J., Yan, R., & Shou, M. Z. (2023). *UniVTG: Towards Unified Video-Language Temporal Grounding* (arXiv:2307.16715). arXiv. http://arxiv.org/abs/2307.16715

15. Liu, Y., Li, S., Wu, Y., Chen, C. W., Shan, Y., & Qie, X. (2022). *UMT: Unified Multi-modal Transformers for Joint Video Moment Retrieval and Highlight Detection* (arXiv:2203.12745). arXiv. http://arxiv.org/abs/2203.12745

16. Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3431–3440. https://doi.org/10.1109/CVPR.2015.7298965

17. Mehrish, A., Majumder, N., Bhardwaj, R., Mihalcea, R., & Poria, S. (2023). *A Review of Deep Learning Techniques for Speech Processing* (arXiv:2305.00359). arXiv. http://arxiv.org/abs/2305.00359

18. Moon, W., Hyun, S., Lee, S., & Heo, J.-P. (2023). *Correlation-guided Query-Dependency Calibration in Video Representation Learning for Temporal Grounding* (arXiv:2311.08835). arXiv. http://arxiv.org/abs/2311.08835

19. Moon, W., Hyun, S., Park, S., Park, D., & Heo, J.-P. (2023). *Query-Dependent Video Representation for Moment Retrieval and Highlight Detection* (arXiv:2303.13874). arXiv. http://arxiv.org/abs/2303.13874

20. Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021). *Learning Transferable Visual Models From Natural Language Supervision* (arXiv:2103.00020). arXiv. http://arxiv.org/abs/2103.00020

21. Savitzky, Abraham., & Golay, M. J. E. (1964). Smoothing and Differentiation of Data by Simplified Least Squares Procedures.

Analytical Chemistry, *36*(8), 1627–1639. https://doi.org/10.1021/ac60214a047

22. Simonyan, K., & Zisserman, A. (2015). *Very Deep Convolutional Networks for Large-Scale Image Recognition* (arXiv:1409.1556). arXiv. http://arxiv.org/abs/1409.1556

23. Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, *13*(4), 600–612. https://doi.org/10.1109/TIP.2003.819861

24. Yang, A., Miech, A., Sivic, J., Laptev, I., & Schmid, C. (2022). *TubeDETR: Spatio-Temporal Video Grounding with Transformers*. https://doi.org/10.48550/ARXIV.2203.16434

25. Zhao, H., Shi, J., Qi, X., Wang, X., & Jia, J. (2017). *Pyramid Scene Parsing Network* (arXiv:1612.01105). arXiv. http://arxiv.org/abs/1612.01105

26. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*. http://arxiv.org/abs/1706.03762

27. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.