PONTIFICAL CATHOLIC UNIVERSITY OF MINAS GERAIS

Graduate Program in Informatics

Leonardo Vilela Cardoso

EXPLORING ATTENTION MECHANISMS AND HIERARCHICAL SUMMARIZATION IN VIDEO CAPTIONING

Belo Horizonte

2026

Leonardo Vilela Cardoso

EXPLORING ATTENTION MECHANISMS AND HIERARCHICAL SUMMARIZATION IN VIDEO CAPTIONING

Thesis presented to the Graduate Program in Informatics of the Pontifical Catholic University of Minas Gerais, as a requirement to obtain a Doctoral degree in Informatics.

Advisor: Prof. Dr. Zenilton Kleber Gonçalves do Patrocínio Júnior

Belo Horizonte

2026

FICHA CATALOGRÁFICA
Elaborada pela Biblioteca da Pontifícia Universidade Católica de Minas Gerais

Cardoso, Leonardo Vilela
C268e Exploring attention mechanisms and hierarchical summarization in vídeo captioning / Leonardo Vilela Cardoso. Belo Horizonte, 2026.
186 f. : il.

Orientador: Zenilton Kleber Gonçalves do Patrocínio Júnior

Tese (Doutorado) - Pontifícia Universidade Católica de Minas Gerais. Programa de Pós-Graduação em Informática

1. Gravações de vídeo - Resumos. 2. Legendas (Cinema, televisão, etc.) – Qualidade. 3. Processamento de imagens – Técnicas digitais. 4. Redes neurais (Computação). 5. Aprendizado do computador. 6. Algorítmos. 7. Teoria dos grafos. I. Patrocínio Júnior, Zenilton Kleber Gonçalves do. II. Pontifícia Universidade Católica de Minas Gerais. Programa de Pós-Graduação em Informática. III. Título.

CDU: 681.3.093

Ficha catalográfica elaborada por Fabiana Marques de Souza e Silva - CRB 6/2086

Leonardo Vilela Cardoso

EXPLORING ATTENTION MECHANISMS AND HIERARCHICAL SUMMARIZATION IN VIDEO CAPTIONING

Thesis presented to the Graduate Program in Informatics of the Pontifical Catholic University of Minas Gerais, as a requirement to obtain a Doctoral degree in Informatics.

Prof. Dr. Zenilton Kleber Gonçalves do Patrocínio Júnior

Prof. Dr. João Paulo Papa

Prof. Dr. David Menotti Gomes

Prof. Dr. Silvio Jamil Ferzoli Guimarães

Profa. Dra. Cristiane Neri Nobre

Belo Horizonte, February 12, 2026.

ACKNOWLEDGMENTS

I thank God for the blessing of completing this stage. There are no words to express my gratitude for the support of my wife, Joice, and my children, Miguel and little Micael, who was born during my doctoral studies, and who were the motivation throughout this entire journey to complete this challenge. The joy on their faces when we meet at the end of the day has always been the fuel to continue, and overcoming moments of weakness has become commonplace.

To the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) for the financial support and encouragement for research, which is so necessary for technological and scientific innovations.

These four years have been complex, but the path has become a polynomial and optimal solution with the help of my advisor, Zenilton. The conversations and advice went beyond mere guidance and gave me direction, both personally and professionally. Finally, I thank my laboratory colleagues and participants in the Postgraduate Program at the Pontifical Catholic University for the conversations and discussions that generated valuable moments of reflection.

Thank you all very much!

"Walk on, walk on
With hope in your heart
And you'll never walk alone
You'll never walk alone"

Gerry & The Pacemakers

ABSTRACT

The addition of attention mechanisms can improve the quality of descriptions generated in the video captioning task, promoting the creation of more coherent paragraphs. However, in long videos, information relevant to the context may be discarded, resulting in redundant descriptions that compromise the quality of the sentences produced. To mitigate this problem, it is possible to employ complementary techniques that smooth the reduction imposed by the video sampling process. While sequential approaches do not consider the temporal distribution of information, summarization-based methods evaluate the semantic importance of the content, prioritizing the selection of a larger number of significant events. In this context, the use of summarization as a pre-processing step is still little explored in the video captioning task. However, different summarization strategies tend to generate different results. To investigate this aspect, this work applies different summarization techniques in the selection of frames, dynamically or statically, in order to assist the transformer enhanced with adaptive attention mechanisms. Three new approaches are proposed: Adaptive Transformer, **H**ierarchical time-aware **S**ummarization with an **A**daptive **T**ransformer (HSAT), and SkimCap. The Adaptive Transformer model adopts a sequential frame selection policy; HSAT employs a graph hierarchy to select a fixed number of key frames representative of the video; and SkimCap presents variations in its selection policy, being trained with features obtained by dynamic hierarchical clustering and tested with frames chosen by the same strategy or by a supervised model trained on the SumMe dataset. Unlike traditional approaches, which depend on uniform sampling or predefined temporal segments, SkimCap performs unsupervised hierarchical clustering to identify and extract semantically relevant scenes. These condensed representations offer a compact yet information-rich input, enabling the generation of more accurate and contextualized captions. The model was evaluated in the ActivityNet dataset, achieving scores for CIDEr-D of 25.44, BLEU@4 of 10.77, and Repetition@4 of 5.84, indicating consistent improvements in caption quality and relevance. An ablation study confirms the effectiveness of hierarchical summarization as a feature selection mechanism, highlighting its contribution to overall performance. SkimCap thus establishes a new direction for incorporating structured visual summarization into end-to-end captioning systems.

Keywords: Video captioning; Memory-augmented transformer; Attention mechanism; Video summarization.

RESUMO

O acréscimo de mecanismos de atenção pode aprimorar a qualidade das descrições geradas na tarefa de video captioning, promovendo a criação de parágrafos mais coerentes. No entanto, em vídeos longos, informações relevantes para o contexto podem ser descartadas, resultando em descrições redundantes que comprometem a qualidade das sentenças produzidas. Para mitigar esse problema, é possível empregar técnicas complementares que suavizam a redução imposta pelo processo de amostragem dos vídeos. Enquanto abordagens sequenciais não consideram a distribuição temporal das informações, métodos baseados em sumarização avaliam a importância semântica do conteúdo, priorizando a seleção de um maior número de eventos significativos. Neste contexto, o uso de sumarização como etapa de pré-processamento ainda é pouco explorado na tarefa de video captioning. Contudo, diferentes estratégias de sumarização tendem a gerar resultados distintos. Para investigar esse aspecto, este trabalho aplica diferentes técnicas de sumarização na seleção de frames, de forma dinâmica ou fixa, a fim de auxiliar o transformador aprimorado com mecanismos adaptativos de atenção. São propostas três novas abordagens: Adaptive Transformer, HSAT e SkimCap. O modelo Adaptive Transformer adota uma política sequencial de escolha de frames; o HSAT emprega uma hierarquia de grafos para selecionar um número fixo de quadros-chave representativos do vídeo; e o SkimCap apresenta variações em sua política de seleção, sendo treinado com features obtidas por agrupamento hierárquico dinâmico e testado com frames escolhidos pela mesma estratégia ou por um modelo supervisionado treinado no conjunto SumMe. Diferentemente das abordagens tradicionais, que dependem de amostragem uniforme ou de segmentos temporais pré-definidos, o SkimCap realiza agrupamento hierárquico não supervisionado para identificar e extrair cenas semanticamente relevantes. Essas representações condensadas oferecem uma entrada compacta, porém rica em informações, permitindo a geração de legendas mais precisas e contextualizadas. O modelo foi avaliado no conjunto de dados ActivityNet, alcançando pontuações de 25.44 em CIDEr-D, 10.77 em BLEU@4 e Repetition@4 de 5.84, indicando melhorias consistentes na qualidade e relevância das legendas. Um estudo de ablação confirma a eficácia da sumarização hierárquica como mecanismo de seleção de características, destacando sua contribuição para o desempenho geral. O SkimCap estabelece, assim, uma nova direção para a incorporação de sumarização visual estruturada em sistemas de legendagem ponta a ponta.

Palavras-chave: Video captioning; Memory-augmented transformer; Attention mechanism; Video summarization.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

AM Attention Mechanism

ANC ActivityNet Captions

ANN Artificial Neural Network

BiLSTM Bidirecional Long Short-Term Memory

BLEU Bilingual Evaluation Understudy

BoW Bag of Words

CIDEr Consensus-based Image Description Evaluation

CNN Convolutional Neural Network

DL Deep Learning

DNN Deep Neural Network

DVC Dense Video Captioning

FPS Frames Per Seconds

GRU Gated Recurrent Unit

GT Ground Truth

hGB Hierarchical Graph Based

HieTaSkim Hierarchical Time-aware Skimming

HieTaSumm Hierarchical Time-aware Summarization

HSAT **H**ierarchical time-aware **S**ummarization with an **A**daptive **T**ransformer

HC Histogram of Colors

KF KeyFrame

KDE Kernel Density Estimation

LSTM Long Short-Term Memory

ML Machine Learning

MT Machine Translation

METEOR Metric for Evaluation of Translation with Explicit Ordering

MSF minimum spanning forests

MST Minimum Spanning Tree

NLP Natural Language Processing

PoC Proof of Concept

QFZ Quasi-Flat Zones

ReLU Rectified Linear Units

RL Reinforcement Learning

RNN Recurrent Neural Network

ROUGE-L Recall-Oriented Understudy for Gisting Evaluation - Longest Common Subsequence

S2S Setence-To-Setence

SL Supervised Learning

USL UnSupervised Learning

VC Video Captioning

VFC Video Frame Captioning

LIST OF SYMBOLS

$\odot$ - Hadamard product

$|.|$ - Cardinality

β - Weight of recall in the F-measure

γ - Parameter denoting the allowed variability

δ_t - Time threshold of a graph

$\mu_w(e)$ - Average in the connected component

$\sigma_w(e)$ - Standard deviation in the connected component

φ and ψ - activation functions

$\Phi_{\mathcal{H}}$ - Saliency map of a hierarchy

ω_z - Input weight matrix cell input

ω_i - Input weight matrix input gate

ω_f - Input weight matrix forget gate

ω_o - Input weight matrix output gate

$\mathbb{A}$ - Set of frames of a video

AS - Automatic Summary

b_c^l, b_a^l, and b_z^l - Trainable bias

b_z, b_i, b_f, and b_o - Bias terms

c - Memory cell of mLSTM

$\mathcal{C}$ - Horizontal cuts of a hierarchy

C - Memory matrix of mLSTM

$CC(G_i)$ - Set of connected components of G_i

c_t - multiple scalar memory cells

C_t^l - Internal cell state

$d(f)$ - Video global descriptor

$\mathcal{D}(d(f_{t_1}), d(f_{t_2}))$ - Dissimilarity function

d - Embedding size

E - Set of edges of a graph

$|E|$ - Number of edges in a graph

E_δ - Set of edges of a similarity graph

$|E_\delta|$ - Number of edges in a similarity graph

E_T - Set of edges of a spanning tree

f - A Frame of a video

$\mathbf{F}(e)$ - Equilibrium measure function

FN - False Negative

FP - False Positive

f_{t_1} and f_{t_2} - Video frames

$f(x, y)$ - Color value

$\mathcal{G}$ - Sequence of subgraphs

$\mathbf{G}$ - Dynamic fusion gate

G - Graph

(G, w) - Weighted graph

(G_δ, w) - Time-aware frame similarity graph

$\mathcal{H}$ - Hierarchy

h - Parallel instances of scaled dot-product attention

H - Height of a frame

$\mathbf{H}^{(se)}$ - Enhanced focus on informative channels

$\tilde{H}_t^l$ - Intermediate hidden state vector

H_{video}^0 - Encoded video

H_{text}^0 - Encoded text

H^0 - Transformer input

h_{t-1} - hidden state

i - Index

K - Key matrix of the transformer

l - layer in step t

L_2 - L_2 norm between deep features extracted from video frames

m_A - Number of matching keyframes generated from the Automatic Summary

$\overline{m}_A$ - Non-matching keyframes

MH_t^l - Multi-head attention results

(MST) T_G^* - Minimum spanning tree

m_t - Additional gate stabilization state of LSTM

M_t^l - Memory state

M_{t-1}^l - Last memory position

$M(X, Y)$ - Maximum matching between two sets of different elements X and Y

N - Number of frames in a video

$\mathbb{N}^2$ - Dimensions of a video

$\mathcal{NC}$ - Number of keyshots

n_U - Number of keyframes selected for the user to represent the user summary

P - Precision

p - Maximum relative size (set by the user) of the final skim to the original video

$\mathbf{P}$ - nonempty disjoint subsets of V whose union equals V

Q - Query matrix of the transformer

R - Recall

r_z, r_i, r_f, and r_o - recurrent weight matrices

$\mathcal{S}$ - Maximum size of each sequence of frames

S - Sequence of minima of w

s_t - Attention focus

$\mathbb{T}$ - Set of frames in a video

tanh - hyperbolic tangent function

T_c - Embedding lengths

T_G - Spanning tree

T_m - Memory length

TN - True Negative

TP - True Positive

T_{text} - Text lengths

T_{video} - Video lengths

t_1 and t_2 - Location of the frame's position

U - User summary

$\mathcal{V}$ - Value matrix of the transformer

V - Set of nodes of a graph

$|V|$ - Number of nodes in a graph

V_N - A sequence of frames on the $\mathbb{A} \times \mathbb{T}$ domain

x_t - input vector of LSTM

(x, y) - Pixel location

W - Width of a frame

w - Weight function

$w(T_G)$ - Weight of a spanning tree

W_i^Q, W_i^K, $W_i^\mathcal{V}$, and W^O - Linear projections

W_{mc}^l, W_{sc}^l, W_{mz}^l, W_{sz}^l, W_{mhl}^l, and W_{mhr}^l - Trainable weights of the Transformer

Z_t^l - Update gate

CONTENTS

1 INTRODUCTION

Video captioning is the task of concisely describing a video through text (LEI et al., 2020; CARDOSO; GUIMARÃES; PATROCÍNIO JR, 2021). However, a significant challenge within this realm revolves around effectively capturing the video's essence, particularly when constructing a Ground Truth (GT) involving inputs from multiple annotators (KRISHNA et al., 2017). This collaborative GT often encompasses a range of perspectives, resulting in an emphasis on diverse moments throughout the video. It's important to note that the efficacy of video captioning outcomes is intricately linked with the successful execution of two interconnected sub-tasks: (i) the identification of temporal events, and (ii) the subsequent generation of descriptive content.

Videos can be defined by the overlap of visual and acoustic content. The visual domain transmits information through time-sampled frames, which fill temporal gaps to compose a scene. A scene contains information related to location, action, and objects or people; this data can be translated into compact descriptions that convey the present content without losing the transmitted message. Thus, it is possible to add value to the object in focus with environmental information, giving meaning to the described scene in the video and transforming the entire visual content into text (LI; GONG, 2019; BELO et al., 2016).

Data on the increasing number of users and hours per minute on streaming platforms such as Google's YouTube highlight the need for methods to process visual content. According to Youtube (2020), the platform has over 2 billion active users, up from approximately 1 billion in 2014 (CRANWELL et al., 2015). According to Omnicore (2020), in 2019, the service received around 500 hours of video per minute, compared to only 300 hours per minute in 2015 (SILVA et al., 2017). In December 2024, Youtube (2024) reported that there are over 20 billion videos available on the platform and a total of 20 million videos published per day by users.

Based on these data, coupled with the amount of information per video, the growth of video content available on the Internet becomes notable, which will eventually be accessed by active network users. Consequently, some research areas in the literature direct their projects towards processing this plethora of information, enabling methods to convert this visual content into text through video descriptions (LI; GONG, 2019; XU et al., 2015).

In this context, it becomes feasible to process large quantities of frames, which are

initially connected sequentially in time and reduced through summarization techniques, enabling more coherent divisions of content. Different summarization methods can result in varying data outputs. This is evident when considering random frame selection, where random sampling arbitrarily selects the occurrence of shots, while hierarchical application enables the definition of uniform and homogeneous segments, resulting in coherent cuts. Thus, the expected outcome of the description is likely to be impacted by summarization due to the guidelines and steps of each method for shot selection on a graph-based approach (BELO et al., 2016).

Video event detection can be accomplished through three main strategies: (i) random selection; (ii) time-sliding window; and (iii) scene (or shot) detection. Irrespective of the method employed, the underlying aim is to refine the video's content, facilitating the subsequent caption generation by highlighting the most pertinent segments of the video. This process gives rise to two primary types of video summaries: static video summaries, comprised of keyframes, and dynamic video summaries, assembled from keyshots. Consequently, the challenge lies in meticulously selecting video frames (or shots) that comprehensively encompass the video's narrative without omitting any crucial content (AAFAQ et al., 2019; LEI et al., 2020; CARDOSO; GUIMARÃES; PATROCÍNIO JR, 2021).

The intricate task of video summarization has captivated researchers for decades, resulting in numerous attempts to address its complexities. Notably, the literature has witnessed various surveys focused on video summarization, further highlighting the significance of this initiative (MEENA; KUMAR; Kumar Yadav, 2023; NARWAL; DUHAN; Kumar Bhatia, 2022; APOSTOLIDIS et al., 2021a). Video summarization can serve as a potent input for the subsequent video caption generation step, as it possesses the potential to construct an informative overview of the video, characterized by maximum representativeness, minimal redundancy, and optimal diversity (MEENA; KUMAR; Kumar Yadav, 2023).

1.1 Motivation

One of the most significant aspects of human vision is the ability to focus on different actions and moments. The visual adaptation to the smallest stimulus is instantaneous and varies among individuals. However, when describing what occurs in front of them, it becomes necessary to shift the point of analysis, and with each new word, the gaze becomes fixed, almost paralyzed, on a single point (XU et al., 2015).

As with humans, video description can base the result of video analysis on specific points. Thus, levels of attention can be applied, with the considered stochastic level resembling a gaze fixed on each detail of an old photograph. New details are discovered

Figure 1 – An example of the result obtained by the method HSAT

HSAT
A camera pans around a large group of people sitting on a bus and leads into people riding on bikes. Several shots are shown of people riding in the water as well as swimming around the area. More clips are shown of people swimming around the ocean as well as swimming around the ocean.

Ground-Truth
Several shots are shown of a man speaking to various groups of people and leads into people wearing wet suits and walking. The people walk down a beach and are seen swimming around in the water. More shots of fish are shown and ends with the people walking out of the water and high fiving the camera man.

Source: Elaborated by the author

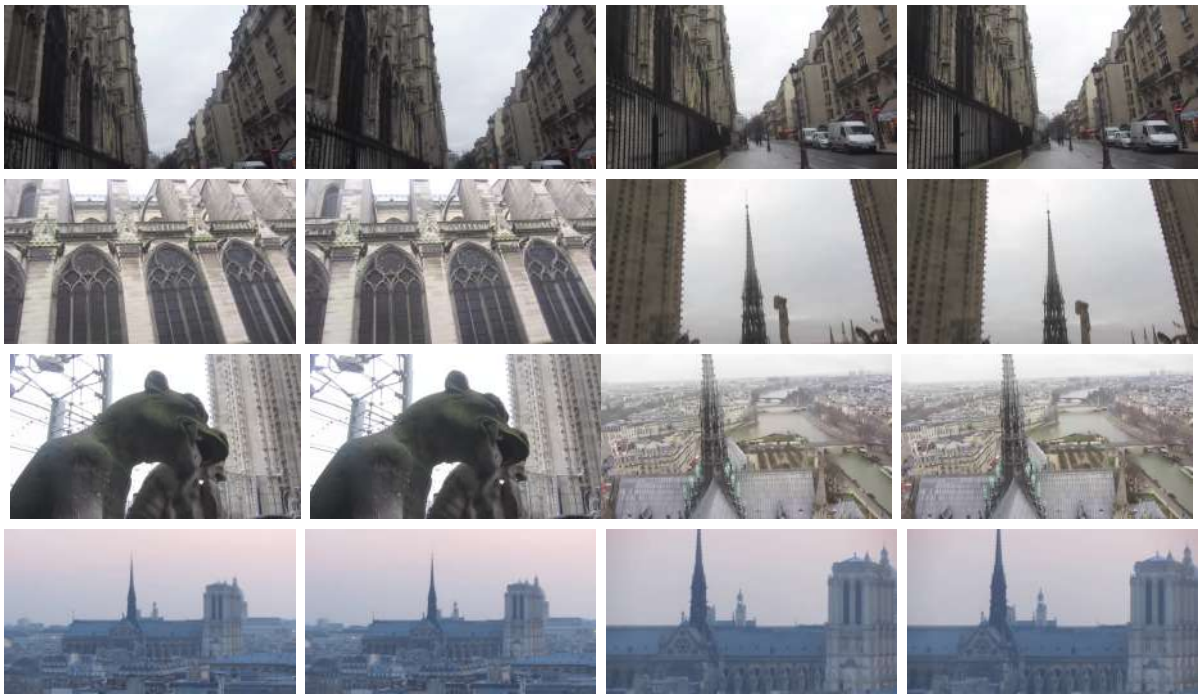
due to the extracted level of detail, and with higher levels of detail, new aspects may emerge. Consequently, the level of detail in Attention Mechanism (AM) is defined by the amount of available information (XU et al., 2015; WANG et al., 2018; LI; GONG, 2019).

The interaction between humans and their environment characterizes the learning process, which is marked by constant expectation and experimentation. Through trial and error, the nervous system, since its earliest stages, learns by sampling, generating a new behavior pattern with each new experience. However, in some cases, a single sample is insufficient to solidify learning, necessitating practice to reinforce and make the experience enduring (SUTTON; BARTO, 2018).

Regarding description generation, transformers (VASWANI et al., 2017) have recently shown to be very useful for many sequence-related tasks, such as machine translation (VYDANA et al., 2021), information retrieval (YATES; NOGUEIRA; LIN, 2021), text classification (GUO et al., 2020), document summarization (ZHANG; WEI; ZHOU, 2019), image classification (CHEN; FAN; PANDA, 2021), image captioning (PAN et al., 2020; HUANG et al., 2019), video captioning (LEI et al., 2020; CARDOSO; GUIMARÃES; PATROCÍNIO JR, 2021), and others tasks (TANG et al., 2020). In video captioning, the authors of (LEI et al., 2020) proposed a memory-augmented transformer to cope with text repetition, while Cardoso, Guimarães and Patrocínio Jr (2021) explored a re-weighting of the importance of data in the memory module and the self-attention module, to directly influence the amount of information used by a memory-augmented transformer to learn.

Figure 1 shows an example of the results obtained by one of the proposed methods, along with the expected GT description with a static summary. It is easy to observe the high correlation among frames, which also appears in the GT result. Unlike other methods

Figure 2 – Example of a "large number of transitions and perspective changes" in a video from the SumMe dataset.



Source: Elaborated by the author

that tend to present the same sentence several times, the use of attention mechanisms made in this proposal enhances coherence among generated sentences and the numerous events within a video. Thus, the final description can adapt even with the presence of similar events and be more concise, meaningful, and intelligible. In addition, Figure 1 demonstrates the great difficulty in describing the characteristics of the database, since the description of daily activities in a real scenario is often done in situations with few variations of information, and the modifications are usually not enough to differentiate the agent due to video characteristics such as perspective and distance.

Videos have a high amount of information, and at times, the arrangement of similar data is recurrent, and even with the reduction in frames in a sampled way, similar data can be repeated, an action that does not generate added value for the base. Analyzing the structure proposed by minimum spanning trees, the transformation of similar information into grouped data becomes clear, but it turns out that unweighted tactics lose the temporal sensitivity of the occurrence of facts in the video; it turns out that thresholds can be applied to reduce this occurrence. Thus, segmentation techniques based on partitioning seem like candidates to add value, reduce the similarity between data, and reduce the amount of information lost due to neglecting data distance in time (BELO et al., 2016; MARTINS et al., 2020).

Another possibility that has gained prominence is the selection of important frame

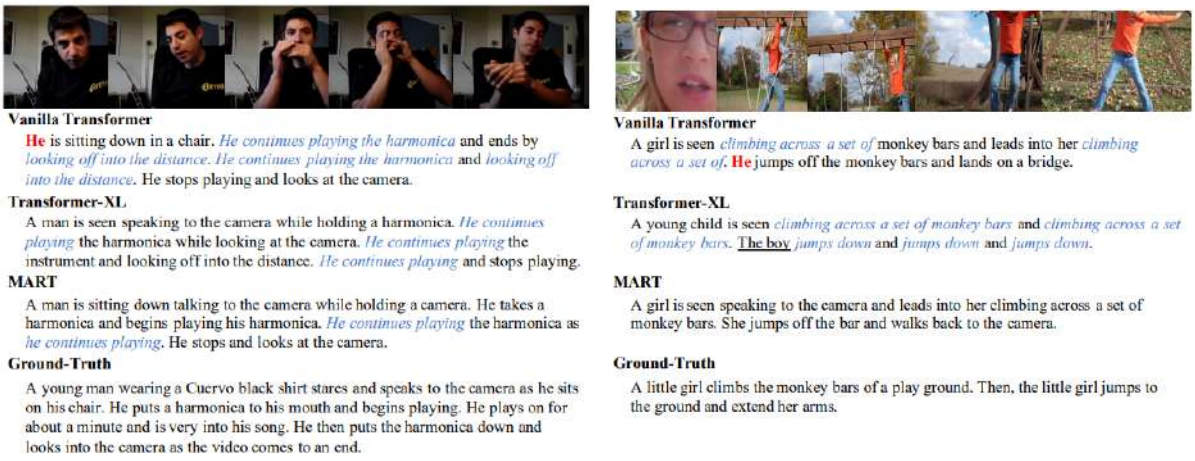
sequences and generating a summary with temporal connection and movement between frames (VIVEKRAJ; DEBASHIS; BALASUBRAMANIAN, 2019). In this case, it is necessary to process the video and produce a group of frames representing each scene that appears throughout the video. The generation of groups of frames representing scenes is called *dynamic video summarization* or video skimming, and it processes each scene as a skim (VIVEKRAJ; DEBASHIS; BALASUBRAMANIAN, 2019). The grouping of skims returns a more representative result that allows the perception of changes in the represented scenes. This process requires a greater number of frames. While more computationally intensive than static summarization, it avoids the need to process the entire video and better preserves the evolution of visual content (TIWARI; BHATNAGAR, 2021). To address both local and global temporal dependencies in video sequences and thanks to the recent advances involving the Extended Long Short-Term Memory (xLSTM) (BECK et al., 2024), one can argue if it could be employed to extract the most informative and representative frames from a video, given the sequential nature of video data.

The Long Short-Term Memory (LSTM) (HOCHREITER; SCHMIDHUBER, 1997) manages to overcome the vanishing gradient problem of Recurrent Neural Networks (RNNs) and has been successfully applied to various domains until the appearance of the Transformer. Its effectiveness has been demonstrated in numerous sequence-related tasks, but it has three main limitations: (i) inability to revise storage decisions; (ii) limited storage capacities; and (iii) lack of parallelization. Recently, the extended Long Short-Term Memory (xLSTM) (BECK et al., 2024) has emerged as a powerful tool for capturing temporal dependencies, performing favorably in natural language processing (NLP) tasks compared to state-of-the-art Transformers and State Space Models, both in performance and scaling. Figure 2 presents a difficult scenario with a video containing a large number of transitions and perspective changes. By leveraging past observations to anticipate future frames, LSTM-based models can effectively capture structural and semantic patterns, facilitating the selection of keyframes that represent the most salient video aspects. This predictive ability enables the identification of abrupt scene transitions, motion dynamics, and recurring visual elements, all of which are critical for generating concise yet informative summaries.

1.2 Problem

A critical concern within video captioning pertains to mitigating frame similarity, especially among temporally close frames employed in caption generation. Addressing this issue is crucial to diminishing redundancy in the ultimate output. Simultaneously, it is necessary to maintain temporal coherence between the video content and the generated text, which may necessitate processing and describing a larger number of frames,

Figure 3 – Example of Literature Methods for the ActivityNet Captions dataset, in which blue/bold indicates the presence of repetition and red/bold indicates a possible pronoun different from the GT.



(a) (b)

Source: Lei et al. (2020)

potentially affecting the final output's quality adversely. To ameliorate these challenges, attention mechanisms come into play. These mechanisms enable the generation of attention distributions, prioritizing video segments of greater significance while potentially downplaying others, thus enhancing the efficacy of video captioning. Traditional attention approaches, however, tend to distribute attention uniformly across all feasible regions. However, highlighting regions might also incur some failures and omissions (AAFAQ et al., 2019; DAI et al., 2019). Therefore, some methods reduce the frames into smaller distributions to check local interest and reassess the importance of features. These strategies avoid learning data with little significance for the final result. By doing that, unobserved semantic aspects can be explored (DEVLIN et al., 2018).

In the process of crafting coherent paragraphs for video captions, a notable problem arises where sentences often show significant similarities. This redundancy problem interferes with the comprehensive understanding of the described content, representing a disadvantage for the generation of discriminative content. Figure 3 shows two examples of methods taken from the literature and their results, demonstrating a high repetition occurrence and some context loss. In addition, results tend to lose the continuity between the events. The Vanilla Transformer is the method with the highest repetition and loss of context. The Transformer-XL produces repeated sentences but tends to be better at information continuity. The MART method had the most similar result to GT and a low rate of repetition, but repetition occurs in one case. Therefore, these methods have problems, and this work seeks to improve those results, reducing the occurrence of repetition, and loss of context, maintaining the continuity of sentences, and being as close as possible to the GT.

Thesis statement: *The concepts of video representations as graph-based structures that consider time to generate a hierarchical arrangement, along with a deep learning methodology can be used together to automatically propose descriptions to be used in a scenario that uses static or dynamic summarization as a way of guaranteeing observations of points that are not perceptible in sequential approaches, producing concise descriptions, resulting in a reduction in the repetition of terms. It is also expected that the proposed attention mechanisms will have an impact on reducing repetition due to the reweighting and observation of specific points in the frames. Furthermore, while improving the quality of descriptions, the combination of attention models proposed for inclusion in transformers with summarization techniques that evaluate temporal coherence, whether static or dynamic, tends to reduce the computational cost due to the reduction of frames that must be processed.*

1.3 Questions and Hypotheses

During the development of this work, it is intended to answer three questions that give rise to three research hypotheses. They are:

Question 1 – What is the impact of different video summarization techniques on the final result to generate a dense paragraph in the video captioning task?

Hypothesis 1 – Considering that different summarization techniques can produce different results and that distinct aspects/events present in videos can be explored, modifying the importance of the relationship between frames. Thus, varying summarization techniques can allow observing different aspects/events in videos.

Question 2 – Does a hierarchical video summarization approach as a component of the video captioning task impact differently from other strategies such as sequential selection of keyframes (or keyshots)?

Hypothesis 2 – Sequential techniques make it possible to evaluate a set of frames that occur temporally interconnected; however, as it is necessary to limit the number of frames processed, some aspects are ignored due to their occurrence after a time limit. On the other hand, techniques that use keyframes (or keyshots) to describe content may fail to capture temporal changes present in some scenes. Both issues can be addressed by a hierarchical video summarization method that maintains temporal consistency, captures the changes in video aspects/events more accurately, and describes the variations in video perspectives.

Question 3 – Is it possible to adopt a more sophisticated attention mechanism to improve the behavior of a transformer applied to video captioning?

Hypothesis 3 – Attention mechanisms are used in video captioning techniques to capture variations in perspectives and reweight learned features. Traditional methods tend to represent the most important regions more comprehensively, but the application of sophisticated methods appears to be a possibility for improving the quality of the regions of interest. In this way, improving the attention previously applied to transformer-based mechanisms can directly impact the improvement of the descriptions produced.

1.4 Goals

The main idea of this study is to assess the importance of data with attention mechanisms (in video segments) to improve readability while minimizing redundancy. However, in scenarios involving multiple events, there is a propensity for repeated sentence fragments, particularly in event-based descriptions, in which correlated segments increase the likelihood of duplicate (or highly similar) sentence components. Although certain methods attempt to resolve this issue, it remains a non-trivial challenge. This challenge depends on the complex evaluation of the interaction between a piece of text and another text generated simultaneously to avoid the repetition of the event described in the final result.

1.4.1 Specific Goals

The specific goals of this work are:

a) Develop, validate, and test a prototype to demonstrate the effectiveness of applying different summarization methods on the selection of frames that cover aspects/events previously neglected, as expected by Hypothesis 1;

b) Use a hierarchical method to create a video summary that respects the temporal coherence of the video and represents the disjoint segments (or not) in a video shot, to prove Hypothesis 2;

c) Implement an attention-based approach to evaluate crucial information that can improve descriptions, especially when it appears sporadically but has great importance at multiple points in the video, according to Hypothesis 3;

d) Generate models that are capable of being used in video captioning based on static video summarization and dynamic video summarization (or skimming); and

e) Analyze the experimental test results aiming to validate the generated methods with the metrics used in the literature, and compare them to the state-of-the-art.

1.5 Justification

Video captioning methods emerged as a way to assist in video processing. The automatic generation of texts for videos is a way to increase the understanding of the content described in the videos. Thus, one of the biggest applications of video captioning is content summarization due to the large amount of frames with little or no modification. Therefore, the description generated must represent a compilation of important information in the video, distributed in keyframes or keyshots, allowing the user to understand the content presented by the video with a quick analysis of the description without the need to watch the entire video in a graph-based approach (SHAO; SHEN; ZHOU, 2008; AAFAQ et al., 2019).

In addition, they can be applied in information retrieval techniques, as they produce relevant textual information directly linked to the video content, enabling the location of different aspects present in the scenes.

On the other hand, several works discuss the application of video captioning techniques to increase social inclusion, as it facilitates the understanding of the content for people with different types of visual and cognitive disabilities. As a result, the generated description can be used in applications that convert, for example, text into audio, making it easier for blind people to perceive part of the content displayed in the video. According to Li and Gong (2019), one of the great potentials in improving video description methods is the possibility of social inclusion for people with little or no vision. This can be explained by the enhanced capabilities of information retrieval techniques, which help mitigate such difficulties by expanding the range of retrievable video details and promoting greater interaction with technology.

Aafaq et al. (2019) say that:

> The advancement of video description opens up enormous opportunities in many application domains. It is envisaged that in the near future, we will be able to interact with robots in the same manner as humansRohrbach et al. (2013). If video description is advanced to the stage of being able to comprehend events unfolding in the real world and render them in spoken words, then Service Robots or Smartphone Apps will be able to understand human actions and other events to converse with humans in a much more meaningful and coherent manner. For example, they could answer a user's question as to where they left their wallet or discuss what they should cook for dinner.

To enhance the results obtained, methods based on AM evaluate the distribution related to the content under analysis. Unlike edge sampling or salience techniques, these methods assess which part is the optimal point for composing the description of each word. However, this approach is only feasible if the evaluation considers the probability

distributions provided by the generated model (SHAO; SHEN; ZHOU, 2008; XU et al., 2015; GAO et al., 2017; LI; GONG, 2019).

1.6 Main Contributions

This work validates the hypotheses that guide the approach of this work to video captioning. The main contributions are:

- An extension of the shared Memory-Augmented Recurrent Transformer. Unlike the original model, which selects features sequentially and may overlook later information, this approach incorporates (i) an unsupervised graph-based strategy for static or dynamic summarization, and (ii) a supervised Long Short-Term Memory (LSTM)-based method to generate summaries dynamically. The static strategy extracts sparsely distributed key content, while the dynamic strategy captures temporal transitions, producing coherent video skims.
- An extended transformer architecture equipped with a new attention mechanism, Adaptive Attention, which reweights previously applied attention scores to emphasize salient features. Experimental results show that this mechanism matches or surpasses the performance obtained using frames selected by the proposed summarizers.
- A novel graph-based summarization model for selecting keyframes or keyshots. By employing hierarchical partitioning techniques, the model captures temporal patterns and important content that are often overlooked by sequential or uniform strategies.
- Two supervised video summarization approaches for estimating the importance of frames in each video segment. These methods incorporate user annotations to guide training and integrate diverse viewpoints, enabling the detection of patterns that would otherwise remain unnoticed.
- A transformer architecture designed to process the keyframes or keyshots selected during summarization, allowing the model to attend to frames that sequential strategies may miss due to temporal constraints.
- Two recurrent blocks that capture temporal dynamics and key events, enabling the generation of high-quality supervised video summaries.

1.7 Document Organization

The remainder of this text is organized into six chapters. Chapter 2 reviews concepts related to video summarization, memory-augmented recurrent transformers, and traditional LSTM methods. Chapter 3 discusses related work on video summarization, video captioning, attention mechanisms, and transformer models. Chapter 4 presents the summarization task as a precursor to video captioning and describes the additional attention modules used in the proposed method. Chapter 5 reports the experimental results and provides analysis. Finally, Chapter 6 presents concluding remarks and discusses potential extensions.

2 THEORETICAL BACKGROUND

The machine learning area has experienced rapid advancements, significantly impacting various domains such as computer vision, natural language processing, and autonomous systems. Central to these advancements are sophisticated algorithms capable of learning from data, thereby enabling systems to make predictions, recognize patterns, and improve performance over time. This Chapter provides a comprehensive overview of the theoretical foundations and practical applications of video captioning, highlighting their architecture.

2.1 Video

Let $\mathbb{A} \subset \mathbb{N}^2$, where $\mathbb{A} = \{0, \ldots, \mathrm{H}-1\} \times \{0, \ldots, \mathrm{W}-1\}$, with H and W representing the width and height of each frame, respectively. Let $\mathbb{T} \subset \mathbb{N}$, where $\mathbb{T} = \{0, \ldots, N-1\}$, and N denotes the number of frames in a video. A frame f is a function from $\mathbb{A}$ to $\mathbb{R}^3$, where for each spatial position (x, y) in $\mathbb{A}$, $f(x, y)$ represents the color value at pixel location (x, y). A video V_N, defined on the domain $\mathbb{A} \times \mathbb{T}$, can be viewed as a sequence of frames f. This can be expressed as $\mathrm{V}_N = (f)_{t \in \mathbb{T}}$, where N is the number of frames in the video.

A frame f is typically described using a global descriptor $d(f)$. Consider two video frames, f_{t_1} and f_{t_2}, located at positions t_1 and t_2, respectively. The (dis)similarity between f_{t_1} and f_{t_2} can be assessed through a distance measure $\mathcal{D}(d(f_{t_1}), d(f_{t_2}))$ between their descriptors. Various options exist for $\mathcal{D}(d(f_{t_1}), d(f_{t_2}))$, the distance measure between two frames based on the global descriptor, such as histogram/frame difference, histogram intersection, difference of histogram means, and even the L_2 norm between deep features extracted from video frames.

2.2 Graph and Hierarchy of Partitions

Let $G = (V, E)$ be an graph in which V is a finite set of vertices and E a finite set of (undirected) edges defined by $E \subseteq \{\{u, v\} \subseteq V \mid u \neq v\}$. A weighted graph (G, w) is a pair of a graph and a weight function $w : E \mapsto \mathbb{R}$ that assigns a value to each edge of G. Moreover, a spanning tree $T_G = (V, E_T)$ of (G, w) is a connected acyclic subgraph of G, in which $E_T \subseteq E$ and the weight of T_G to be equal to the sum of weights of all edges in E_T, denoted as $w(T_G) = \sum_{e \in E_T} w(e)$. The minimum spanning tree (MST) T_G^* is defined

as a spanning tree of (G, w) with minimal weight.

Given a finite set V (e.g., the set of vertices of a graph), a partition of V refers to a set $\mathbf{P}$ containing nonempty disjoint subsets of V whose union equals V. Each element of $\mathbf{P}$ is termed a region of $\mathbf{P}$. For two partitions $\mathbf{P}$ and $\mathbf{P}'$ of V, $\mathbf{P}'$ is deemed a (total) refinement of $\mathbf{P}$, symbolized by $\mathbf{P}' \preceq \mathbf{P}$, if every region of $\mathbf{P}'$ is encompassed within a region of $\mathbf{P}$. A hierarchy (on V) of depth ℓ is a sequence $\mathcal{H} = (\mathbf{P}_0, \ldots, \mathbf{P}_\ell)$ of partitions of V such that $\mathbf{P}_{i-1} \preceq \mathbf{P}_i$ for all $i \in \{1, \ldots, \ell\}$. A hierarchy $\mathcal{H}$ is called complete if $\mathbf{P}_\ell = \{V\}$ and if $\mathbf{P}_0$ contains every singleton of V (i.e., $\mathbf{P}_0 = \{\{x\} \mid x \in V\}$).

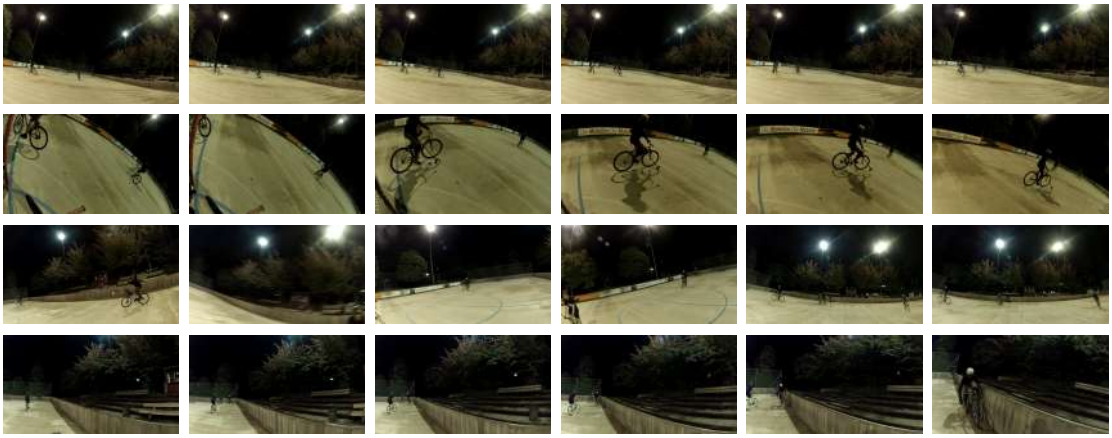
Given a graph $G = (V, E)$, a partition of V is connected (for G) if every one of its regions is connected, and a hierarchy $\mathcal{H}$ on V is connected (for G) if every one of its partitions is connected. According to (COUSTY et al., 2018), a connected hierarchy can be equivalently treated using a saliency map represented through a weighted graph (e.g., a re-weighted MST). Given a connected hierarchy $\mathcal{H}$ for (G, w), the saliency map of $\mathcal{H}$ is the map from E into $\{0, \ldots, \ell\}$, denoted by $\Phi_{\mathcal{H}}$, such that, for any edge $u = \{x, y\} \in E$, the value $\Phi_{\mathcal{H}}(u)$ is the lowest value i in $\{0, \ldots, \ell\}$ such that x and y belong to a same region of $\mathbf{P}_i$.

2.3 Video Summarization

Video summarization is a complex area of research that has attracted considerable interest within the computer vision and multimedia communities (APOSTOLIDIS et al., 2021a; TIWARI; BHATNAGAR, 2021). One of the difficulties of video summarization is to extract the most important information from a video and present it concisely (EJAZ; TARIQ; BAIK, 2012; PANDEY et al., 2017; AVILA et al., 2011; SONG et al., 2015). Summarizing videos presents a significant challenge due to the inherently subjective nature of the content, besides the user's perspective and context-specific nuances observed by him (CARDOSO et al., 2023). This complexity is compounded by the dynamic and unstructured environment often present in such videos, making it difficult to extract relevant information without losing essential contextual details. Video summarization is particularly beneficial when managing a video collection replete with repeated or redundant information dispersed across various temporal points. Video summaries may be crucial for several applications, such as video browsing and retrieval, content analysis and processing, surveillance monitoring, and medical diagnosis.

There are two types of video summaries: static video summaries composed of keyframes (also known as a storyboard) and dynamic video summaries composed of keyshots (also known as a video skim). Static video summarization has become a powerful tool for navigating vast video libraries, but some methods relying exclusively on keyshot

Figure 4 – Example of "redundancy issues" on a video in the Summe dataset.



Source: Elaborated by the author

extraction have limitations (JADON; JASIM, 2020). These summaries, presented as a series of static frames, can miss the temporal flow and neglect some crucial parts within the video, hindering comprehension of the overall narrative or event (PHAPHUANGWITTAYAKUL et al., 2021).

Video skimming strategies will incorporate elements beyond keyshots to create more informative summaries (APOSTOLIDIS et al., 2021a). Video skimming is the task of automatically generating dynamic video summaries. These summaries are represented by a small part of the original video and combine more than one scene (or shot) belonging to the video (APOSTOLIDIS et al., 2021a). Scene (or shot) clustering is essential for the quality of the dynamic summary. Some strategies use clustering according to the neighborhood and others select sparse frames. While the neighborhood-based choice allows shots to be grouped according to similarity and cuts to be made to represent scenes, sparse strategies tend to cover a larger part of the video (APOSTOLIDIS et al., 2021a; TIWARI; BHATNAGAR, 2021).

Although video skimming improves the detection of key video information, data redundancy can still hamper it, similar to static video summarization (APOSTOLIDIS et al., 2021a). Figure 4 illustrates a difficult scenario. By incorporating the temporal flow it is possible to evaluate the relationship between frames that relate to each other (PHAPHUANGWITTAYAKUL et al., 2021). Identifying keyshots can help select important sequences and focus on their representativeness (JADON; JASIM, 2020; KUMARI; DASH; SAHU, 2023). Thus, it is possible to capture the main actions/events distributed throughout the video, making it possible to reduce redundancy.

Characterized as a collection of images with subtle variations, often imperceptible to the human eye, these images, with or without the aid of other descriptors, depict a scene and convey content. Even the mere superposition of images can effectively propagate

messages. The amount of information extracted from videos is directly proportional to their length; thus, when analyzing the visual domain, it is expected that the content addressed will be highly relevant to the video. Consequently, it can be said that the volume of information extracted from a video necessitates content processing to optimize searches (LIU et al., 2010).

Video descriptors facilitate the measurement of similarity between frames extracted from each video. This process enables the assessment of cuts between frames, as a decrease in similarity above a defined threshold between frame pairs will indicate a scene change. The scene change is characterized by partial or total differences between the frames under evaluation (LIU et al., 2010).

Given the substantial amount of information in videos, it is often necessary to segment them into smaller scene shots. Scene shots are characterized by the similarity between their frames and minimal changes in visual information. However, a video is not static and contains scene modifications. Scene change, defined as a cut boundary, marks the point where one scene transitions to another. Thus, a video may contain several cuts and varied environments, or repetitions of environments separated by scenes with different settings.

Every frame is similar to another frame according to a given metric; the more different the aspects between them, the more dissimilar they will be. Figure 5 presents two scenarios: in the first, the video appears to continue with repeated similar scene shots between the first and fifth frames, while in the second, no repetition occurs. In both cuts, there are similar elements between the frames, but the change in the video's focus indicates a possible cut. Therefore, changes in camera position, recording angle, and recording focus characterize a shift in perspective, making cuts perceptible.

The presence of cuts in a video does not preclude the evaluation of disjoint similar information. In Figure 5 (a), identical frames separated by others with low correlation are observable. This scenario appears to involve a sports news broadcast where the presenter introduces a segment and resumes at the end. The existence of two highly similar frames at different points in the video, separated by a time limit, facilitates the evaluation of disjoint video segments.

According to Belo et al. (2016), it is possible to simulate cuts in videos using the following strategy: starting with a graph to which the Minimum Spanning Tree (MST) is applied, where edges compute the similarity between each vertex, the removal of edges will render the graph disconnected. Consequently, the connected components can represent cuts of scene shots in the video. By applying edge removal criteria based on cost, connected components will emerge, representing smaller scenes within the video. The number of components is relative to the number of edges removed. By removing the most expensive

Figure 5 – Example of a static summary. Each summary presents a fixed number of frames used to represent the other one in the video. Thus, five frames are selected per video. In which, (a) represents a summary with similar information distributed along the video. In (b) is shown an example of the results for an approach for summarization without the time analysis, returning a set of frames without repetition, but neglecting the importance of disjointing similar scenes.

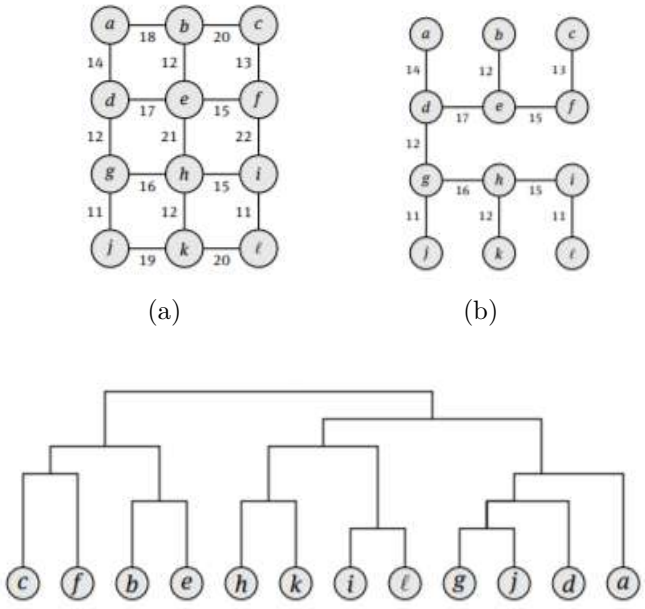
Source: Belo et al. (2016)

edge, two connected components will appear, and the similarity between the frames within each component will be lower than that of the removed edge. Thus, considering e as the number of edges to remove and $\mathcal{C}$ as the total number of components, the number of components will be expressed as $\mathcal{C} = e + 1$.

The application of hierarchical partitioning to videos enables uniform and consistent scene cuts, as it adheres to causality and locality rules. To apply the hierarchical method, it is necessary to represent the information in graph format. From this graph, the Minimum Spanning Tree (MST) is extracted, and the desired hierarchy is applied to it (GUIMARÃES et al., 2017).

Figure 6 illustrates the application of hierarchy in a graph with vertices and edges in a didactic manner. This representation contains all the necessary information for its definition. Considering that the vertices represent frames and each edge denotes the similarity between a pair of vertices, graph (a) would show the existing connections among all frames. However, for certain approaches, it is not necessary to include all the information from the initial graph. The MST representation maintains data consistency, ensuring that the data remains unmodified while still allowing for its processing. For uniform cuts to occur, it is necessary to apply techniques that level the data and organize it according to its importance. Thus, the data is presented in a nested manner, with data sequences characterized by levels within the hierarchy. According to the example, vertex g is nested with f at the same level, and as the levels increase, more data is nested within the sequence. If the vertices represent frames and two summary frames are to be extracted, cutting the highest level of the dendrogram will separate it into two sets. The first set will contain c, f, b, e, and the second set will contain h, k, i, l, g, j, d, a. Analogously, as

Figure 6 – Example of Aplication of hierarchical Graph-Based (hGB). In (a), the initial graph is presented, in (b), the MST is extracted from (a), and in (c), one presents the Quasi-Flat Zones (QFZ) dendrogram that represents the graph.



(a) (b)

(c)

Source: Guimarães et al. (2017)

the dendrogram is cut, connected components are generated, representing the extracted scenes.

To better explain the watershed hierarchy used in this work, consider a weighted graph (G, w) with $G = (V, E)$, in which V denotes the set of vertices, E the set of edges, and $w : E \to \mathbb{R}$ a weight function on the edges. According to Maia et al. (2020), the watershed hierarchy is a nested sequence of partitions of the graph that can be constructed using minimum spanning forests (MSF)s. Let $\mathcal{G} = \{G_0, G_1, \ldots, G_{n-1}\}$ represent a sequence of subgraphs of G, and $S = (M_1, M_2, \ldots, M_n)$ denote the sequence of minima of w. Thus, the hierarchical watershed of G based on S can be defined as the sequence of subgraphs, in which $\mathcal{G}$ is defined such that each subgraph G_i is an MSF of G rooted in the union of the minima of w that occur after index i, as represented by

$$G_i = MSF(G, \bigcup_{j=i+1}^{n} M_j), \quad \forall i \in \{0, 1, \ldots, n-1\} \tag{2.1}$$

and these subgraphs are represented by

$$G_{i-1} \subseteq G_i, \quad \forall i \in \{1, 2, \ldots, n-1\} \tag{2.2}$$

for each subgraph G_i contains the previous subgraph G_{i-1}. Then a sequence of partitions $\mathcal{H} = \{CC(G_0), CC(G_1), \ldots, CC(G_{n-1})\}$, in which $CC(G_i)$ denotes the set of connected

components of G_i, represents a hierarchical watershed of the weighted graph (G, w) for the minima sequence of S (MAIA et al., 2020). Thus, given a hierarchy $\mathcal{H}$, $\mathcal{H}$ is a hierarchical watershed of (G, w) if there exists a sequence S of minima of w such that $\mathcal{H}$ is equivalent to the hierarchical watershed of (G, w) for S.

A hierarchical segmentation of G_δ into k components is analogous to partitioning a hierarchy $\mathcal{H}$ into k regions, each containing more similar elements. This segmentation can be achieved by removing $k-1$ edges with higher weights (indicative of greater dissimilarity) from the $T^*_{G_\delta}$, as it represents $\mathcal{H}$. This approach integrates a similarity measure between clusters during graph partitioning, offering a more comprehensive methodology compared to traditional methods that solely assess the similarity between isolated frames.

2.4 *Machine Learning* (ML)

An algorithm is classified as a Machine Learning (ML) algorithm if it can extract knowledge from a dataset. This process is feasible using the model described by Bengio, Goodfellow and Courville (2017), which proposes the use of a learning experience E that can be applied to a task T and allows for the measurement of its performance P. ML problems require a sufficient quantity of sample data for training, validation, and testing. These samples are selected, grouped, and made available in the literature as datasets. The use of a dataset facilitates the creation of various methods with different approaches for the same problem, enabling the comparison of obtained results.

ML algorithms are classified into two primary groups: Supervised Learning (SL) and UnSupervised Learning (USL). In USL, it is possible to learn the distribution of unlabeled data and classify them according to the data distribution within the dataset. Conversely, Supervised Learning (SL)requires labeled information to ensure correct data classification. This process involves analyzing the dataset samples and their labels to classify test samples with minimal error (BENGIO; GOODFELLOW; COURVILLE, 2017).

Analogously, USL aims to recognize patterns in data with unknown or random distribution. Given a data point x with its representative vector, USL algorithms explicitly or implicitly learn the ideal distribution $p(x)$ for the data. In contrast, SLnecessitates two pieces of information to predict the probabilistic distribution of the dataset. This involves observing a vector x of samples and another vector y, which, when associated with x, provides meaningful information. Thus, SLaims to estimate the best distribution $p(x|y)$, where the existence of an auxiliary vector characterizes SLmethods. In contrast, USL methods aim to determine the best distribution without any auxiliary input (BENGIO; GOODFELLOW; COURVILLE, 2017).

Furthermore, according to Bengio, Goodfellow and Courville (2017), a third di-

vision of ML methods involves Reinforcement Learning (RL). In RL, the evaluation of data distribution is not solely based on data observation but also on interaction with the environment and feedback between learning and experiences. This feedback loop aids in problem-solving by enhancing evaluation.

According to Sutton and Barto (2018), RL is inspired by human interaction and knowledge acquisition processes. Observing the cognitive development of a child reveals that learning occurs through action, observation, and adaptation. Actions have consequences, which can be measured to determine the necessity of repeating them. Observation involves assimilating the results of each action, and adaptation is the process of adjusting behavior based on the outcomes. Thus, a child learns the cost, whether negative or positive, of performing each action, allowing them to evaluate the likelihood of successful execution. Consequently, RL algorithms focus on learning what to do, how to do it, and the associated cost. These problems typically involve closed-loop systems that consider all information to map the least costly solution. In this context, the cumulative influence of information on future results ensures an exhaustive evaluation to understand the problem and its variations.

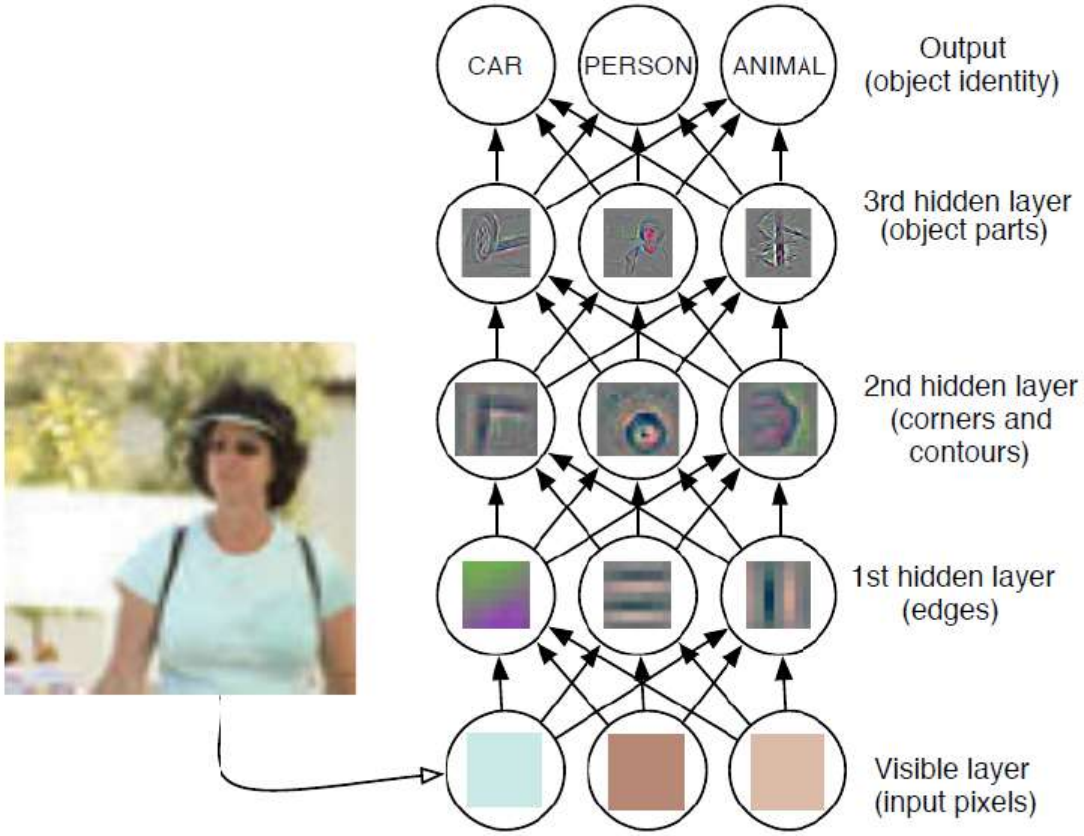
To support the proposition by Bengio, Goodfellow and Courville (2017), which presents RL as another ML paradigm, Sutton and Barto (2018) illustrates the differences between SL, USL, and RL. The primary characteristic of RL is its exploration of data to gain knowledge without an auxiliary vector, distinguishing it from SL. However, this does not categorize RL as USL, since USL only aims to evaluate a suitable data distribution. While this assists RL, RL prioritizes maximizing rewards, thus subordinating the hidden distribution relationship of USL to the pursuit of higher rewards.

2.5 Deep Learning (DL)

A DL method is characterized by its ability to transform input data into an abstract and composite representation of what the network learns from the input. According to Bengio, Goodfellow and Courville (2017), DL is described as follows:

> The idea of learning the right representation for the data provides one perspective on deep learning. Another perspective on deep learning is that it allows the computer to learn a multi-step computer program. Each layer of the representation can be thought of as the state of the computer's memory after executing another set of instructions in parallel. Networks with greater depth can execute more instructions in sequence. Being able to execute instructions sequentially offers great power because later instructions can refer back to the results of earlier instructions. According to this view of deep learning, not all of the information in a layer's representation of the input necessarily encodes factors of variation that explain the input. The representation is also

Figure 7 – Model illustration of DL aplication. This presents a network with three hidden layers after the first visible layer. Thus, the first layer is used to bound detect, the second for the contours, and the last hidden layer is used for the detection of the internal part of the object. With this information, the model's last part is responsible for classifying the input data based on the processed features.



Source: Bengio, Goodfellow and Courville (2017)

used to store state information that helps to execute a program that can make sense of the input. This state information could be analogous to a counter or pointer in a traditional computer program. It has nothing to do with the content of the input specifically, but it helps the model to organize its processing.

Figure 7 illustrates a model utilizing DL methods to classify an input image. This model comprises a visible layer, three hidden layers, and an output layer. The objective is to evaluate the best probabilistic distribution for classifying the input. Therefore, information about edges, contours, and the interior of the object is extracted to enable the model to determine what is likely present in the input frame (BENGIO; GOODFELLOW; COURVILLE, 2017).

According to Bengio, Goodfellow and Courville (2017), models employing DL techniques facilitate the evaluation of small data masses, enabling information generalization. However, there is a potential increase in generalization error, as the model strives to

best approximate the trained characteristics to the test samples. Conversely, as datasets grow, the amount of data available for analysis increases, and DL methods, particularly Deep Neural Networks (DNNs), can handle large quantities of data. The aim is to reduce generalization errors by utilizing more data for analysis.

Multiple layers between the input and output layers distinguish a DNN from an Artificial Neural Network (ANN). Each layer traversed in a DNN is responsible for evaluating the probabilistic relationship for each output. Therefore, the hidden layers that evaluate the best distribution to compute the output characterize the application of DL methods (BENGIO; GOODFELLOW; COURVILLE, 2017).

Liu et al. (2017) emphasize that DNNs enable the representation of functions with greater complexity by increasing the number of layers and units. Thus, it becomes feasible to establish mapping functions for non-trivial problems from well-defined models. The use of DNNs ensures the evaluation of information without concern for the order of information, as the correlation between the input data does not influence the network.

2.6 Convolutional Neural Network (CNN)

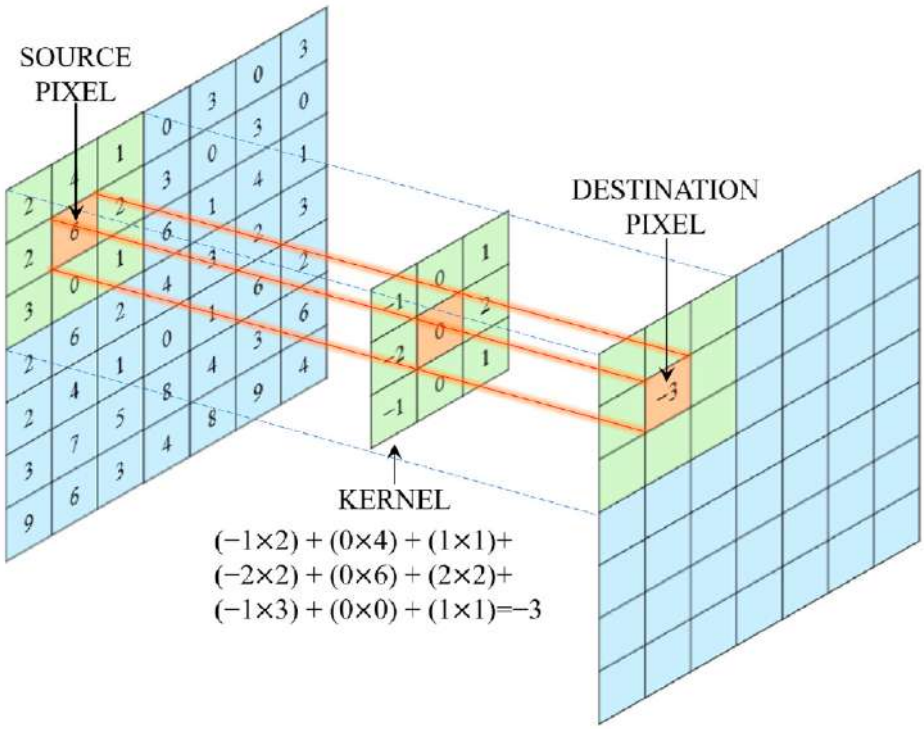
The use of Convolutional Neural Networks (CNNs) is widely prevalent for processing data that can be represented in grid-like topologies. This topology is exemplified by the distribution of an image that can be approximated in a 2D grid format of pixels (BENGIO; GOODFELLOW; COURVILLE, 2017).

According to (BENGIO; GOODFELLOW; COURVILLE, 2017), CNNs employ convolution operations to replace the previously used multiplication operations, which significantly increase computational cost. Convolution is a linear mathematical operation occurring in at least one of the network layers. This also leads to a reduction in network complexity due to the decrease in the number of weights to be managed. This reduction is further reinforced by the decrease in the number of parameters through the evaluation of spatial relationships, thus enhancing performance through the use of backpropagation techniques.

Figure 8 presents an example of applying a 3×3 filter on a 5×5 input frame, resulting in a new 3×3 frame. This reduction is due to the narrowing imposed by the operation. To prevent this reduction, it is possible to resize the frame and add lines and columns with zeros to the new pixels to prevent this reduction. However, a slight distortion at the corners may occur due to this operation. This newly generated frame characterizes the activation map of the input.

The attempt to reduce the flattening of the frame is known as padding. Its occurrence aims to avoid excessive reduction, which could lead to problems during the model's

Figure 8 – An example of the application of filter 3×3 on an image with 7×7 dimensions. This process is used to produce a new image based on the filter. In this case, the new image has a dimension of 5×5.



Source: Selvaraj et al. (2025)

learning process. Thus, by applying filters, it is possible to prevent the input from being drastically reduced, impacting the model's results and smoothing the edges. Considering that the filter is applied at a fixed size, the application of padding tends to reduce its effect as the filter moves towards the center of the frame (BENGIO; GOODFELLOW; COURVILLE, 2017).

Furthermore, according to Bengio, Goodfellow and Courville (2017), CNNs incorporate a pooling stage, which simplifies information in a region of the frame. This process can be divided into three stages: the first involves repeated convolution operations to produce an activation map with more information about the input data; the second stage applies an activation function to each activation map to introduce non-linearity into the system; and in the third stage, the frame is further modified by the pooling function, simplifying the information obtained from the second stage. Commonly, the activation function used is ReLU (Rectified Linear Unit), and max pooling is used for pooling. The ReLU function zeros all negative values, acting similarly to a ramp function, while max pooling selects the highest value in a region as the definitive value to simplify the region. For instance, considering max pooling for a 2×2 region, this process would return only the region with the highest value among the four possible values. The purpose of pooling is to reduce the input size, allowing for the evaluation of less information without quality loss, resulting in lower memory consumption and potentially improving probabilities. The

final step is the use of a fully connected layer with the probabilistic distribution for each class in the dataset. This layer is connected to all possible network outputs to enable classification.

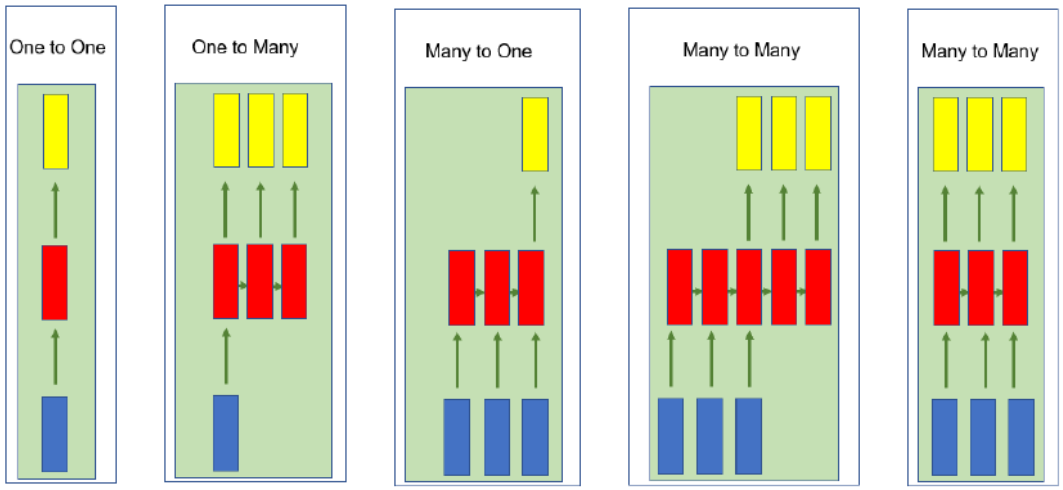
2.7 Recurrent Neural Network (RNN)

According to Bengio, Goodfellow and Courville (2017), the most widely adopted technique for working with sequential data is Recurrent Neural Networks (RNNs), as they can handle data of variable lengths. This is particularly evident in the translation of words between different languages, where a word in one language may correspond to several words in another. Such cases involve information from the same domain but of different lengths, requiring context for accurate translation, whereas individual word analysis may not yield the expected result. Consequently, during sentence translation, generating a new word directly depends on the network's modeling, often necessitating reading the entire input sentence to produce a consistent translation. This process is facilitated by evaluating hidden states that feed subsequent stages to generate the output. Thus, an RNN is characterized by feeding the next step with data from the previous step, making the output of one step the input for the next, ensuring that each step considers the most recent information produced.

Figure 9 illustrates five scenarios where RNNs effectively analyze sequential information. Each situation can be associated with a computational problem and metaphorically adapted to the described scenario. For instance, (a) represents a classification problem where an image requires a single label; (b) involves the textual description of images, expecting a sentence with several words from a single image; (c) addresses the binary classification of sentiments in a sentence as positive or negative, where multiple words lead to a single word response; (d) entails translating sentences from one language to another, such as Mandarin to Portuguese, where textual structure and grammatical rules differ; and (e) involves frame-by-frame video labeling, where each frame requires a single label. These examples academically illustrate the potential applications of RNNs, which are not limited to the discussed cases.

According to Bengio, Goodfellow and Courville (2017), a significant issue with traditional RNNs is generating very long sentences, as the relevance of earlier tokens diminishes with an increasing number of tokens, eventually rendering the initial terms nearly insignificant.

Figure 9 – Example of an application of a Recurrent Neural Network (RNN), where rectangles represent vectors and arrows denote the operations performed. The blue rectangles signify input vectors, the red vectors indicate the application of the RNN, and the yellow vectors represent the outputs. In this context, panel (a) shows an operation where a single input vector produces only one output; panel (b) illustrates a scenario where one input vector can generate multiple RNN states and multiple outputs; panel (c) depicts multiple input vectors processed by the RNN to produce a single output vector; panel (d) demonstrates multiple input vectors being fully read by the RNN before generating several outputs from new RNN states; and panel (e) shows multiple input states producing multiple RNN states and outputs.



Source: Elaborated by the author

2.8 Long Short-Term Memory (LSTM)

Traditional Long Short-Term Memory (LSTM) networks have been extensively studied in sequence modeling, particularly for tasks requiring long-range dependency learning. (HOCHREITER; SCHMIDHUBER, 1997) introduced LSTMs as a solution to the vanishing gradient problem in recurrent neural networks (RNNs), enabling effective learning over extended time sequences. In video-related applications, LSTMs have been widely adopted for action recognition (DONAHUE et al., 2015) and captioning (VENUGOPALAN et al., 2015), demonstrating their ability to capture temporal dependencies in visual data. These foundational contributions provide the basis for advanced variations, including xLSTM, which seeks to improve memory retention and feature abstraction.

The LSTM cell consists of three main gates: the forget gate, the input gate, and the output gate. According to (BECK et al., 2024), the updated mathematical formulation

can be summarized by Equations (2.3)–(2.8):

$$C_t = f_t C_{t-1} + i_t z_t \tag{2.3}$$

$$h_t = o_t \tilde{h}, \quad \tilde{h} = \psi(C_t) \tag{2.4}$$

$$z_t = \varphi(\tilde{z_t}), \quad \tilde{z_t} = w_z^\top x_t + r_z h_{t-1} + b_z \tag{2.5}$$

$$i_t = \sigma_t(\tilde{i_t}), \quad \tilde{i_t} = w_i^\top x_t + r_i h_{t-1} + b_i \tag{2.6}$$

$$f_t = \sigma_t(\tilde{f_t}), \quad \tilde{f_t} = w_f^\top x_t + r_f h_{t-1} + b_f \tag{2.7}$$

$$o_t = \sigma_t(\tilde{o_t}), \quad \tilde{o_t} = w_o^\top x_t + r_o h_{t-1} + b_o \tag{2.8}$$

in which Equation 2.3 represents the cell states, Equation 2.4 represents the hidden states, Equation 2.5 represents the cell input, Equation 2.6 represents the input gate, Equation 2.7 represents the forget gate and Equation 2.8 represents the output gate. The weight vectors ω_z, ω_i, ω_f, and ω_o represent the input weight matrices associated with the cell input, input gate, forget gate, and output gate, respectively, establishing the connections between the input vector x_t and the corresponding gating mechanisms. Similarly, the recurrent weight matrices r_z, r_i, r_f, and r_o define the connections between the hidden state h_{t-1} and the respective gating units, facilitating the propagation of temporal dependencies within the network. The bias terms b_z, b_i, b_f, and b_o provide additional flexibility in the gating computations. The activation functions φ and ψ, typically chosen as the hyperbolic tangent function (tanh), govern the transformations applied to the cell input and hidden state. The function ψ plays a crucial role in normalizing or constraining the cell state, preventing it from growing unbounded. All gate activations employ the sigmoid function, defined as $\sigma(x) = \frac{1}{1+e^{-x}}$, which ensures that the outputs remain within a bounded range, thus regulating the flow of information. In more advanced formulations, multiple scalar memory cells $c_t \in \mathbb{R}^d$ are combined into a vector representation, allowing for the integration of recurrent weight matrices $R \in \mathbb{R}^{d \times d}$, which facilitate the interaction and transformation of memory cell outputs within the network.

2.9 Extended Long Short-Term Memory (xLSTM)

Although traditional LSTMs have demonstrated effectiveness in modeling sequential dependencies, their ability to process long sequences is restricted by their limited memory capacity and difficulties in revisiting stored information. These challenges hinder performance in complex, temporally dependent tasks such as video summarization, where capturing fine-grained temporal structures is essential. To overcome these challenges, xLSTM introduces a novel architecture designed to enhance sequence modeling through parallelism and improved memory mechanisms. This model integrates sLSTM and mLSTM memory cells, where sLSTM employs an advanced memory mixing strategy, and mLSTM introduces a matrix-based memory state with a fully parallelizable covariance update rule.

Using these innovations, xLSTM blocks improve long-range dependency modeling, enable dynamic memory updates, and enhance computational efficiency. These advances are particularly valuable in video summarization, where efficiently capturing temporal patterns and key events is crucial for generating high-quality summaries (BECK et al., 2024).

2.9.1 sLSTM

The sLSTM is used to create a block that revises store decisions with exponential gates in combination with normalization and stabilization. Specifically, input and forget gates can have exponential activation functions. A normalizer state is proposed that sums up the product of the input gate times all future gates. Thus, Equations 2.4, 2.6, and 2.7 are replaced, respectively, by Equations 2.10, 2.11, and 2.12, and a normalizer state is proposed in Equation 2.9 as proposed by (BECK et al., 2024).

$$n_t = f_t n_{t-1} + i_t \tag{2.9}$$
$$h_t = o_t \tilde{h}, \quad \tilde{h} = C_t / n_t \tag{2.10}$$
$$i_t = exp(\tilde{i_t}), \quad \tilde{i_t} = w_i^\top x_t + r_i h_{t-1} + b_i \tag{2.11}$$
$$f_t = \sigma_t(\tilde{f_t}) \; OR \; exp(\tilde{f_t}), \tilde{f_t} = w_f^\top x_t + r_f h_{t-1} + b_f \tag{2.12}$$

In (BECK et al., 2024), the original LSTM gating mechanisms, including input and hidden-dependent gating with bias terms, are extended to the new architectures. To prevent overflow caused by exponential activation functions, gate stabilization is achieved through an additional state m_t, similar to that proposed by (MILAKOV; GIMELSHEIN, 2018).

$$m_t = max(log(f_t) + m_{t-1}, log(i_t)) \tag{2.13}$$
$$i'_t = exp(log(i_t) - m_t) = exp(\tilde{i_t} - m_t) \tag{2.14}$$
$$f'_t = exp(log(ft) + m_{t-1} - m_t) \tag{2.15}$$

in which Equations 2.13, 2.14, and 2.15, are stabilizer states, stabilizer input gate, and stabilizer forget gate.

2.9.2 mLSTM

The mLSTM memory increases the LSTM memory cell to enhance storage capacities, to do this, a new scalar $c \in \mathbb{R}$ is proposed to a matrix $\mathrm{C} \in \mathbb{R}^{\times}$, in which the retrieval

is performed using matrix multiplication. The expected result is storing a pair of vectors key and value, $k_t \in \mathbb{R}$ and $v_t \in \mathbb{R}$, respectively. Using the query vector ($q_{t+\tau} \in \mathbb{R}$) its possible to retrieve the v_t in a $t + \tau$ time.

Unlike sLSTM, the mLSTM cells abandon the memory mixing in the hidden-hidden recurrent connections to enhance the speedup of the model, allowing parallelism. However, similarly to the attention mechanism, the mLSTM has quadratic complexity as a result of a higher separability when limiting retrieval pairwise interactions. Thus, the mLSTM forward pass proposed to (BECK et al., 2024) is:

$$
\begin{aligned}
C_t &= f_t C_{t-1} + i_t v_t K_t^\top && (2.16)\\
n_t &= f_t n_{t-1} + i_t k_t && (2.17)\\
h_t &= o_t \odot \tilde{h}, \quad \tilde{h} = C_t q_t / max\{\left| n_t^\top q_t \right|, 1\} && (2.18)\\
q_t &= W_q x_t + b_q && (2.19)\\
k_t &= \frac{1}{(\sqrt{d})} W_k x_t + b_k && (2.20)\\
v_t &= W_v x_t + b_v && (2.21)\\
i_t &= exp(\tilde{i_t}), \quad \tilde{i_t} = w_i^\top x_t + b_i && (2.22)\\
f_t &= \sigma_t(\tilde{f_t})\ OR\ exp(\tilde{f_t}), \quad \tilde{f_t} = w_f^\top x_t + b_f && (2.23)\\
o_t &= \sigma_t(\tilde{o_t}), \quad \tilde{o_t} = w_o^\top x_t + b_o && (2.24)
\end{aligned}
$$

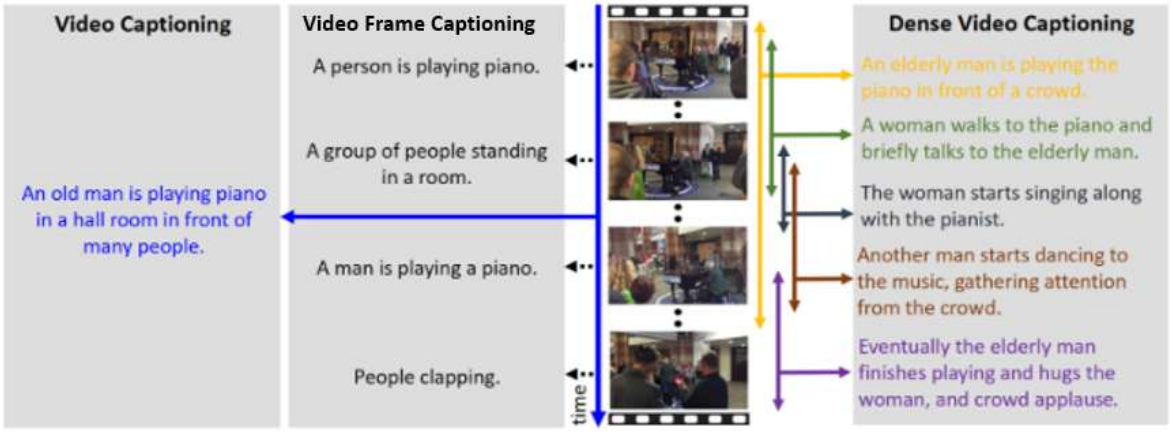
in which Equations 2.16 to 2.24 refer to cell state, normalizer state, hidden state, query input, key input, value input, input gate, forget gate, and output gate, respectively.

2.10 Video Captioning

The video captioning task consists of producing diverse sentences capable of describing numerous events in a video in a dense paragraph of description that lists all of them (LEI et al., 2020; CARDOSO; GUIMARÃES; PATROCÍNIO JR, 2021). One of the biggest challenges in video captioning is generating a coherent descriptive sentence for a database of videos (segmented or unsegmented). Content segmentation can lead to redundancy in the generated descriptions. On the other hand, processing long videos without segmenting becomes a complicated task due to the high amount of visual content to be processed (KRISHNA et al., 2017; PARK et al., 2019).

The description of events can happen in two ways, one being the separate description of every event, and, in this case, they may (or may not) be related; the other way is to group the related events and produce at least one coherent sentence about them in

Figure 10 – Aplication examples of Video Captioning (VC), Video Frame Captioning (VFC) e Dense Video Captioning (DVC)



Source: Adapted from Aafaq et al. (2019)

the paragraph. The problem in grouping events into paragraphs is linked to coherence because the paragraph must present its elements, avoiding repetition. A dense description can narrate separate events (with or without relation), but in this case, since they are separate descriptions, redundancy is usually not an issue. However, in the generation of a paragraph for a video, each sentence is possibly related to the others (AAFAQ et al., 2019; PARK et al., 2019). Therefore, if some events present similarities, the generated sentences can be repeated. Thus, the method for video captioning should prevent the final description from having the same (or almost the same) sentence many times. The real intention behind this process is to capture other aspects that, by chance, have not been described yet. Thus, describing a video through a paragraph consists of ensuring cohesion while maintaining similarity to the expected GT results (AAFAQ et al., 2019; LEI et al., 2020).

Considering Figure 10 and the definitions proposed by Aafaq et al. (2019), Dense Video Captioning (DVC) generates descriptions based on overlapping shot frames (SF), where the same frame contributes to multiple described sentences. By capturing aspects related to the movement of objects through the analysis of multiple frames, it is possible to produce sentences that do not rely on frame intersections to generate descriptions. This process is more aligned with DVC and deviates from Video Frame Captioning (VFC), which captures static information within each frame. This work aims to operate between VFC and DVC, generating Dense Video Captioning with Disjoint Sentences as coherent paragraphs.

Based on the information available in the literature and motivated by the challenges and potential causes discussed in this chapter, this work aims to explore techniques for video-to-text description using deep learning (DL) methods to enhance the quality of sentences generated for each segment. Therefore, the research seeks to apply a model

based on attention mechanisms (AM) that minimizes method errors.

Some feature extraction methods use one strategy based on sequential data sampling. In video captioning, this process consists of choosing frames at regular intervals within the video. Thus, some relevant frames for the video content (and also for its description) might not be considered. Generally, a sequential selection policy extracts a limited number of frames (for instance, 100 frames) just from the beginning of the video. Therefore, the remaining video frames (along with all the events they represent) are completely ignored (KRISHNA et al., 2017; ZHOU et al., 2019; PARK et al., 2019).

2.11 Attention Mechanisms

The attention mechanisms were first applied to the machine translation task (BAHDANAU; CHO; BENGIO, 2015) and, after that, they were applied to other tasks such as object detection, image classification, and image content description (ZHANG et al., 2021). The use of attention mechanisms seeks to highlight the importance of more prominent content, which tends to be neglected in conventional methods (VASWANI et al., 2017).

The impact on the application of attention mechanisms to description tasks was first explored by (XU et al., 2015), in which the authors evaluated the impact of attention in image captioning. They explored *soft* and *hard* attention, and the results achieved showed improvement for both, but with high training costs for *hard* attention. Thus, other variants of attention were proposed and studied in the literature. But in (VASWANI et al., 2017), the use of attention reached another level, and it began to be considered a method (called transformer) capable of producing results by itself without the need for other techniques.

According to Bengio, Goodfellow and Courville (2017), the AM arose for Machine Translation (MT) problems, in which a sentence was translated from one language to another. Thus, AM is applied to prevent the evaluation of large amounts of data by modifying the data of an array with all data to a sequence of variable lengths. In this way, AM can walk through the entire sequence and assist in the correct translation.

Analogously to the one applied in MT, Ba, Mnih and Kavukcuoglu (2014) proposes the adaptation of AM for object detection, enabling its use in various problems of visual information processing, such as the one proposed by this work.

The division of AM into hard-attention and soft-attention, as presented by Xu et al. (2015), demonstrates the potential to increase the level of perception of models, since when applying hard attention, a tightening of the rules occurs. This is due to the proposed region selection formulation, in which the location variable s_t represents the

Figure 11 – Illustration of Attention Mechanism (AM) applied to soft-attention and hard-attention techniques. The upper image presents the probability distribution for soft and the lower one for hard. Each image represents the region that will be used to generate a word to describe the image.

Source: Xu et al. (2015)

place where the model will focus when producing the t-th word. Thus, the value of 1 is assigned to the indicated $s_{t,i}$ if the i-th is used for extracting visual resources, and for the other cases, it will receive the value 0. On the other hand, soft attention is represented by a probability distribution, in which the value will be set according to the composition of an average sampling vector for each region. The composition of the vector will be the region resulting from soft attention.

Figure 11 shows the use of an image that was processed by soft attention and hard attention. According to what was proposed by Xu et al. (2015), it is possible to see the stiffening in hard attention by selecting only one point in the image with a region for recognition of visual resources; on the other hand, in soft attention, it was possible to perceive a region greater than is represented by a vector of probabilities. This vector represents how much each point will contribute to the extraction of visuals.

Thus, according to Ba, Mnih and Kavukcuoglu (2014), it is possible to use AM for various problems and add information with a certain level of focus. This process makes it possible to use it for evaluating frames, increasing the quality of information used to define video text description models.

A challenge for attention mechanisms lies in the long-range dependencies. This problem is related to the network's capacity to learn from all the previous states. However, the adoption of a self-attention mechanism creates the possibility to circumvent those difficulties and, at the same time, allows more efficient use of available resources through extensive use of parallelism (VASWANI et al., 2017).

To re-weight the importance of certain data, attention mechanisms can be applied to the network backbone to increase the importance of specific information about others. Techniques, as presented by (HU; SHEN; SUN, 2018), tend to increase the importance of some features, compared to others, and could be explored to increase the relationship between data that previously could not be easily related (HU; SHEN; SUN, 2018; CARDOSO; GUIMARÃES; PATROCÍNIO JR, 2021).

Experiments on machine translation tasks showed transformers as superior models in quality while being more parallelizable and requiring significantly less time to train than others. A transformer achieves good results because it can capture the relationship between tokens and the generated vocabulary. After that, transformer models have been successfully applied to several distinct tasks, such as machine translation, information retrieval, text classification, document summarization, image classification, image captioning, and video captioning (CARDOSO; GUIMARÃES; PATROCÍNIO JR, 2021).

2.12 Transformers

The proposal of using attention mechanisms to compose a new model called transformer first appeared in (VASWANI et al., 2017) to reduce the computational cost without quality loss. Following Vaswani et al. (2017), the main component of a transformer is the *scaled dot-product attention*. Given query matrix Q, key matrix K, and value matrix $\mathcal{V}$. The attention output is given by Equation 2.25:

$$Attention(Q, K, \mathcal{V}) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)\mathcal{V}. \tag{2.25}$$

Combining h parallel instances of scaled dot-product attention, obtaining the multi-head attention (VASWANI et al., 2017) represented by Equation 2.26:

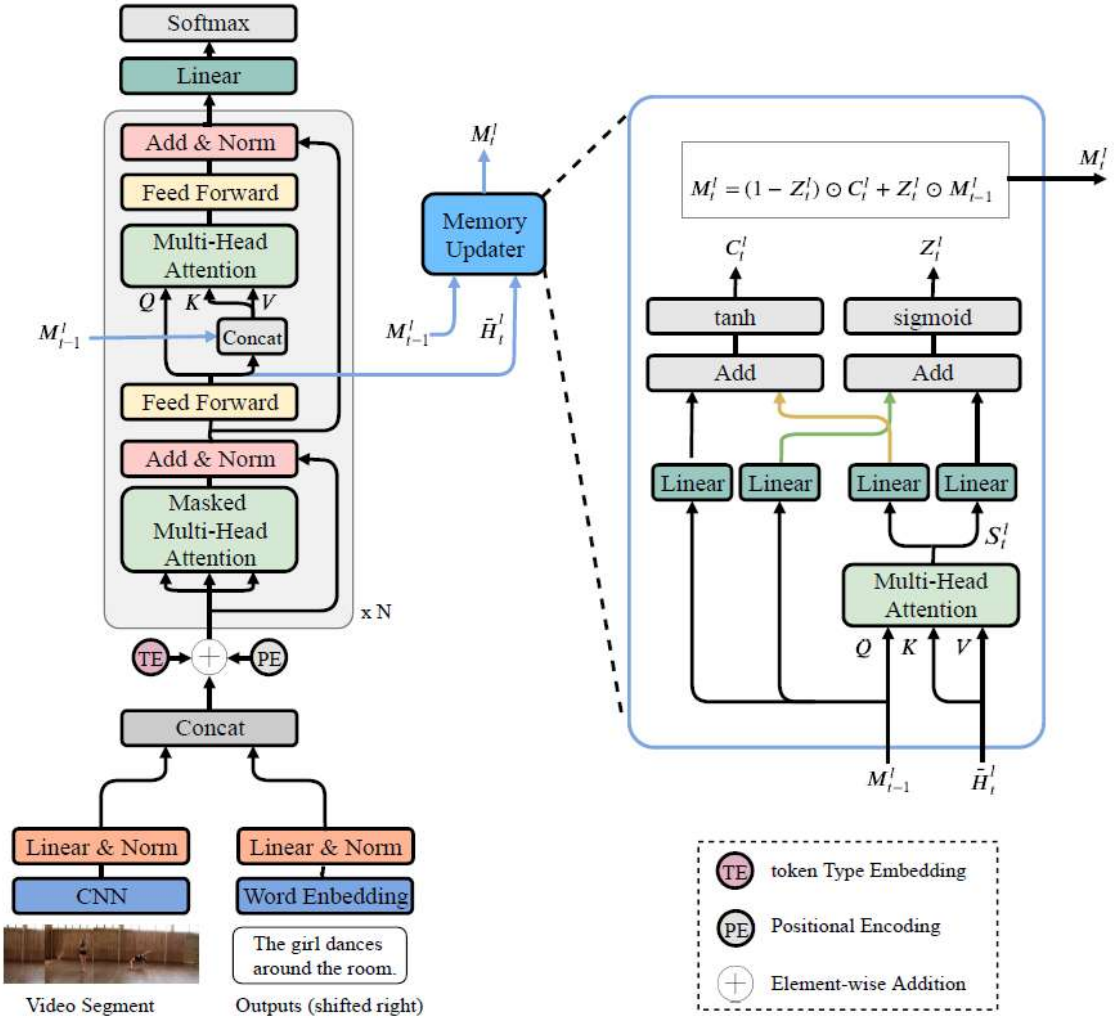
$$MultiHeadAtt(Q, K, \mathcal{V}) = Concat(head_1; \ldots; head_h)W^O, \tag{2.26}$$

in which $head_i = Attention(QW_i^Q, KW_i^K, \mathcal{V}W_i^{\mathcal{V}})$, and linear projections W_i^Q, W_i^K, $W_i^{\mathcal{V}}$, and W^O are learned during training.

Currently, some methods exploit the traditional encoder-decoder model, while others use a shared model. In a traditional encoder-decoder model, the encoder is separate from the decoder, and the video-encoded information must be passed and used in some of the decoder steps. So, the decoder uses the data produced by the encoding network to generate each word/sentence (ZHOU et al., 2018; PAN et al., 2020; HUANG et al., 2019). For shared models, as the one shown in Figure 12, video data and generated text tokens are passed as input to the same module, which (after operating on them) is responsible for both encoding and decoding. That model is generally used due to the reduction of operations, resulting in time and processing savings.

A memory-augmented recurrent transformer (MART) was proposed in (LEI et al., 2020) using a shared architecture, see Figure 12. To use this model, initially, video and text are encoded and normalized separately. Encoded video and text embeddings are denoted as $H^0_{video} \in \mathbb{R}^{T_{video} \times d}$ and $H^0_{text} \in \mathbb{R}^{T_{text} \times d}$, respectively, in which T_{video} and T_{text} represent video and text lengths, while d is the embedding size. They are used after a concatenation

Figure 12 – Memory-augmented recurrent transformer (MART) architecture with shared encoder and decoder applied in video paragraph captioning.



Source: Lei et al. (2020)

and passed to the transformer as input $H^0 \in \mathbb{R}^{T_c \times d}$, i.e., $H^0 = Concat(H^0_{video}; H^0_{text})$, in which $T_c = T_{video} + T_{text}$, following the proposal of (SUN et al., 2019; CHEN et al., 2019).

The use of a memory module has been gaining prominence in the literature. Unlike conventional methods, the use of memory mechanisms helps to reduce the redundancy by recurrently evaluating information from previous states (LEI et al., 2020; DAI et al., 2019) . In this way, a memory module becomes a mechanism for assessing the importance of sentences (and video events). So, the information that is passed between states is used to assess their relevance degree. Memory data is encoded during video processing, and it should be updated to keep track of relevant information. Considering that $M_t^l \in \mathbb{R}^{T_m \times d}$ represents memory state at layer l in step t in which T_m denotes memory length, and $\tilde{H}_t^l \in \mathbb{R}^{T_c \times d}$ is the intermediate hidden state vector, the memory update

process (proposed in (LEI et al., 2020)) can be summarized by Equations (2.27)–(2.30):

$$S_t^l = MultiHeadAtt(M_{t-1}^l, \tilde{H}_t^l, \tilde{H}_t^l), \quad (2.27)$$

$$C_t^l = tanh(W_{mc}^l M_{t-1}^l + W_{sc}^l S_t^l + b_c^l), \quad (2.28)$$

$$Z_t^l = sigmoid(W_{mz}^l M_{t-1}^l + W_{sz}^l S_t^l + b_z^l), \quad (2.29)$$

$$M_t^l = (1 - Z_t^l) \odot C_t^l + Z_t^l \odot M_{t-1}^l, \quad (2.30)$$

in which $\odot$ denotes Hadamard product, W_{mc}^l, W_{sc}^l, W_{mz}^l, and W_{sz}^l are trainable weights, b_c^l and b_z^l are trainable bias. $C_t^l \in \mathbb{R}^{T_m \times d}$ is the internal cell state, while $Z_t^l \in \mathbb{R}^{T_m \times d}$ is the update gate that controls which information to retain from the previous memory state. Equation 2.30 presents Z_t^l as the information regulator and C_t^l as the new information sample. In this way, the update of M_t^l will occur through a linear combination of new information represented by C_t^l and the information already present in the memory, i.e., M_{t-1}^l.

The memory updater module seeks to assist in assessing video segments' importance in generating new sentences. Its adoption allows the transformer to recurrently evaluate longer sentences. Conceptually, that proposal uses similar strategies to those defined by LSTM (HOCHREITER; SCHMIDHUBER, 1997) and GRU (CHO et al., 2014) modules. The difference between them and the memory updater module is the high capacity of the latter to model complex data provided by the transformer employing a multi-head attention stage. Consequently, it enables the memory to capture/model different concepts and, therefore, to better understand and deal with similarities among video events.

2.13 Evaluation Metrics

The evaluation of sentences is a separate challenge, as there are several ways to write sentences, but with the same meaning, whether using synonyms or emphasizing information. This process is intuitive for humans. However, there is no specific approach for evaluating the video captioning task. So, what is usually done is the adaptation of machine translation metrics that are extended for this task (AAFAQ et al., 2019; VEDANTAM; ZITNICK; PARIKH, 2015; PAPINENI et al., 2002).

In training, textual descriptions are generated for each video, but as the model is learning the features, some descriptions returned may not match what was expected. During training, an improvement in the quality of descriptions is expected and after the last training season is completed, more accurate descriptions are expected. This process makes it possible to check, at each time, whether the model tends to improve.

To assess the improvement of the descriptions, in each new model generated, the obtained score will be computed. With the application of Bilingual Evaluation Understudy (BLEU), Consensus-based Image Description Evaluation (CIDEr), and Reduction, seek to calculate the proximity between the descriptions generated by the models and those belonging to the GT. Thus, similar to the works of Xu et al. (2015), Wang et al. (2018), Li and Gong (2019), Lei et al. (2020), it will be possible to measure the quality of all text sentences generated at the end of the description process.

2.13.1 Bilingual Evaluation Understudy (BLEU)

This work will use the variation of BLEU proposed by Papineni et al. (2002) which proposes the use of n-gram to improve the evaluation made in function only of the precision (unigram). Using $n = 4$, the size of the rating window corresponds to a grouping of 4 words. In addition, the selection for $n = 4$ will allow the comparison with other works that use MT to evaluate VC.

According to Papineni et al. (2002), the use of BLEU unigram compares the evaluation of the simple precision of the method, characterized by the simple count of correct words divided by the total number of words in the sentence.

Table 1 presents two sentences translated from Chinese to English for evaluation with three considered GT. They are talking about the same subject, but the quality of them is not the same. Comparing the sentences with the GT, it is noticeable that the number of words that are in *Candidate* 1 but not in 2. The analysis n-gram allows the counting of the adequate score for both candidates and helps in the choice of the best sentence.

Table 1 – Chinese to English Language Sentence Translation Example.

Legend	Sentence
Candidate 1	It is an action guide which ensures that the military always obeys the commands of the party.
Candidate 2	It is to ensure the troops forever hear the activity guidebook that party direct.
Reference 1	It is a guide to action that ensures that the military will forever heed Party commands.
Reference 2	It is the guiding principle which guarantees the military forces always being under the command of the Party.
Reference 3	It is the practical guide for the army always to heed the directions of the party.

Source: Adapted from Papineni et al. (2002)

The calculation of BLEU-4 measures the proximity of the automatically generated description and the human reference as GT. For this, it will consider its length, each

word chosen, and the order in which they were generated (PAPINENI et al., 2002).

2.13.2 Consensus-based Image Description Evaluation (CIDEr)

The CIDEr metric proposes to measure the best textual sentence among the candidates by the majority of simple votes. This process was considered by Vedantam, Zitnick and Parikh (2015) as an evaluation of the translation consensus, since if the sentence is very close to most GT sentences used as a reference, the probability is greater that the sentence is correct. This technique seeks to bring the human description closer to that described by MT, as human evaluation is inherent to the perception of the person describing the scene in focus.

According to Vedantam, Zitnick and Parikh (2015) the main objetiveo of the metric CIDEr is:

> Given an image and a collection of human-generated reference sentences describing it, the goal of our consensus-based protocol is to measure the similarity of a candidate sentence to a majority of how most people describe the image (i.e. the reference sentences).

Vedantam, Zitnick and Parikh (2015) propose the use of CIDEr to assess the quality of image descriptions by evaluating the consensus between various descriptions of the GT and the textual description of the method. Thus, using CIDEr to evaluate the descriptions of videos in the proposed models.

For improving CIDEr, CIDEr-D is created for image captioning, more specifically for the MS-COCO dataset, and used for tasks such as video captioning. The CIDEr-D first executes the process of stemming for mapping the word of the same radical into one single token. The process of stemming is used to ensure that the correct form of the word is used for calculating the CIDEr-D score with more precision. The second action is to penalize the words with high confidence repeated in long sentences. For these cases, a Gaussian penalty based on the difference between candidate and reference sentence lengths is applied to this sentence. Finally, the CIDEr-D can penalize the sequences with a lot of repetition of sentences.

2.13.3 Repetition R@4

Those previous metrics cannot penalize the repetition that may happen. So, it is necessary to adopt another metric for evaluating how diverse the description is. Thus, the Repetition-4 score (R@4) (LEI et al., 2020; XIONG; DAI; LIN, 2018; PARK et al., 2019) was applied, and its objective is to emphasize the reduction of repetition of words in the description. Both R@4 and B@4 scores use 4-grams to increase word grouping. Unlike

other metrics, the purpose of this one is to find values closer to zero, so it is a metric where your score decreases while the others increase.

The evaluation of R@4 in isolation does not reflect improvement, because when considering a sentence without any repeated words, created randomly, the value of this metric tends to zero, but the result may not be as expected. Thus, the use of this metric is included with other metrics that assess the similarity of sentences produced with GT, as the focus is on the representation of sentences related to GT, but with a focus on diversity (LEI et al., 2020; XIONG; DAI; LIN, 2018; PARK et al., 2019).

2.13.4 Metrics for Video Summarization

Assessing frame quality in the context of video summarization poses a distinct challenge because of the many ways in which frames can be constructed while conveying similar meanings. These variations can arise from using different analyses of resources from different informational aspects. Although humans have an intuitive understanding of this process, abstract evaluation remains an open question without a specific framework. As a result, the conventional practice involves adapting similar metrics that have been stretched to accommodate the specific requirements of the video summary task. By repurposing and customizing these metrics, researchers and practitioners can assess the effectiveness and fidelity of summaries generated in video summarization, despite the inherent complexities and subjectivity involved in sentence evaluation (AVILA et al., 2011; BELO et al., 2016).

To compute the improvement of the frame selection. They reported their results using metrics widely disseminated in the literature, such as CUSa, CUSe (AVILA et al., 2011; BELO et al., 2016), and COV (BELO et al., 2016), defined by the Equations 5.1–5.3, respectively, to evaluate the similarity between the frames generated by their summarization method and the GT results.

$$\text{CUSa} = \frac{m_A}{n_U} \tag{2.31}$$

$$\text{CUSe} = \frac{\overline{m}_A}{n_U} \tag{2.32}$$

in which m_A denotes the number of matching keyframes generated from the Automatic Summary (AS), $\overline{m}_A$ represents non-matching keyframes from AS, and n_U is the number of keyframes selected for the user to represent the user summary (U) to each video.

$$\text{COV} = \frac{\sum_{U \in US} |M(AS, U)|}{\sum_{U \in US} |U|} \tag{2.33}$$

in which $M(X, Y)$ and $|.|$ are the maximum matching between two sets of different ele-

ments X and Y, and the cardinality of a set, respectively.

While those first two metrics provide valuable insights, they often fail to measure the diversity displayed in user summaries as COV does. Furthermore, the calculation of averages for each user's measurements can introduce distortions and inaccuracies. Specifically, the CUSa, which is commonly employed to assess user opinions, fails to effectively capture the diversity of these opinions. To illustrate, consider two users, A and B, providing summaries for the same video. Let the summary of user A be $U_A = \{X, Y\}$ while the summary of user B is $U_B = \{M, N, O, P, Q, R, S, T, U, V\}$, in which each character denotes a single frame of video. Now suppose that three distinct methods generate summaries: $AS_1 = \{X, Y\}$, $AS_2 = \{M, N, O, P, Q, R, S, T, U, V\}$, and $AS_3 = \{X, M, N, O, P, Q\}$. Despite these summaries being completely different, they provide the same accuracy rate (i.e., CUSa = 0.5). This highlights the limitations of CUSa in accurately assessing divergence of opinion and the need for more comprehensive assessment metrics (AVILA et al., 2011; BELO et al., 2016).

Unlike CUSa, COV assesses the extent to which an automatic summary covers all user-generated summaries. This measure takes into account both the diversity of opinions expressed by users and the degree of agreement among them. Specifically, the CUSa measure calculates the average ratio between each user's summary and an automatic summary, thus capturing the level of agreement between the two. In contrast, COV assesses the proportion of an automatic summary that aligns with all user summaries, providing a measure of overall coverage. COV is used as the first metric to compute the effectiveness of the HieTaSum (see Algorithm 3 in Chapter 4) and HieTaSkim (see Algorithm 4 in Chapter 4). The reader should refer to Avila et al. (2011) and Belo et al. (2016) for more information about those metrics.

The F-score, also known as the F_1-score, is a measure used in statistical analysis to evaluate the accuracy of a binary classification system. It is particularly useful when the dataset is imbalanced, meaning that the classes are not represented equally. The F-score considers both the precision P and the recall R of the test to compute the score.

Precision is defined as the ratio of correctly predicted positive observations to the total predicted positives, as shown in Equation 2.34,

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.34}$$

in which, TP (True Positives) are the number of correctly predicted positive cases, and FP (False Positives) are the number of incorrectly predicted positive cases.

Recall, also known as Sensitivity or True Positive Rate, is the ratio of correctly predicted positive observations to all the observations in the actual class, as presented by

Equation 2.35,

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.35}$$

in which, FN (False Negatives) are the number of incorrectly predicted negative cases.

The F-score can be interpreted as a weighted average of the precision and recall, where an F-score reaches its best value at 1 and worst at 0. The general formula for the F-score is represented by the Equation 2.36:

$$F_\beta = (1 + \beta^2) \times \frac{\text{Precision} \times \text{Recall}}{(\beta^2 \times \text{Precision}) + \text{Recall}} \tag{2.36}$$

in which β is a parameter that determines the weight of recall in the combined score.

When $\beta = 1$, precision and recall are equally weighted. The F_1-score is thus given by the Equation 2.37:

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.37}$$

The F_1-score, which is the harmonic mean of precision and recall, provides a balance between precision and recall. It is particularly useful when the class distribution is imbalanced, as it provides a single metric that considers both false positives and false negatives.

In practical terms, the F_1-score is crucial for evaluating the performance of classification models, especially in scenarios where both precision and recall are important to the task at hand. For example, in information retrieval, the F_1-score can provide a balanced measure of a search algorithm's effectiveness.

Understanding the relationship between precision, recall, and the F_1-score is critical for developing robust Machine Learning (ML) models that can accurately predict outcomes in various real-world scenarios.

3 RELATED WORKS

The problem of dense video captioning is related to the ability to extract knowledge from a dataset and the way this information is used. Thus, other challenges are presented, such as the need to reduce the prediction error. Another problem with working with video is the amount of similar information laid out sequentially with little or no variation. The relationship between the distribution of frames directly impacts the sentences generated and implies an increase or not in repetition. That said, in this Chapter, the articles that form the background to maintain consistency in the description and maintain the quality of the sentences will be addressed, as well as similar tactics used by other authors who faced similar problems to this work will be presented.

3.1 Video Summarization

To better understand the video summarization process, exploring the related terms and ideas fundamental to this field is crucial. This section examines some of the concepts that are essential to understanding the video summarization task used as the first step of video captioning.

Despite the challenges associated with the subjectivity of ground truth generated by multiple users, numerous unsupervised methods have been developed over the years. In (FURINI et al., 2007), the authors presented a platform for customizing video summaries. Utilizing clustering techniques, they introduced a method, named VISTO, which analyzed low-level features to determine frame similarity. Keyframe selection was achieved by choosing the center of each cluster, followed by a post-processing step to remove potential frame redundancies. In (AVILA et al., 2011), the authors propose a clustering-based strategy for video summarization named VSUMM. Initially, a sampling process reduced the number of frames under analysis. Frames represented by color histograms were then grouped into similar sets using the k-means algorithm. VSUMM results are often grouped in temporally dispersed frames.

The work of Belo et al. (2016) presents an approach for video summarization based on graph theory. The proposed hierarchical approach comprises keyframe extraction, scene segmentation, and video summarization stages. In the keyframe extraction stage, the method selects representative frames based on image quality and diversity. In the scene segmentation stage, the video is divided into different scenes based on the visual similarity between frames. Finally, in the video summarization stage, the keyframes

and scenes are combined to generate a summary of the video. The proposed approach employs a graph-based hierarchical approach that is capable of generating effective video summaries. The approach is evaluated on various datasets, and the results demonstrate its efficacy in terms of summary quality and efficiency. The authors analyze the impact of different parameters on the performance of the proposed approach, such as the number of clusters, the threshold for scene segmentation, and the size of the summary. The proposed approach is also evaluated on different types of videos, such as sports, news, and movies, showing its adaptability to various video genres. However, one limitation of the proposed approach is that it does not consider time-aware modification, which may be a critical criterion in some applications.

In (PANDA; MITHUN; ROY-CHOWDHURY, 2017), the authors presented an unsupervised approach for summarizing a collection of videos, developing a diversity-aware optimization method for multi-video summarization by exploring the complementarity of the videos.

The video summarization field has advanced significantly in recent years, particularly with the advent of deep learning algorithms. The study in (MOLINO et al., 2017) addressed the challenges of egocentric video summarization. In (BASAVARAJAIAH; SHARMA, 2019), the authors focused on summarization methods applied directly to the compressed domain. Furthermore, Vivekraj, Debashis and Balasubramanian (2019) provided an extensive bibliography on dynamic video summarization. According to Apostolidis et al. (2021a), deep-learning-based video summarization methods represent video content using deep feature vectors extracted by pre-trained neural networks. These features are then utilized by a deep summarizer network, producing either a set of keyframes (a static summary) or a set of video fragments (a dynamic summary).

The hierarchical process of the proposed method outlined in (COUSTY et al., 2018) is a significant contribution to the field of image segmentation. The method employs a bottom-up approach, starting with the generation of initial regions and gradually merging regions to form larger ones. Quasi-flat zones are then utilized to identify regions with similar characteristics and merge them to create larger segmentations. The method then creates a minimum spanning tree to capture the most salient edges of the image, and the salience maps are generated to highlight the most significant regions. The final result is a hierarchical segmentation of the image, which captures the details at multiple scales and provides a comprehensive understanding of the image's structure. This hierarchical process gives the method a significant advantage over traditional image segmentation techniques, as it can generate highly accurate and detailed segmentations that capture the essential features of the image.

Supervised video summarization has gained significant attention in recent years,

particularly with the advancement of deep learning techniques. Among these, methods leveraging attention mechanisms, Convolutional Neural Networks (CNNs), and Recurrent Neural Networks (RNNs) have shown remarkable performance in generating concise and informative summaries of lengthy video content (APOSTOLIDIS et al., 2021a; TIWARI; BHATNAGAR, 2021). Attention mechanisms, in particular, modify the way temporal dependencies and salient features are identified and utilized in video summarization (AAFAQ et al., 2019; APOSTOLIDIS et al., 2021a). By dynamically focusing on the most relevant parts of the video frames, attention-based models can effectively capture key events and important details, ensuring that the generated summary is both comprehensive and representative of the original content (APOSTOLIDIS et al., 2021a; TIWARI; BHATNAGAR, 2021).

Another kind of supervised technique is the adoption of Convolutional Neural Network (CNN)s and Recurrent Neural Network (RNN)s, which has also been extensively employed in supervised video summarization (APOSTOLIDIS et al., 2021a). CNNs are adept at extracting spatial features from individual frames, making them crucial for identifying visual patterns and salient objects within the video. When combined with RNNs, which are proficient at modeling temporal dependencies, the resulting models can efficiently process video sequences to generate summaries that preserve the narrative structure of the original content (APOSTOLIDIS et al., 2021a). For instance, CNN-RNN architectures can analyze both the spatial features of frames and their temporal relationships, enabling the detection of key moments and transitions within the video. The use of CNNs and RNNs in the same model, along with the integration of attention mechanisms, has led to the development of sophisticated video summarization methods that are capable of producing high-quality, context-aware summaries (APOSTOLIDIS et al., 2021a; TIWARI; BHATNAGAR, 2021).

The application of supervised techniques to video summarization presents significant challenges, particularly due to the substantial amount of labeled data required to train these models effectively (APOSTOLIDIS et al., 2021a). Supervised learning algorithms rely on annotated datasets, which necessitate extensive human effort to label keyframes or important segments in the video content accurately (TIWARI; BHATNAGAR, 2021). This process is not only time-consuming but also subject to inconsistencies, as different annotators may have varying interpretations of what constitutes a salient moment in a video. Furthermore, the diverse and dynamic nature of video data, spanning multiple genres, scenes, and actions, requires large-scale and varied datasets to ensure the robustness and generalization capabilities of the model. The sheer volume of information in video data – comprising thousands of frames and complex temporal dependencies – further complicates the annotation process, making gathering enough high-quality labeled data for training a difficult task. Consequently, the requirement for extensive labeled

datasets often becomes a bottleneck, limiting the scalability and applicability of supervised video summarization techniques in practical scenarios (APOSTOLIDIS et al., 2021a; TIWARI; BHATNAGAR, 2021).

3.2 Video Skimming

Although current developments in video summarization are tightly related to the growing use of deep neural network architectures, one can still find room for non-deep unsupervised approaches, especially to multiscale graph-based methods like the one proposed here. Usually, video skimming techniques can be categorized into supervised, unsupervised, reinforcement-based, and adversarial methods (CARDOSO et al., 2025a, 2025b)

Supervised learning utilizes videos pre-labeled by humans, offering segments or frames based on their importance. This information trains the model, enabling it to predict the crucial frames highlighting the video content (APOSTOLIDIS et al., 2021a). In (PUTHIGE et al., 2023), the authors present a supervised method for dynamic video summary using different attention mechanisms to assist the selection of frames. The variant using channel attention followed by spatial attention achieved better in the SumMe dataset. However, providing reliable and large amounts of annotations could be hard for a wide range of domains.

On the other hand, unsupervised techniques select the most important frames' sequences without needing pre-labeled data. These methods often rely on visual features such as colors, movements, or sliding windows to automatically detect scenes, making them versatile and applicable to various domains. In (JADON; JASIM, 2020), various techniques for creating dynamic summaries are explored. The main difference is the adopted clustering strategy. Gaussian and K-means clustering techniques are employed to select the frames, and the most promising results are those from the variants utilizing Gaussian clusters. Kumari, Dash and Sahu (2023) introduce an approach that chooses a representative frame to generate a keyshot. Within each cluster, frames are assessed, and their similarity is used for comparison. Thus, applying a similarity cutoff threshold, the k most similar frames are chosen per cluster. The video skim is then generated according to the temporal sequence of these selected frames. In (NAIR; MOHAN, 2023), video frames are grouped and separated into shots before the feature extraction process using CNNs. The shot separation strategy involves clustering similar frames into distinct groups. Keyshots are selected by excluding ambiguous frames and considering only the similarity between frames. However, this method overlooks the frames' temporal ordering and any structure within the video.

Reinforcement learning differs from supervised and unsupervised learning by using agents to evaluate the importance of each frame. The creation of the skim considers a reward function aimed at grouping concise and informative frames. Although reinforcement learning can consume significant resources due to the need to evaluate all skim possibilities through trial and error, it allows for including user perspectives to improve results (ZANG et al., 2023; PHAPHUANGWITTAYAKUL et al., 2021). The work in (ZANG et al., 2023) evaluates temporal modification to detect drastic changes occurring in the video. The modifications are used to delimit the video shots. To improve learning, this method uses a reward function that seeks to minimize the selection of the midpoints of each video segment. After, the k-medoids algorithm is applied to cluster the frames and generate the video skim.

Generative adversarial networks (GANs) represent another approach, creating frames related to existing ones through the competition between the two networks. One network processes the original video to create synthetic summaries, while the other distinguishes synthetic results from real ones. This process enhances the creation of summaries, not limited to existing frames. Although they can be seen as unsupervised approaches, GANs should be classified apart because they differ from the previously described unsupervised methods since they explore deep generative techniques not relying only on the frame features extracted at the beginning of the process. In (MINAIDI; PAPAIOANNOU; POTAMIANOS, 2023), the authors present a model that uses GANs to produce dynamic summaries. The model has generators and discriminators and uses attention mechanisms to determine which frames are more important than others. The method applies self-attention to deep features obtained with GoogleNet and creates the video summary through an LSTM. They do not specify the maximum limit for their summaries but use the selection of keyshots to represent the regions of interest in the summary.

3.3 Traditional Methods based on LSTM for Video Captioning

Given the amount of information processed in pattern recognition methods, objects, and description of images and videos, Krizhevsky, Sutskever and Hinton (2012), Ba, Mnih and Kavukcuoglu (2014), Xu et al. (2015) discuss the application of CNN and the advantages of processing images and videos, as it allows the network to learn the filters that will be used to solve the problem and reduce the need for manual implementation. Thus, there is a reduction in human effort and the maximization of computational resources to solve problems related to object recognition in images. Due to the methodology used by convolutional algorithms, currently processing a high number of images, the error rate is increasingly reduced, enabling an approximation of less than 0.3% compared to human performance.

However, the discretionary use of a CNN, as Krizhevsky, Sutskever and Hinton (2012) does not prevent the occurrence of prediction errors. To minimize this occurrence, it is possible to use two methods to prevent the occurrence of errors in CNN, namely *Dropout* and Data Augmentation. Dropout, despite the high processing cost in large databases, consists in discarding neurons that do not bring gain to the network, for that, some neurons must be zeroed. Data Augmentation, on the other hand, works by artificially increasing the base to improve your dataset. Due to the high complexity of object recognition, CNNs need the base to be prepared, seeking to reduce efforts related to ideal pattern matching during recognition. Not even ImageNet can solve this problem; therefore, the amount of data needed for processing would be gigantic, being necessary that during the pre-processing the algorithm can learn about some relative patterns of the base.

Ba, Mnih and Kavukcuoglu (2014) indicate that the use of deep learning makes it possible to improve the results obtained by object recognition applications. Thus, it presents a methodology for recognizing multiple objects in the same image. Using AM to define which object should be statically described. However, the description of an image depends directly on the point under analysis, called AM, as the reference used to describe an object concerns its meaning in the medium. Therefore, it can be said that at a given moment, the object that is the focus of object detection becomes a characteristic for other information that has more importance in the scope of the image. The author presents, in a static way, the use of AM in the context of object detection. This concept helped in the process of understanding the application of the method introduced for pattern recognition.

Xu et al. (2015) propose the evaluation of AM considering the comparative analysis between “hard” and “soft”, applying comparative metrics like BLEU to verify the behavior in four different image bases, checking the quality of the transcripts found in each AM. Thus, the production of the word that describes the next frame is characterized by the distribution of weights that define the center of each region. With the application of recurrent neural networks, the characteristic vectors of each frame are computed in a weighted way, allowing the application of multiple layers in an AM. Still, information loss can occur. This is due to the amount of information that passes through the network and the possibility of the method disregarding the weights used in the production of sentences. The use of LSTM is a way to preserve the error; its use guarantees the operation as a network decoder to produce the words in a vector of subtitles, due to the possibility of storing more information with its long-term memory. deadline. Therefore, the quality of the textual sentence generated with the analysis of the hidden states and words created previously is increased. The approach proposed by the author made it possible to compare the application of two methods of AM, demonstrating the improvement achieved by soft.

Considering the improvement in adding RL, this work will apply in the video context the hard approach to increase the quality obtained in the textual descriptions of this method.

Already Chan et al. (2015) apply Bidirecional Long Short-Term Memory (BiLSTM) for the treatment of spoken audio; however, it proved to be costly to train and reach relevant results. This fact was observed due to the high number of acoustic samples existing in an audio frame. To solve this problem, it was necessary to use a pyramid topology, applying non-linear reductions, seeking to reduce sampling time and frames.

According to Ordóñez and Roggen (2016) AM tries to approach the human visual system that can define, according to the scope, the best point or region to apply attention, taking the entire context of the image, avoiding describing from static form what is presented at the entrance. The application of recurrent neural networks and combining them with convolutional neural networks becomes a possibility to arrive at representations of natural language sentences with the application of AM, defining the most significant focus for the image.

According to Wang et al. (2018), the video description process may similarly use tactics applied to images. Still, it is necessary to use techniques to reduce the number of repeated image evaluations because, in a one-second video, several images overlap with approximate characteristics and few variations. Thus, one proposal is to use hierarchical summarization to list the best combination of image frames that present the video content without losing information. This tactic reduces the number of ratings needed to create video descriptors. However, it is necessary to pay attention to the loss of temporal information, as the generated tree, if it does not have the time variation, disregards any information referring to the moment that occurred in the original video.

3.4 Methods based on Transformers for Video Captioning

Experiments on machine translation tasks showed transformers as superior models in quality while being more parallelizable and requiring significantly less time to train than others. A transformer achieves good results because it can capture the relationship between tokens and the generated vocabulary. After that, transformer models have been successfully applied to several distinct tasks, such as machine translation, information retrieval, text classification, document summarization, image classification, image captioning, and video captioning (LEI et al., 2020; VYDANA et al., 2021; GUO et al., 2020; ZHANG; WEI; ZHOU, 2019; CHEN; FAN; PANDA, 2021; PAN et al., 2020; HUANG et al., 2019; SUN et al., 2019).

Some transformer models exploit the traditional encoder-decoder architecture, while others use a shared one. In a traditional encoder-decoder architecture, the en-

coder is separate from the decoder, and the encoded information about the video must be passed and used in some of the decoder steps. So, the decoder uses the data produced by the encoding network to generate each word/sentence (ZHOU et al., 2018). For shared architectures, video data and generated text tokens are passed as input to the same module, which (after operating on them) is responsible for both encoding and decoding. The latter architecture is generally used due to the reduction of operations, resulting in time and processing savings (LEI et al., 2020; CARDOSO et al., 2025).

Considering the results found by Cardoso, Guimarães and Patrocínio Jr (2021), it may be interesting to explore the use of different attention mechanisms in other parts within the transformer. The idea of reinforcing the characteristics learned during the current training period appears as a way of helping the transformer to learn new characteristics, increasing the quality of the results of the multi-head attention mechanism. Thus, the attention regions parallelized by the transformer (through multi-head attention) can be improved through adaptive attention to refine the attention distribution computed by each head.

Ging et al. (2020) presents two transformers to process the features, one for visual features and the other for textual features. Furthermore, both transformers use hierarchical strategies to capture the temporal relationship between frames. The concatenation of processed features is processed through an interaction process, called cross-modal cycle-consistency loss, which encourages semantic alignment between vision and text features in the joint embedding space

In (YAMAZAKI et al., 2022), a model is presented that uses visual features similar to a residual network. The approach used separates the encoder from the decoder, and during the encoder process, the visual features are fused with the output produced by the attention-based language model. During decoding, the transformer is separated into two transformers, the first being used to predict sentences and the second to predict descriptions. In the first transformer, a Gated Recurrent Unit (GRU) is used as a way to simulate a memory module in a transformer. The second uses a loss function to guide the descriptions according to the GT during the training process. The model presented by Yamazaki et al. (2023) consists of a strategy, based on an encoder and decoder, multimodal that processes visual, textual, and human detector features in the encoder in parallel. This strategy aims to produce features that tend to describe the events present in the video in a more discriminative way. Furthermore, the features processed by two transformers are fused in the decoder, which aims to increase coherence between the events distributed throughout the video. The models presented by Yamazaki et al. (2022) and Yamazaki et al. (2023) present an end-to-end video processing strategy.

Table 2 – Main Features of Related Works × Proposed Approach (Det indicates whether detection features are used; while Rec indicates whether sentence-level recurrence is used).

Method	Det	Rec	LSTM	Transformer	Memory	Hierarchy
MFT (XIONG; DAI; LIN, 2018)		✓	✓			
HSE (ZHANG; HU; SHA, 2018)		✓	✓			
GVD (ZHOU et al., 2019)	✓		✓			
GVDsup (ZHOU et al., 2019)	✓		✓			
AdvInf (PARK et al., 2019)	✓	✓	✓			
VTransformer (ZHOU et al., 2018)		✓		✓		
Transformer-XL (DAI et al., 2019)		✓		✓		
Transformer-XLRG (DAI et al., 2019)		✓		✓		
MART (LEI et al., 2020)		✓		✓	✓	
EMT (CARDOSO; GUIMARÃES; PATROCÍNIO JR, 2021)		✓		✓	✓	
COOT (GING et al., 2020)		✓		✓		✓
VLCAP (YAMAZAKI et al., 2022)		✓		✓	✓	
VLTINT (YAMAZAKI et al., 2023)		✓		✓	✓	
Proposed Approach		✓		✓	✓	✓

Source: Elaborated by the author

3.5 Summary on Video Captioning

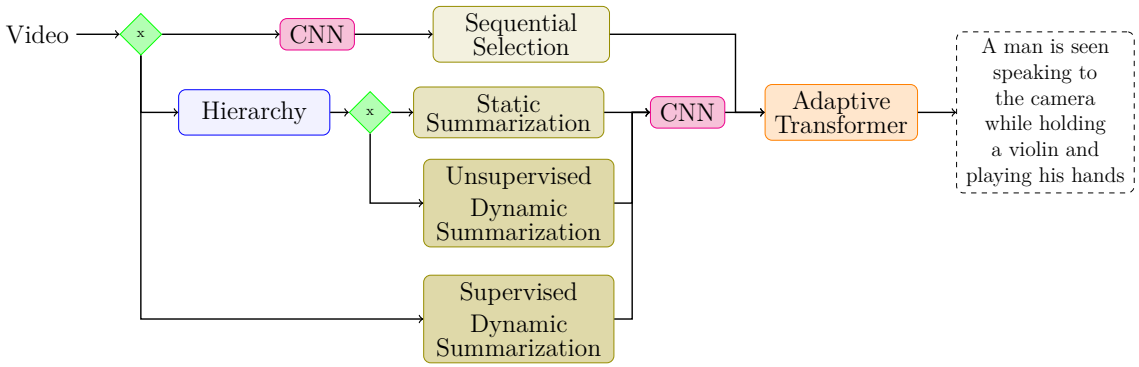
Table 2 summarizes the main characteristics of related work and the proposed approach of this work. This includes standard techniques, whether frame selection is done considering hierarchical techniques, and whether the approach uses a memory module in description generation. Unlike approaches presented in the literature, this work presents a new transformer that uses hierarchy to select more discriminative and representative frames, memory to help reduce the repetition of terms already produced, and a new attention mechanism to help re-weighting learned weights. It is expected to observe a reduction in repetition while maintaining coherence using the proposed transformer.

4 OUTLINE OF THE PROPOSED METHODS

In this Chapter, the strategies used for developing this work will be discussed, raising the decision strategies and the other tools used for making the proposed dense video captioning solution for coherent paragraphs, without loss of coherence and low repetition. The proposed architecture uses a shared transformer. Models that only use transformers do not use a recurrent network. To solve global dependency problems, transformers only use attention mechanisms. The transformer is a technique that allows for significant parallelization. However, the traditional model based entirely on attention to video captioning cannot solve repetition problems for coherent paragraphs. Therefore, it is necessary to add recurrent valuation strategies. Thus, this proposal has a memory module to guarantee the recurrent evaluation of sentences. The memory module uses an approach similar to networks like LSTM for long-term model dependencies.

Figure 13 illustrates the proposed method for video captioning. This work adopts three major components: (i) a video static or dynamic summary generator; (ii) a feature extractor; and (iii) a memory-augmented transformer with adaptive attention. The process of creating skims uses two different approaches, one supervised and the other unsupervised. The unsupervised approach is inspired by Belo et al. (2016) and copes better with (dis)similarities among video frames, producing a more valuable frame selection for the description process. Unlike Belo et al. (2016), it uses a watershed-based hierarchical method applied to a frame-similarity graph constructed from CNN-based frame descriptors and cosine similarity. Besides, frames are only considered similar if the difference

Figure 13 – Outline of the proposed process to generate captioning, considering sequential frames, static video summary, or dynamic video summary as the pass before the training model.



Source: Elaborated by the author

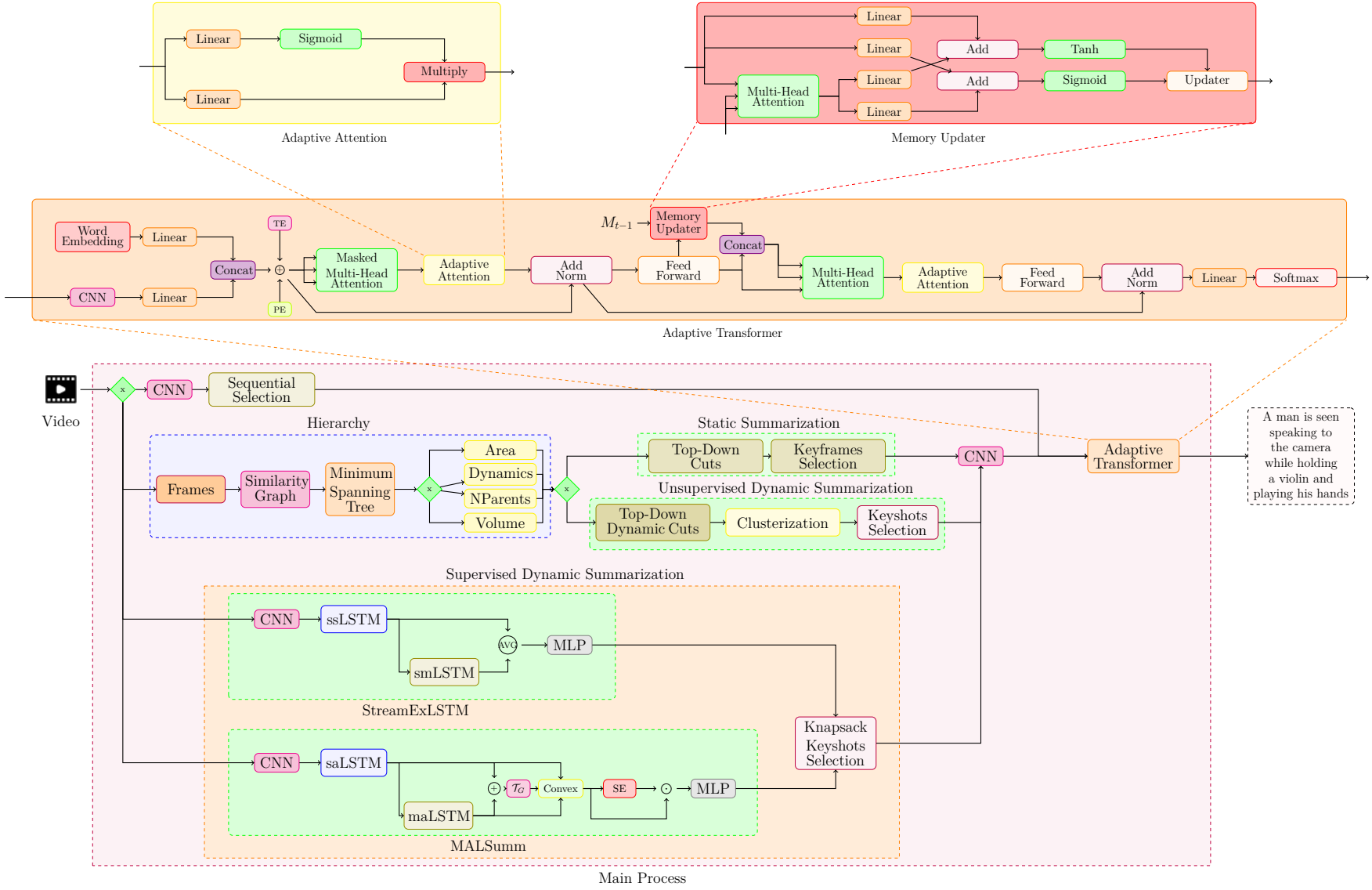
between their time indices is less than a fixed threshold. This can be used to restrict the relationship between video frames far away and implies that the hierarchical graph-based summarization approach is time-aware (which also differs from many works in the literature, including (BELO et al., 2016)).

Figure 14 presents a detailed overview of the proposed pipeline, which integrates video summarization and caption generation into a unified architecture. The model is initialized by processing the video through sequential frame selection, hierarchical graph-based selection, or LSTM-based selection. Unlike the sequential option, which uses the first available frames, the summarized frames can be represented by individual frames (keyframes) or dynamically grouped (keyshots). The resulting visual features are used as input to a transformer with augmented memory and equipped with adaptive attention, responsible for generating coherent textual descriptions. Figure 14 consolidates the interactions between all modules, providing a global perspective that will be decomposed into specialized components throughout the following sections.

In the lower left of the diagram, the process begins with the extraction of frame-level features using a convolutional neural network. These representations can follow a sequential path, where all frames are preserved and processed in their original temporal order. This basic path serves both as a benchmark for comparison and as a contingency mechanism in cases where summarization modules are unnecessary or less informative. By including the sequential alternative, the architecture maintains compatibility with traditional approaches while enabling adaptive selection strategies.

The static summarization module is presented as the first alternative to sequential processing. This component is based on the analysis of similarities between frames through graphs, using structures such as minimum spanning trees and top-down hierarchical cuts to identify visually representative keyframes. These operations highlight semantically important content that is dispersed over time. The figure shows how these key elements are selected and forwarded to the top of the architecture, offering a compact yet informative representation that reduces redundancy and preserves essential content.

Figure 14 – Detailed overview of the proposed architecture. The diagram illustrates the complete processing flow, from feature extraction to caption generation using the Adaptive Transformer with memory augmentation.



Source: Elaborated by the author

The second summarization path focuses on dynamic analysis, capturing temporal transitions that static methods may not perceive. This module supports two strategies: a graph-based dynamic partitioning mechanism and LSTM-based recurrent models (ssLSTM, smLSTM, saLSTM, and maLSTM). The graph-based strategy produces video summaries by grouping temporally coherent segments, while the LSTM-based approaches learn the importance of frames directly from user annotations, allowing the model to detect subtle temporal patterns. As illustrated in the figure, these paths converge on a set of selected keyframes that encode both movement and semantic evolution throughout the video.

Beyond the general workflow, the summarization stage incorporates four distinct hierarchical criteria to structure the video content and guide the selection of representative frames: watershed by area, which prioritizes segments based on their spatial influence; dynamics, which emphasizes the intensity of movement and temporal changes; number of parent frames, which reflects the structural relationships between the merging levels in the hierarchy; and volume, which captures the overall persistence of regions throughout the hierarchy. These complementary perspectives allow the model to analyze the video at multiple semantic scales, enriching the diversity of selected frames and improving the quality of the subsequent subtitling process.

Once the summarization module produces a compact set of visual inputs, the features enter the upper part of the architecture, where the Adaptive Transformer processes them for caption generation. This block incorporates two mechanisms highlighted in the figure: Adaptive Attention, responsible for reweighting previously calculated attention scores, and the Memory Updater, which maintains long-term contextual information between frames. These components allow the model to modulate its focus on relevant clues and accumulate temporal evidence, ultimately improving the coherence and context of the generated captions.

In the final step, the processed visual representations are combined with word embeddings within the transformer, allowing the decoder to produce natural language descriptions. The diagram illustrates how the summarization, attention, and memory mechanisms collectively shape the input to the captioning model, forming a cohesive workflow. Therefore, this figure serves as a reference point for understanding how each component interacts within the larger architecture. The following sections will further decompose these building blocks, presenting their mathematical formulations, design decisions, and experimental motivations.Thus, this work is divided into two sections. The first section focuses on the video summarization process, describing the strategies used to identify and select keyframes or key scenes that capture the essential temporal dynamics of each video. The second subsection presents the video captioning module, which uses the summarized visual content to generate coherent and semantically meaningful textual

descriptions. Together, these subsections provide a comprehensive overview of the techniques, architectural components, and design choices employed in the development of the proposed method.

4.1 Hierarchical Video Representation

Static and unsupervised dynamic summarization approaches adopt the same graph-based modeling that uses a weighted graph (G_δ, w), named time-aware frame similarity graph, to represent a video V_N, in which $G_\delta = (V, E_\delta)$ (see step 1 in Fig. 15). Each node $v_t \in V$ represents a frame $f_t \in \mathrm{V}_N$. There is an edge $e \in E_\delta$ with a weight $w(e) = \mathcal{D}(d(f_{t_1}), d(f_{t_2}))$ between two nodes v_{t_1} and v_{t_2} if the difference between their time indexes falls below a specified threshold δ_t, i.e.,

$$E_\delta = \{(v_{t_1}, v_{t_2}) | v_{t_1}, v_{t_2} \in V, v_{t_1} \neq v_{t_2}, |t_2 - t_1| \leq \delta_t\}. \tag{4.1}$$

This constraint on the frames' time indices limits the connections between temporally distant ones, ensuring that the summarization methods consider two frames as similar only if they are close in time. This contrasts with other approaches in the literature that consider two frames similar regardless of their temporal distance. By imposing this constraint, the summarization approach can evaluate the importance of frames over time, even if they reappear throughout the video.

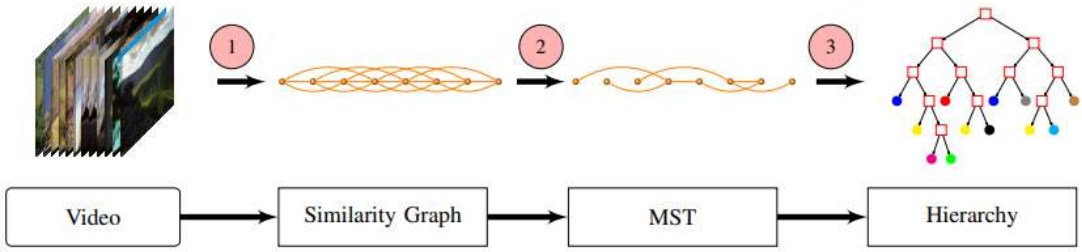
According to (COUSTY; NAJMAN, 2011), an MST $T^*_{G_\delta}$ generated from a weighted graph (G_δ, w) is the minimal graph representation of a hierarchy of that graph (step 2 in Fig. 15) . After generating $T^*_{G_\delta}$, a re-weighting process is applied to obtain a new weight map for it. That map is enough to represent any connected hierarchy $\mathcal{H}$ since it corresponds to the saliency map $\Phi_\mathcal{H}$ as stated by (COUSTY et al., 2018) (step 3 in Fig. 15) .

Once $\mathcal{H}$ is constructed, a k-sized hierarchical partition of (G_δ, w) would be given by the connected components obtained by removing $k-1$ edges from $T^*_{G_\delta}$ with the highest weights according to $\Phi_\mathcal{H}$.

Figure 15 presents the main steps to create a hierarchy from a video, and Algorithm 1 presents the hierarchical video summarization approach used for frame selection with the following steps: (i) construction of a frame similarity graph for a video (lines 1–14); (ii) calculation of a minimum spanning tree (MST) for the graph (line 15); and (iii) generation of a hierarchy through a re-weighting process based on that MST (line 16).

Inspired by Belo et al. (2016), this approach also uses a minimum spanning tree

Figure 15 – Outline of the proposed model for creating a hierarchy from a video.



Source: Elaborated by the author

(MST) to generate and represent a hierarchy. One possibility is the Kruskal method application to obtain the MST (line 15 of Algorithm 1) (KRUSKAL, 1956). After that, a hierarchy is calculated by re-weighting all edges belonging to the MST. Finally, unlike Belo et al. (2016), watershed hierarchy following the definition of Cousty and Najman (2011) was constructed (line 16 of Algorithm 1).

The complexity of the proposed approaches for static or dynamic summarization depends on the time spent on (i) graph construction; (ii) MST and hierarchy generation; and (iii) hierarchical graph cuts calculation. For a video with N frames, the construction of a time-aware frame similarity graph (G_δ, w) depends on δ_t and is given by $O(|V| + |E_\delta|) = O(|V| \times \delta_t)$ with $|V| = O(N)$ and $\delta_t << N$. According to (NAJMAN; COUSTY; PERRET, 2013), considering that edges can be sorted in linear time, the computation of an MST $T^*_{G_\delta}$ and generation of a hierarchy $\mathcal{H}$ based on $T^*_{G_\delta}$ is $O(|E_\delta| \times \alpha(|V|))$ in which $\alpha()$ is the extremely slowly growing inverse of the single-valued Ackermann function. Thus, the complexity of computing the MST and the hierarchy is related to the $O(|V| \times \delta_t)$. Finally, the calculation of hierarchical graph cuts is $O(V)$. So, the total complexity of the method is $O(|V| \times \delta_t) = O(\delta_t N)$.

After the summarization step, aligned features for appearance and optical flow are extracted only for the selected keyframes or keyshots and used both to induce a video captioning model during training time and to feed the trained model during inference time to generate a description for a new unseen video. One of the proposed models presented for this work for the task of video caption is named Adaptive Transformer, in which a memory-augmented transformer with a shared architecture explores the attention mechanism to re-weight the importance of data generated by the self-attention module. The motivation for that is to explore the results generated by the self-attention module to improve readability through diminishing repetition.

Algorithm 1: Hierarchical video representation

Input: A video $\mathcal{V}$, threshold value δ_t
Output: A hierarchy $\mathcal{H}$

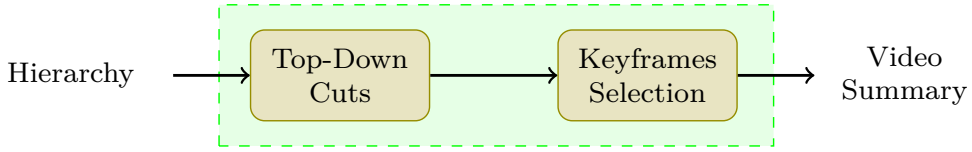
Create a graph $G = (V, E)$ with a vertice set $V = \emptyset$ and an edge set $E = \emptyset$
for all *frame* $f \in \mathcal{V}$ **do**
  if $f \notin V$ **then**
    $V := V \cup \{f\}$ // Insert f in V if f does not belong to it
  end if
  $d_f := GenerateDescriptor(f)$ // Obtain a descriptor for frame f
  for all *frame* $f' \in \mathcal{V}$ such that $f' \neq f$ and $|t_f - t_{f'}| < \delta_t$ **do**
    if $f' \notin V$ **then**
      $V := V \cup \{f'\}$ // Insert f' in V if f' does not belong to it
    end if
    $d_{f'} := GenerateDescriptor(f')$ // Obtain a descriptor for frame f'
    $G.AddEdge(f, f')$ // Insert edge (f, f') using frame similarity as weight
  end for
end for
$T := G.Obtain\_MST\_from\_Graph()$
$\mathcal{H} := T.Generate\_Hierarchy\_from\_MST()$
Return $\mathcal{H}$;

4.2 Static Video Summarization

After constructing the hierarchy based on one of the four considered criteria, Figure 16 presents the static summarization process based on keyframe selection. The Top-Down Cuts module initiates the structural reduction process to identify the most representative levels of the hierarchical tree. The central idea of this module is to traverse the hierarchy from the top to progressively more granular levels, interrupting the process when a threshold of the nodes exceeds a certain limit. This avoids the selection of redundant regions and preserves only the groupings that capture relevant aspects of the visual organization of the video. This procedure allows transforming a dense hierarchical structure into a reduced set of candidate segments for summarization, serving as a basis for the subsequent selection of keyframes.

The Keyframe Selection module evaluates the resulting nodes from the cuts to extract the frames that best represent each identified region. This process can operate in

Figure 16 – Static Video Summarization



Source: Elaborated by the author

Algorithm 2: Hierarchical video summarization

Input: A Hierarchy H, summary size k
Output: A list of keyframes $\mathcal{K}$
1: $\mathcal{K} := H.Find\_Keyframes(k)$ // Remove $k-1$ edges from H to generate k sets // and select the central vertice of each set as keyframe
2: Return $\mathcal{K}$;

two distinct ways. One is statically defined, where the user predefines the desired number of keyframes, and the algorithm selects the most representative frames based on this fixed constraint. Another approach occurs when the number of frames is not specified; the algorithm uses an adaptive threshold determined by the standard deviation of the distribution of the visual characteristics of the nodes. This variable threshold allows the selection to automatically adjust to the level of variability in the video, producing more coherent summaries in scenarios with different degrees of visual complexity.

4.2.1 Hierarchical Time-Aware Graph-Based Summarization for Static Keyframe Selection

Different summarization techniques face a challenge in how they choose key elements. Unsupervised methods can't guarantee the same results every time and can be influenced by individual interpretations of a subjective result. To improve frame selection, one promising approach is to use hierarchical methods to identify the most important scenes. Unlike the traditional approach that uses a sequential selection policy for frame selection, the static summarization step chooses frames based on their similarity. It adopts a hierarchical graph-based summarization method to obtain the most valuable frames (as keyframes). In (BELO et al., 2016), the authors provide a unified framework for video summarization that uses a minimum spanning tree of frames and a weight map based on hierarchical observation scales computed over that tree. The weight map is generated from the frame similarity graph in which the clusters (or connected components) can easily be inferred. Moreover, the use of this strategy allows the application of a similarity measure between clusters during graph partition, instead of considering only the similarity between isolated frames.

Algorithm 2 presents the process for generating keyframes in a static approach to choosing frames. In this way, receives as input the hierarchy generated by Algorithm 1 and the value k that represents the number of cuts that will be made, and presents the process for generating keyframes in a static approach to choosing frames, in which the value k represents the number of cuts that will be made. Thus, with the horizontal cuts in the hierarchy, the components that represent the video are generated according to the temporal coherence applied in Algorithm 1. For each component, only one frame is chosen

as the keyframe; this choice is made considering the temporal center of the frames of the video. After that, the selection of the central vertice of each set as a keyframe is a simple task (line 1 of Algorithm 2).

At first, the use of an extra step in the video captioning pipeline, such as the proposed hierarchical summarization approach, may appear to increase the total computational time. Choosing a smaller set of more informative frames reduces computational time. This is because the description generator processes a much smaller number of frames without any loss of quality in the final result.

4.2.2 Hierarchical Time-Aware Graph-Based Summarization for Dynamic Keyframe Selection

In the video summarization domain, keyframe selection is a critical task to ensure that the generated summary captures the essential content of the video effectively. Traditionally, methods employ a static number of selected frames to create these summaries. This static approach involves predetermining a fixed number of frames to be included in the summary, regardless of the content or complexity of the video, as used in Algorithm 2. Although simple, this method is often insufficient when dealing with videos that feature numerous similar scenes, as it can lead to redundancy and the inclusion of repetitive content, ultimately diluting the quality and informative content of the summary.

To fix these limitations, a dynamic frame selection based on the specific characteristics of the video content is proposed. This approach involves dividing the hierarchical structure of the video into several connected components, from which central frames, selected in terms of their chronological order, are identified as keyframes for the summary, as presented in Algorithm 1. By dynamically determining the number of these components, the method adapts the size of the video summary according to the diversity and complexity inherent in the video content. This flexibility allows the summarization process to be more responsive to video variations, such as differences in scene frequency and similarity, thus producing more concise and relevant summaries.

The dynamic approach offers significant advantages over the static method by ensuring that the size of the summary is proportional to the actual informational content of the video. Videos with numerous similar scenes benefit from this method, as it avoids the inclusion of redundant frames, increasing the effectiveness and overall coherence of the summary. Consequently, this approach adapts the summarization process to the unique attributes of each video, providing a more accurate and contextually relevant representation of the original content. By leveraging these adaptive techniques, video summarization can achieve higher quality and greater utility in diverse applications, from media content management to efficient information retrieval and consumption.

Algorithm 3: Hierarchical time-aware video summarization – HieTaSum

Input: A hierarchy $\mathcal{H}$
Output: A list of keyframes $\mathcal{K}$
1: $\mathcal{K} := \mathcal{H}.Dynamic\_Selection\_of\_Keyframes()$ // Remove edges from $\mathcal{H}$ to obtain // frame sets and select the central // vertice of each set as keyframe
2: Return $\mathcal{K}$;

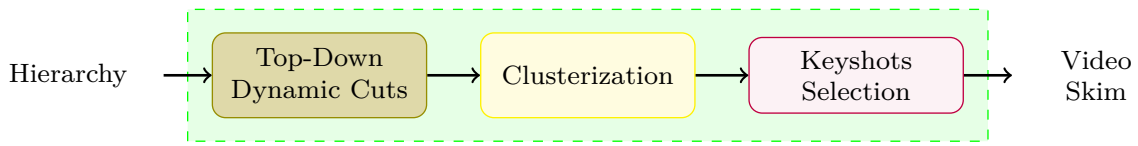
This approach (see Algorithm 3) uses the hierarchy generated in Algorithm 1. Once the hierarchy $\mathcal{H}$ is established, a hierarchical segmentation of G_δ produces a video summary of size k. This is achieved by removing the $k-1$ edges with the highest weights from $\mathcal{H}$. Rather than generating a fixed-size video summary, it adopts a strategy to identify the moment when stability is reached during the edge removal process, which is similar to, but distinct from, the method used in (BELO et al., 2016).

4.3 Unsupervised Video Summarization

Starting from a previously constructed hierarchy using watershed by area, dynamics, volume, or number of parents, the Figure 17 presents the dynamic summarization process, which begins with the Top-Down Dynamic Cuts module. Unlike static summarization based on keyframes, which aims to select isolated points in time, here the focus is on preserving temporal continuity and movement between frames. In this module, the hierarchy is traversed from the top to progressively more specific levels, seeking to identify components that represent coherent events in the video. These dynamic cuts allow the detection of regions that maintain visual and temporal consistency, serving as a basis for the formation of keyshots.

After the dynamic cuts, the Clustering module organizes the resulting frames into structured groupings, in order to construct continuous segments that represent actions or events in the video. This step can operate in a way that is oriented towards the internal similarity of each component found. In one strategy, each component of the cut hierarchy is associated with its central frame, and the other frames are grouped based on their visual similarity to this centroid. This process tends to create compact keyshots, aligned

Figure 17 – Unsupervised Dynamic Video Summarization



Source: Elaborated by the author

with the average behavior of the event that each component represents, prioritizing the internal coherence of the segments.

The second strategy deepens the hierarchy to more granular levels, making continuous cuts until reaching the leaves. This approach seeks to capture fine variations within each event, resulting in more detailed groupings that are sensitive to local transitions, even if this produces segments that are more spaced out along the timeline. Finally, the Keyshots Selection module uses the generated groupings to define which sets of frames make up the final keyshots. As a result, the keyshots preserve the dynamics of the scenes, offering a summarization that maintains complete events and temporal flow, an essential characteristic for applications that depend on visual evolution between frames.

4.3.1 Hierarchical Time-Aware Graph-Based Video Skimming

Video skimming is a critical process when dealing with large video datasets that contain a significant amount of similar frames. The sheer volume of data can make it challenging to extract valuable information efficiently, especially when redundancy is present. Video skimming techniques can help mitigate the effects of redundancy by reducing the amount of redundant data, allowing for more effective analysis and decision-making. Moreover, skimming can improve the accessibility of video content by providing a condensed and easy-to-follow summary that can be quickly reviewed, thereby unlocking its full potential for research and commercial applications. Efficient and effective video analysis is crucial for a broad range of fields, including healthcare, security, and entertainment, among others. As such, video skimming techniques are becoming increasingly popular, and their development is a highly active area of research. Thus, superficial video analysis represents a crucial process that can help extract valuable information from video data more efficiently.

The proposed approach used to create the summary of the video, (called **Hie**rarchical **T**ime-**a**ware **Skim**ming – HieTaSkim), instead of creating a fixed-size video summary, an adaptive strategy is adopted to identify the point of stability during the edge removal process, similar to (BELO et al., 2016). The result of that process is a set of keyshots that would be the basis for video skim generation, represented by the Algorthm 4. Thus, a given edge e with the highest weight according to $\Phi_{\mathcal{H}}$ is removed from $T^*_{G_\delta}$ only when $w(e) \geq \mathbf{F}(e)$ in which $\mathbf{F}(e)$ represents an equilibrium measure function given by

$$\mathbf{F}(e) = \mu_w(e) + \gamma \times \sigma_w(e) \tag{4.2}$$

in which $\mu_w(e)$ and $\sigma_w(e)$ represent the average and the standard deviation, respectively, of all edge weights in the connected component containing edge e, and γ is a parameter denoting the allowed variability. Let $\mathcal{NC}$ be the number of keyshots that this adaptive

Algorithm 4: Hierarchical time-aware video skimming – HieTaSkim

Input: A hierarchy $\mathcal{H}$
Output: A list of keyshots $\mathcal{S}$
1: $\mathcal{S} := \mathcal{H}.Dynamic\_Selection\_of\_Keyshots()$ // Remove edges from $\mathcal{H}$ to obtain // a set of frames to represents // the keyshots
2: Return $\mathcal{S}$;

approach generates. To avoid a summary with little diversity (containing only the central portion of the original video), $\mathcal{NC}$ should be at least $\mathcal{NC}_{min}$. So, the adaptive approach should be applied until it produces at least $\mathcal{NC}_{min}$ keyshots.

After producing keyshots through hierarchical graph cuts, temporally central frames for every keyshot are selected as a basis for the video skimming. Some frames are selected around every appointed central frame to form the video skim. The maximum size $\mathcal{S}$ of each sequence of frames is given by

$$\mathcal{S} = \frac{N \times p}{\mathcal{NC}} \tag{4.3}$$

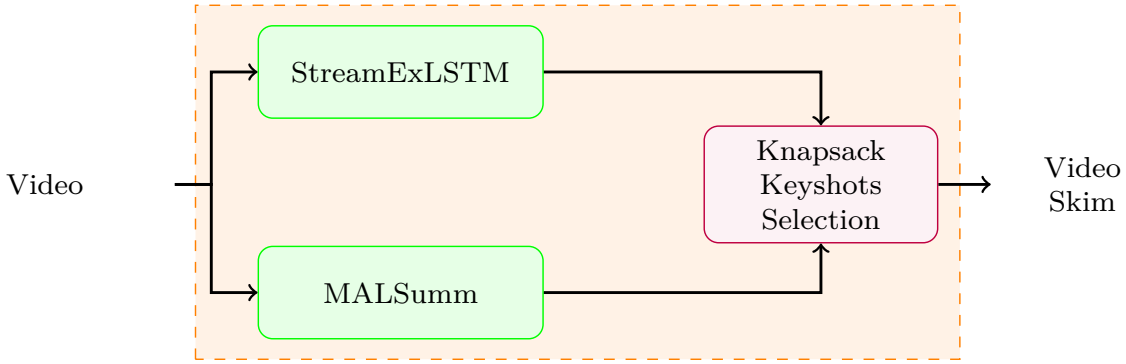
in which p is the maximum relative size (set by the user) of the final skim to the original video.

4.4 Supervised Video Summarization

In contrast to unsupervised approaches, which rely exclusively on the hierarchical structure derived from similarity relationships between frames, the supervised method incorporates recurrent models trained directly to learn relevant temporal and semantic patterns. The Figure 18 presents the supervised dynamic summarization process, where the first version, StreamExLSTM, represents a simplification of the original xLSTM architecture (BECK et al., 2024), retaining only the recurrent core necessary to capture long-term dependencies in the video stream. Its goal is to produce continuous scores throughout the video, indicating the most relevant regions for subsequent keyshot selection. This reduced version prioritizes training efficiency and stability while maintaining good generalization capability.

The second supervised variant, called MALSumm, expands on xLSTM by introducing an enhanced memory module that seeks to better preserve the information accumulated throughout the sequence. While traditional recurrent models suffer from memory degradation over time, the additional module reinforces the retention of significant patterns, especially in long videos or those with spaced events. Furthermore, MALSumm incorporates a Squeeze-and-Excitation (SE) module, responsible for dynamically weight-

Figure 18 – Supervised Dynamic Video Summarization



Source: Elaborated by the author

ing the ratio between newly observed information and that retrieved from memory. This adaptive weighting improves the model's ability to identify key events without losing temporal context.

After processing the video stream, both StreamExLSTM and MALSumm produce sequences of scores that indicate the temporal relevance of each frame or local group of frames. Unlike unsupervised versions, which rely on top-down cuts and clustering to detect events, supervised versions learn directly from annotated examples the structure of an appropriate summary. This allows capturing more abstract semantic patterns, such as actions, narrative transitions, or context changes, which purely similarity-based methods may miss. As a result, supervised models tend to generate keyshots more aligned with human annotation behavior.

The keyshot selection process adopted in this work follows the classical 0-1 Knapsack formulation similarly to the employed in (APOSTOLIDIS et al., 2021b). Each shot s_i is treated as an item characterized by a VALUE v_i, corresponding to the relevance score predicted by the supervised model, and a WEIGHT w_i, representing its temporal duration in frames. The objective is to select the subset of segments that maximizes the total importance without exceeding the maximum allowed duration W. Formally, the problem is defined as:

$$\max_{\mathbf{x}} \sum_{i=1}^{N} v_i x_i \quad \text{subject to} \quad \sum_{i=1}^{N} w_i x_i \leq W, \quad x_i \in \{0,1\}.$$

Here, $x_i = 1$ indicates that shot s_i is included in the final summary. The 0-1 formulation is particularly suitable for video summarization because it enforces discrete selection decisions and prevents undesired fragmentation within the resulting summary.

In addition to its conceptual simplicity, this formulation provides a globally optimal solution through dynamic programming, differing from heuristic methods that rely solely on local similarity criteria. By evaluating all feasible segment combinations, the knapsack ensures that the final summary maximizes total relevance while respecting the

temporal constraint. This property is especially important when the supervised model produces multiple relevance peaks close in time, which could lead to redundant selections if thresholding or heuristic rules were used. Thus, the knapsack operates as a structural regularizer on the temporal layout of the summary.

Finally, the knapsack-based selection establishes a clear connection between the supervised relevance scores and the temporal organization of the resulting video skim. Whereas the unsupervised approaches depend on hierarchical structures derived from similarity graphs (area, dynamics, number of parents, or volume), the knapsack formulation allows the model to directly prioritize patterns learned from human annotations. Consequently, even when shots differ in duration or exhibit internal variability, the optimization ensures that only the most informative segments, considering the trade-off between value and cost, are included in the final summary. The result is a compact, semantically coherent set of data aligned with keyword evaluation.

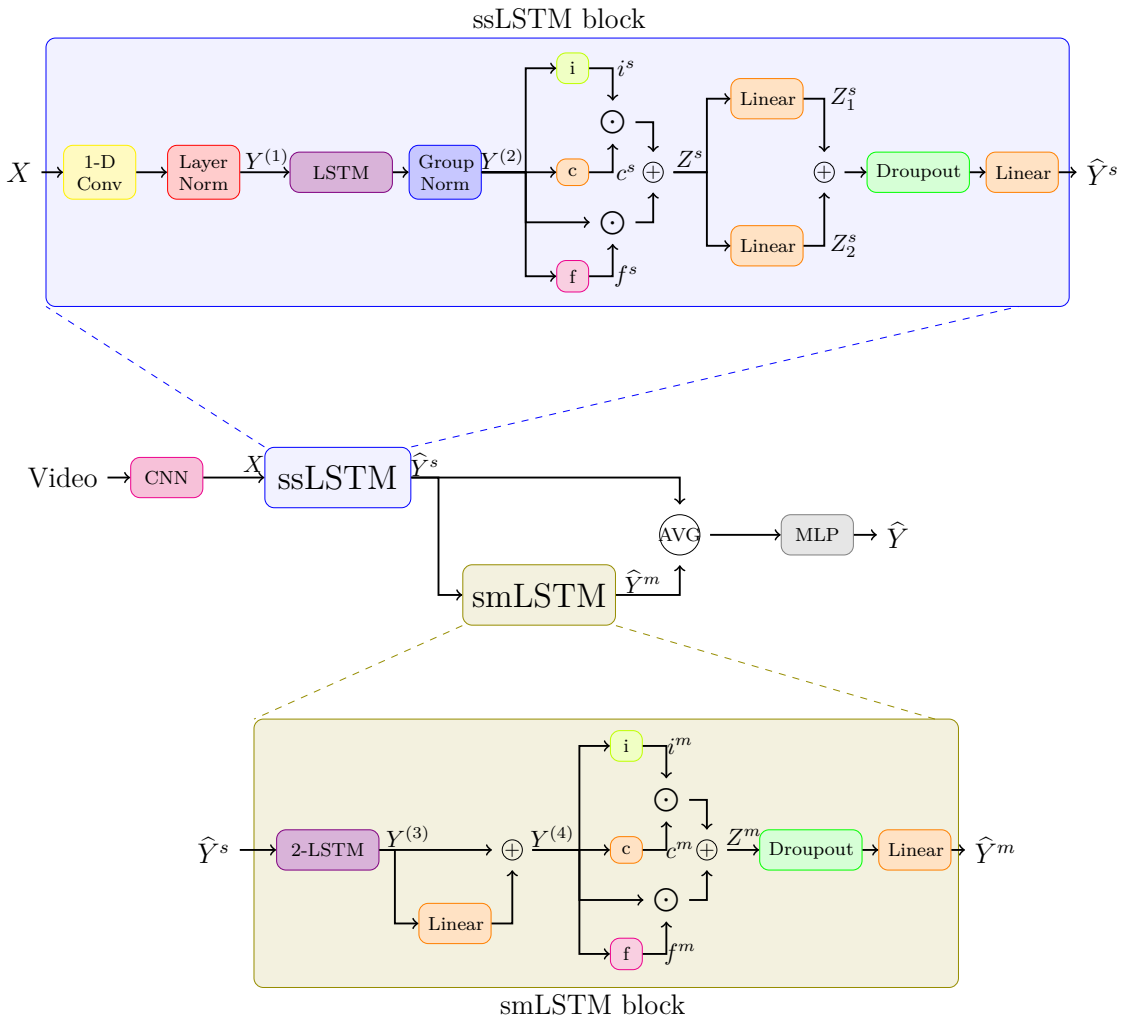
4.4.1 Simplified Extended LSTM Supervised Dynamic Video Summarization (StreamExLSTM)

The problem of video skimming is related to the amount of similar information laid out sequentially with little or no variation. The relationship between the distribution of frames directly impacts the sentences generated and implies an increase or not in repetition.

This work uses a simplified xLSTM model that integrates structural and attention-based enhancements in video skimming to address the challenges of generating concise yet representative video summaries. The approach leverages two key modules, ssLSTM and smLSTM, to capture local and global temporal dependencies within video frames. These components work in synergy to refine the selection of important frames while preserving the summary's temporal coherence.

Figure 19 illustrates the proposed process, The first step is, given a video consisting of T frames, the proposed simplified xLSTM model initially extracts a sequence of deep feature representations, denoted as $X = \{x_t\}_{t=1}^{T}$, where each feature vector $x_t \in \mathbb{R}^D$ encodes a high-dimensional representation of the corresponding frame. These feature vectors are obtained using a Convolutional Neural Network (CNN), named GoogleNet (SZEGEDY et al., 2015), and pre-trained on ImageNet, which captures spatial and semantic features crucial for understanding the video content. Thus, each feature vector is defined as $x_t = \{x_{t,i}\}_{i=1}^{D}$, in which D represents the dimensionality of the extracted feature space. The generated deep features are used as input to the simplified xLSTM that models temporal dependencies, ensuring that key patterns and the most discriminative information are effectively retained and scene transitions are smoothed.

Figure 19 – Overview of the proposed StreamExLSTM model. (a) Overall pipeline: frame-level features extracted by a pre-trained CNN are processed in parallel by two specialized modules – ssLSTM and smLSTM – designed to capture local structure and global temporal dependencies, respectively. Their outputs are averaged and passed through a multilayer perceptron (MLP) to produce frame-level importance scores. (b) The ssLSTM block models short-term dependencies using a combination of convolutional operations, normalization, and gated LSTM transformations to highlight salient local patterns. (c) The smLSTM block enhances global sequence modeling by applying stacked LSTMs with attention-aware gating, supported by residual and dropout connections to maintain stability and generalization.



Source: Elaborated by the author

The ssLSTM module illustrated in Figure 19 serves as the initial processing layer, transforming input feature representations through convolutional, recurrent, and normalization layers. Specifically, the ssLSTM employs a 1D convolutional layer to extract local patterns, followed by a layer normalization step to stabilize activations. The core of this module is a single-layer LSTM network, which encodes sequential dependencies within

the video frames. Additionally, a Group Normalization layer refines the activations before passing them through gated transformations, ensuring a balanced selection of salient features. Let $X \in \mathbb{R}^{T \times C_{\text{in}}}$ be an input sequence with T time steps and C_{in} channels.

A 1D convolution with kernel size K produces an output $Y^{(1)} \in \mathbb{R}^{T' \times C_{\text{out}}}$ computed as:

$$y_{t,j}^{(1)} = \sum_{i=1}^{C_{\text{in}}} \sum_{k=1}^{K} w_{k,i,j} \cdot x_{t+k-1,i} + b_j, \tag{4.4}$$

for $t = 1, \ldots, T'$ and $j = 1, \ldots, C_{\text{out}}$, where $w \in \mathbb{R}^{K \times C_{\text{in}} \times C_{\text{out}}}$ and $b \in \mathbb{R}^{C_{\text{out}}}$ are the convolutional weights and biases, respectively.

Layer normalization is applied to each time step t of $Y^{(1)}$. For the feature vector $y_t^{(1)} \in \mathbb{R}^{C_{\text{out}}}$, the normalized output is given by:

$$\hat{y}_t^{(1)} = \frac{y_t^{(1)} - \mu_t}{\sqrt{\sigma_t^2 + \epsilon}} \odot \gamma + \beta, \tag{4.5}$$

where

$$\mu_t = \frac{1}{C_{\text{out}}} \sum_{j=1}^{C_{\text{out}}} y_{t,j}^{(1)}, \quad \sigma_t^2 = \frac{1}{C_{\text{out}}} \sum_{j=1}^{C_{\text{out}}} \left(y_{t,j}^{(1)} - \mu_t \right)^2, \tag{4.6}$$

and $\gamma, \beta \in \mathbb{R}^{C_{\text{out}}}$ are learnable parameters, and ϵ is a small constant for numerical stability. Denote the output of this stage as $Y^{(2)} \in \mathbb{R}^{T' \times C_{\text{out}}}$.

The normalized sequence $Y^{(2)}$ is then processed by a traditional LSTM. For each time step t, the LSTM computes $H_t = \text{LSTM}(Y_t^{(2)})$. Let the LSTM output be represented as $H = \{h_t\}_{t=1}^{T'}$. Group normalization is applied to the LSTM outputs. Assume each hidden state $h_t \in \mathbb{R}^H$ is partitioned into G groups, each of size $\frac{H}{G}$. For the g-th group at time t, denote the corresponding subvector as $h_t^{(g)}$. The group-normalized output is computed as:

$$\hat{h}_t^{(g)} = \frac{h_t^{(g)} - \mu_t^{(g)}}{\sqrt{\sigma_t^{(g)2} + \epsilon}} \odot \gamma^{(g)} + \beta^{(g)}, \tag{4.7}$$

where

$$\mu_t^{(g)} = \frac{1}{H/G} \sum_{j \in \mathcal{G}_g} h_{t,j}, \quad \sigma_t^{(g)2} = \frac{1}{H/G} \sum_{j \in \mathcal{G}_g} \left(h_{t,j} - \mu_t^{(g)} \right)^2, \tag{4.8}$$

with $\mathcal{G}_g$ denoting the set of indices for group g, and $\gamma^{(g)}, \beta^{(g)} \in \mathbb{R}^{H/G}$ are learnable parameters for that group. The final output of the process is $Y^{(4)} \in \mathbb{R}^{T' \times H}$, obtained by concatenating the normalized groups.

After applying group normalization, let the resulting signal be denoted by $Y^{(4)} \in \mathbb{R}^{T' \times H}$. This normalized signal is then fed into four distinct operations to compute the intermediate signals $i = \mathcal{T}_i(Y^{(4)})$, $c = \mathcal{T}_c(Y^{(4)})$, $f = \mathcal{T}_f(Y^{(4)})$, and $r = \mathcal{T}_r(Y^{(4)})$ as residual connection, where $\mathcal{T}_i$, $\mathcal{T}_c$, $\mathcal{T}_f$, and $\mathcal{T}_R$ denotes learnable transformations applied to the group-normalized output. The output Z is then computed as the sum of two element-wise multiplications: the product of i and c and the product of the residual signal (R) with f $Z = i \odot c + r \odot f$, where $\odot$ represents the Hadamard (element-wise) product. et $Z \in \mathbb{R}^{N \times d}$ denote the output signal from the previous operations, where N is the number of samples and d is the feature dimension. The signal Z is processed by two parallel linear transformations $Z_1 = W_1 Z + b_1$, and $Z_2 = W_2 Z + b_2$, where $W_1, W_2 \in \mathbb{R}^{d' \times d}$ and $b_1, b_2 \in \mathbb{R}^{d'}$ are the learnable weight matrices and bias vectors, respectively.

The outputs of the two linear transformations are then summed element-wise $Z_{\text{sum}} = Z_1 + Z_2$. To prevent overfitting, a dropout operation is applied to $Z_{\text{drop}} = \text{Dropout}(Z_{\text{sum}})$, where (Dropout$(\cdot)$) randomly zeroes some of the elements during training according to a specified dropout probability. Finally, a fully connected layer is applied to $\hat{Y} = W_{\text{fc}}\, Z_{\text{drop}} + b_{\text{fc}}$, with ($W_{\text{fc}} \in \mathbb{R}^{d'' \times d'}$) and ($b_{\text{fc}} \in \mathbb{R}^{d''}$) representing the weights and bias of the final layer, and (d'') being the dimension of the final output. The proposed formulation ensures that the model captures both short-term dependencies (via convolution) and long-term dependencies (via LSTM) while normalization techniques and residual connections enhance stability and performance.

Following the ssLSTM, the processed frame representations are forwarded to the smLSTM module, which extends the temporal modeling capacity through multi-layer recurrent processing as presented in Figure 19. This module consists of a two-layer LSTM network that enhances long-range dependencies across video sequences. A linear transformation introduces a residual connection, enhancing learned representations while maintaining stability. Moreover, a dropout mechanism is incorporated to prevent overfitting and improve generalization. The output of the smLSTM retains the most informative patterns while mitigating the effects of redundant frames.

The proposed smLSTM memory module follows a structured sequence of operations. The output of ssLSTM is used as input of the smLSTM module $H_t = \text{LSTM}(\hat{Y}_t)$, where H_t represents the hidden state output of the LSTM given input $\hat{Y}_t$. The output of the LSTM inside of smLSTM was summed $S_t = R_t + L_t$ as a residual connection $R_t = H_t$ with a linear projection of the same LSTM $L_t = W_l H_t + b_l$, where R_t is the residual connection and L_t is the linearly transformed output with weight matrix W_l and bias b_l. The output S_t was used as input to four components $I_t^m, C_t^m, F_t^m, R_t^m$. The update of the memory was represented by $M_t = (I_t \cdot C_t) + (F_t \cdot R_t)$, where M_t is the memory update mechanism. To prevent overfitting in the memory updater mechanism, a dropout operation is used to regularize the signal $M_t' = \text{Dropout}(M_t)$. The final output of the

smLSTM module is a fully connected layer $O_t = W_f M'_t + b_f$, where O_t is the output after applying a fully connected transformation with weight W_f and bias b_f.

The output of the simplified XLSTM was computed using the average of the smLSTM and ssLSTM that balances the contributions of ssLSTM and smLSTM. Figure 19 shows the proposed process, let L_1 and L_2 denote the ssLSTM and smLSTM, respectively. Their average is computed as $L_{\text{avg}} = \frac{1}{2}(L_1 + L_2)$. The output is used as input to two sequential fully connected layers as $V_1 = W_1\, L_{\text{avg}} + b_1$, and $V_2 = W_2\, V_1 + b_2$, where W_1 and W_2 are the weight matrices and b_1 and b_2 are the bias vectors for the respective layers. The final output of the pipeline is given by V_2.

By combining convolutional pre-processing, recurrent feature extraction, dynamic channel re-weighting, and attention-based refinement, the simplified xLSTM model effectively captures both local and global temporal dependencies in video sequences. This structured approach ensures that the generated summaries maintain high representativeness while significantly reducing computational overhead.

4.4.2 Memory-Augmented LSTM for Dynamic Video Summarization (MAL-Summ)

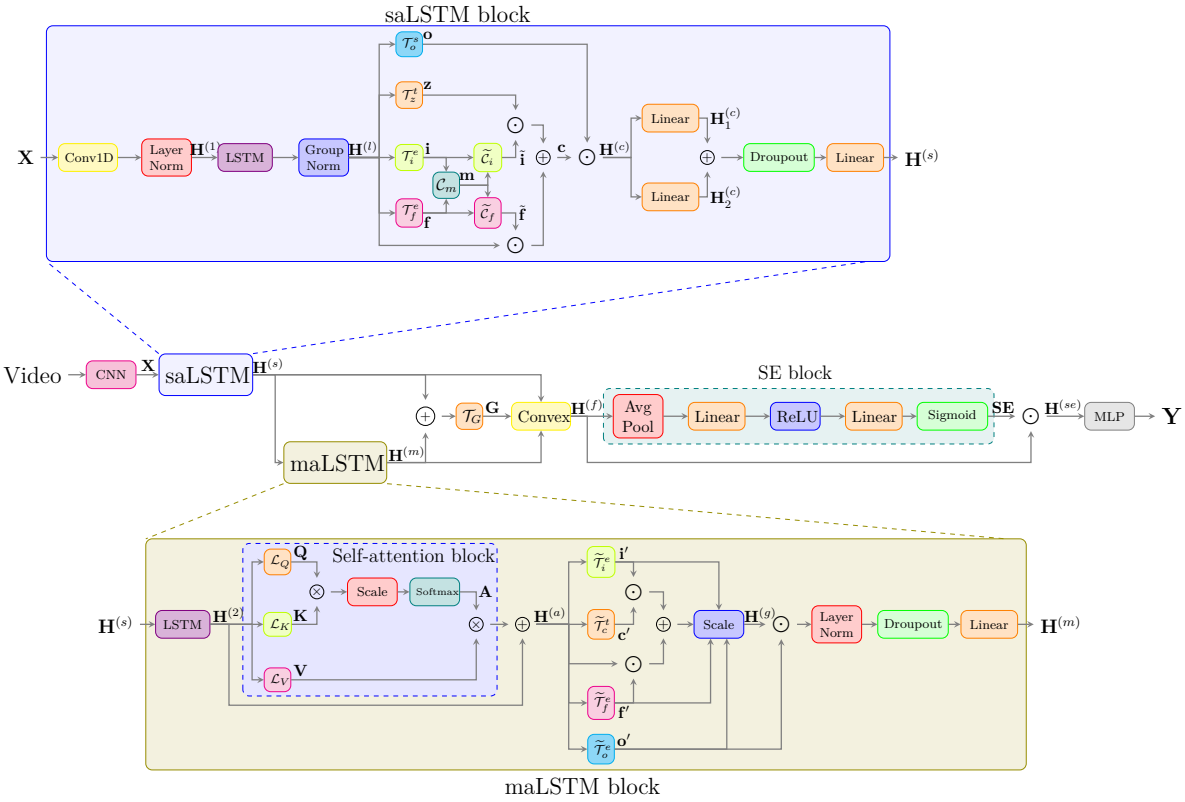
To deal with high amounts of repetitive or low-variation content in videos, video skimming techniques tend to produce a temporally coherent summary that seeks to represent the video content in small video shots. The occurrence of similar scenes distributed throughout the video sequence tends to impact the ability to produce diverse and concise summaries.

Let $\mathbf{X} \in \mathbb{R}^{T \times d}$ denote the input sequence of frame-level features, where T is the number of frames and d is the feature dimension. The proposed model (Figure 20) processes $\mathbf{X}$ using two complementary LSTM-based branches a structural-aware LSTM (saLSTM) and a memory-augmented LSTM with attention (maLSTM).

The proposed saLSTM module (Figure 20) first enhances the input sequence $\mathbf{X}$ through a temporal convolution, capturing local structural patterns across time, followed by layer normalization to stabilize the feature distribution. This intermediate representation $\mathbf{H}^{(1)}$ is then processed by a standard LSTM, which models long-range dependencies and produces a contextualized representation.

To further refine these representations, Group Normalization is applied to the LSTM outputs, where the feature channels are partitioned into groups and normalized independently. This step reduces the dependence on batch statistics and ensures stable feature distributions even with small batch sizes, yielding the final contextualized output

Figure 20 – Overview of the proposed Memory-Augmented LSTM for Dynamic Video Summarization (MALSumm) model. The architecture integrates two complementary recurrent blocks: one specialized in preserving essential temporal information and another in learning new temporal dependencies. Additionally, an adaptive clustering-based strategy is employed to generate video skims and define summary lengths, ensuring concise yet informative video representations.



Source: Elaborated by the author

$\mathbf{H}^{(l)}$.

$$\mathbf{H}^{(1)} = \text{LayerNorm}(\text{Conv1D}(\mathbf{X})) \tag{4.9}$$

$$\mathbf{H}^{(l)} = \text{GroupNorm}(\text{LSTM}(\mathbf{H}^{(1)})) \tag{4.10}$$

Gating mechanisms refine the LSTM output $\mathbf{H}^{(l)}$ via elementwise operations. Gates ($\mathbf{i}$, $\mathbf{f}$, $\mathbf{o}$, $\mathbf{z}$) are computed through learned transformations, enabling dynamic control of information flow. A modulation term $\mathbf{m}$, derived from log-scaled gates and bias λ via a

max operation, helps emphasize salient features while suppressing noise.

$$\mathbf{i} = \mathcal{T}_i^e(\mathbf{H}^{(l)}) = \exp(\mathbf{W}_i\,\mathbf{H}^{(l)}), \tag{4.11}$$
$$\mathbf{f} = \mathcal{T}_f^e(\mathbf{H}^{(l)}) = \exp(\mathbf{W}_f\,\mathbf{H}^{(l)}), \tag{4.12}$$
$$\mathbf{z} = \mathcal{T}_z^t(\mathbf{H}^{(l)}) = \tanh(\mathbf{W}_z\,\mathbf{H}^{(l)}), \tag{4.13}$$
$$\mathbf{o} = \mathcal{T}_o^s(\mathbf{H}^{(l)}) = \text{sigmoid}(\mathbf{W}_o\,\mathbf{H}^{(l)}), \tag{4.14}$$
$$\mathbf{m} = \mathcal{C}_m(\mathbf{f}, \mathbf{i}) \;\; = \max(\log \mathbf{f} + \lambda, \log \mathbf{i}) \tag{4.15}$$

The learnable scalar λ stabilizes gating by balancing input and forget signals. Recalibrated gates $\tilde{\mathbf{i}}$ and $\tilde{\mathbf{f}}$ are normalized via the modulation term $\mathbf{m}$, guiding the cell state update $\mathbf{c}$ from past outputs and new candidates. Next, the output $\mathbf{H}^{(c)}$ is generated by combining the cell state with the output gate $\mathbf{o}$, effectively integrating both temporal and structural dependencies.

$$\tilde{\mathbf{i}} = \widetilde{\mathcal{C}}_i(\mathbf{i}, \mathbf{m}) = \exp(\log \mathbf{i} - \mathbf{m}), \tag{4.16}$$
$$\tilde{\mathbf{f}} = \widetilde{\mathcal{C}}_f(\mathbf{f}, \mathbf{m}) = \exp(\log \mathbf{f} + \lambda - \mathbf{m}), \tag{4.17}$$
$$\mathbf{c} = \tilde{\mathbf{f}} \odot \mathbf{H}^{(l)} + \tilde{\mathbf{i}} \odot \mathbf{z}, \tag{4.18}$$
$$\mathbf{H}^{(c)} = \mathbf{o} \odot \mathbf{c} \tag{4.19}$$

Next, $\mathbf{H}^{(c)}$ is processed through two parallel linear projections, followed by dropout regularization and a final linear transformation:

$$\mathbf{H}_1^c = \text{Linear}(\mathbf{H}^{(c)}), \quad \mathbf{H}_2^c = \text{Linear}(\mathbf{H}^{(c)}) \tag{4.20}$$
$$\mathbf{H}^{(s)} = \text{Linear}\left(\text{Dropout}\left(\mathbf{H}_1^c + \mathbf{H}_2^c\right)\right) \tag{4.21}$$

The refined output $\mathbf{H}^{(s)}$ is subsequently fed into a second LSTM layer, named maLSTM (Figure 20), producing the representation $\mathbf{H}^{(2)}$, which is further enriched through a self-attention mechanism. This involves projecting $\mathbf{H}^{(m)}$ into query ($\mathbf{Q}$), key ($\mathbf{K}$), and value ($\mathbf{V}$) spaces using learned linear transformations. Attention weights $\mathbf{A}$ are computed via scaled dot-product attention, capturing contextual relevance among temporal positions. The attended output $\mathbf{H}^{(a)}$ is then derived by weighting the value vectors,

enabling the model to selectively emphasize informative segments across the sequence.

$$\mathbf{H}^{(2)} = \text{LSTM}(\mathbf{H}^{(s)}) \tag{4.22}$$
$$\mathbf{Q} = \mathcal{L}_Q(\mathbf{H}^{(2)}) = \mathbf{W}_q\,\mathbf{H}^{(2)} \tag{4.23}$$
$$\mathbf{K} = \mathcal{L}_K(\mathbf{H}^{(2)}) = \mathbf{W}_k\,\mathbf{H}^{(2)} \tag{4.24}$$
$$\mathbf{V} = \mathcal{L}_V(\mathbf{H}^{(2)}) = \mathbf{W}_v\,\mathbf{H}^{(2)} \tag{4.25}$$
$$\mathbf{A} = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right) \tag{4.26}$$
$$\mathbf{H}^{(a)} = \mathbf{H}^{(2)} + (\mathbf{A}\mathbf{V}) \tag{4.27}$$

To further refine the attention-enhanced features, a second gating mechanism is applied to $\mathbf{H}^{(m)}$ using learned transformations for input, forget, and output gates. These gates modulate the flow of information through elementwise operations, and a candidate state $\mathbf{c}'$ is introduced via a nonlinearity. The gated representation $\mathbf{H}^{(g)}$ is computed by weighting the current state and candidate, normalized with a small constant ϵ for stability. The output is modulated by an output gate and refined through Layer Normalization, Dropout, and a Linear transformation, which together enhance convergence, prevent overfitting, and project features into a new space.

$$\mathbf{i}' = \widetilde{\mathcal{T}}_i^e(\mathbf{H}^{(a)}) = \exp(\mathbf{W}'_i\,\mathbf{H}^{(a)}) \tag{4.28}$$
$$\mathbf{f}' = \widetilde{\mathcal{T}}_f^e(\mathbf{H}^{(a)}) = \exp(\mathbf{W}'_f\,\mathbf{H}^{(a)}) \tag{4.29}$$
$$\mathbf{o}' = \widetilde{\mathcal{T}}_o^e(\mathbf{H}^{(a)}) = \exp(\mathbf{W}'_o\,\mathbf{H}^{(a)}) \tag{4.30}$$
$$\mathbf{c}' = \widetilde{\mathcal{T}}_c^t(\mathbf{H}^{(a)}) = \tanh(\mathbf{W}_c\,\mathbf{H}^{(a)}) \tag{4.31}$$
$$\mathbf{H}^{(g)} = \frac{\mathbf{f}' \odot \mathbf{H}^{(a)} + \mathbf{i}' \odot \mathbf{c}'}{\mathbf{i}' + \mathbf{f}' + \mathbf{o}' + \epsilon} \tag{4.32}$$
$$\mathbf{H}^{(m)} = \text{Linear}(\text{Dropout}(\text{LayerNorm}(\mathbf{o}' \odot \mathbf{H}^{(g)}))) \tag{4.33}$$

Finally, a dynamic fusion gate $\mathbf{G}$ adaptively combines the outputs of the structural-aware LSTM ($\mathbf{H}^{(s)}$) and the memory-augmented LSTM ($\mathbf{H}^{(m)}$). The gate uses a sigmoid over combined representations to weigh each branch's contribution. The fused output $\mathbf{H}^{(f)}$ is a convex combination that flexibly integrates structural and memory-aware features, enhancing representation quality.

$$\mathbf{G} = \mathcal{T}_G(\mathbf{H}^{(s)}, \mathbf{H}^{(m)}) = \text{sigmoid}(\mathbf{W}_g\,(\mathbf{H}^{(s)} + \mathbf{H}^{(m)})) \tag{4.34}$$
$$\mathbf{H}^{(f)} = \mathit{Convex}(\mathbf{G}, \mathbf{H}^{(s)}, \mathbf{H}^{(m)}) = \mathbf{G} \odot \mathbf{H}^{(s)} + (1 - \mathbf{G}) \odot \mathbf{H}^{(m)} \tag{4.35}$$

A squeeze-and-excitation (SE) mechanism recalibrates channel-wise importance of fused features $\mathbf{H}^{(f)}$. Global average pooling captures temporal statistics, which pass through a bottleneck with ReLU and sigmoid to produce channel weights $\mathbf{s}$. These weights

modulate the features via elementwise multiplication, yielding $\mathbf{H}^{(se)}$ with enhanced focus on informative channels. Final scores $\mathbf{Y} \in \mathbb{R}^{T\times 1}$ are obtained by projecting $\mathbf{H}^{(se)}$ through a two-layer network, mapping each timestep to a scalar that reflects its relevance for summarization. The resulting scores serve as the model's predictions for keyframe selection or segment importance.

$$\mathbf{SE} = \text{sigmoid}(\mathbf{W}_2\,\text{ReLU}(\mathbf{W}_1\,\text{AvgPool}(\mathbf{H}^{(f)}))) \tag{4.36}$$

$$\mathbf{H}^{(se)} = \mathbf{H}^{(f)} \odot \mathbf{SE} \tag{4.37}$$

$$\mathbf{Y} = \text{MLP}(\mathbf{H}^{(se)}) \tag{4.38}$$

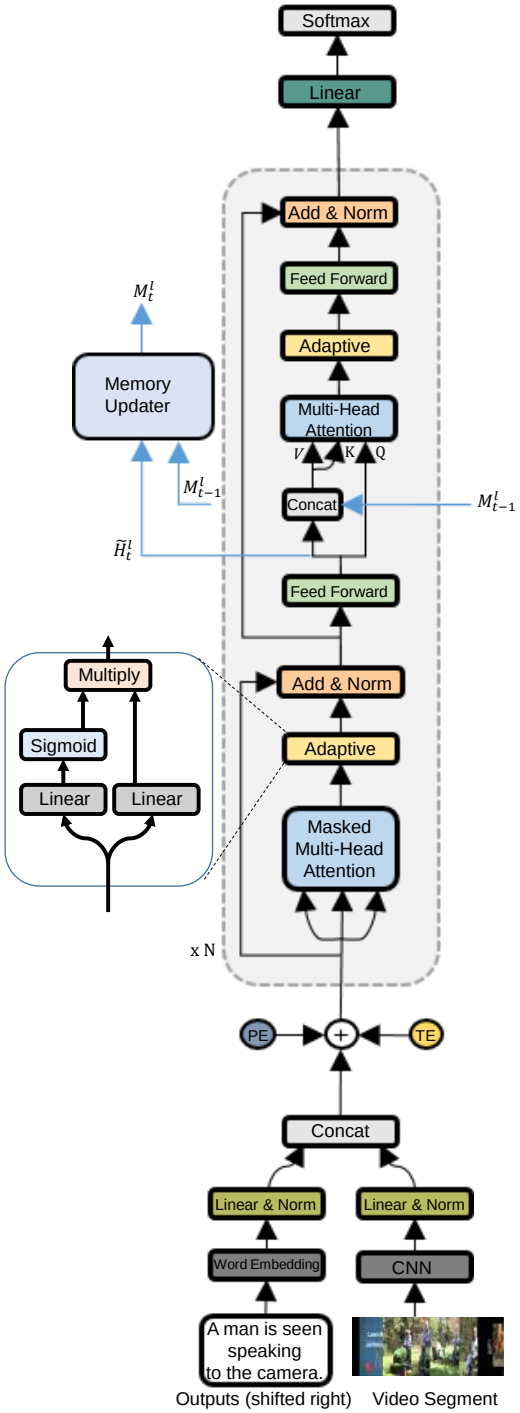
4.5 Video Captioning

To deal with coherence issues, this work proposes to explore an improvement to the self-attention module within the transformer's main backbone (just after the multi-head attention modules). So, was adopted additional attention blocks to emphasize the data generated by self-attention and cross-attention. Figure 21 presents the inclusion of Adaptive Attention, used to enhance multi-head attention, focusing on reducing repetition. The memory module is still an important part of the shared transformer module, which is responsible for capturing the long-term dependency of the sentences.

Similar to that proposed by Huang et al. (2019), which uses attention mechanisms to reinforce smooth attention in the image captioning task. This work includes a simplified version of that mechanism after the existing multi-head attention in the transformer. This simplification was first introduced by Cardoso, Guimarães and Patrocínio Jr (2021) inside of the memory updater after the multi-head attention. This process seeks to highlight the characteristics that are considered important in smoothing out others (CARDOSO; GUIMARÃES; PATROCÍNIO JR, 2021). Figure 21 shows the Adaptive Transformer with the presence of two modules (called Adaptive), both are identical. This work uses a version of adaptive attention to apply re-weighting to the results used as input to the memory updater, and adaptive attention is also applied to the results of cross-attention. It is worth mentioning that adaptive attention was used before only inside the memory update module, which differs from its use in this work (CARDOSO; GUIMARÃES; PATROCÍNIO JR, 2021).

Figure 22 presents a comparison between the attention-on-attention mechanism shown in Figure 22(a) and the adaptive attention mechanism depicted in Figure 22(b). The attention-on-attention mechanism is designed to be applied in various parts of the model, needing its application alongside skip connections. In contrast, adaptive attention is intended for just after multi-head attention but before normalization. This approach

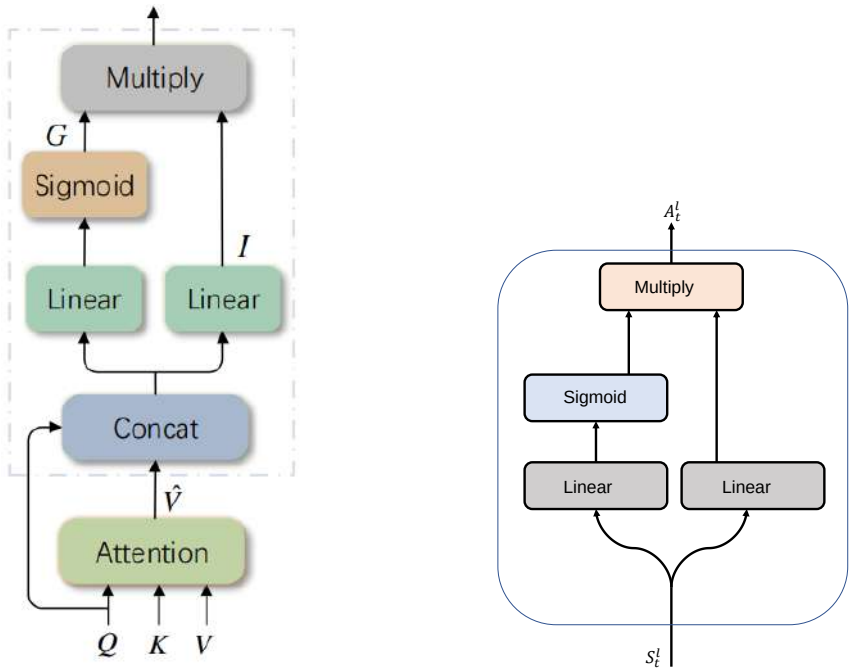
Figure 21 – An illustration of the Adaptive Transformer architecture highlighting one of its adaptive attention modules (and showing a detailed representation of it).



Source: Elaborated by the author

leverages the multi-head attention skip connection to eliminate the need for an additional concatenation step, which the transformer will perform during normalization. The skip

Figure 22 – Comparision between the attention module proposed by Huang et al. (2019) and the adaptive attention proposed by Cardoso, Guimarães and Patrocínio Jr (2021).



Source: Elaborated by the author

connections are crucial for preserving information flow and addressing the vanishing gradient problem, while the attention mechanism enables the network to capture long-range dependencies and understand the input text's context. The role of the skip connection in adaptive attention becomes evident upon evaluating Figure 21.

The detailed outline of the adaptive attention module applied after the multi-head attention used in the results on $A_t^l \in \mathbb{R}^{T_m \times d}$ is given by Equation 4.39:

$$A_t^l = sigmoid(W_{mhl}^l MH_t^l) \odot W_{mhr}^l MH_t^l + b_a^l, \tag{4.39}$$

in which W_{mhl}^l, and W_{mhr}^l are trainable weights, b_a^l are trainable bias, and MH_t^l stands for multi-head attention results.

The strategy employed in the adaptive attention mechanism involves evaluating data that has already been processed by another attention mechanism. Thus, the quality of the distribution is enhanced, making the descriptions more discriminating and more related to the content they intend to represent. The use of second attention seeks to amplify the importance of the information. Thus, the application of adaptive attention acts as a refinement process on the results of previous attention, and this has an impact on the data learned on the memory updater module both on the regulator Z_t and on the

newly learned information C_t. So, as new information is highlighted, it tends to replace the data stored in memory.

4.5.1 Sequential Selection in the Video Captioning task

Generating textual descriptions for video content has seen significant advancements with the advent of transformer models. These models excel at capturing long-range dependencies and contextual information, making them particularly suitable for sequential data such as videos. In recent approaches, transformers have been employed to process up to 100 sequential frames in each video, leveraging their self-attention mechanism to generate coherent and contextually relevant subtitles. By considering a fixed number of frames, the model can effectively manage computational resources and maintain a consistent input size, which simplifies training and increases the model's ability to learn intricate temporal relationships within the video segment.

The proposed adaptive transformer model that uses sequential frames has two main steps. The first consists of extracting features from the first fixed number of frames and disregarding any other frames, if any. The second step consists of training a model based on the adaptive transformer with the extracted features. The first step are repeated to test the model, but in the second, the features are used as input for the trained model to generate descriptions for the test dataset.

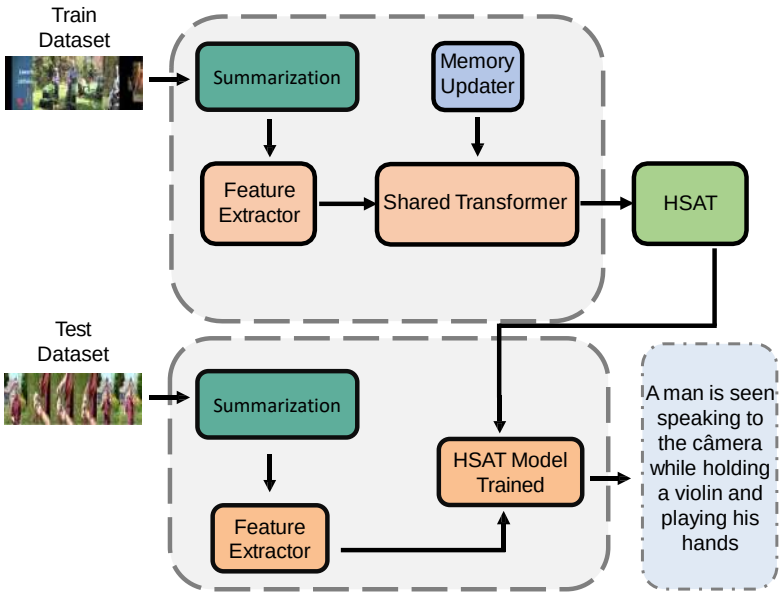
However, videos often contain more than a fixed number of frames, which poses a challenge for these models. When the number of frames exceeds this limit, critical information from parts of the video may be omitted, potentially leading to incomplete or inaccurate subtitles. This truncation can result in the loss of essential contextual details beyond the first fixed number of frames, thus failing to capture the full narrative of the video. To address this issue, researchers should explore strategies for dynamic frame selection, efficient sampling techniques, or hierarchical modeling approaches that can encapsulate the broader temporal context without taxing the computational capacity of the transformer model. Balancing the fixed frame limit and the need for a comprehensive understanding of the video remains a key area of focus for improving the robustness and accuracy of video captioning systems.

4.5.2 Using Static Summarizer in the Video Captioning task

Unlike the traditional approach that uses a sequential selection policy for frame selection, the static method chooses frames based on their similarity. It adopts a hierarchical graph-based summarization method to obtain the most valuable frames (as keyframes)

The problem of dense video captioning is related to the amount of similar infor-

Figure 23 – One of the proposed methods for using the summarization process as preprocessing to the video captioning task, as (i) a hierarchical graph-based summarizer; (ii) a feature extractor; and (iii) a shared memory-augmented transformer with adaptive attention.



Source: Elaborated by the author

mation laid out sequentially with little or no variation. The relationship between the distribution of frames directly impacts the sentences generated and implies an increase or not in repetition.

To deal with that issue, the video captioning process based on the static keyframes is divided into three steps (see Fig. 23). The first is the selection of the best set of frames for each video through a hierarchical summarization approach which splits a video into subsets of similar frames and selects the central frame (using similarity among frames) as the keyframe. The second step extracts features for the appearance and optical flow of previously selected keyframes. Finally, the third step is description generation using an Adaptive Transformer, which was trained to work with smaller sets of keyframes (but describing the most important video contents).

4.5.3 Using Unsupervised Dynamic Summarizer in the Video Captioning task

The problem of dense video captioning is closely related to the presence of semantically similar content arranged sequentially with minimal variation (CARDOSO; GUIMARÃES; PATROCÍNIO JR, 2021, 2022). The distribution of such frames directly affects the quality of generated sentences and may lead to increased repetition across the

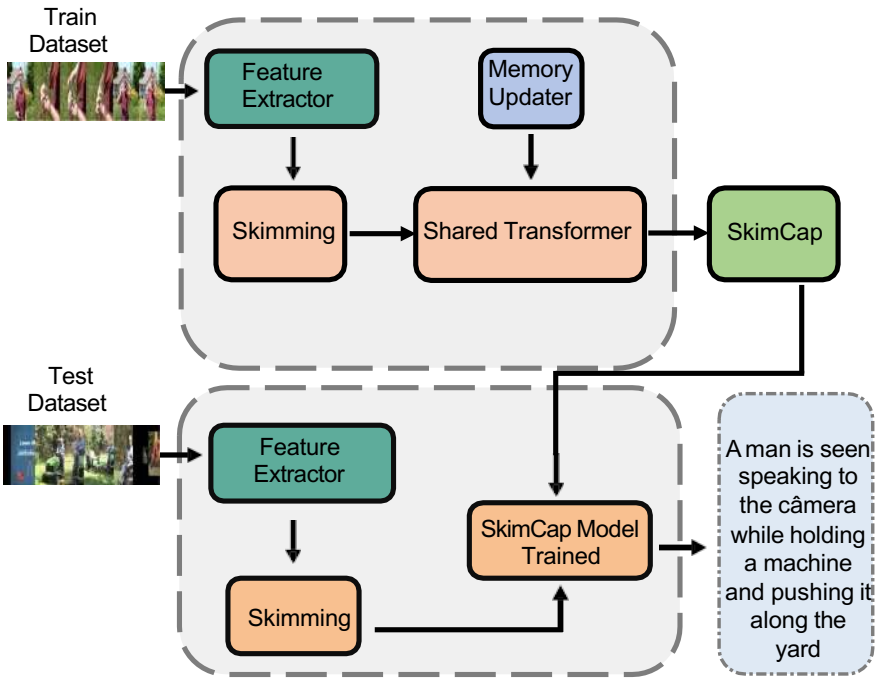
output descriptions (CARDOSO; GUIMARÃES; PATROCÍNIO JR, 2023). To deal with large video content, especially those with more than a fixed number of frames, presents significant challenges. To address this, several approaches have been explored, including the use of hierarchical graph-based methods for video skimming, which have shown promise in effectively summarizing videos for captioning tasks.

The hierarchical graph-based approach to video skimming leverages a structured representation of video content by constructing a graph where nodes represent video frames and edges signify the temporal and contextual relationships between these frames. This method allows dynamic frame selection based on hierarchical segmentation of the graph, ensuring that critical and diverse segments of the video are included in the summary. By capturing essential features through this hierarchical structure, the transformer model can generate more coherent and contextually accurate captions as it processes a distilled version of the video that retains meaningful content and narrative flow.

To address coherence issues, this work proposes SkimCap, presented in Figure 24, a hierarchical feature selection framework designed to overcome the limitations of conventional frame sampling strategies in video captioning. Existing methods frequently rely on sequential sampling, selecting the first k frames and discarding the remainder of the video, regardless of its semantic content. This approach becomes particularly problematic in long videos, where relevant events may occur beyond the initial portion. SkimCap addresses this challenge by identifying and selecting the most informative frames based on content diversity and contextual relevance, rather than temporal position. Sequential selection methods, widely adopted in prior works, implicitly assume that the early segments of a video contain the most relevant content (CARDOSO; GUIMARÃES; PATROCÍNIO JR, 2023). However, this assumption breaks down in scenarios where essential information appears after a certain frame threshold. The proposed method mitigates this bias by assessing frame importance globally, ensuring that semantically rich segments are preserved even when they occur later in the video. SkimCap is particularly effective in datasets where videos exceed a given length, preventing premature truncation and improving the richness and coherence of generated descriptions.

In comparison, traditional approaches that use sequential fixed frames often face limitations when dealing with videos with more than a fixed number of frames. These methods typically truncate the video to the first a fixed number of frames, which can lead to omitting vital information that appears later in the video. This truncation can result in incomplete subtitles that do not accurately represent the entire context of the video. While these approaches are computationally efficient and straightforward, their inability to adapt to different video lengths and content complexity reduces their effectiveness for comprehensive video captioning tasks.

Figure 24 – The outline of the proposed method with (i) a feature extractor; (ii) a hierarchical graph-based skimming; and (iii) a shared memory-augmented transformer with adaptive attention.



Source: Elaborated by the author

The SkimCap framework operates by assigning a relevance score to each frame using a hierarchical clustering technique over pre-extracted visual features. These features are derived from both appearance (ResNet-based) and motion (optical flow) descriptors. For videos with more than 100 frames, the visual content was modeled as a graph, and top-down cuts were applied along a hierarchy derived from the feature space. This strategy allows retaining a representative subset of frames, maximizing coverage without redundancy. (CARDOSO et al., 2024). Videos with 100 frames or fewer are processed in full, preserving all available information. This mechanism avoids the arbitrary exclusion of semantically relevant but temporally delayed content.

SkimCap supports multiple hierarchical schemes to guide the selection process. Specifically, four variants of watershed hierarchies are evaluated: area, dynamics, volume, and number of parents. These hierarchies enable the structuring of videos into coherent clusters, facilitating the identification of both dominant and subtle visual segments (COUSTY; NAJMAN, 2011). The selection policy operates over these hierarchies to identify the candidate frames that best summarize the video at varying levels of granularity. This work explore two selection policies for extracting representative frames. In the *central strategy*, the method selects the central frame of each clustered shot (or segment) produced by the hierarchy, yielding a locally coherent representation of each

region. This approach promotes contextual continuity by focusing on the most representative point within each cluster. The *spaced strategy*, in contrast, samples non-contiguous frames across multiple clusters, resulting in broader temporal coverage. This is particularly advantageous for videos with heterogeneous or sparsely distributed content, as it increases the likelihood of capturing infrequent or dispersed events.

On the other hand, hierarchical approaches that use only keyframes as input for training also present an attractive alternative. By selecting keyframes based on significant visual changes or predefined criteria, these methods aim to capture the most informative parts of the video while disregarding redundant or less critical frames. This strategy reduces the computational load and focuses the model's attention on crucial moments. However, relying solely on keyframes can lead to a loss of temporal coherence, as the transition between keyframes can miss intermediate actions or contextual build-up crucial to generating accurate and fluent captions. Thus, the hierarchical graph-based approach offers a balanced solution by dynamically selecting frames through a structured representation while maintaining content diversity and temporal coherence. This method contrasts with the limitations of sequential fixed-frame approaches, which can truncate essential information, and hierarchical keyframe approaches, which can miss the intermediate context. By leveraging the strengths of hierarchical graph structures, video captioning with transformers can achieve more accurate and comprehensive descriptions, accommodating the complexities of longer video content while optimizing computational efficiency.

By replacing sequential sampling with a hierarchy-aware, content-driven selection mechanism, SkimCap significantly enhances the quality of input features for downstream video-language models. The proposed method improves temporal diversity, reduces semantic redundancy, and avoids the systematic omission of important events due to their temporal position, ultimately contributing to more expressive and coherent video descriptions.

4.5.4 Using Supervised Dynamic Summarizer in the Video Captioning task

In addition to the previously described sequential, static and dynamic unsupervised strategies, this work incorporates a supervised summarization pipeline as an alternative mechanism for selecting representative temporal segments. In this formulation, summaries produced by supervised models, such as StreamExLSTM and MALSumm, serve as structured inputs for the Adaptive Transformer used in caption generation. To this end, this approach seeks to investigate how supervised learning of frame importance, derived from human-labeled data, can influence the captioning step when applied to a different dataset. Thus, the summarizers are trained exclusively on the SumMe dataset and then applied directly to ActivityNet, allowing the framework to isolate the behavior

of supervised summarization under cross-domain conditions.

A practical challenge arises from the incompatibility between the feature domains in the two datasets. While the supervised summarization models trained on SumMe expect GoogleNet-style features, the ActivityNet captioning pipeline relies on high-dimensionality ResNet and optical flow descriptors. To reconcile these representations, Principal Component Analysis (PCA) is employed to project ActivityNet features onto a lower-dimensional space aligned with the GoogleNet domain. This projection preserves the underlying structure necessary for summarization while allowing the use of pre-trained supervised models without modifications. After projection, the summarization model generates keyshots whose indices are integrated into the Adaptive Transformer as temporal guides.

Supervised summarization differs from unsupervised approaches by leveraging patterns learned directly from human annotations. While unsupervised hierarchical methods rely solely on statistical properties of the feature space, such as cluster topology or structural salience, supervised models encode notions of relevance that reflect subjective human judgments. This introduces an additional semantic layer to the frame selection process. However, it also means that supervised methods can implicitly encode dataset-specific biases, and the transfer of these biases to a new domain must be considered when interpreting their behavior. To reduce this bias, supervised summarization offers richer prior semantic information, but less independence from the characteristics of the source dataset.

The supervised strategy contrasts with sequential frame selection, a common method in older captioning systems. Sequential sampling adopts the simplifying premise that the first frames sufficiently represent the video. This approach benefits from low computational cost and guaranteed temporal continuity, but does not take into account the possibility of relevant events occurring in later positions. Supervised summarization, by selecting frames based on predicted importance rather than temporal order, circumvents this limitation and provides inputs that better align with the narrative organization of the video. However, it introduces additional computational overhead compared to sequential heuristics.

Compared to unsupervised hierarchical summarization, which emphasizes structural diversity and coverage across the feature space, supervised methods emphasize patterns of relevance that align with human annotations. Thus, unsupervised approaches exploit the inherent structure of the data, while supervised approaches impose an external notion of importance. Therefore, unsupervised methods are domain-agnostic and adapt naturally to new datasets without the need for retraining, while supervised methods can capture more subtle semantic nuances that may not emerge solely from feature simi-

larity. Their combination with a Transformer-based model therefore, enables a spectrum of design choices that balance generality, semantic richness, and computational cost.

The integration of supervised summarization with the Adaptive Transformer provides a flexible methodological framework in which the captioning model can operate on a set of temporally and semantically oriented filtered inputs. Although no fine-tuning is performed in the summarization step, the resulting summaries still function as structured guides that direct attention to segments considered most relevant. This configuration allows the pipeline to maintain a clear separation between summarization and captioning, while also enabling controlled experimentation with different summarization philosophies.

5 RESULTS

This Chapter will present the results achieved by each proposed strategy. Various tasks were conducted to demonstrate the viability of this work. This strategy aims to evaluate the impact of data preprocessing on the descriptions created for video captioning models. Therefore, this caption aims to present the different datasets, the implementation details adopted for each task, and a state-of-the-art comparison, covering the details related to video captioning, static video summarization, dynamic video summarization, and video skimming.

5.1 Baselines

The performance of the Adaptive Transformer was compared with the following methods representing the state-of-the-art: VTransformer (Vanilla transformer) (ZHOU et al., 2018), Transformer-XL (DAI et al., 2019), Transformer-XLRG (DAI et al., 2019), AdvInf (PARK et al., 2019), GVD (ZHOU et al., 2019), GVDsup (ZHOU et al., 2019), MFT (XIONG; DAI; LIN, 2018), HSE (ZHANG; HU; SHA, 2018), MART (LEI et al., 2020), and EMT (CARDOSO; GUIMARÃES; PATROCÍNIO JR, 2021).

The works used for comparison are divided according to the technique used. Thus, MFT (XIONG; DAI; LIN, 2018) and HSE (ZHANG; HU; SHA, 2018) are based on LSTM to recurrently evaluate the generated sentences to produce new words. The works GVD (ZHOU et al., 2019), GVDsup (ZHOU et al., 2019), and AdvInf (PARK et al., 2019), in addition to LSTM, also use detection features in an attempt to increase the quality of the obtained scores. Finally, the remaining works, i.e., VTransformer (ZHOU et al., 2018), Transformer-XL (DAI et al., 2019), Transformer-XLRG (DAI et al., 2019), MART (LEI et al., 2020), and EMT (CARDOSO; GUIMARÃES; PATROCÍNIO JR, 2021), use transformers as a technique to reduce recurrence, increasing the quality of descriptions, optimizing performance, and augmenting consistency by combining all the information from elements that represent the context.

5.2 Evaluation Metrics

Assessing frame quality in the context of video summarization poses a distinct challenge because of the many ways in which frames can be constructed while conveying similar meanings. These variations can arise from using different analyses of resources

from different informational aspects. Although humans have an intuitive understanding of this process, abstract evaluation remains an open question without a specific framework. As a result, the conventional practice involves adapting similar metrics that have been stretched to accommodate the specific requirements of the video summary task. By repurposing and customizing these metrics, researchers and practitioners can assess the effectiveness and fidelity of summaries generated in video summarization, despite the inherent complexities and subjectivity involved in sentence evaluation (AVILA et al., 2011; BELO et al., 2016).

To compute the improvement of the frame selection, in this work, the obtained results are evaluated following the same approach used by the authors of (AVILA et al., 2011; BELO et al., 2016). They reported their results using metrics widely disseminated in the literature, such as CUSa, CUSe (AVILA et al., 2011; BELO et al., 2016), and COV (BELO et al., 2016; CARDOSO et al., 2023), defined by the Equations 5.1–5.3, respectively, to evaluate the similarity between the frames generated by their summarization method and the GT results.

$$\text{CUSa} = \frac{m_A}{n_U} \tag{5.1}$$

$$\text{CUSe} = \frac{\overline{m}_A}{n_U} \tag{5.2}$$

in which m_A denotes the number of matching keyframes generated from the Automatic Summary (AS), $\overline{m}_A$ represents non-matching keyframes from AS, and n_U is the number of keyframes selected for the user to represent the user summary (U) to each video.

$$\text{COV} = \frac{\sum_{U \in US} |M(AS, U)|}{\sum_{U \in US} |U|} \tag{5.3}$$

in which $M(X, Y)$ and $|.|$ are the maximum matching between two sets of different elements X and Y, and the cardinality of a set, respectively.

While those two first metrics provide valuable insights, they often fail to measure the diversity displayed in user summaries as COV does. Furthermore, the calculation of averages for each user's measurements can introduce distortions and inaccuracies. Specifically, the CUSa, which is commonly employed to assess user opinions, fails to effectively capture the diversity of these opinions. To illustrate, consider two users, A and B, providing summaries for the same video. Let the summary of user A be $U_A = \{X, Y\}$ while the summary of user B is $U_B = \{M, N, O, P, Q, R, S, T, U, V\}$, in which each character denotes a single frame of video. Now suppose that three distinct methods generate summaries: $AS_1 = \{X, Y\}$, $AS_2 = \{M, N, O, P, Q, R, S, T, U, V\}$, and $AS_3 = \{X, M, N, O, P, Q\}$. Despite these summaries being completely different, they provide the same accuracy rate (i.e., CUSa = 0.5). This highlights the limitations of CUSa in accurately assessing diver-

gence of opinion and the need for more comprehensive assessment metrics (AVILA et al., 2011; BELO et al., 2016; CARDOSO et al., 2023).

Unlike CUSa, COV assesses the extent to which an automatic summary covers all user-generated summaries. This measure takes into account both the diversity of opinions expressed by users and the degree of agreement among them. Specifically, the CUSa measure calculates the average ratio between each user's summary and an automatic summary, thus capturing the level of agreement between the two. In contrast, COV assesses the proportion of an automatic summary that aligns with all user summaries, providing a measure of overall covering. In this work, COV is used as the first metric to compute the effectiveness of the static summarization. The reader should refer to (AVILA et al., 2011; BELO et al., 2016; CARDOSO et al., 2023) for more information about those summarization metrics.

To evaluate the performance of the dynamic summarization, the F-score is employed as the primary metric for assessing the overlap between predicted summaries and human-generated ground truth. Following the frame clustering evaluation strategy proposed by (APOSTOLIDIS et al., 2021b), the The model-generated summaries are matched against user annotations to calculate the degree of correspondence. In the SumMe dataset, the highest F-score obtained for each video is considered representative, and the final result is computed as the mean across all videos. For the TVSum dataset, importance scores predicted by the model are converted into key fragments according to the protocol defined along with the dataset, and the average F-score across all videos is used. This metric is crucial because it reflects the ability of the model to condense video content into concise and informative segments while preserving essential information.

To ensure a fair and comprehensive evaluation, a 5-random protocol is followed with an 80/20 split between training and testing videos. The final scores reported correspond to the average performance over the five splits.

Beyond segment-level agreement, rank-based correlation metrics are also used to assess the alignment between predicted importance scores and human judgments. Specifically, Kendall's τ and Spearman's ρ are computed on the TVSum dataset to evaluate how well the model preserves the relative ordering of frame importance. Kendall's τ measures the consistency of pairwise rankings, offering a robust indication of whether the model identifies similar priority relationships as those indicated by human annotators. Spearman's ρ, on the other hand, captures monotonic relationships between the predicted and reference rankings, providing insight into the global alignment of scores. These correlation metrics are particularly relevant in scenarios where fine-grained ranking consistency is more informative than strict segment overlap, making them an essential complement to F-score in evaluating subjective tasks like video summarization.

In video captioning, the evaluation of sentences is a separate challenge, as there are several ways to write sentences with the same meaning, whether using synonyms or emphasizing distinct information. This process is intuitive for humans. However, there is no specific approach for evaluating the video captioning task. So, what is usually done is the adaptation of machine translation metrics that are extended for this task (AAFAQ et al., 2019; VEDANTAM; ZITNICK; PARIKH, 2015; PAPINENI et al., 2002).

To compute the improvement of the descriptions, the evaluation of the obtained results follows the same approach used by the authors of (LEI et al., 2020; XIONG; DAI; LIN, 2018; PARK et al., 2019). They reported their results using metrics widely disseminated in the literature, such as BLEU-4 (B@4) (PAPINENI et al., 2002) and CIDEr-D (VEDANTAM; ZITNICK; PARIKH, 2015), to evaluate the similarity between the descriptions generated by their models and the GT results. However, these metrics cannot penalize the repetition that may happen, so it is necessary the use other metrics for evaluating how diverse the description is. Thus, the Repetition-4 score (R@4) (LEI et al., 2020; XIONG; DAI; LIN, 2018; PARK et al., 2019) was applied, and its objective is to emphasize the reduction of repetition of words in the description. Both R@4 and B@4 scores use 4-grams to increase word grouping.

5.3 Dataset and Implementation Details

5.3.1 Video Summarization

Similar to (FURINI et al., 2007; BELO et al., 2016), HieTaSumm was applied to the same video collections from the OpenVideo dataset (referred to as the VSUMM dataset in (TIWARI; BHATNAGAR, 2021)). This dataset contains 50 videos of different genres. All videos are in MPEG-1 format (30 fps, 352 $times$240 pixels). The genres are distributed into documentary, educational, ephemeral, historical, and lecture. The time duration of each video varies from 01 to 04 minutes. The process of creating of user summary consists of the collaboration of 50 different persons. Each user is dealing with the task of choosing the keyframes for 5 videos. Thus, 250 were created for the dataset, each video has 05 different user summaries generated manually. And, as a way to pre-process the video dataset is extracted 04 fps from all videos.

For the creation of the frame similarity graph, is use ResNet50 and VGG16 (both pre-trained on ImageNet) to extract frame descriptors. The cosine similarity was used to assess the similarity between two frame descriptors. Anit d, is also set $\delta_t = 32$ (i.e., 08 seconds with 04 fps) and $\gamma = 0.05$, during the experiments. The parameter δ_t plays a crucial role in enhancing the temporal threshold and restricting vertex connections to avoid the creation of edges that span across all frames of the video. This strategy is used

since, if all frames were connected, temporal dependencies may be neglected. Similarly, the parameter γ is employed to regulate the variance amplification in feature differences. Its utilization helps control the level of distinction among features, ensuring a balanced representation of the underlying data.

5.3.2 Video Skimming

The SumMe dataset (GYGLI et al., 2014) is used for evaluating the proposed method. All videos in the SumMe dataset are in 640 × 480 MP4 format at a frame rate of 30 fps. The dataset encompasses several genres, including sports events, travel logs, social activities, and various everyday scenarios, distributed along three categories: 4 egocentric videos, 17 moving videos, and 4 static videos. The duration of each video ranges from 1 to 6 minutes, providing a diverse spectrum of content lengths for comprehensive evaluation. The user-generated summaries were created through the collaboration of 25 distinct individuals. Each participant was tasked with selecting frames for all 25 videos, resulting in multiple summaries per video (on a,verage 15 user-generated summaries per video).

For constructing the frame similarity graph, frame descriptors were extracted using deep features obtained with VGG16 and ResNet50 models, and cosine similarity was employed to assess the similarity between those descriptors to compute w. During preprocessing, frames were extracted at a rate of 2 fps from every video. Experiments were conducted with parameter δ_t set to 2, 4, 8, or 16, γ varying between 25%, 50%, 75%, and 100%, p set to 15%, and $\mathcal{NC}_{min}$ set to 3.

To obtain the MST, the HieTaSkim utilizes the Kruskal algorithm (KRUSKAL, 1956), while *watershed* techniques (COUSTY; NAJMAN, 2011) are employed to generate the hierarchy. During experiments, distinct attribute-based watershed hierarchies were generated and evaluated based on (NAJMAN; COUSTY; PERRET, 2013) using 04 attributes: area, dynamics, volume, and number of parents.

5.3.3 Video Captioning

In the video captioning, are used the ActivityNet Captions (ANC) dataset (KRISHNA et al., 2017; HEILBRON et al., 2015). This dataset contains 10,009 videos for training and 4,917 videos for validation. Videos used during the training step have a single reference paragraph, while validation videos have two reference paragraphs. In (PARK et al., 2019), the authors used the same configuration proposed by (KRISHNA et al., 2017), however, with different divisions, in which both validation and testing were conducted with the same set of videos. Here, is follow (ZHOU et al., 2019), in which authors

proposed a new way of subdividing this dataset to optimize the use of videos and avoid overfitting. They kept the training videos and divided the validation videos into two subsets, namely: AE-VAL with 2,460 videos for validation and AE-TEST with 2,457 videos for testing. And, the ANC dataset comes with annotated segments (for each temporal event) with human-written natural language sentences that represent, on average, there are 3.65 segments per video.

The initial preprocessing follows with minor adjustments to the same procedure described in (LEI et al., 2020; CARDOSO; GUIMARÃES; PATROCÍNIO JR, 2021). The used vocabulary was created based on phrases that occurs in at least 5 instances for the ANC dataset. The resulting vocabulary carries 3,544 words. The unit memory size was defined as 2, and, the memory dimension was set to $1,200$. Videos are represented by extracting 4 FPS. Those frames are used as input for the hierarchical summarization approach (described with CNN-based features extracted with a pre-trained ResNet50 and using cosine similarity) with $\delta_t = 8$ and using the watershed hierarchy by area as described in (NAJMAN; COUSTY; PERRET, 2013; COUSTY; NAJMAN, 2011) to generate a summary of size $k = 10$.

Aligned features for appearance and optical flow were extracted for each frame belonging to the video summary. Specifically, for appearance, 2048-D feature vectors from the "FLATTEN-673" layer in ResNet-200 (HE et al., 2016) are used, while, for the optical flow, 1024-D feature vectors from the "GLOBAL POOL" layer of BNInception (IOFFE; SZEGEDY, 2015) are adopted. Both networks are pre-trained on the ANC dataset for action recognition and they are supplied by (XIONG et al., 2016).

In all experiments for the video captioning task, models were trained for 20 epochs with a learning rate of 10^{-4}. After the training process, the best model was selected, considering the CIDEr-D score since it is considered the ideal assessment metric for content description tasks (AAFAQ et al., 2019; VEDANTAM; ZITNICK; PARIKH, 2015). BLEU-4 score (B@4) is reported which is a common metric for NLP tasks such as machine translation, and the Repetition-4 score (R@4), which measures redundancy. However, the R@4 score should not be used alone because it can prevent a correct assessment since a random text without any word repetition would present an R@4 score close to 0. Therefore, the best way to assess the model's improvement is through the joint evaluation of two or more metrics. In this way, the assessment of the relationship between the GT result and the final description was made using the CIDEr-D score, while the R@4 score was evaluated afterward to point out the diversity achieved (through the reduction of repetition).

To use the Adaptive Transformer, initially, video and text are encoded and normalized separately. Encoded video and text embeddings are denoted as $H^0_{video} \in \mathbb{R}^{T_{video} \times d}$ and

Table 3 – Performance of the proposed method for different levels of precision in evaluation of video summaries. CUSa, CUSe, and COV values were multiplied by 10^2 to improve readability.

Metrics	Precision of Matches (%)												
$(\times 10^2)$	100	99	98	95	90	85	80	75	70	65	60	55	50
ResNet50 + CH													
COV	23.09	35.48	41.81	56.55	68.87	77.87	84.53	87.73	89.55	89.90	90.16	90.33	90.42
CUSa	23.27	35.91	42.42	57.17	69.64	78.90	85.75	88.89	90.68	91.01	91.26	91.46	91.54
CUSe	76.73	64.10	57.58	42.83	30.36	21.11	14.25	11.11	09.32	08.99	08.74	8.54	8.46
VGG16 + CH													
COV	22.85	35.38	42.20	56.74	67.86	77.58	85.65	88.35	89.21	90.08	90.27	90.33	90.42
CUSa	23.01	35.59	42.64	56.34	68.54	78.52	86.85	89.48	90.34	91.22	91.40	91.46	91.54
CUSe	76.99	64.41	57.36	43.66	31.46	21.47	13.15	10.52	09.66	08.78	08.60	08.54	08.46
ResNet50 + ResNet50													
COV	08.29	15.66	20.32	30.67	41.93	49.28	53.66	57.64	60.39	63.16	65.24	67.94	70.60
CUSa	08.45	15.90	20.58	31.12	42.20	49.68	54.17	58.22	60.91	63.70	65.82	68.50	71.30
CUSe	91.55	84.10	79.42	68.88	57.80	50.32	45.83	41.77	39.09	36.30	34.18	31.50	28.70
VGG16 + VGG16													
COV	01.43	12.66	20.90	35.21	49.35	57.92	63.57	69.63	74.92	76.96	78.78	81.24	82.95
CUSa	01.60	12.66	20.93	35.43	50.00	58.32	64.02	70.16	75.38	77.54	79.36	81.89	83.55
CUSe	98.40	87.34	79.07	64.57	50.00	41.68	35.98	29.84	24.62	22.46	20.64	18.11	16.45

Source: Elaborated by the author

$H^0_{text} \in \mathbb{R}^{T_{text} \times d}$, respectively, in which T_{video} and T_{text} represent video and text lengths, while d is the embedding size. They are used after a concatenation and passed to the transformer as input $H^0 \in \mathbb{R}^{T_c \times d}$, i.e., $H^0 = Concat(H^0_{video}; H^0_{text})$, in which $T_c = T_{video} + T_{text}$, following the proposal of (SUN et al., 2019; CHEN et al., 2019).

5.4 Comparison to the State-of-the-Art Methods

5.4.1 Video Summarization

Table 3 presents the HieTaSumm results. The ResNet50 and VGG16 were used to extract frame descriptors for the construction of the frame similarity graph. During the evaluation of the results, ResNet50 and VGG16 were used to extract frame descriptors, but the cosine similarity was used to verify the agreement between the groundtruth and automatic summaries. Color histograms (CH) were used during the assessment of the results.

Table 3 presents the average values of all metrics for the 50 videos belonging to the dataset. The results are presented for different levels of precision (between groundtruth and automatic summaries). It is possible to notice that the use of ResNet50 presents a slight improvement compared to the results with VGG16 (under a greater precision in evaluation), and the VGG16 presented better results (under a lower precision in evaluation). Moreover, it is also possible to observe the high values of COV and CUSa achieved by HieTaSumm method, and even under a higher precision in evaluation, the HieTaSumm method still presents competitive results.

5.4.2 Video Skimming with Unsupervised Approach

Table 5 presents the average F-score results obtained by the HieTaSkim compared to other unsupervised state-of-the-art methods in the SumMe dataset. Considering that HieTaSkim's graph-based approach incorporates temporal ordering, similar points separated by time are considered distinct. This constraint over temporal connections in the frame similarity graph allows the preservation of distinct similar sequences that are far away from one another. Consequently, unlike other techniques in Table 5, the HieTaSkim produces more discriminative summaries. Furthermore, in the works of (JADON; JASIM, 2020) and (KUMARI; DASH; SAHU, 2023), the generation of video skim is based on keyframes, which implies the neglect of information. This occurs because shots of different sizes are represented by just one frame. On the other hand, although (NAIR; MOHAN, 2023) produces the video skim based on shots, it joins similar frames into a single shot, disregarding different perspectives.

Table 4 – Results of the proposed method for each video in the SumMe dataset for unsupervised clustering techniques, in which, E represents the egocentric videos, M stands for the moving videos, and S represents the static videos. For better visualization.

Cat.	Video Name	Human (Avg.)	Uniform Sampling	SIFT	VSUMM (K-means)	VSUMM (Gaussian)	CNN (K-means)	CNN (Gaussian)	Loop	HieTaSkim
E	Base Jumping	25.7	08.5	23.4	08.3	09.4	23.9	24.7	19.0	36.4
	Bike Polo	32.2	07.1	19.6	07.8	06.5	20.4	21.2	19.5	37.5
	Scuba	21.7	01.5	14.4	14.6	17.2	19.5	18.4	33.7	35.6
	Valparaiso Downhill	21.7	19.9	19.0	20.2	19.7	20.7	21.1	29.8	47.9
M	Bearpark Climbing	21.7	16.0	14.6	15.7	14.2	19.6	20.4	28.3	20.1
	Bus in Rock Tunnel	21.7	03.0	17.7	02.9	03.3	12.4	11.9	14.4	36.2
	Car Railcrossing	21.7	36.4	36.0	38.6	39.6	19.7	17.4	31.7	51.7
	Cockpit Landing	21.7	08.9	03.5	90.6	85.6	96.5	98.4	21.7	45.1
	Cooking	21.7	02.4	19.2	02.3	02.6	20.5	19.7	29.0	33.4
	Eiffel Tower	31.2	11.9	00.4	12.3	13.5	15.7	14.6	25.6	28.5
	Excavators River Crossing	30.3	32.8	32.0	32.7	34.5	34.2	35.7	23.1	38.3
	Jumps	48.3	17.6	16.0	17.5	18.5	18.2	17.6	48.9	48.3
	Kids Playing in Leaves	28.9	42.7	36.6	42.4	48.2	37.2	38.4	26.4	71.6
	Notre Dame	23.1	22.9	23.0	22.4	02.1	02.3	02.3	32.9	34.2
	Paluma Jump	50.9	04.9	09.2	04.7	04.8	04.9	04.9	27.4	40.1
	playing Ball	27.1	24.0	22.2	25.8	23.7	25.6	25.8	14.7	55.5
	Playing on Water slide	19.5	16.9	23.2	17.4	18.5	27.8	29.7	15.6	38.8
	Saving Dolphins	18.8	21.2	12.1	22.9	25.7	24.7	21.7	22.8	57.9
	St Maarten Landing	49.6	04.0	12.0	03.9	02.54	05.9	06.8	42.8	24.7
	Statue of Liberty	18.4	06.9	20.8	07.1	07.2	09.5	09.7	15.5	47.7
	Uncut Evening Flight	35.0	25.3	25.6	25.2	27.4	27.8	29.5	29.3	31.6
S	Air Force One	33.2	06.7	07.0	06.5	06.1	06.5	04.8	32.1	36.0
	Car Over Camera	34.6	03.6	04.0	03.8	03.5	04.6	04.8	10.3	18.6
	Fire Domino	39.4	00.3	24.7	00.3	00.2	00.4	00.4	13.1	40.3
	Paintball	39.9	22.4	23.0	23.3	24.5	29.7	30.4	36.6	42.1
	Average F-Score	31.1	01.5	17.1	15.5	18.7	17.7	21.2	25.8	39.9

Source: Elaborated by the author

Table 5 presents the average F-score results obtained by the HieTaSkim compared to other unsupervised state-of-the-art methods in the SumMe dataset. Considering that HieTaSkim's graph-based approach incorporates temporal ordering, similar points separated by time are considered distinct. This constraint over temporal connections in the frame similarity graph allows the preservation of distinct similar sequences that are far away from one another. Consequently, unlike other techniques in Table 5, the HieTaSkim produces more discriminative summaries. Furthermore, in the works of (JADON; JASIM, 2020) and (KUMARI; DASH; SAHU, 2023), the generation of video skim is based on keyframes, which implies the neglect of information. This occurs because shots of different sizes are represented by just one frame. On the other hand, although (NAIR; MOHAN, 2023) produces the video skim based on shots, it joins similar frames into a single shot, disregarding different perspectives.

Figure 25 shows detailed F-score results for distinct attribute-based watershed hierarchies with varying values of δ_t and deep features extracted by different backbones. The attributes used were area, dynamics, volume, and number of parents. For ResNet50 deep features, one can observe that both watershed by area and dynamics with $\delta_t = 8$ and $\gamma = 75\%$ yield a slight performance improvement compared to all other variations using ResNet50 deep features or VGG16 deep features. Conversely, using VGG16 deep features exhibits better performance for lower parameter values. The most significant difference is that for VGG16 deep features, the best results were related to a lower frame number, achieving 39.5 of F-score. While using ResNet50 deep features, the HieTaSkim needs more information to reach 39.9 of the F-score. In both scenarios, watershed by area was used as the hierarchical strategy.

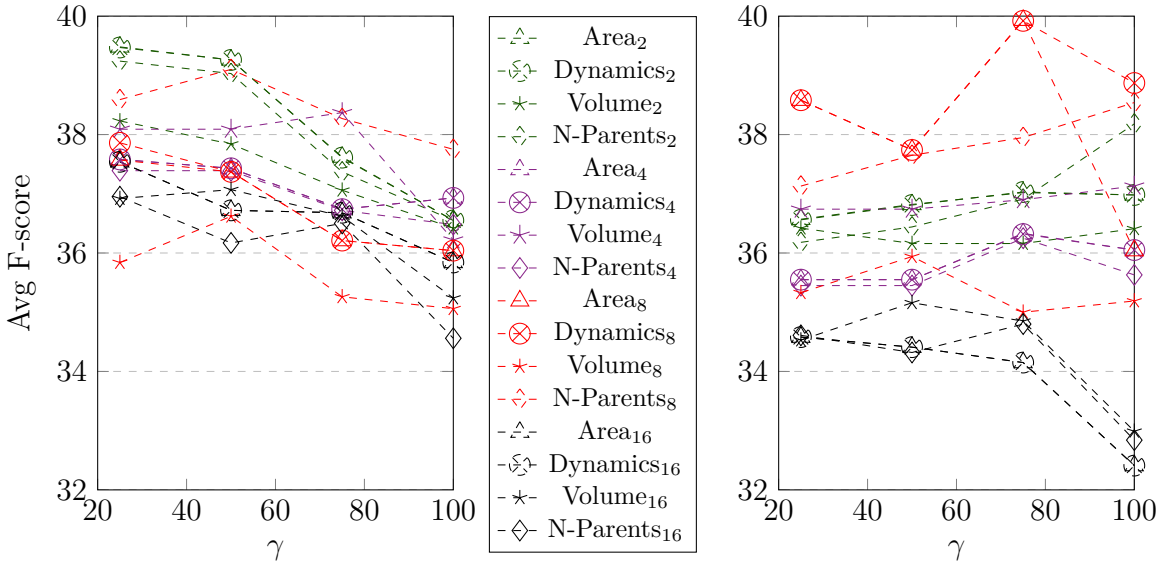
Table 6 presents the results by videos for CUSa, CUSe, COV, and F-score. These measures are used to verify the closeness of each video skim to all annotators' votes. As one should expect, the values of CUSe are high since annotations in the SumMe dataset

Table 5 – Average F-score results for unsupervised methods in the SumMe Dataset.

Method	Avg F-Score
VSUMM(K-means) (JADON; JASIM, 2020)	15.5
SIFT (JADON; JASIM, 2020)	17.1
CNN (K-means) (JADON; JASIM, 2020)	17.7
VSUMM(Gaussian) (JADON; JASIM, 2020)	18.7
CNN (Gaussian) (JADON; JASIM, 2020)	21.2
LOOP (KUMARI; DASH; SAHU, 2023)	25.8
VSMCNN (NAIR; MOHAN, 2023)	36.0
HieTaSkim	**39.9**

Source: Elaborated by the author

Figure 25 – Detailed F-score results for distinct attribute-based watershed hierarchies with deep features extracted by VGG16 or Resnet50 and $\delta_t = 2$, 4, 8, and 16. Legend labels represent the used attribute and δ_t value, e.g., Area_2 stands for watershed hierarchy based on area attribute with $\delta_t = 2$.



(a) VGG16 deep features

(b) Resnet50 deep features

Source: Elaborated by the author

are more suitable for static video summary evaluation. The video 'Jumps', of the moving category, achieved a CUSa and COV of 84.4 and 64.1, respectively, indicating a strong consensus between the selected frames and the annotators. However, the CUSe of 239.8 is attributed to the greater number of frames required to generate a dynamic summary, much exceeding the number of frames selected by the annotators.

Table 7 presents the results of state-of-the-art methods with different approaches to video skimming in the SumMe dataset. Nonetheless, a direct comparison of the HieTaSkim with others seems not to be entirely adequate, because those other methods use data augmentation (and synthetic data), large-time training on the same data, and specialized reward functions. However, the HieTaSkim outperforms similar approaches as shown in Table 5.

5.4.3 Video Skimming with Supervised Approach

In this subsection are presented the results for two different approaches for generating the dynamic summary of each video.

Table 6 – Detailed results of the proposed method for each video in the SumMe dataset. In the first column, E stands for egocentric videos, M represents moving videos, and S is for static ones. For better visualization, all scores were multiplied by 100.

Category	Video Name	CUSa	CUSe	COV	F-Score
E	Base Jumping	4.1	196.7	37.8	36.4
	Bike Polo	33.5	222.4	36.0	37.5
	Scuba	33.7	298.6	43.3	35.6
	Valparaiso Downhill	31.7	219.4	32.0	47.9
M	Bearpark Climbing	9.1	175.7	11.3	20.1
	Bus in Rock Tunnel	33.0	313.2	35.8	36.2
	Car Railcrossing	53.0	264.5	49.6	51.7
	Cockpit Landing	16.8	128.4	19.0	45.1
	Cooking	10.9	232.3	23.4	33.5
	Eiffel Tower	9.2	194.5	20.3	28.5
	Excavators River Crossing	38.5	138.3	33.2	38.3
	Jumps	84.4	239.8	64.1	48.3
	Kids Playing in Leaves	51.2	131.0	26.7	71.7
	Notre Dame	24.9	220.8	30.3	34.2
	Paluma Jump	23.4	234.9	32.7	40.0
	Playing Ball	30.5	157.4	31.6	55.5
	Playing on Water slide	51.9	260.6	45.2	38.8
	Saving Dolphins	38.2	142.3	28.4	57.9
	St Maarten Landing	23.5	284.6	30.4	34.7
	Statue of Liberty	28.3	209.8	21.9	47.7
	Uncut Evening Flight	16.6	110.3	17.6	31.6
S	Air Force One	27.7	158.4	24.2	36.0
	Car Over Camera	18.4	434.5	34.1	18.6
	Fire Domino	42.9	194.4	24.8	40.3
	Paintball	62.4	356.9	64.3	42.0
	Average	33.4	220.7	32.7	39.9

Source: Elaborated by the author

5.4.3.1 Streamlined Video Skimming

Table 8 presents the results of state-of-the-art methods with different approaches for video skimming on the SumMe and TVSum datasets. However, a direct comparison of the proposed model with others seems not to be entirely adequate, because these other

Table 7 – Comparison of Top F-score results between different approaches for video skimming in the SumMe Dataset.

Method	Approach	Avg F-Score
AoA(ca+sa)	Supervised	46.0
DN-VSN	Reinforcement	52.0
SUM-GAN-AED	GAN	64.9
HieTaSkim	Unsupervised	**39.9**

Source: Elaborated by the author

methods use data augmentation (and synthetic data). Considering that the StreamExLSTM incorporates temporal ordering by recurrent evaluation of frames, the trained model uses updated memory to distinguish possible scene modifications that occur at different similar points separated by time from being considered distinct. The memory update appears as a way to prevent similar frames from being disregarded that appear at different points in the video. Consequently, the StreamExLSTM produces more discriminative and concise summaries. Even so, StreamExLSTM outperforms the state-of-the-art unsupervised approach on the SumMe dataset and presents results very close to the reinforcement and GAN-based approaches on the TVSum dataset.

To further evaluate the effectiveness of the proposed model, a series of experiments were conducted to investigate its ability to generalize across different dataset configurations. For this purpose, StreamExLSTM was trained with one dataset and tested on the other. Additionally, training was performed by combining both datasets, with each dataset used separately for testing.

Table 9 presents the variation in results as a function of changes in training and testing configurations. In particular, training in the SumMe dataset while testing in TVSum led to an increase in the F-score obtained (61.11% $\rightarrow$ 66.96%). In contrast, when training on TVSum and evaluating on SumMe, a decrease in performance was observed (48.80% $\rightarrow$ 20.87%), indicating a disparity in generalization.

Furthermore, unlike experiments in which training and testing were conducted on the same dataset, resulting in high variability between random splits, training on SumMe and testing on TVSum yielded superior results compared to training and testing

Table 8 – Results of the state-of-the-art approaches for video skimming on the SumMe and TVSum datasets. In the approach type, S, U, SS, G, and RL stand for Supervised, Unsupervised, Semi-supervised, Generative Adversarial Network (GAN), and Reinforcement Learning.

Method	Approach	Avg F-Score ↑		#Params (M) ↓	Test Method
		SumMe	TVSum		
HieTaSkim (CARDOSO et al., 2024)	U	39.9	–	–	5 Random
SF-CVS (HUANG; WANG, 2019)	S	46.0	58.0	–	5 Random
CA + SA (PUTHIGE et al., 2023)	S	46.0	57.3	–	5 Random
SUM-FCN (ROCHAN; YE; WANG, 2018)	S	47.5	56.8	36.58	M Random
M-AVS (JI et al., 2019)	S	44.4	61.0	4.40	5 Random
RSGN (ZHAO et al., 2021)	S	45.0	60.1	–	5 Random
DHAVS (LIN; ZHONG; FARES, 2022)	S	45.6	60.8	–	5 Random
LMHA (ZHU et al., 2022)	SS	51.1	61.0	–	5 Random
HMT (ZHAO; GONG; LI, 2022)	SS	44.1	60.1	–	5 Random
VJMHT (LI et al., 2022)	SS	50.6	60.9	–	5 FCV
VSS-Net (ZHANG et al., 2023)	SS	51.5	61.0	23.12	5 FCV
DN-VSN (ZANG et al., 2023)	RL	52.0	62.8	–	5 Random
CAAN (LIANG et al., 2022)	G	50.6	59.3	–	5 FCV
SUM-GAN-AED (MINAIDI; PAPAIOANNOU; POTAMIANOS, 2023)	G	64.9	63.1	–	5 Random
StreamExLSTM	S	48.8	61.1	14.5	5 Random

Source: Elaborated by the author

Table 9 – Results of StreamExLSTM for different training configurations.

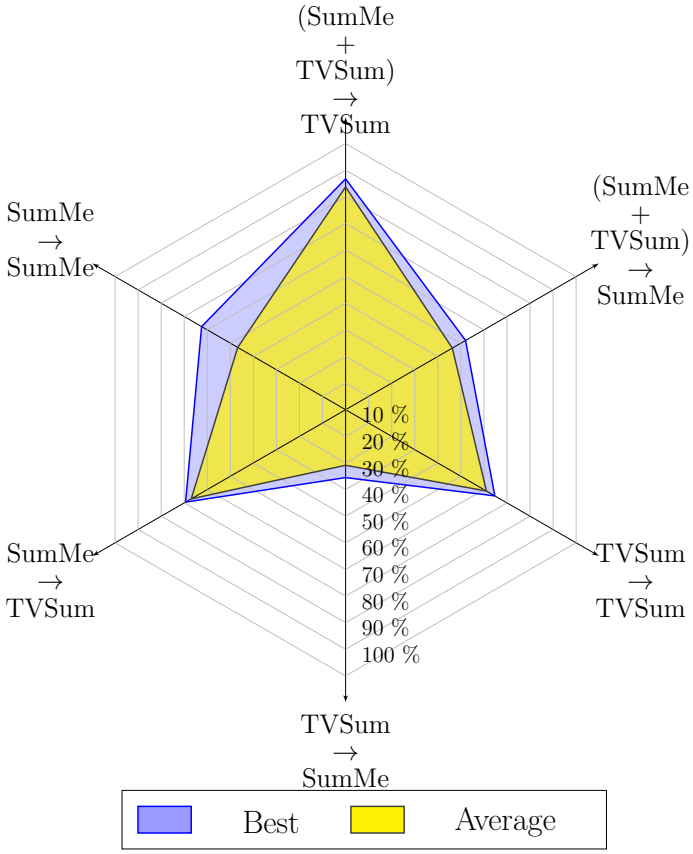
Training	Testing	Avg F-Score (%) ↑	Best F-Score (%) ↑
SumMe	SumMe	48.80	62.46
SumMe	TVSum	66.96	69.42
TVSum	SumMe	20.87	25.50
TVSum	TVSum	61.11	64.76
SumMe + TVSum	SumMe	46.29	52.03
SumMe + TVSum	TVSum	83.68	86.74

Source: Elaborated by the author

exclusively on SumMe. This was evidenced by a closer alignment between the maximum and average values of the F-score, suggesting a reduction in standard deviation between splits.

Similarly, Figure 26 highlights an increase in both average and maximum values of F-score when using a combined training set, with variations in the test set. The SumMe + TVSum combination achieved an average F-score of 83.68%, reflecting enhanced coverage of keyframes in the TVSum dataset. However, as SumMe's characteristics were overshadowed by those of TVSum, a discrepancy emerged between the selected regions

Figure 26 – Best and average results of StreamExLSTM for different training configurations.



Source: Elaborated by the author

Table 10 – Comparison with the state-of-the-art methods under rank-based evaluation. Results are reported on the TVSum dataset using the canonical experimental setting.

Method	Kendall's τ (↑)	Spearman's ρ (↑)
Random (OTANI et al., 2019)	0.000	0.000
Human (OTANI et al., 2019)	0.177	0.204
CAAN (LIANG et al., 2022)	0.038	0.050
DHAVS (LIN; ZHONG; FARES, 2022)	0.082	0.089
RSGN (ZHAO et al., 2021)	0.083	0.090
HMT (ZHAO; GONG; LI, 2022)	0.096	0.107
VJMHT (LI et al., 2022)	0.097	0.105
SSPVS (LI et al., 2023)	0.177	0.233
SSPVS+Text (LI et al., 2023)	0.181	0.238
VSS-Net (ZHANG et al., 2023)	0.190	0.249
StreamExLSTM	0.173	0.232

Source: Elaborated by the author

and the ground-truth annotations of SumMe, underscoring dataset-specific biases in summarization performance.

Table 10 presents the experimental results based on Kendall's τ and Spearman's ρ correlation coefficients. As proposed by (OTANI et al., 2019), these rank-based metrics evaluate the consistency between the predicted importance scores and the frame-level annotations provided by human annotators. StreamExLSTM achieves correlation scores that are competitive with VSS-Net (ZHANG et al., 2023), a semi-supervised approach, and SSPVS (LI et al., 2023), which utilizes multimodal cues to enhance summary context. In particular, the proposed method outperforms human-generated summaries and several recent models reported in the literature. These results emphasize the effectiveness of StreamExLSTM's dual-level (local and global) temporal modeling, which improves the semantic alignment of the generated summaries with human judgment. StreamExLSTM captures high-value segments more accurately, reflecting the underlying temporal structure and improving summary relevance to the user.

To evaluate the individual contributions of the ssLSTM and smLSTM modules within the StreamExLSTM architecture, an ablation study is performed, isolating the effects of each component. When tested independently, both modules demonstrated significant performance contributions. However, the most significant gains were observed when both modules were combined. The fusion of local and global temporal modeling led to higher F-Scores for both datasets, validating the complementary nature of the two modules.

While this dual-module configuration naturally increased the total number of trainable parameters, empirical results showed that it did not introduce significant variations in training or inference time, as shown in Table 11. This efficiency is attributed to the

Table 11 – Average results for the ablation study of the proposed StreamExLSTM on the Summe and TVSum datasets. ✓ indicates the presence of smLSTM or ssLSTM.

Dataset	smLSTM	ssLSTM	Average F-Score ↑	Average Time ↓		#Param ↓
				Training	Test per Video	
SumMe	✓	✓	48.80	19'42"	0'02"	14.05M
	✓		44.62	23'37"	0'03"	9.72M
		✓	41.35	21'41"	0'02"	7.62M
TVSum	✓	✓	61.11	13'41"	0'05"	14.05M
	✓		57.76	15'02"	0'05"	9.72M
		✓	59.92	11'45"	0'04"	7.62M

Source: Elaborated by the author

fusion of ssLSTM results into smLSTM, similar to a skip connection. The ablation study confirms that the integration of ssLSTM and smLSTM is not only synergistic in terms of accuracy but also computationally feasible for practical deployment in video summarization tasks.

5.4.3.2 MalSumm Video Skimming

Table 12 presents the average F-scores achieved by the proposed model and recent state-of-the-art methods. On the SumMe dataset, the proposed model achieved an F-score of 49.7, surpassing all supervised and unsupervised methods. For the TVSum dataset, the model reached 62.1, also positioning itself among the top-performing approaches, outperforming all supervised and semi-supervised approaches. These results confirm the effectiveness of the proposed strategy in generating coherent and concise summaries that reflect the annotated user preferences.

To further evaluate the alignment between the predicted frame importance and the human annotations, Kendall's τ and Spearman's ρ coefficients values are also calculated on the TVSum dataset, following the evaluation protocol proposed by (OTANI et al., 2019). Table 13 compares the rank-based correlation results obtained by the proposed model with those reported in the literature using the canonical setting. This proposed model achieved a Kendall's τ of 0.180 and Spearman's ρ of 0.242, values comparable to the upper human agreement baseline. These results demonstrate that the proposed method not only achieves high F-score scores but also preserves the ordinal relationships between frame importances as perceived by human annotators.

The ablation study investigates the impact of the proposed blocks on MALSumm applied to SumMe and TVSum datasets. On the SumMe dataset, the full model achieves the best F-score (49.70), while removing the SE block or the saLSTM block slightly reduces performance, indicating the SE block has a stronger effect than the other. Excluding both

Table 12 – Results of the state-of-the-art methods for video skimming on the SumMe and TVSum datasets. Here, S, U, SS, G, and RL stand for Supervised, Unsupervised, Semi-supervised, Generative Adversarial Network (GAN), and Reinforcement Learning.

Method	Approach	Avg F-Score ↑		#Params (M) ↓	Test Method
		SumMe	TVSum		
HieTaSkim (CARDOSO et al., 2024)	U	39.9	–	–	5 Random
SF-CVS (HUANG; WANG, 2019)	S	46.0	58.0	–	5 Random
CA + SA (PUTHIGE et al., 2023)	S	46.0	57.3	–	5 Random
SUM-FCN (ROCHAN; YE; WANG, 2018)	S	47.5	56.8	36.58	M Random
M-AVS (JI et al., 2019)	S	44.4	61.0	4.40	5 Random
RSGN (ZHAO et al., 2021)	S	45.0	60.1	–	5 Random
DHAVS (LIN; ZHONG; FARES, 2022)	S	45.6	60.8	–	5 Random
LMHA (ZHU et al., 2022)	SS	51.1	61.0	–	5 Random
HMT (ZHAO; GONG; LI, 2022)	SS	44.1	60.1	–	5 Random
VJMHT (LI et al., 2022)	SS	50.6	60.9	–	5 FCV
VSS-Net (ZHANG et al., 2023)	SS	51.5	61.0	23.12	5 FCV
DN-VSN (ZANG et al., 2023)	RL	52.0	62.8	–	5 Random
CAAN (LIANG et al., 2022)	G	50.6	59.3	–	5 FCV
SUM-GAN-AED (MINAIDI; PAPAIOANNOU; POTAMIANOS, 2023)	G	64.9	63.1	–	5 Random
MALSumm	S	49.7	62.1	15.23	5 Random

Source: Elaborated by the author

Table 13 – Comparison with the state-of-the-art methods under rank-based evaluation. Results are reported on the TVSum dataset using the canonical experimental setting.

Method	Kendall's τ (↑)	Spearman's ρ (↑)
Random (OTANI et al., 2019)	0.000	0.000
Human (OTANI et al., 2019)	0.177	0.204
CAAN (LIANG et al., 2022)	0.038	0.050
DHAVS (LIN; ZHONG; FARES, 2022)	0.082	0.089
RSGN (ZHAO et al., 2021)	0.083	0.090
HMT (ZHAO; GONG; LI, 2022)	0.096	0.107
VJMHT (LI et al., 2022)	0.097	0.105
SSPVS (LI et al., 2023)	0.177	0.233
SSPVS+Text (LI et al., 2023)	0.181	0.238
VSS-Net (ZHANG et al., 2023)	0.190	0.249
MALSumm	0.180	0.242

Source: Elaborated by the author

blocks (SE and saLSTM) causes a sharp drop to 44.69, confirming their critical roles. For the TVSum dataset, the full model scores 62.08, with minimal loss when SE or saLSTM blocks are removed (61.17 $\sim$ 61.77); even lightweight versions without the maLSTM block maintain a score of around 60.71. Overall, when used alone, both maLSTM and saLSTM blocks show lower performance than their combination with the SE block. The combination maLSTM + SE appears to be better than saLSTM + SE and maLSTM + saLSTM, but it still falls short of the full model (maLSTM + saLSTM + SE).

5.4.4 Video Captioning with Adaptive Transformer

Tables 15 and 16 present the results found for the Adaptive Transformer model along with other *state-of-the-art* methods. Table 15 summarizes the performance of the proposed models, comparing them with *state-of-the-art* models in AE-VAL split of the ANC dataset. The results reported were evaluated mainly according to the CIDEr-D metric to choose the best model. The results shown in Table 15 represent models based on transformers, LSTM-only, and LSTM with detection features. The results achieved by the proposed model show an improvement compared to the others in relation to the CIDEr-D metric. The adoption of the adaptive attention module as a reinforcement strategy for previous attention results contributes to increasing the quality of the learned feature. With this, the results of its modified attention tend to better weight data with greater importance. It turns out that just the addition of the *adaptive attention* block increases the CIDEr-D score, but this does not guarantee the reduction of the R@4 score.

Despite the superior result for BLEU-4, achieved by the GVDsup method, to those found in Table 15, as it is not considered the best metric for the video captioning task, the results do not represent a marked improvement as found by CIDEr-D. According to (PAPINENI et al., 2002), the use of BLEU unigram compares the evaluation of the simple precision of the method, characterized by the simple count of correct words divided by the total number of words in the sentence. On the other hand, the CIDEr-D score proposes to measure the best textual sentence among the candidates by the majority of simple votes. However, the B@4 score is considered an interesting metric for some NLP tasks, such as machine translation, since, if the sentence is very close to most GT sentences used as a reference, the probability is greater that the sentence is correct. This evaluation method seeks to bring the human description closer to that described by machine translation, as human evaluation is inherent to the perception of the person describing the scene in focus

Table 14 – Average results for the ablation study of MALSumm.

	maLSTM	saLSTM	SE	Average F-Score ↑	Average Time ↓		#Param ↓
					Training	Test per Video	
SumMe	✓	✓	✓	49.70	4'46"	37"	14.71M
	✓	✓		48.61	4'31"	30"	14.58M
	✓		✓	49.60	3'51"	34"	9.85M
	✓			44.69	3'48"	32"	9.72M
		✓	✓	47.80	1'26"	25"	7.09M
		✓		45.50	1'18"	24"	6.96M
TVSum	✓	✓	✓	62.08	11'13"	47"	14.71M
	✓	✓		61.17	10'24"	45"	14.58M
	✓		✓	62.01	9'50"	45"	9.85M
	✓			61.77	9'39"	43"	9.72M
		✓	✓	60.94	2'43"	30"	7.09M
		✓		60.71	2'25"	33"	6.96M

Source: Elaborated by the author

Table 15 – Performance of the Adaptive Transformer model and other state-of-the-art methods in ae-val split of ActivityNet Captions (Det indicates whether detection features are used; while Rec indicates whether sentence-level recurrence is used).

	Det	Rec	B@4	CIDEr-D	R@4 ↓
LSTM based methods					
MFT (XIONG; DAI; LIN, 2018)	χ	$\surd$	10.27	19.12	17.71
HSE (ZHANG; HU; SHA, 2018)	χ	$\surd$	9.84	18.78	13.22
LSTM based methods with detection feature					
GVD (ZHOU et al., 2019)	$\surd$	χ	11.04	21.95	8.76
GVDsup (ZHOU et al., 2019)	$\surd$	χ	**11.30**	22.94	7.04
AdvInf (PARK et al., 2019)	$\surd$	$\surd$	10.04	20.97	5.76
Transformer based methods					
VTransformer (ZHOU et al., 2018)	χ	χ	9.75	22.16	7.79
Transformer-XL (DAI et al., 2019)	χ	$\surd$	10.39	21.67	8.54
Transformer-XLRG (DAI et al., 2019)	χ	$\surd$	10.17	20.40	8.85
MART (LEI et al., 2020)	χ	$\surd$	10.33	23.42	5.18
EMT $\boldsymbol{SE_d}$	χ	$\surd$	10.24	23.66	**4.27**
EMT $\boldsymbol{SE_{T_m}}$	χ	$\surd$	10.33	23.61	4.74
EMT $\boldsymbol{AdA_{S_t}}$	χ	$\surd$	10.49	23.58	7.55
EMT $\boldsymbol{AdA_{S_t} + SE_d}$	χ	$\surd$	10.34	23.78	6.08
EMT $\boldsymbol{AdA_{S_t} + SE_{T_m}}$	χ	$\surd$	10.37	23.80	6.23
Adaptive Transformer	χ	$\surd$	10.38	**24.22**	5.84

Source: Elaborated by the author

Table 16 – Performance of the Adaptive Transformer model and other transformer-based methods in ae-test split of ActivityNet Captions (Rec indicates whether sentence-level recurrence is used).

	Rec	B@4	CIDEr-D	R@4 ↓
VTransformer (ZHOU et al., 2018)	χ	9.31	21.33	7.45
Transformer-XL (DAI et al., 2019)	$\surd$	**10.25**	21.71	8.79
Transformer-XLRG (DAI et al., 2019)	$\surd$	10.07	20.34	9.37
MART (LEI et al., 2020)	$\surd$	9.78	22.16	5.44
EMT $\boldsymbol{SE_d}$	$\surd$	10.00	22.84	**4.55**
EMT $\boldsymbol{SE_{T_m}}$	$\surd$	9.91	22.74	4.68
EMT $\boldsymbol{AdA_{S_t}}$	$\surd$	9.97	21.07	7.48
EMT $\boldsymbol{AdA_{S_t} + SE_d}$	$\surd$	10.10	22.78	5.91
EMT $\boldsymbol{AdA_{S_t} + SE_{T_m}}$	$\surd$	**10.25**	22.17	7.18
Adaptive Transformer	$\surd$	10.00	**23.04**	5.29

Source: Elaborated by the author

(VEDANTAM; ZITNICK; PARIKH, 2015).

The results shown by Table 16 present the comparison of the proposed model and relation to the performance of models based on transformers in AE-TEST split of the ANC dataset (similar to what was done in (LEI et al., 2020; CARDOSO; GUIMARÃES;

Table 17 – Performance of the Adaptive Transformer, HSAT, and other state-of-the-art methods in ae-val split of ActivityNet Captions (Det indicates whether detection features are used; while Rec indicates whether sentence-level recurrence is used).

Method	Det	Rec	B@4 ↑	CIDEr-D ↑	R@4 ↓
LSTM based methods					
MFT (XIONG; DAI; LIN, 2018)	χ	√	10.27	19.12	17.71
HSE (ZHANG; HU; SHA, 2018)	χ	√	9.84	18.78	13.22
LSTM based methods with detection feature					
GVD (ZHOU et al., 2019)	√	χ	11.04	21.95	8.76
GVDsup (ZHOU et al., 2019)	√	χ	**11.30**	22.94	7.04
AdvInf (PARK et al., 2019)	√	√	10.04	20.97	5.76
Transformer based methods					
VTransformer (ZHOU et al., 2018)	χ	χ	9.75	22.16	7.79
Transformer-XL (DAI et al., 2019)	χ	√	10.39	21.67	8.54
Transformer-XLRG (DAI et al., 2019)	χ	√	10.17	20.40	8.85
MART (LEI et al., 2020)	χ	√	10.33	23.42	5.18
EMT (CARDOSO; GUIMARÃES; PATROCÍNIO JR, 2021)	χ	√	10.24	23.66	**4.27**
Adaptive Transformer	χ	√	10.38	**24.22**	5.84
HSAT	χ	√	10.31	23.76	5.85

Source: Elaborated by the author

PATROCÍNIO JR, 2023)). Again, the Adaptive Transformer presents better results than the others, mainly for the CIDEr-D metric.

In summation, as one can see in Table 15 and 16, the results achieved by the proposed model are superior. The changes achieved imply an increase in similarity with the GT results, but with a reduction of repetition without losing cohesion among the generated sentences for describing each video.

5.4.5 Video Captioning with HSAT

Tables 17 and 18 present the results found for HSAT along with other *state-of-the-art* methods. Table 17 summarizes the performance of the Adaptive Transformer, comparing it with *state-of-the-art* models in AE-VAL split of the ANC dataset. The results reported were evaluated mainly according to the CIDEr-D metric to choose the best model. The results shown in Table 17 represent models based on transformers, LSTM-only, and LSTM with detection features. The results achieved by the Adaptive Transformer show an improvement compared to the others with the CIDEr-D metric, except when compared to the Adaptive Transformer. The adoption of the adaptive attention module as a reinforcement strategy for previous attention results contributes to increasing the quality of the learned feature. With this, the results of its modified attention tend to better weight data with greater importance. It turns out that just the addition of the *adaptive attention* block increases the CIDEr-D score, but this does not guarantee the reduction of the R@4 score.

Table 18 – Performance of the Adaptive Transformer, HSAT, and other transformer-based methods in ae-test split of ActivityNet Captions (Rec indicates whether sentence-level recurrence is used).

Method	Rec	B@4 ↑	CIDEr-D ↑	R@4 ↓
VTransformer (ZHOU et al., 2018)	χ	9.31	21.33	7.45
Transformer-XL (DAI et al., 2019)	$\surd$	**10.25**	21.71	8.79
Transformer-XLRG (DAI et al., 2019)	$\surd$	10.07	20.34	9.37
MART (LEI et al., 2020)	$\surd$	9.78	22.16	5.44
EMT (CARDOSO; GUIMARÃES; PATROCÍNIO JR, 2021)	$\surd$	10.00	22.84	**4.55**
Adaptive Transformer	$\surd$	10.00	**23.04**	5.29
HSAT	$\surd$	9.94	22.97	5.35

Source: Elaborated by the author

Despite the superior result for BLEU-4, achieved by the GVDsup method, to those found in Table 17, as it is not considered the best metric for the video captioning task, the results do not represent a marked improvement as found by CIDEr-D. According to (PAPINENI et al., 2002), the use of BLEU unigram compares the evaluation of the simple precision of the method, characterized by the simple count of correct words divided by the total number of words in the sentence. On the other hand, the CIDEr-D score proposes to measure the best textual sentence among the candidates by the majority of simple votes. However, the B@4 score is considered an interesting metric for some NLP tasks, such as machine translation, since, if the sentence is very close to most GT sentences used as a reference, the probability is greater that the sentence is correct. This evaluation method seeks to bring the human description closer to that described by machine translation, as human evaluation is inherent to the perception of the person describing the scene in focus (VEDANTAM; ZITNICK; PARIKH, 2015).

The results shown by Table 18 present the comparison of the Adaptive Transformer and relation to the performance of models based on transformers in AE-TEST split of the ANC dataset (similar to what was done in (LEI et al., 2020; CARDOSO; GUIMARÃES; PATROCÍNIO JR, 2021)). Again, the Adaptive Transformer presents better results than the others, mainly for the CIDEr-D metric, followed closely by HSAT.

In summation, as one can see in Table 17 and 18, the results achieved by the HSAT are superior, except when compared to the Adaptive Transformer, whose results HSAT follows closely. The changes achieved imply an increase in similarity with the GT results, but with a reduction of repetition without losing cohesion among the generated sentences for describing each video.

5.4.6 Video Captioning with SkimCap

The Skimcap method is evaluated on the ActivityNet Captions (ANC) dataset (KRISHNA et al., 2017; HEILBRON et al., 2015), which contains 10,009 training videos and 4,917 validation videos. Each training video is annotated with a single reference paragraph, while validation videos include two reference paragraphs to provide richer supervision. Although (PARK et al., 2019) followed the original configuration of (KRISHNA et al., 2017), their protocol reused the same videos for both validation and testing, potentially introducing evaluation bias. In contrast, this work adopt the revised split proposed by (ZHOU et al., 2019), which mitigates overfitting by dividing the validation set into two disjoint subsets: AE-VAL, with 2,460 videos used for validation, and AE-TEST, with 2,457 videos reserved for testing. On average, each video comprises 3.65 annotated temporal segments, each described by a human-written sentence.

The preprocessing pipeline follows previous works (LEI et al., 2020; CARDOSO; GUIMARÃES; PATROCÍNIO JR, 2021, 2022, 2023), with minor modifications. The vocabulary was built by selecting words that appear in at least five training instances, resulting in 3,544 unique tokens. In this work, the memory size is set to 1,200 and uses a single memory unit per transformer layer. Video frames are sampled at 2 FPS and passed through a hierarchical skimming strategy to select representative content. This selection relies on cosine similarity between CNN-based features extracted using the feature extraction procedure proposed in (LEI et al., 2020).

To maintain temporal consistency, a stride of $\delta_t = 4$ and use four distinct hierarchies as proposed by (NAJMAN; COUSTY; PERRET, 2013; COUSTY; NAJMAN, 2011), watershed by area, dynamics, number of parents, and volume. For each hierarchy, two variations based on different frame selection policies are generated, one favoring grouped segments and another promoting sparse, widely distributed selections. For videos longer than 50 seconds, the hierarchy is set to select 100 frames that are aligned with the ground-truth annotations. For videos containing 100 frames or fewer, all frames are retained.

Video and textual inputs are independently encoded and normalized before fusion. Let $H^0_{\text{video}} \in \mathbb{R}^{T_{\text{video}} \times d}$ and $H^0_{\text{text}} \in \mathbb{R}^{T_{\text{text}} \times d}$ represent the video and text embeddings, respectively, where T_{video} and T_{text} denote sequence lengths and d the embedding dimension. The embeddings are concatenated to form the combined sequence $H^0 \in \mathbb{R}^{T_c \times d}$, where $T_c = T_{\text{video}} + T_{\text{text}}$, which is then fed into the transformer encoder following the joint modeling strategy described in (SUN et al., 2019; CHEN et al., 2019).

Table 19 presents a performance comparison between the proposed model and state-of-the-art baselines on the AE-VAL split of the ANC dataset, with a primary focus on the CIDEr-D metric, which aligns well with human judgment in content description

Table 19 – Performance of the Adaptive Transformer, HSAT, SkimCap and other state-of-the-art methods in ae-val split of ActivityNet Captions. Det indicates the use of detection features, Rec indicates the use of sentence-level recurrence.

Method	Det	Rec	B@4 ↑	CIDEr-D ↑	R@4 ↓
LSTM based methods					
MFT (XIONG; DAI; LIN, 2018)	χ	$\surd$	10.27	19.12	17.71
HSE (ZHANG; HU; SHA, 2018)	χ	$\surd$	9.84	18.78	13.22
LSTM based methods with detection feature					
GVD (ZHOU et al., 2019)	$\surd$	χ	11.04	21.95	8.76
GVDsup (ZHOU et al., 2019)	$\surd$	χ	**11.30**	22.94	7.04
AdvInf (PARK et al., 2019)	$\surd$	$\surd$	10.04	20.97	5.76
Transformer based methods					
VTransformer (ZHOU et al., 2018)	χ	χ	9.75	22.16	7.79
Transformer-XL (DAI et al., 2019)	χ	$\surd$	10.39	21.67	8.54
Transformer-XLRG (DAI et al., 2019)	χ	$\surd$	10.17	20.40	8.85
MART (LEI et al., 2020)	χ	$\surd$	10.33	23.42	5.18
EMT (CARDOSO; GUIMARÃES; PATROCÍNIO JR, 2021)	χ	$\surd$	10.24	23.66	**4.27**
Adaptive Transformer	χ	$\surd$	10.38	24.22	5.84
HSAT	χ	$\surd$	10.31	23.76	5.85
SkimCap	χ	$\surd$	10.72	**25.44**	5.84

Source: Elaborated by the author

tasks. The results cover transformer-based methods, LSTM architectures, and LSTM augmented with detection features. For all hierarchies, the results remain competitive as shown in Table 20, indicating the impact of selecting more representative features. This shows that the inclusion of preprocessing techniques, such as hierarchical skimming and adaptive attention modulation, played a decisive role in increasing caption relevance without increasing redundancy.

In addition, although the GVDsup model achieves the highest B@4 score, it does not reflect improvements in coherence or contextual diversity for video captioning, since BLEU is an NLP-centric metric. This reinforces the argument that metrics such as CIDEr-D and R@4 are more appropriate for evaluating dense video descriptions, as they better capture relevance and redundancy, respectively. Overall, the results highlight the importance of informed preprocessing in reducing overlap between event-level captions while preserving descriptive fidelity.

Table 20 – Performance of the SkimCap model in ae-val split of ActivityNet captions with different types of hierarchies and clustering.

Watershed	Clustering	B@4 ↑	CIDEr-D ↑	R@4 ↓
Area	Central	10.72	**25.44**	5.84
	Spaced	10.89	25.34	5.83
Dynamics	Central	10.77	24.56	5.39
	Spaced	**11.03**	25.03	7.37
NParents	Central	10.21	24.45	**4.35**
	Spaced	10.34	24.97	5.17
Volume	Central	10.84	25.21	5.62
	Spaced	10.84	24.91	8.24

Source: Elaborated by the author

Table 21 – Performance of the Adaptive Transformer, HSAT, and SkimCap models and other transformer-based methods in ae-test split of ActivityNet Captions. Rec indicates the use of sentence-level recurrence.

Method	Rec	B@4 ↑	CIDEr-D ↑	R@4 ↓
VTransformer (ZHOU et al., 2018)	χ	9.31	21.33	7.45
Transformer-XL (DAI et al., 2019)	$\surd$	10.25	21.71	8.79
Transformer-XLRG (DAI et al., 2019)	$\surd$	10.07	20.34	9.37
MART (LEI et al., 2020)	$\surd$	9.78	22.16	5.44
EMT (CARDOSO; GUIMARÃES; PATROCÍNIO JR, 2021)	$\surd$	10.00	22.84	**4.55**
Adaptive Transformer	$\surd$	10.00	23.04	5.29
HSAT	$\surd$	9.94	22.97	5.35
SkimCap	$\surd$	**10.47**	**23.77**	6.13

Source: Elaborated by the author

5.5 Ablation Study

To demonstrate the adaptive transformer architectures' effectiveness, the memory size is modified from 2 to 1 and 4. And, to ensure the effectiveness of the adaptive attention after the previous attention module, the removal (in an alternate way) of the adaptive attention modules from the proposed positions was also analyzed. In addition, with the modification of memory size, for each variation, the adaptive attention module was reassessed. In this way, it was possible to measure the impact achieved by the inclusion of adaptive attention as a way to modify both multi-head attention modules. Those changes provide an ablation study, allowing the identification of the best configuration of the Adaptive Transformer architecture.

The evaluation of the results obtained for the CIDEr-D score demonstrates the effectiveness of the proposed configuration at the detriment of the others. Furthermore, one can notice that the reduction of the Repetition-4 score is relatively low, but for the BLEU-4, there was a slight increase in the score; however, as discussed before, it is not a

Table 22 – Performance in the ae-val split of ActivityNet Captions during the ablation study for verifying the quality of results achieved for the proposed architectures in which (*) denotes transformer without adaptive attention after the second Multi-Head Attention, and (+) denotes transformer without adaptive attention after the first Multi-Head Attention.

	Mem Size	Hidden Size	B@4	CIDEr-D	R@4 ↓
Original versions					
EMT $\boldsymbol{SE_d}$	1	768	10.24	23.66	4.27
EMT $\boldsymbol{SE_{T_m}}$	1	768	10.33	23.61	4.74
EMT $\boldsymbol{AdA_{S_t}}$	1	768	10.49	23.58	7.55
EMT $\boldsymbol{AdA_{S_t} + SE_d}$	1	768	10.34	23.78	6.08
EMT $\boldsymbol{AdA_{S_t} + SE_{T_m}}$	1	768	10.37	23.80	6.23
Adaptive Transformer	2	1200	10.38	**24.22**	5.84
HSAT	2	1200	10.31	23.76	5.85
EMT- Modified versions					
EMT $\boldsymbol{SE_d}$	2	1200	10.30	23.14	4.80
EMT $\boldsymbol{SE_d}$	4	1200	10.40	23.29	4.58
EMT $\boldsymbol{SE_d}$	8	1200	10.31	23.74	5.31
EMT $\boldsymbol{SE_d}$	16	1200	10.14	23.53	**3.56**
EMT $\boldsymbol{AdA_{S_t}}$	2	1200	10.42	23.54	7.19
EMT $\boldsymbol{AdA_{S_t}}$	4	1200	10.30	23.58	7.55
EMT $\boldsymbol{AdA_{S_t}}$	8	1200	10.35	23.30	7.74
EMT $\boldsymbol{AdA_{S_t}}$	16	1200	10.41	23.74	8.12
Adaptive Transformer- Modified versions					
Adaptive Transformer	1	1200	**10.55**	23.58	**5.48**
Adaptive Transformer*	1	1200	10.28	22.90	6.35
Adaptive Transformer$^+$	1	1200	10.23	22.25	6.45
Adaptive Transformer	2	1200	10.38	**24.22**	5.84
Adaptive Transformer*	2	1200	10.10	23.55	5.89
Adaptive Transformer$^+$	2	1200	10.52	23.88	5.97
Adaptive Transformer	4	1200	10.53	23.01	6.69
Adaptive Transformer*	4	1200	10.16	22.74	5.80
Adaptive Transformer$^+$	4	1200	10.43	22.64	6.86

Source: Elaborated by the author

good metric for the video captioning task.

Table 22 presents the obtained results in the ablation study of the proposed transformer architecture, along with the original results to facilitate comparison. Variations of the best models were present with the SE block to assess if any improvement occurs in any configuration. Thus, the memory and hidden size variations are represented by T_m and d. The configurations that were evaluated are **EMT** $\boldsymbol{SE_d}$ and **EMT** $\boldsymbol{AdA_{S_t}}$ in the same validation set used to produce the results shown in Table 17. The results are superior to the literature in many cases, but not superior to the best one presented in Table 17.

It is possible to observe that, while memory size reduces, the B@4 score obtained for the method is higher, and when memory size is larger, the results get further away

from the original EMT. The relationship between memory size and the obtained results is a good point to highlight. The CIDEr-D results were more similar to those in the literature.

5.6 Dual Adaptive Attention on Transformer

Table 23 reports the performance of the proposed SkimCap model on the ae-val split of the ActivityNet Captions dataset, evaluating the impact of incorporating a second adaptive module as a dual-attention mechanism under different hierarchy and pooling configurations. The analysis focuses on how the position of adaptive attention within the multi-head attention layers affects overall captioning performance.

The results indicate that adding a second adaptive attention generally improves the model's ability to align visual and linguistic representations. In particular, the SkimCap^{+} configuration, where adaptive attention is applied to both multi-head attentions, achieves the highest scores across most metrics. This suggests that distributing adaptive modulation across both attention blocks enhances contextual reasoning during caption generation.

In contrast, the SkimCap^{-} variant, which introduces adaptive attention only after the first multi-head attention, maintains competitive performance with slightly reduced improvements, indicating that early-stage adaptation contributes more to visual grounding than late-stage refinement. The SkimCap* configuration, applied only after the last attention, presents lower scores on both CIDEr-D and BLEU@4, suggesting that late-stage adaptation is less effective when contextual information has already been compressed by previous layers.

Regarding hierarchical structures, models based on the Area and Dynamic hierarchies tend to produce more stable results, while the Volume and NParents configurations present marginally higher variance. Among the clustering strategies, the spaced variant consistently produces lower R@4 values compared to the central one, implying that clustering sparse features can improve retrieval accuracy and temporal coverage.

Table 23 – Performance of the SkimCap model on the ae-val split of the ActivityNet Captions dataset, incorporating a second adaptive module as a dual-attention mechanism with different hierarchy and clustering configurations. In which $^+$, $^-$, and * indicate the application of adaptive attention to both multi-head attentions, only after the first, and only after the last multi-head attention, respectively.

Method	Watershed	Clustering	B@4 ↑	CIDEr-D ↑	R@4 ↓
SkimCap$^+$	Area	Central	11.33	26.71	6.34
		Spaced	11.19	26.48	5.40
SkimCap$^+$	Dynamics	Central	11.23	26.67	6.18
		Spaced	11.35	26.93	5.87
SkimCap$^+$	NParents	Central	11.18	26.62	5.77
		Spaced	11.35	26.73	6.66
SkimCap$^+$	Volume	Central	11.35	26.27	6.50
		Spaced	11.37	27.01	5.80
SkimCap*	Area	Central	10.99	25.05	7.03
		Spaced	10.84	25.31	6.72
SkimCap*	Dynamics	Central	10.72	24.51	7.26
		Spaced	10.84	25.78	7.00
SkimCap*	NParents	Central	10.80	24.92	7.96
		Spaced	10.88	25.29	6.33
SkimCap*	Volume	Central	10.57	24.75	6.07
		Spaced	10.84	25.62	6.18
SkimCap$^-$	Area	Central	11.36	26.87	6.61
		Spaced	11.20	26.72	7.06
SkimCap$^-$	Dynamics	Central	11.27	26.35	6.12
		Spaced	11.35	26.93	5.87
SkimCap$^-$	NParents	Central	11.18	26.62	5.77
		Spaced	11.19	26.48	5.40
SkimCap$^-$	Volume	Central	11.35	26.29	6.49
		Spaced	11.35	27.26	5.62
SkimCap	Area Dynamics	Central	10.72	25.44	5.84
		Spaced	11.03	25.03	7.37

Source: Elaborated by the author

5.7 Evaluation of the Adaptive Transformer trained on SkimCap with MalSumm Summarizer

To evaluate the generalization capacity of the proposed model, experiments were performed with the supervised training model (MalSumm) on the ActivityNet dataset. Frame selection was done by choosing 100 frames from videos longer than 50 seconds, to maintain comparability with the results obtained by the SkimCap model, which used HieTaSkim for frame selection. Thus, the videos from the ae-val and ae-test partitions were

Table 24 – Performance of the Adaptive Transformer, HSAT, and SkimCap models and other state-of-the-art methods in ae-val split of ActivityNet Captions. Det indicates the use of detection features, Rec indicates the use of sentence-level recurrence.

Method	Det	Rec	B@4 ↑	CIDEr-D ↑	R@4 ↓
LSTM based methods					
MFT (XIONG; DAI; LIN, 2018)	χ	$\surd$	10.27	19.12	17.71
HSE (ZHANG; HU; SHA, 2018)	χ	$\surd$	9.84	18.78	13.22
LSTM based methods with detection feature					
GVD (ZHOU et al., 2019)	$\surd$	χ	11.04	21.95	8.76
GVDsup (ZHOU et al., 2019)	$\surd$	χ	**11.30**	22.94	7.04
AdvInf (PARK et al., 2019)	$\surd$	$\surd$	10.04	20.97	5.76
Transformer based methods					
VTransformer (ZHOU et al., 2018)	χ	χ	9.75	22.16	7.79
Transformer-XL (DAI et al., 2019)	χ	$\surd$	10.39	21.67	8.54
Transformer-XLRG (DAI et al., 2019)	χ	$\surd$	10.17	20.40	8.85
MART (LEI et al., 2020)	χ	$\surd$	10.33	23.42	5.18
EMT (CARDOSO; GUIMARÃES; PATROCÍNIO JR, 2021)	χ	$\surd$	10.24	23.66	**4.27**
Adaptive Transformer	χ	$\surd$	10.38	24.22	5.84
HSAT	χ	$\surd$	10.31	23.76	5.85
SkimCap	χ	$\surd$	10.72	**25.44**	5.84
SkimCap$_{sup}$	χ	$\surd$	10.80	25.30	6.56

Source: Elaborated by the author

Table 25 – Performance of the Adaptive Transformer, HSAT, and SkimCap models and other transformer-based methods in ae-test split of ActivityNet Captions. Rec indicates the use of sentence-level recurrence.

Method	Rec	B@4 ↑	CIDEr-D ↑	R@4 ↓
VTransformer (ZHOU et al., 2018)	χ	9.31	21.33	7.45
Transformer-XL (DAI et al., 2019)	$\surd$	10.25	21.71	8.79
Transformer-XLRG (DAI et al., 2019)	$\surd$	10.07	20.34	9.37
MART (LEI et al., 2020)	$\surd$	9.78	22.16	5.44
EMT (CARDOSO; GUIMARÃES; PATROCÍNIO JR, 2021)	$\surd$	10.00	22.84	**4.55**
Adaptive Transformer	$\surd$	10.00	23.04	5.29
HSAT	$\surd$	9.94	22.97	5.35
SkimCap	$\surd$	10.47	23.77	6.13
SkimCap$_{sup}$	$\surd$	**10.5**	**24.16**	6.53

Source: Elaborated by the author

summarized using the model generated by MalSumm, previously trained on the SumMe dataset. This process allows the evaluation of whether the model can maintain performance when processing frames obtained from an independent, yet consistent, selection mechanism.

Table 24 presents the results on the ae-val split. The use of other frames from the ActivityNet dataset achieved similar results to SkimCap trained with frames selected

by HieTaSkim, achieving similar CIDEr-D, superior BLEU-4, and increased Repetition-4. Despite this, the model proved efficient by achieving consistent results, even with frames chosen based on a supervised model.

To highlight the quality of the SkimCap model trained with hierarchically selected frames, the results presented in Table 25 are from frames selected with the MalSumm summarizer. Even for different frames from the same dataset, the model maintained consistency in the qualitative evaluations, surpassing the results found by the original SkimCap for the CIDEr-D metric of 24.16 and BLEU@4 of 10.5. Even with an increase in repetition frequency, the results for Repetition-4 remained close, reinforcing that the hierarchical frame selection process promotes stable generalization even when applied to unseen data.

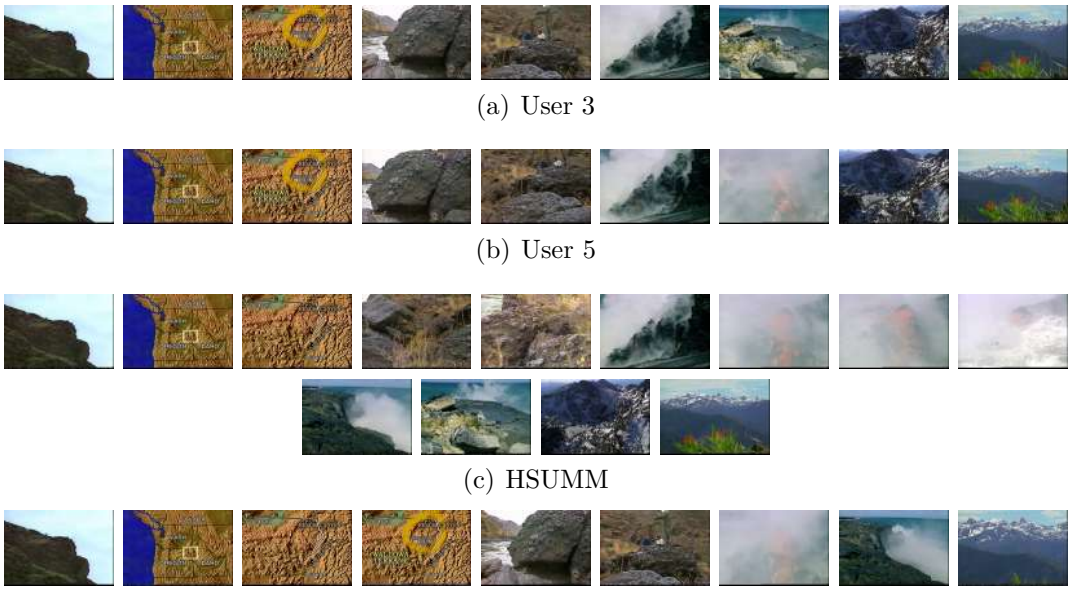
The results presented in Tables 24 and 25 validate the effectiveness of the hierarchical approach proposed in this work. Consistent performance in both the validation and test sets indicates that the proposed model not only benefits from the temporal abstraction introduced during training but also preserves its descriptive capacity when evaluated with frames obtained through independent selection methods.

5.8 Qualitative Analysis of Video Summarization

To provide a better understanding of the results obtained and their improvements, Figures 27 and 28 present samples of summaries generated by various approaches in the literature, including HSUMM (BELO et al., 2016), VSUMM1 (AVILA et al., 2011), VSUMM2 (AVILA et al., 2011), VISTO (FURINI et al., 2007) and Open Video summaries (referred to like OVSummary), and the groundtruth (GT) results, alongside those generated by the HieTaSumm method. This comparison enables the evaluation of time awareness, similarity with the GT results, and the rate of the frames selected by the HieTaSumm method and others.

Figure 27 shows the results generated by the HieTaSumm method along with HSUMM (BELO et al., 2016) results, and the summaries generated by two users. Each frame list created for each user encapsulates a distinct selection of frames, reflecting individual preferences and perspectives. Employing cosine similarity, quantifying the degree of similarity between the GT of the users and that generated for HieTaSumm method. However, it is essential to recognize that similarity is subjective and may vary among observers. Factors such as the weighting of different frames, the level of granularity in frame selection, and the specific context of the video all influence perceived similarity. Therefore, when evaluating the cosine similarity between two lists of frames, it is crucial to consider the subjective nature of the perception and the different perspectives that

Figure 27 – Comparative example of HieTaSumm results compared with HSUMM results and with the frames selected by the User 3 and User 5 (both selected 9 frames). The video summary generated by HieTaSumm contains 9 frames.



(a) User 3

(b) User 5

(c) HSUMM

(d) HieTaSumm

Source: Elaborated by the author

individuals bring to the comparison. The result obtained for the HSUMM has a much higher number of frames than the others, and, due to this, they present a large number of frames with high similarity. Furthermore, HSUMM results may not preserve chronological order.

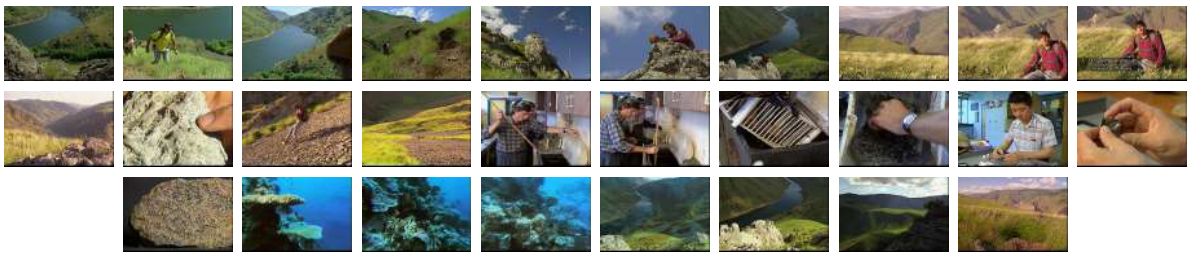
On the other hand, HieTaSumm method presents a fluid and coherent result. Furthermore, the selected keyframes are very similar to those frames in GT. For all keyframes selected by HieTaSumm method, only one frame does not have another directly correlated with those selected by the two users. But, in all cases, even with different keyframes selected by the users, the automatic summary generated by HieTaSumm the method is very close to theirs (especially for Users 3 and 5 shown in Figure 27). In addition, the unrelated keyframe preserves temporal order and, when looking at the three keyframes in which the map is present, it is possible to observe that a refinement process takes place to identify the correct highlighted region, starting from a global visualization to an analysis local that identifies the region in focus as the most important point of location on the map of the region presented in the video.

Figure 28 also presents some subjective characteristics for the keyframes selected by users 2 and 3. Considering the number of frames selected, 15 and 17 respectively, it tends to suggest the existence of a larger number of scene modifications. This variation

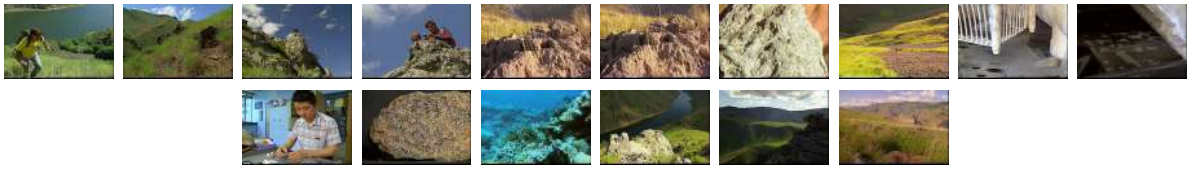
Figure 28 – Comparative example of results compared with the results of VSUMM1 (AVILA et al., 2011), VSUMM2 (AVILA et al., 2011), VISTO (FURINI et al., 2007), OVSummary and with the frames selected by the User 2 and User 3.

(a) User 2

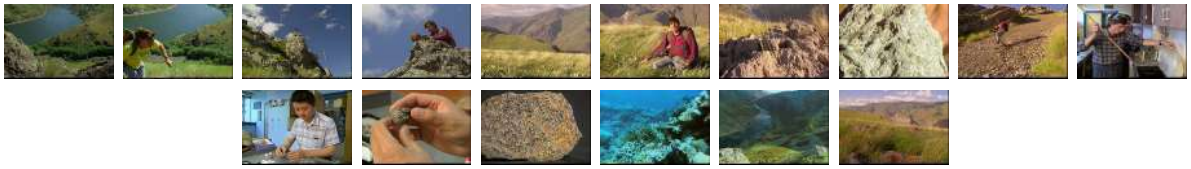
(b) User 3



(c) OVSummary



(d) Visto



(e) VSUMM1

(f) VSUMM2

(g) HieTaSumm

Source: Elaborated by the author

can cause the selection of a greater number of frames returned by automatic methods,

Figure 29 – An example of HieTaSkim result compared to the ratings of all users. The video skim generated by HieTaSkim represents 15% of the total length video_13 in the SumMe dataset, named 'Kids Playing in Leaves'. This result uses ResNet50 deep features, $\delta_t = 8$, and $\gamma = 75\%$, and contains 24 frames

Source: Elaborated by the author

but the increase in the number of scenes can cause frames to be repeated by automatic methods. In this way, the returned summaries have a great challenge of maintaining temporal coherence, but without two highly similar frames being selected without the presence of other events. With this difficulty in mind, OVSummary presents a series of repeated frames side by side. Seen displays some repeated frames, but a reduced number of frames with more similar information. VSUMM1 observes more scene modification and has some information that tends to be more similar. VSUMM2 tends to keep the results without redundancy, but without the presence of some scenes more relevant to the user. Finally, the hierarchical approach used by HieTaSumm tends to reduce the redundancy of information with a lot of similarities. HieTaSumm results have a smaller number of keyframes, but these keyframes are more related to user summaries. Moreover, keyframes selected by HieTaSumm method keeps the temporal ordering and shows that the dynamic selection of summary size helps to capture the changing scenes more smoothly.

5.9 Qualitative Analysis of Video Skimming

Figure 29 compares the result obtained by HieTaSkim with all users' scores compiled jointly. This comparison encompasses the ground-truth (GT) results alongside those generated by the HieTaSkim method. It facilitates evaluating temporal awareness, the similarity with the GT results, and assessing which frames are selected by the HieTaSkim method. The result presented in Figure 29(a) received an F-Score of 71.5 and, in addition to having high similarity with one of the users, it was also able to identify scenes that

intersect with many users' votes, especially in the beginning of the video that appears to be of great importance by the majority, as seen in Figure 29(b). Even so, it is possible to notice that the temporal relationship is respected, and the sequences in the skim are sparse and representative.

Figure 30 shows two video summaries generated by the proposed method, in which each skim corresponds to approximately 15% of the original. In Figure 30, the video 'Air Force One' is presented with an F-Score equal to 59.97%, representing one of the best summary results for the SumMe dataset. In Figure 30, video 16 represents the fluidity of the generated skim in the TVSum dataset with an F-Score of 66.55%. The overlap between the predicted and annotated frame-level scores demonstrates that StreamExLSTM effectively identifies semantically relevant segments, maintaining consistency with human preferences.

Figure 31 illustrates qualitative examples from SumMe and TVSum. For the Paintball video from SumMe, the model excluded an uninformative segment caused by a camera fall and generated a summary with three shots (start, middle, and end) capturing key moments. For the truck breakdown video 6 from TVSum, despite frequent camera changes, the model focused on the central segment, aligning with annotators, while adding a few initial frames for context. These examples demonstrate the method's ability to follow human judgment, filter irrelevant content, and detect meaningful events even under visual variability.

To evaluate the skimming model's ability to correctly represent the events present in the videos and demonstrate the relationship between information neglected in videos processed by sequential frame selection, this work uses temporal density representations using Kernel Density Estimation (KDE). Figure 33 shows the temporal density of the selected frames along the video timeline. The peaks indicate segments with a higher concentration of frames, highlighting the most relevant temporal regions according to the hierarchical summarization process. Thus, Figure 33 presents a density analysis comparing the distribution of frames selected by the Hierarchical Skimming and Sequential (Baseline) approaches over the video's duration. It is observed that the sequential strategy concentrates the selection of frames in the initial regions, resulting in limited coverage of the content. In contrast, the proposed method distributes the selection more evenly throughout the video's duration, demonstrating its ability to identify relevant moments in a broader and more representative manner.

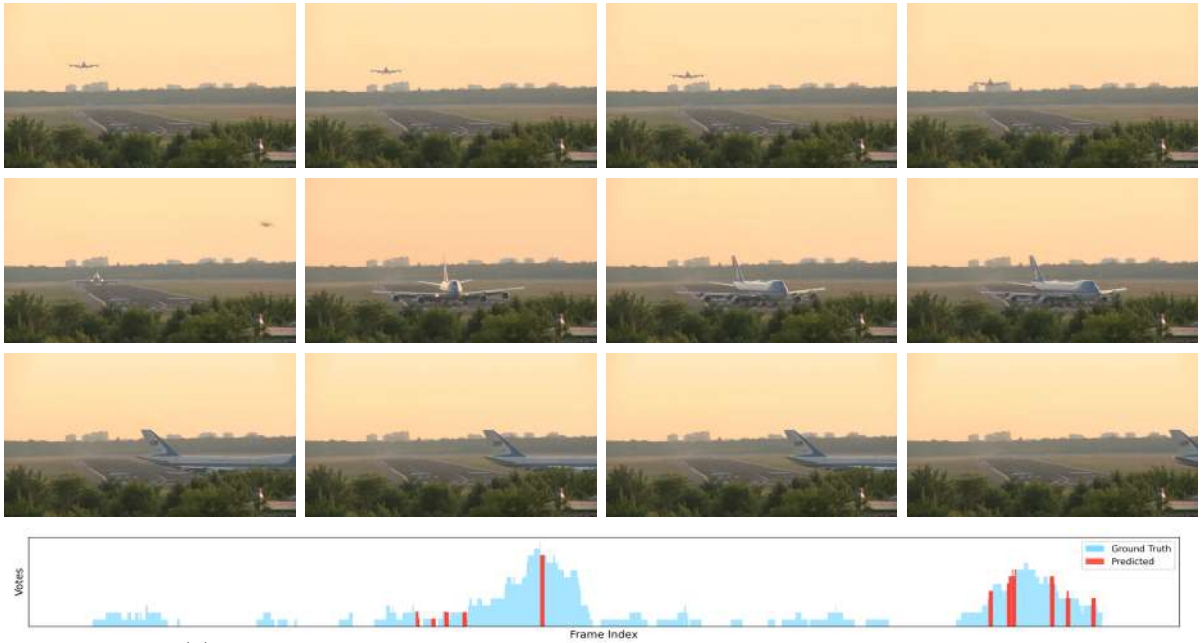
The curve corresponding to Skimming exhibits multiple peaks, indicating that the method can capture different events distributed over time. This behavior suggests greater sensitivity to transitions and changes in visual context, which is desirable for summarization and captioning of complex videos. The baseline curve, on the other hand,

presents a concentrated and abrupt density, demonstrating less temporal diversity in frame selection.

To compare the information generated in the summarization with the ground truth events, this work represents the frames in a two-dimensional space using t-SNE, as shown in Figure 32. Thus, each point represents a projected video frame. Different colors indicate distinct semantic events derived from annotated time markers, while the circled points correspond to the frames selected by the summarization method. Thus, Figure 32 shows the two-dimensional projection obtained by t-SNE, which enables visualizations of the grouping of video features according to the annotated events. Each color represents a distinct event, while the uncolored points correspond to frames without annotation. Events form well-defined regions, indicating separability between the visual patterns associated with each type of event.

The circles outlined in black highlight the frames selected by the Hierarchical Skimming method. These points are distributed in different regions of the embedding space, covering multiple clusters and demonstrating that the selection is not limited to dense or redundant areas. This distribution reinforces that the proposed method prioritizes diversity without losing contextual coherence, resulting in a more comprehensive representation of the video content.

Figure 30 – Qualitative comparison between ground-truth and generated results for the SumMe and TVSum datasets. The results display the representative frames selected by StreamExLSTM, capturing key moments of the video compared to the ground-truth importance scores (in light blue) annotated by human users and the predicted scores (in red) generated by the proposed model. The overlap between the predicted and annotated peaks demonstrates that the proposed model effectively identifies semantically relevant segments, maintaining consistency with human preferences.



(a) Frames selected for video 'Air Force One' from the SumMe dataset.

(b) Frames selected for video 16 from the TVSum dataset.

Source: Elaborated by the author

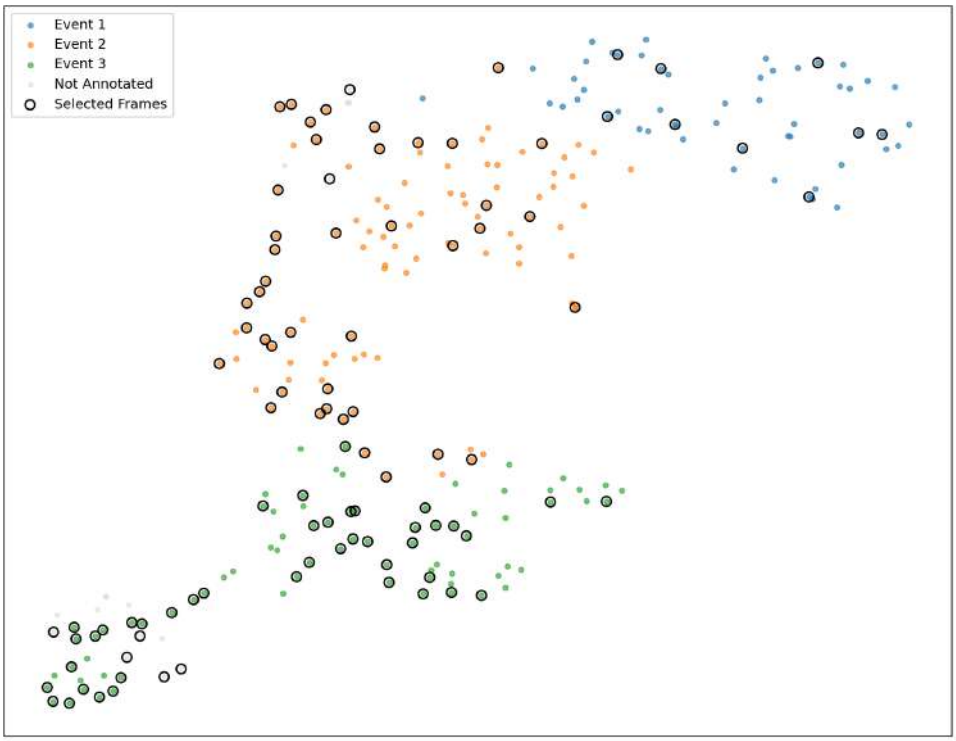
Figure 31 – Qualitative comparison between ground-truth (blue) and predicted summaries (red) for two videos. In (a), the model selects key moments in Paintball (SumMe) while skipping irrelevant segments. In (b), for TVSum video 6, it highlights central events aligned with annotator preferences. Sample frames show the semantic relevance of selected segments.

(a) Frames selected for video 'Paintball' from the SumMe dataset.

(b) Frames selected for video 6 from the TVSum dataset.

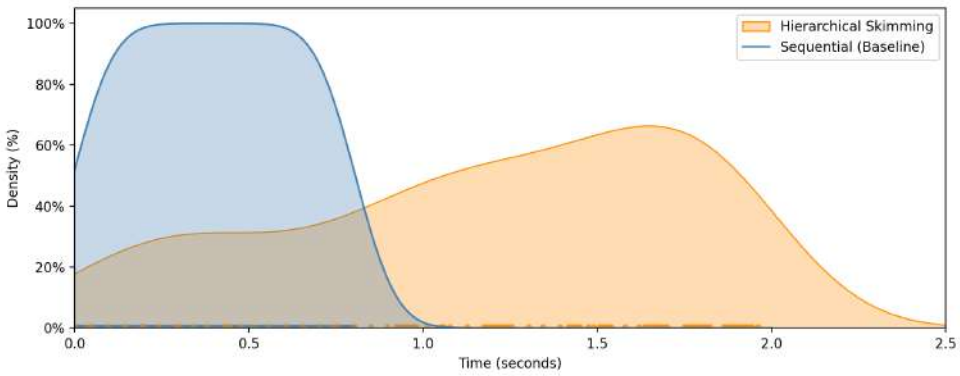
Source: Elaborated by the author

Figure 32 – t-SNE visualization of frame-level feature embeddings for the video v_4Lu8ECLHvK4 of the ActivityNet dataset.



Source: Elaborated by the author

Figure 33 – Estimated frame density distribution using Kernel Density Estimation (KDE) for the video v_4Lu8ECLHvK4 of the ActivityNet dataset.



Source: Elaborated by the author

5.10 Qualitative Analysis of Adaptive Transformer

To further promote the perception of the results obtained and their improvements, Figure 34 and 35 presents samples of paragraphs generated by the Adaptive Transformer (without any summarization) and those produced by other approaches from the literature, i.e., Vanilla Transformer (ZHOU et al., 2018), Transformer -XL (DAI et al., 2019), MART (LEI et al., 2020) and EMT (CARDOSO; GUIMARÃES; PATROCÍNIO JR, 2021), in addition to the GT results. With this, it is possible to compare the different results found by each approach, making it possible to evaluate the coherence, similarity with the GT results, and the repeatability rate of the descriptions generated by the Adaptive Transformer and the other methods.

The paragraph descriptions generated by the Vanilla Transformer cannot prevent repetition, and, in many cases, there is no similarity between the produced text and the GT result. In Figure 34 and 35, it is easy to observe that the paragraph produced by Vanilla Transformer does not have fluidity, and the notion of continuity is lost. The Vanilla Transformer is the method with the highest repetition rate among those presented. The Transformer-XL can return fluid and continuous paragraphs, but it has a high repeatability rate, as illustrated in Figure 34 and, in some cases does not return a discriminating description, as one can see in Figure 35. MART manages to maintain coherence among generated sentences and a lower repetition rate. However, in some cases, MART appears to produce less detailed descriptions, as shown in Figure 35. In the results for EMT is possible to observe that it preserves coherence and context with a low repeatability rate. In Figure 35, the generated paragraph maintains the context but doesn't get the second pronoun right. The qualitative results for the Adaptive Transformer are closer to the GT, it is possible to notice that the descriptions have coherence and fluidity. In addition, repeatability is reduced. Figure 34 presents a very discriminating result and captures the continuity of the scene and, despite the description being longer than the GT, there is no repetition. Despite the repetition present in Figure 35, it is possible to notice that the result presents continuity in the description.

Figure 34 – Examples (for qualitative analysis) of results obtained by Adaptive Transformer, compared to Vanilla Transformer, Transformer-XL, MART, EMT, and GT results for the video v_993xtlhuVII, in which blue/bold indicates the presence of repetition and red/bold indicates a possible pronoun different from the GT. Best viewed in color.

Vanilla Transformer

He continues speaking while holding the violini and showing how to play his hands. He continues playing the instrument while looking down at the camera. He continues playing the violin and then stops to speak to the camera.

Transformer-XL

A man is seen speaking to the camera while holding a violin. The man continues playing the instrument while moving his hands up and down. The man continues playing the instrument and ends by looking back to the camera

MART

A man is seen speaking to the camera while holding a violin and begins playing the instrument. The man continues to paly the instrument while moving his hands up and down. He continues to play and ends by moving his hands up and down.

EMT

A man is seen speaking to the camera while holding a violin and playing his hands. He then moves the instrument all around his hands as well as the other hand movements. He continues playing the instrument and ends by looking back to the camera.

Adaptive Transformer

A man is seen speaking to the camera while holding a musical instrument and begins playing the instrument. The man continues to play the instrument while looking off into the distance and smiling to the camera. He continues moving his hands around to play and showing off the proper hand as well as showing how to properly play.

Ground-Truth

A man is seen looking to the camera while holding a violin. The man then begins playing the instrument while the camera zooms in on his fingers. The man continues to play and stops to speak to the camera.

Source: Elaborated by the author

Figure 35 – Examples (for qualitative analysis) of results obtained by Adaptive Transformer, compared to Vanilla Transformer, Transformer-XL, MART, EMT, and GT results for the video v_GkwkHQJifDU, in which blue/bold indicates the presence of repetition and red/bold indicates a possible pronoun different from the GT. Best viewed in color.

Vanilla Transformer
She continues moving around the room and leads into her **speaking to the camera**. **She continues moving around** on the step and ends by **speaking to the camera**.

Transformer-XL
A woman is standing in a gym. She begins to do a step

MART
A woman is standing in a room talking. She starts working out on the equipment

EMT
A woman is seen speaking to the camera while standing in front of a board. **The woman** then begins moving her arms and legs around while still speaking to the camera.

Adaptive Transformer
A woman is in a room in front of a step and performs a routine while speaking to the camera. She steps up and down on a blue mat.

Ground-Truth
A woman is seen speaking to the camera and leads into her walking up and down the board. She then stands on top of the beam while speaking to the camera continuously.

Source: Elaborated by the author

5.11 Qualitative Analysis of HSAT

The results found for HSAT demonstrate that it presents an improvement related to detecting video events. Compared to the sequential selection of frames, the amount of information the method does not observe/process is large.

In some cases, when the sequential selection of frames is used, only the first one hundred frames at a rate of 2 FPS are used to represent the video. Thus, for A video with a length greater than 50 seconds, all information after the first 50 seconds is ignored. On the ANC dataset, the videos do not have the same length, the number of events varies from 2 to 6, and, in some cases, one video has more than two hundred seconds. Because of this, summarization appears as a better way to evaluate the content distributed in the entire video. Thus, the neglected information due to time limitations adopted in a sequential selection of frames does not exist with the hierarchical summarization approach.

Despite that, HSAT selects a relatively small number of frames (only 10), which is sufficient to cover all videos of the dataset (since the number of events in the ANC dataset varies from 2 to 6).

Figure 36 – A result example of HSAT showing fluidity in movement variation.

Source: Elaborated by the author

Figures 36 and 37 show the diversity of video content present in the dataset. Figure 36 shows the summarization result in a short video that has 25 seconds. Since it is a short video, the summarization process returns similar frames; however, with some minor variations in perspective. In video summarization, the number of frames remains the same for all videos, and the fluidity of the video is maintained. In addition, it is possible to correctly follow the actions over time without neglecting the video context.

Figure 37 shows the frames selected as Keyframes for the HSAT method. Figure 38 illustrates the selected frames when a sequential selection (with time constraints) is made. One can observe that in Figure 38 some content is not present at all. In contrast, HSAT

Figure 37 – A result example of HSAT with a greater number of distinct keyframes. In this case, the result should cover more than one point of view. Even so, the video summarization approach managed to capture frames that did not appear in a sequential selection of frames.

Source: Elaborated by the author

manages to obtain a greater variety of video content, making it easier to describe different moments of the video. In the sequential selection of frames, since that video has 80 seconds, it disregards any information that occurs in the final 30 seconds. In turn, HSAT uses hierarchical summarization to cover a greater variety of instances. In this way, HSAT only disregards very similar frames that are direct neighbors in time to include more distinct and meaningful frames for the video description. Due to the summarization process, the number of frames used can be reduced, generating results as significant as for techniques with large amounts of frames.

Figures 39 and 40 present a qualitative comparison between the result obtained by the HSAT with the Adaptive Transformer. As one can see, the result is very discriminative and does not have many repetitions of terms. The results presented in both situations (39 and 40) show very approximate descriptions but with some points described from another perspective. Thus, as HSAT uses features taken from different regions of the video and analyzes the importance of each frame in time, the modifications related to the description are due to the presence of points that may not be visualized in the same set of frames used to illustrate the video content. In this way, hierarchical summarization presents itself as a great candidate to improve the descriptions produced for the video captioning task.

Figure 38 – An example of selected frames by a sequential selection (with time constraints).

Source: Elaborated by the author

Figure 39 – Examples (for qualitative analysis) of results obtained by HSAT, compared to Adaptive Transformer and GT results for the video v_993xtlhuVII. The same set of frames is used only to exemplify how videos are described.

Adaptive Transformer

A man is seen speaking to the camera while holding a musical instrument and begins playing the instrument. The man continues to play the instrument while looking off into the distance and smiling to the camera. He continues moving his hands around to play and showing off the proper hand as well as showing how to properly play.

HSAT

A man is seen speaking to the camera while holding a violin and the instrument. The man continues to play and then pauses to speak to the camera. He continues moving his hands around to play and is seen speaking to the camera.

Ground-Truth

A man is seen looking to the camera while holding a violin. The man then begins playing the instrument while the camera zooms in on his fingers. The man continues to play and stops to speak to the camera.

Source: Elaborated by the author

Figure 40 – Examples (for qualitative analysis) of results obtained by HSAT, compared to Adaptive Transformer and GT results for the video v_GkwkHQJifDU. The same set of frames is used only to exemplify how videos are described.

Adaptive Transformer
A woman is in a room in front of a step and performs a routine while speaking to the camera She steps up and down on a blue mat.

HSAT
A woman is seen speaking to the camera while standing in front of a board. She begins moving up and down the board while speaking to the camera.

Ground-Truth
A woman is seen speaking to the camera and leads into her walking up and down the board She then stands on top of the beam while speaking to the camera continuously

Source: Elaborated by the author

5.12 Qualitative Analysis of SkimCap

Figures 41 and 42 present a qualitative comparison between the results obtained using SkimCap and those produced by the Adaptive Transformer (CARDOSO; GUIMARÃES; PATROCÍNIO JR, 2022) and HSAT (CARDOSO; GUIMARÃES; PATROCÍNIO JR, 2023). The captions generated by SkimCap are more discriminative and exhibit fewer repetitive expressions. The examples in Figure 41 and Figure 42 show similar overall descriptions, yet some details are conveyed from different narrative perspectives, revealing subtle shifts in semantic focus.

These differences arise from the way SkimCap processes information; it leverages hierarchical selection to extract features from diverse temporal regions of the video and evaluates frame importance globally. Consequently, the generated descriptions reflect visual cues that may not be captured when relying on fixed, sequential frame sets. This reinforces the effectiveness of hierarchical skimming in enhancing the quality and expressiveness of video captioning, especially in scenarios with long or content-rich video segments.

Figure 43 presents a result generated by the proposed model. Although the model incorrectly refers to the male subject in the video, the model successfully identifies the sequence of actions, maintaining temporal and semantic alignment with the events shown. This indicates that adaptive attention and temporal alignment of the hierarchy to select skims effectively capture the relevant movement patterns and dynamics of the scene, even if refined gender-related cues are not fully integrated into the linguistic output. Therefore, despite this mismatch in pronoun usage, the generated description remains coherent, contextually appropriate, and consistent with the visual narrative. The caption accurately conveys what is happening in the scene, demonstrating the model's ability to generalize the semantics of actions and preserve the narrative flow.

Figure 41 – Examples (for qualitative analysis) of results obtained by SkimCap, compared to Adaptive Transformer, HSAT, and GT results for the video v_993xtlhuVII. The same set of frames is presented only to exemplify how videos are described.

Adaptive Transformer
A man is seen speaking to the camera while holding a musical instrument and begins playing the instrument. The man continues to play the instrument while looking off into the distance and smiling to the camera. He continues moving his hands around to play and showing off the proper hand as well as showing how to properly play.
HSAT
A man is seen speaking to the camera while holding a violin and the instrument. The man continues to play and then pauses to speak to the camera. He continues moving his hands around to play and is seen speaking to the camera.
SkimCap
A man is seen standing in front of a camera holding up a violin. The man then begins playing the instrument while looking back to the camera. He continues playing the violin and ends by speaking to the camera.
Ground-Truth
A man is seen looking to the camera while holding a violin. The man then begins playing the instrument while the camera zooms in on his fingers. The man continues to play and stops to speak to the camera.

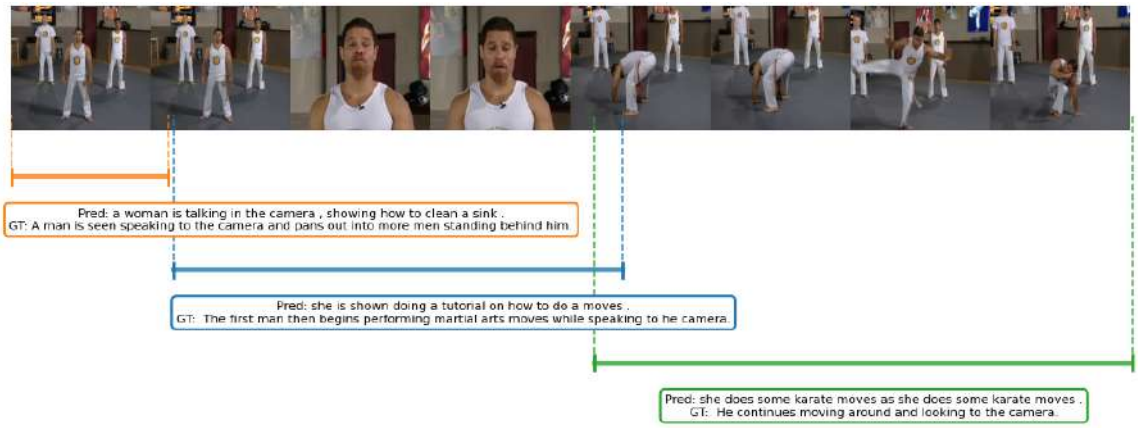
Source: Elaborated by the author

Figure 42 – Examples (for qualitative analysis) of results obtained by SkimCap, compared to Adaptive Transformer, HSAT, and GT results for the video v_GkwkHQJifDU. The same set of frames is presented only to exemplify how videos are described.

Adaptive Transformer
A woman is in a room in front of a step and performs a routine while speaking to the camera. She steps up and down on a blue mat.
HSAT
A woman is seen speaking to the camera while standing in front of a board. She begins moving up and down the board while speaking to the camera
SkimCap
A woman is seen speaking to the camera while standing in front of a large group of exercise equipment. She continues moving around on the beam while moving her arms up and down.
Ground-Truth
A woman is seen speaking to the camera and leads into her walking up and down the board. She then stands on top of the beam while speaking to the camera continuously.

Source: Elaborated by the author

Figure 43 – Comparison between event-predicted results and ground truth for the ActivityNet dataset.



Source: Elaborated by the author

5.13 Discussion

The results presented in Section 5 validate the three hypotheses proposed in this study and reveal broader implications for the design of video captioning systems. Overall, the findings suggest that the structure and quality of frame selection have a foundational effect on caption generation, often surpassing the influence of deeper architectural refinements in the transformer decoder. This emphasizes that the encoder's input distribution is a key determinant of downstream linguistic expressiveness, resonating with discussions in recent literature that highlight the tendency of video-language models to overfit redundant frames, lose sensitivity to motion cues, and focus excessively on short-term correlations.

5.13.1 Interpretation of Hypothesis 1: Complementary Effects of Summarization

The comparison across supervised and unsupervised summarization strategies supports *Hypothesis 1* by illustrating how different models capture complementary aspects of video scenes. Unlike sequential sampling, which is inherently biased toward uniform temporal spacing, summarization-based approaches emphasize semantic relevance, leading to captions that incorporate a wider and more meaningful set of visual cues. This benefit is particularly visible in videos with high scene variability, where uniform sampling fails to capture transitions or subtle events.

Moreover, the diversity of summarization strategies tested in this work reveals that their strengths depend on the temporal structure of the video. Unsupervised cluster-based methods performed best in scenarios with multiple distinct events, while supervised summarization exhibited more stability on videos characterized by continuous motion or homogeneous activities. These findings align with prior studies on video abstraction and show that combining different summarization paradigms can produce complementary coverage. This interplay between architecture and content structure reinforces the importance of adaptive frame selection in modern video-language models.

5.13.2 Interpretation of Hypothesis 2: Balancing Coverage and Temporal Continuity

The results related to *Hypothesis 2* demonstrate that summarization improves representativeness but must be carefully calibrated to preserve temporal coherence. For long videos, summarized representations consistently improved caption diversity and accuracy by prioritizing salient segments over redundant intervals. However, the experiments with shorter videos make clear that aggressive summarization compresses the video excessively, eliminating motion cues essential for describing actions, object interactions, or subtle tran-

sitions.

This tension between coverage and continuity reflects a well-known challenge in video understanding: models require enough variation to avoid redundancy but enough continuity to infer movement reliably. Our findings echo earlier work in event detection and keyframe extraction that warned against using fixed summarization thresholds across heterogeneous datasets. In practice, this suggests that summarization modules should incorporate video-length–aware or entropy-aware constraints to prevent information loss. The fact that representation quality improves when motion cues are preserved highlights the structural role of temporal reasoning in caption generation, even when only a subset of frames is available.

5.13.3 Interpretation of Hypothesis 3: Effects of Adaptive Attention

The evaluation of the Adaptive Transformer confirms *Hypothesis 3*, showing that a secondary attention mechanism enhances the internal calibration of multi-head attention outputs. This recalibration helps recover latent features that sequential attention alone underweights, leading to captions that better reflect local dependencies and nuanced visual patterns. The gains observed in the ablation study further demonstrate that the adaptive attention is most effective when positioned after both multi-head attention modules, suggesting that the mechanism works best as a refinement step rather than as a replacement for the primary attention.

Nevertheless, introducing additional attention layers introduces new risks. Deeper recalibration makes the decoder more sensitive to correlated features and may amplify repetitive linguistic patterns. This was evident in certain hierarchical configurations, where double adaptive attention increased CIDEr-D scores but slightly reduced lexical diversity. Similar behaviors have been reported in architectures such as SE-ResNet and stacked cross-attention models, where overemphasis of specific features can distort the semantic balance of the output. These observations highlight the importance of regularization techniques and architectural constraints to maintain stability in transformer-based captioning models.

5.13.4 Broader Challenges: Reinforcement Learning and Multimodal Architectures

Despite the consistent improvements obtained through summarization and adaptive attention, the proposed methods face structural challenges when compared to reinforcement-learning-based systems and multimodal foundation models. RL-based captioning approaches, which directly optimize reward functions derived from evaluation metrics, often

outperform supervised systems by reshaping the output distribution during training. In contrast, the models presented in this work rely exclusively on supervised learning, which limits their ability to align directly with CIDEr-D metric. This distinction explains part of the performance gap observed in the literature and highlights a methodological limitation of the approach.

Furthermore, multimodal architectures with large-scale pretraining enjoy advantages beyond model size or backbone depth. Their extensive exposure to diverse visual and textual contexts enables richer grounding and better generalization across unseen scenarios. While the approaches proposed here yield competitive results within the supervised Transformer family, matching or surpassing multimodal foundation models remains difficult without comparable training resources, aligned pretraining objectives, or multi-sensor inputs such as audio or speech. These structural factors underscore the broader challenge facing captioning models that rely primarily on visual-only supervised learning.

5.13.5 Implications and Limitations

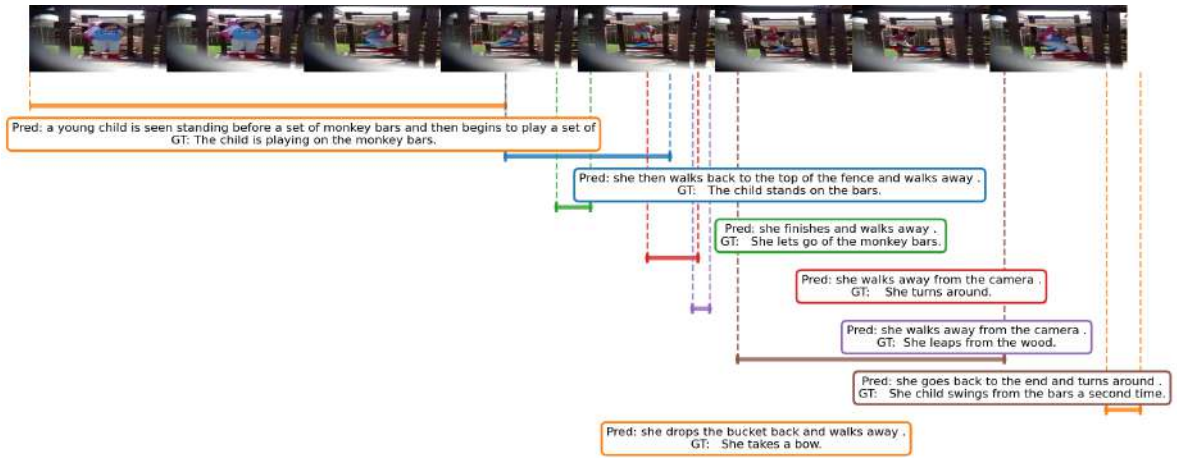
The collective findings demonstrate that summarization serves not only to reduce redundant frames but also to reorganize the video into a more semantically aligned representation for caption generation. This also suggests that hierarchical and event-based summarization strategies should be viewed as integral components of the captioning pipeline, rather than optional preprocessing steps. In practice, the improved representativeness provided by summarization enhances feature consistency, reduces the burden on the encoder, and leads to higher-quality attention distributions within the transformer.

However, some limitations must be acknowledged. The performance of the summarization models depends heavily on the quality of the frame-level features, which are sensitive to occlusion, motion blur, and low-light conditions. The summarization thresholds used in this study are fixed, which restricts their ability to adapt to variability across video lengths and content types. Additionally, the increased repetitiveness observed in deeper adaptive attention configurations indicates that further work is needed to calibrate the trade-off between expressive refinement and linguistic diversity. These limitations highlight the complexity of designing systems that must simultaneously handle variability in content, preserve motion cues, and generate coherent, non-redundant captions.

Figure 44 presents another limitation that arises when reference annotations contain a large number of events for videos that exhibit minimal visual or semantic variation. In these cases, the dense event structure can excessively segment the narrative, forcing the model to pay attention to transitions that are not visually significant and increasing the complexity of the captioning task. This incompatibility can lead to unnecessary linguistic elaborations, weakening of temporal coherence, and an increased chance of repetitive

phrases as the model attempts to justify each annotated segment. Consequently, a more compact event representation would help generate captions that are more natural, concise, and semantically faithful to the underlying content.

Figure 44 – Comparison between event-predicted results and ground truth for the video v_wZgBJlWqWWI of the ActivityNet dataset.



Source: Elaborated by the author

An additional limitation concerns the long training time required by the proposed architecture, a problem that is exacerbated when working with long, densely annotated videos, such as those found in large-scale benchmarks. Even with a high-end GPU like the RTX Quadro A6000, the combination of extensive sequences, adaptive attention modules, and components with increased memory results in slow convergence and substantial computational overhead. This prolonged training cycle not only delays iterative experimentation due to the need to adjust hyperparameters, ablation studies, and refinements in attention mechanisms, but also restricts the practical scalability of the approach. Reducing the complexity of the architecture, optimizing batch processing strategies, or incorporating more efficient sequence modeling techniques may therefore be necessary to make the training process more sustainable without compromising the quality of the captions.

6 CONCLUSION

This work presents three distinct methods for video captioning, all incorporating an Adaptive Transformer. The first method employs sequential frames, using a limited number of frames in their original order. The second method applies a hierarchical selection of a fixed number of keyframes, while the third method selects a variable number of key segments based on detected events, also using a hierarchical strategy.

The Adaptive Transformer uses additional attention modules after the multi-head attention modules within the transformer. This adaptive mechanism aims to reinforce the attention applied previously. With this, the second attention tends to capture closer adjacent information with higher quality. When applied in the approach that selects frames sequentially, even with a limited amount of information, experiments have shown that this approach increases the quality of the generated captioning while maintaining coherence when compared to supervised sequential approaches in the literature. This improvement comes from the local analysis and refinement of the attention outputs generated in previous layers, resulting in coherent, diverse captions aligned with the ground truth, in addition to reducing repetition, which supports *Hypothesis 3*.

The second approach extends the use of the Adaptive Transformer and evaluates the impacts of improved frame selection. Thus, a hierarchical strategy was adopted to select a fixed number of keyframes that represented the events, but without the movement present in the video scenes. By evaluating the importance of the frames and prioritizing the most informative ones, the HSAT variation reduces the volume of data processed during caption generation without compromising the content. This targeted selection used HieTaSumm with a fixed number of keyframes in a graph hierarchy-based tool. Choosing frames based on the hierarchy optimizes the summarization process without losing the overall context, demonstrating *Hypothesis 2*.

The third approach (SkimCap) modifies the frame selection policy to ensure that the movement information of the agents in the videos is preserved. Different frame selection policies were applied. Thus, the selection of frames was improved so that the selected frames correctly represented the video. Unlike the approach used in HSAT, in this variant, the videos are dynamically processed using different summarization techniques, one

of which is a modification of HieTaSumm to select segments rather than single frames. Another summarization approach was the use of a tool based on an extended LSTM with enhanced memory. Thus, both summarization approaches are applied to select key segments of variable length, adapting the number of frames to represent distinct events in the video. Both hierarchical selection (HieTaSkim) and supervised selection (Streamlined or MalSumm) are event-sensitive and capture content changes over time more effectively, increasing coherence between video segments, reducing redundancy, and increasing diversity among the selected frames, and supporting *Hypothesis 1*.

In addition to frame selection, the summarization methods reduce computational cost by focusing on highly informative subsets of frames. The integration of adaptive attention further improves the quality of the captions. Experimental results indicate that the proposed variants achieve comparable performance to the Adaptive Transformer when keyframes are selected and outperform the baseline when summarization is applied to recognize events. Despite the increased summarization time, the selection of events produces coherent, diverse captions aligned with the ground truth. These results are evident when observing the quantitative and qualitative results. The application of summarization strategies in frame selection also improves feature consistency, leading to higher BLEU-4 and CIDEr-D scores, less repetition, and overall competitive performance. In particular, area-based segmentation summarization achieves the most significant gains.

Finally, this work performed ablation studies on the proposed modules to verify the contribution of each component of the model. Thus, the adaptive attention added to the transformer was evaluated after each multi-head attention, but proved most efficient when aligned after both simultaneously. In addition, evaluations were conducted in the summarization process to measure the impact of modifying hierarchies, disabling modules, and assessing data dispersion. The ablation study confirmed the gain in using the proposed architecture with summarization as pre-processing and the benefits of the Adaptive Transformer.

6.1 Future Works

As directions for future research, the use of reinforcement learning emerges as a promising strategy to improve both the summarization stage and the training of video captioning models. This approach allows the model to learn interactively, adjusting its predictions based on rewards defined by the quality of the results. Thus, the learning process becomes more dynamic and aligned with the desired behavior, favoring better adaptation to different types of videos and narrative contexts.

Another relevant line of investigation involves the application of generative adver-

sarial networks (GANs) to expand the generalization capacity of the models. By generating new synthetic video segments, these networks can enrich the training set with more varied and challenging examples. This additional diversity can help the model identify complex patterns and deal with infrequent situations in the original data, overcoming typical limitations of purely supervised methods.

Semi-supervised models also represent a valuable alternative, especially in scenarios with limited labeled data. By simultaneously exploring supervised and unsupervised information, these approaches are able to better capture the underlying structure of the data and leverage clues present in unlabeled samples. The integration of these three lines, reinforcement learning, generative networks, and semi-supervised models, tends to significantly expand the potential of video summarization and captioning systems, consolidating advances towards more robust, flexible, and generalizable models.

Finally, the post-processing stage of text in video captioning tasks appears as a candidate for reducing repetitiveness. This refinement helps ensure fluency, clarity, and greater informative value in the final caption. Furthermore, it opens up opportunities to integrate different large-scale language models. While one model might generate the initial caption focusing on temporal coherence, another could act as a reviewer of the proposed sentence, eliminating redundancies and improving textual quality. In this way, the combination of the proposed model and a new post-processing stage may be contribute to more natural, concise, and consistent captions.

6.2 Published Papers

This study resulted in the following published papers:

- Cardoso, L. V. et al. Exploring adaptive attention in memory transformer applied to coherent video paragraph captioning. In: IEEE. 2022 IEEE Eighth International Conference on Multimedia Big Data (BigMM). [S.l.], 2022. p. 37–44
- Cardoso, L. V. et al. Hierarchical time-aware summarization with an adaptive transformer for video captioning. International Journal of Semantic Computing, v. 17, n. 04, p. 569–592, 2023.
- Cardoso, L. V. et al. Hierarchical time-aware approach for video summarization. In: SPRINGER. Brazilian Conference on Intelligent Systems. [S.l.], 2023. p. 274–288.
- Cardoso, L. V. et al. Unsupervised video skimming with adaptive hierarchical shot detection. In: IEEE. 2024 37th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI). [S.l.], 2024. p. 1–6.

- Cardoso, L. V. et al. Skimcap: A transformer-based video captioning method with adaptive attention and hierarchical skimming features. In: 2025 38th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI). [S.l.: s.n.], 2025. p. 1–6.

- Cardoso, L. V. et al. Streamlined extended long short-term memory for video skimming. Pattern Recognition Letters, v. 198, p. 132–139, 2025.

- Cardoso, L. V. et al. Memory-augmented long short-term memory for dynamic video summarization. In: 2025 38th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI). [S.l.: s.n.], 2025. p. 1–6

6.3 Awards

This work received an honorable mention for the following papers:

- Cardoso, L. V. et al. Unsupervised video skimming with adaptive hierarchical shot detection. In: IEEE. 2024 37th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI). [S.l.], 2024. p. 1–6.

- Cardoso, L. V. et al. Memory-augmented long short-term memory for dynamic video summarization. In: 2025 38th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI). [S.l.: s.n.], 2025. p. 1–6

REFERENCES

AAFAQ, N.; MIAN, A.; LIU, W.; GILANI, S. Z.; SHAH, M. Video description: A survey of methods, datasets, and evaluation metrics. ACM Computing Surveys (CSUR), ACM New York, NY, USA, v. 52, n. 6, p. 1–37, 2019.

APOSTOLIDIS, E.; ADAMANTIDOU, E.; METSAI, A. I.; MEZARIS, V.; PATRAS, I. Video summarization using deep neural networks: A survey. Proceedings of the IEEE, v. 109, n. 11, p. 1838–1863, 2021.

APOSTOLIDIS, E.; BALAOURAS, G.; MEZARIS, V.; PATRAS, I. Combining global and local attention with positional encoding for video summarization. In: IEEE. 2021 IEEE international symposium on multimedia (ISM). [S.l.], 2021. p. 226–234.

AVILA, S. E. F. D.; LOPES, A. P. B.; JR, A. da L.; ARAÚJO, A. de A. Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method. Pattern Recognition Letters, Elsevier, v. 32, n. 1, p. 56–68, 2011.

BA, J.; MNIH, V.; KAVUKCUOGLU, K. Multiple object recognition with visual attention. arXiv preprint arXiv:1412.7755, 2014.

BAHDANAU, D.; CHO, K.; BENGIO, Y. Neural machine translation by jointly learning to align and translate. In: ICLR. [S.l.: s.n.], 2015.

BASAVARAJAIAH, M.; SHARMA, P. Survey of compressed domain video summarization techniques. ACM Comput. Surv., Association for Computing Machinery, New York, NY, USA, v. 52, n. 6, 2019. ISSN 0360-0300.

BECK, M. et al. xLSTM: Extended long short-term memory. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems. [S.l.: s.n.], 2024.

BELO, L. S.; CAETANO JR, C. A.; PATROCÍNIO JR, Z. K. G.; GUIMARÃES, S. J. F. Summarizing video sequence using a graph-based hierarchical approach. Neurocomputing, Elsevier, v. 173, p. 1001–1016, 2016.

BENGIO, Y.; GOODFELLOW, I.; COURVILLE, A. Deep learning. [S.l.]: MIT press, 2017.

CARDOSO, L. V.; AZEVEDO, B. P. B. V. C.; GUIMARÃES, S. J. F.; PATROCÍNIO JR, Z. K. G. Skimcap: A transformer-based video captioning method with adaptive attention and hierarchical skimming features. In: 2025 38th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI). [S.l.: s.n.], 2025. p. 1–6.

CARDOSO, L. V.; GOMES, G. O. R.; GUIMARÃES, S. J. F.; PATROCÍNIO JR, Z. K. G. Hierarchical time-aware approach for video summarization. In: SPRINGER. Brazilian Conference on Intelligent Systems. [S.l.], 2023. p. 274–288.

CARDOSO, L. V.; GUIMARÃES, S. J. F.; PATROCÍNIO JR, Z. K. G. Enhanced-memory transformer for coherent paragraph video captioning. In: IEEE. 2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI). [S.l.], 2021. p. 836–840.

CARDOSO, L. V.; GUIMARÃES, S. J. F.; PATROCÍNIO JR, Z. K. G. Exploring adaptive attention in memory transformer applied to coherent video paragraph captioning. In: IEEE. 2022 IEEE Eighth International Conference on Multimedia Big Data (BigMM). [S.l.], 2022. p. 37–44.

CARDOSO, L. V.; GUIMARÃES, S. J. F.; PATROCÍNIO JR, Z. K. G. Hierarchical time-aware summarization with an adaptive transformer for video captioning. International Journal of Semantic Computing, v. 17, n. 04, p. 569–592, 2023. Disponível em: <https://doi.org/10.1142/S1793351X23640031>.

CARDOSO, L. V.; SORAGGI, B. H. P.; GUIMARÃES, S. J. F.; PATROCÍNIO JR, Z. K. G. Memory-augmented long short-term memory for dynamic video summarization. In: 2025 38th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI). [S.l.: s.n.], 2025. p. 1–6.

CARDOSO, L. V.; SORAGGI, B. H. P.; GUIMARÃES, S. J. F.; PATROCÍNIO JR, Z. K. G. Streamlined extended long short-term memory for video skimming. Pattern Recognition Letters, v. 198, p. 132–139, 2025. ISSN 0167-8655. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167865525002806>.

CARDOSO, L. V.; WERNECK, J. F. M.; GUIMARÃES, S. J. F.; PATROCÍNIO JR, Z. K. G. Unsupervised video skimming with adaptive hierarchical shot detection. In: IEEE. 2024 37th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI). [S.l.], 2024. p. 1–6.

CHAN, W.; JAITLY, N.; LE, Q. V.; VINYALS, O. Listen, attend and spell. arXiv preprint arXiv:1508.01211, 2015.

CHEN, C.-F.; FAN, Q.; PANDA, R. Crossvit: Cross-attention multi-scale vision transformer for image classification. arXiv preprint arXiv:2103.14899, 2021.

CHEN, Y.-C. et al. Uniter: Learning universal image-text representations. ICLR, 2019.

CHO, K. et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: EMNLP. [S.l.]: ACL, 2014. p. 1724–1734.

COUSTY, J.; NAJMAN, L. Incremental algorithm for hierarchical minimum spanning forests and saliency of watershed cuts. In: SPRINGER. Mathematical Morphology and Its Applications to Image and Signal Processing: 10th International Symposium, ISMM 2011, Verbania-Intra, Italy, July 6-8, 2011. Proceedings 10. [S.l.], 2011. p. 272–283.

COUSTY, J.; NAJMAN, L.; KENMOCHI, Y.; GUIMARÃES, S. Hierarchical segmentations with graphs: Quasi-flat zones, minimum spanning trees, and saliency maps. J. Math. Imaging Vis., Kluwer Academic Publishers, USA, v. 60, n. 4, p. 479–502, may 2018. ISSN 0924-9907.

CRANWELL, J. et al. Adolescents' exposure to tobacco and alcohol content in youtube music videos. Addiction, Wiley Online Library, v. 110, n. 4, p. 703–711, 2015.

DAI, Z. et al. Transformer-XL: Attentive language models beyond a fixed-length context. In: 57th Annual Meeting of the ACL. [S.l.: s.n.], 2019. p. 2978–2988.

DEVLIN, J.; CHANG, M.-W.; LEE, K.; TOUTANOVA, K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

DONAHUE, J. et al. Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition. [S.l.: s.n.], 2015. p. 2625–2634.

EJAZ, N.; TARIQ, T. B.; BAIK, S. W. Adaptive key frame extraction for video summarization using an aggregation mechanism. Journal of Visual Communication and Image Representation, Elsevier, v. 23, n. 7, p. 1031–1040, 2012.

FURINI, M.; GERACI, F.; MONTANGERO, M.; PELLEGRINI, M. Visto: visual storyboard for web video browsing. In: Proceedings of the 6th ACM International Conference on Image and Video Retrieval. [S.l.: s.n.], 2007. p. 635–642.

GAO, L.; GUO, Z.; ZHANG, H.; XU, X.; SHEN, H. T. Video captioning with attention-based lstm and semantic consistency. IEEE Transactions on Multimedia, IEEE, v. 19, n. 9, p. 2045–2055, 2017.

GING, S.; ZOLFAGHARI, M.; PIRSIAVASH, H.; BROX, T. Coot: Cooperative hierarchical transformer for video-text representation learning. Advances in neural information processing systems, v. 33, p. 22605–22618, 2020.

GUIMARÃES, S.; KENMOCHI, Y.; COUSTY, J.; PATROCÍNIO, Z.; NAJMAN, L. Hierarchizing graph-based image segmentation algorithms relying on region dissimilarity: the case of the felzenszwalb-huttenlocher method. Mathematical Morphology-Theory and Applications, De Gruyter, v. 2, n. 1, p. 55–75, 2017.

GUO, Q.; QIU, X.; LIU, P.; XUE, X.; ZHANG, Z. Multi-scale self-attention for text classification. In: AAAI. [S.l.: s.n.], 2020. v. 34, n. 05, p. 7847–7854.

GYGLI, M.; GRABNER, H.; RIEMENSCHNEIDER, H.; GOOL, L. V. Creating summaries from user videos. In: SPRINGER. Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII 13. [S.l.], 2014. p. 505–520.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. In: IEEE CVPR. [S.l.: s.n.], 2016. p. 770–778.

HEILBRON, F. C.; ESCORCIA, V.; GHANEM, B.; NIEBLES, J. C. Activitynet: A large-scale video benchmark for human activity understanding. In: IEEE CVPR. [S.l.: s.n.], 2015. p. 961–970.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. Neural computation, MIT Press, v. 9, n. 8, p. 1735–1780, 1997.

HU, J.; SHEN, L.; SUN, G. Squeeze-and-excitation networks. In: IEEE CVPR. [S.l.: s.n.], 2018. p. 7132–7141.

HUANG, C.; WANG, H. A novel key-frames selection framework for comprehensive video summarization. IEEE Transactions on Circuits and Systems for Video Technology, IEEE, v. 30, n. 2, p. 577–589, 2019.

HUANG, L.; WANG, W.; CHEN, J.; WEI, X.-Y. Attention on attention for image captioning. In: IEEE ICCV. [S.l.: s.n.], 2019. p. 4634–4643.

IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: PMLR. ICML. [S.l.], 2015. p. 448–456.

JADON, S.; JASIM, M. Unsupervised video summarization framework using keyframe extraction and video skimming. In: IEEE. 2020 IEEE 5th International Conference on computing communication and automation (ICCCA). [S.l.], 2020. p. 140–145.

JI, Z.; XIONG, K.; PANG, Y.; LI, X. Video summarization with attention-based encoder–decoder networks. IEEE Transactions on Circuits and Systems for Video Technology, IEEE, v. 30, n. 6, p. 1709–1717, 2019.

KRISHNA, R.; HATA, K.; REN, F.; FEI-FEI, L.; NIEBLES, J. C. Dense-captioning events in videos. In: IEEE ICCV. [S.l.: s.n.], 2017. p. 706–715.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, v. 25, 2012.

KRUSKAL, J. B. On the shortest spanning subtree of a graph and the traveling salesman problem. Proceedings of the American Mathematical society, v. 7, n. 1, p. 48–50, 1956.

KUMARI, T. M.; DASH, D. K.; SAHU, A. An efficient video summarization technique using texture feature and spectral clustering. In: IEEE. 2023 OITS International Conference on Information Technology (OCIT). [S.l.], 2023. p. 266–271.

LEI, J. et al. MART: Memory-augmented recurrent transformer for coherent video paragraph captioning. In: 58th Annual Meeting of the ACL. [S.l.: s.n.], 2020. p. 2603–2614.

LI, H.; KE, Q.; GONG, M.; ZHANG, R. Video joint modelling based on hierarchical transformer for co-summarization. IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE, v. 45, n. 3, p. 3904–3917, 2022.

LI, H.; KE, Q.; GONG, M.; DRUMMOND, T. Progressive video summarization via multimodal self-supervised learning. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. [S.l.: s.n.], 2023. p. 5584–5593.

LI, L.; GONG, B. End-to-end video captioning with multitask reinforcement learning. In: IEEE WACV. [S.l.: s.n.], 2019. p. 339–348.

LIANG, G.; LV, Y.; LI, S.; ZHANG, S.; ZHANG, Y. Video summarization with a convolutional attentive adversarial network. Pattern Recognition, Elsevier, v. 131, p. 108840, 2022.

LIN, J.; ZHONG, S.-h.; FARES, A. Deep hierarchical lstm networks with attention for video summarization. Computers & Electrical Engineering, Elsevier, v. 97, p. 107618, 2022.

LIU, W. et al. A survey of deep neural network architectures and their applications. Neurocomputing, Elsevier, v. 234, p. 11–26, 2017.

LIU, Y.; ZHAO, W.-L.; NGO, C.-W.; XU, C.-S.; LU, H.-Q. Coherent bag-of audio words model for efficient large-scale video copy detection. In: ACM. Proceedings of the ACM International Conference on Image and Video Retrieval. [S.l.], 2010. p. 89–96.

MAIA, D. S.; COUSTY, J.; NAJMAN, L.; PERRET, B. Characterization of graph-based hierarchical watersheds: Theory and algorithms. Journal of Mathematical Imaging and Vision, v. 62, n. 5, p. 627–658, Jun 2020. ISSN 1573-7683. Disponível em: <https://doi.org/10.1007/s10851-019-00936-6>.

MARTINS, G. B.; PEREIRA, D. R.; ALMEIDA, J. G.; ALBUQUERQUE, V. H. C. de; PAPA, J. P. Opfsumm: on the video summarization using optimum-path forest. Multimedia Tools and Applications, Springer, v. 79, n. 15, p. 11195–11211, 2020.

MEENA, P.; KUMAR, H.; Kumar Yadav, S. A review on video summarization techniques. Engineering Applications of Artificial Intelligence, v. 118, p. 105667, 2023. ISSN 0952-1976. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0952197622006571>.

MILAKOV, M.; GIMELSHEIN, N. Online normalizer calculation for softmax. arXiv preprint arXiv:1805.02867, 2018.

MINAIDI, M. N.; PAPAIOANNOU, C.; POTAMIANOS, A. Self-attention based generative adversarial networks for unsupervised video summarization. In: 2023 31st European Signal Processing Conference (EUSIPCO). [S.l.: s.n.], 2023. p. 571–575.

MOLINO, A. G. del; TAN, C.; LIM, J.-H.; TAN, A.-H. Summarization of egocentric videos: A comprehensive survey. IEEE Transactions on Human-Machine Systems, v. 47, n. 1, p. 65–76, 2017.

NAIR, M. S.; MOHAN, J. Vsmcnn-dynamic summarization of videos using salient features from multi-cnn model. Journal of Ambient Intelligence and Humanized Computing, Springer, v. 14, n. 10, p. 14071–14080, 2023.

NAJMAN, L.; COUSTY, J.; PERRET, B. Playing with kruskal: algorithms for morphological trees in edge-weighted graphs. In: SPRINGER. Mathematical Morphology and Its Applications to Signal and Image Processing: 11th International Symposium, ISMM 2013, Uppsala, Sweden, May 27-29, 2013. Proceedings 11. [S.l.], 2013. p. 135–146.

NARWAL, P.; DUHAN, N.; Kumar Bhatia, K. A comprehensive survey and mathematical insights towards video summarization. Journal of Visual Communication and Image Representation, v. 89, p. 103670, 2022. ISSN 1047-3203. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1047320322001900>.

OMNICORE. YouTube by the Numbers: Stats, Demographics & Fun Facts. 2020. Disponível em: <https://www.omnicoreagency.com/youtube-statistics>. Acesso em: 04 mai 2020.

ORDÓÑEZ, F.; ROGGEN, D. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. Sensors, Multidisciplinary Digital Publishing Institute, v. 16, n. 1, p. 115, 2016.

OTANI, M.; NAKASHIMA, Y.; RAHTU, E.; HEIKKILA, J. Rethinking the evaluation of video summaries. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. [S.l.: s.n.], 2019. p. 7596–7604.

PAN, Y.; YAO, T.; LI, Y.; MEI, T. X-linear attention networks for image captioning. In: IEEE CVPR. [S.l.: s.n.], 2020. p. 10971–10980.

PANDA, R.; MITHUN, N. C.; ROY-CHOWDHURY, A. K. Diversity-aware multi-video summarization. IEEE Transactions on Image Processing, v. 26, n. 10, p. 4712–4724, 2017.

PANDEY, S.; DWIVEDY, P.; MEENA, S.; POTNIS, A. A survey on key frame extraction methods of a mpeg video. In: 2017 International Conference on Computing, Communication and Automation (ICCCA). [S.l.: s.n.], 2017. p. 1192–1196.

PAPINENI, K.; ROUKOS, S.; WARD, T.; ZHU, W.-J. Bleu: a method for automatic evaluation of machine translation. In: 40th annual meeting of the ACL. [S.l.: s.n.], 2002. p. 311–318.

PARK, J. S.; ROHRBACH, M.; DARRELL, T.; ROHRBACH, A. Adversarial inference for multi-sentence video description. In: IEEE CVPR. [S.l.: s.n.], 2019. p. 6598–6608.

PHAPHUANGWITTAYAKUL, A.; GUO, Y.; YING, F.; XU, W.; ZHENG, Z. Self-attention recurrent summarization network with reinforcement learning for video summarization task. In: IEEE. 2021 IEEE International Conference on Multimedia and Expo (ICME). [S.l.], 2021. p. 1–6.

PUTHIGE, I.; HUSSAIN, T.; GUPTA, S.; AGARWAL, M. Attention over attention: An enhanced supervised video summarization approach. Procedia Computer Science, Elsevier, v. 218, p. 2359–2368, 2023.

ROCHAN, M.; YE, L.; WANG, Y. Video summarization using fully convolutional sequence networks. In: Proceedings of the European conference on computer vision (ECCV). [S.l.: s.n.], 2018. p. 347–363.

ROHRBACH, M. et al. Translating video content to natural language descriptions. In: Proceedings of the IEEE International Conference on Computer Vision. [S.l.: s.n.], 2013. p. 433–440.

SELVARAJ, J.; ALMUTAIRI, F.; ASLAM, S. M.; UMAPATHY, S. Binary and multi-class classification of colorectal polyps using crp-vit: A comparative study between cnns and qnns. Life, MDPI, v. 15, n. 7, p. 1124, 2025.

SHAO, J.; SHEN, H. T.; ZHOU, X. Challenges and techniques for effective and efficient similarity search in large video databases. Proceedings of the VLDB Endowment, VLDB Endowment, v. 1, n. 2, p. 1598–1603, 2008.

SILVA, A. da et al. O youtube como plataforma de marketing: Um estudo bibliográfico. Revista Educação, Gestão e Sociedade, ISSN, p. 2179–9636, 2017.

SONG, Y.; VALLMITJANA, J.; STENT, A.; JAIMES, A. Tvsum: Summarizing web videos using titles. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR). [S.l.: s.n.], 2015. p. 5179–5187.

SUN, C.; MYERS, A.; VONDRICK, C.; MURPHY, K.; SCHMID, C. Videobert: A joint model for video and language representation learning. In: IEEE ICCV. [S.l.: s.n.], 2019. p. 7464–7473.

SUTTON, R. S.; BARTO, A. G. Reinforcement learning: An introduction. [S.l.]: MIT press, 2018.

SZEGEDY, C. et al. Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. [S.l.: s.n.], 2015. p. 1–9.

TANG, H.; JI, D.; LI, C.; ZHOU, Q. Dependency graph enhanced dual-transformer structure for aspect-based sentiment classification. In: 58th Annual Meeting of the ACL. [S.l.: s.n.], 2020. p. 6578–6588.

TIWARI, V.; BHATNAGAR, C. A survey of recent work on video summarization: approaches and techniques. Multimedia Tools and Applications, Springer, v. 80, n. 18, p. 27187–27221, 2021.

VASWANI, A. et al. Attention is all you need. In: Advances in Neural Information Processing Systems. [S.l.: s.n.], 2017. p. 5998–6008.

VEDANTAM, R.; ZITNICK, C. L.; PARIKH, D. Cider: Consensus-based image description evaluation. In: IEEE CVPR. [S.l.: s.n.], 2015. p. 4566–4575.

VENUGOPALAN, S. et al. Sequence to sequence-video to text. In: Proceedings of the IEEE international conference on computer vision. [S.l.: s.n.], 2015. p. 4534–4542.

VIVEKRAJ, V.; DEBASHIS, S.; BALASUBRAMANIAN, R. Video skimming: Taxonomy and comprehensive survey. ACM Comput. Surv., Association for Computing Machinery, New York, NY, USA, v. 52, n. 5, 2019. ISSN 0360-0300.

VYDANA, H. K.; KARAFIÁT, M.; ZMOLIKOVA, K.; BURGET, L.; ČERNOCKỲ, H. Jointly trained transformers models for spoken language translation. In: IEEE ICASSP. [S.l.: s.n.], 2021. p. 7513–7517.

WANG, X.; CHEN, W.; WU, J.; WANG, Y.-F.; WANG, W. Y. Video captioning via hierarchical reinforcement learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. [S.l.: s.n.], 2018. p. 4213–4222.

XIONG, Y.; DAI, B.; LIN, D. Move forward and tell: A progressive generator of video descriptions. In: ECCV. [S.l.: s.n.], 2018. p. 468–483.

XIONG, Y. et al. CUHK & ETHZ & SIAT submission to activitynet challenge 2016. arXiv preprint arXiv:1608.00797, 2016.

XU, K. et al. Show, attend and tell: Neural image caption generation with visual attention. In: PMLR. ICML. [S.l.], 2015. p. 2048–2057.

YAMAZAKI, K. et al. Vlcap: Vision-language with contrastive learning for coherent video paragraph captioning. In: IEEE. 2022 IEEE International Conference on Image Processing (ICIP). [S.l.], 2022. p. 3656–3661.

YAMAZAKI, K.; VO, K.; TRUONG, Q. S.; RAJ, B.; LE, N. Vltint: Visual-linguistic transformer-in-transformer for coherent video paragraph captioning. In: Proceedings of the AAAI Conference on Artificial intelligence. [S.l.: s.n.], 2023. v. 37, n. 3, p. 3081–3090.

YATES, A.; NOGUEIRA, R.; LIN, J. Pretrained transformers for text ranking: Bert and beyond. In: 14th ACM International Conference on Web Search and Data Mining. [S.l.: s.n.], 2021. p. 1154–1156.

YOUTUBE. YouTube em números. 2020. Disponível em: <https://www.youtube.com/about/press>. Acesso em: 04 mai. 2020.

YOUTUBE. Uma plataforma onde a criatividade impulsiona o crescimento: Uma análise sobre o impacto do YouTube no Brasil em 2024. 2024. Disponível em: <https://services.google.com/fh/files/misc/brasil_impact_report.pdf>. Acesso em: 12 mai. 2026.

ZANG, S.-S.; YU, H.; SONG, Y.; ZENG, R. Unsupervised video summarization using deep non-local video summarization networks. Neurocomputing, Elsevier, v. 519, p. 26–35, 2023.

ZHANG, A.; LIPTON, Z. C.; LI, M.; SMOLA, A. J. Dive into deep learning. arXiv preprint arXiv:2106.11342, 2021.

ZHANG, B.; HU, H.; SHA, F. Cross-modal and hierarchical modeling of video and text. In: ECCV. [S.l.: s.n.], 2018. p. 374–390.

ZHANG, X.; WEI, F.; ZHOU, M. HIBERT: Document level pre-training of hierarchical bidirectional transformers for document summarization. In: 57th Annual Meeting of the ACL. [S.l.: s.n.], 2019. p. 5059–5069.

ZHANG, Y.; LIU, Y.; KANG, W.; TAO, R. Vss-net: Visual semantic self-mining network for video summarization. IEEE Transactions on Circuits and Systems for Video Technology, IEEE, v. 34, n. 4, p. 2775–2788, 2023.

ZHAO, B.; GONG, M.; LI, X. Hierarchical multimodal transformer to summarize videos. Neurocomputing, Elsevier, v. 468, p. 360–369, 2022.

ZHAO, B.; LI, H.; LU, X.; LI, X. Reconstructive sequence-graph network for video summarization. IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE, v. 44, n. 5, p. 2793–2801, 2021.

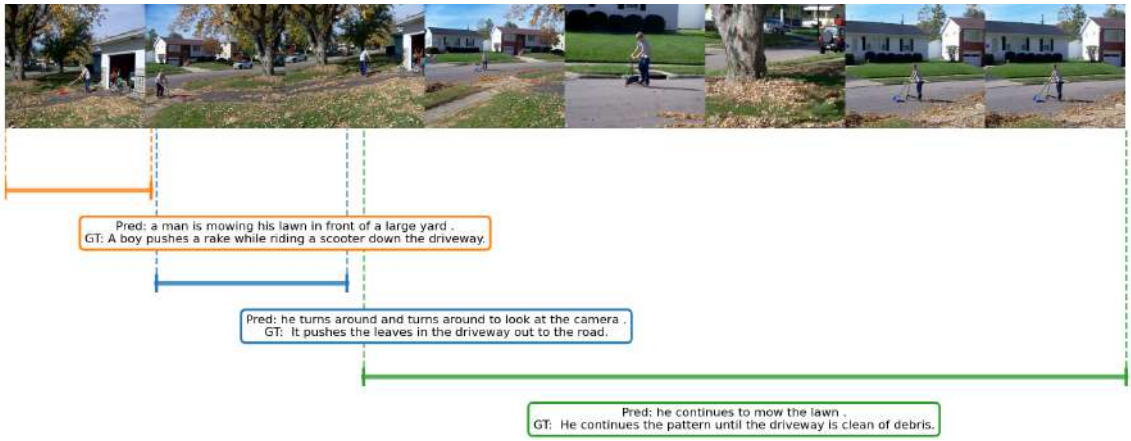
ZHOU, L.; KALANTIDIS, Y.; CHEN, X.; CORSO, J. J.; ROHRBACH, M. Grounded video description. In: IEEE CVPR. [S.l.: s.n.], 2019. p. 6578–6587.

ZHOU, L.; ZHOU, Y.; CORSO, J. J.; SOCHER, R.; XIONG, C. End-to-end dense video captioning with masked transformer. In: IEEE CVPR. [S.l.: s.n.], 2018. p. 8739–8748.

ZHU, W.; LU, J.; HAN, Y.; ZHOU, J. Learning multiscale hierarchical attention for video summarization. Pattern Recognition, Elsevier, v. 122, p. 108312, 2022.

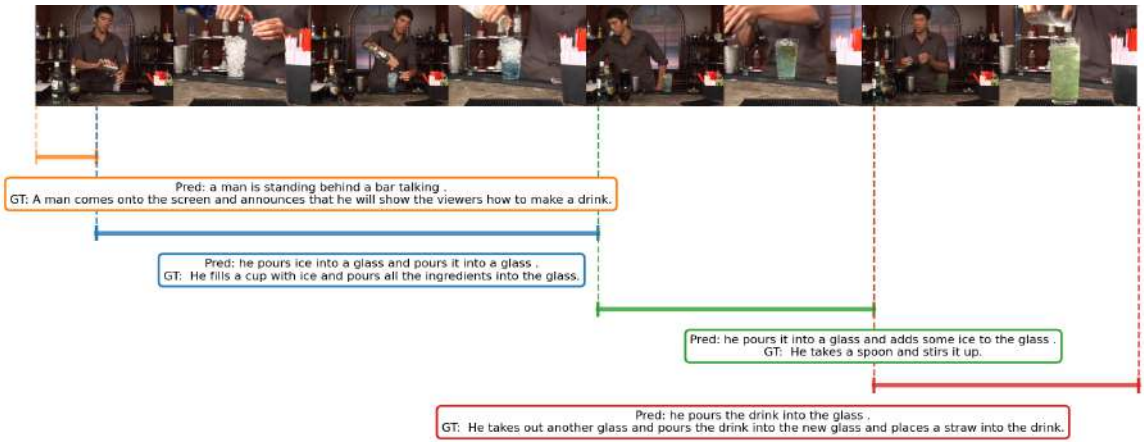
APPENDIX A – ADDITIONAL QUALITATIVE RESULTS

Figure 45 – Comparison between event-predicted results and ground truth for the video v_90vop6PS2Y0 of the ActivityNet dataset.



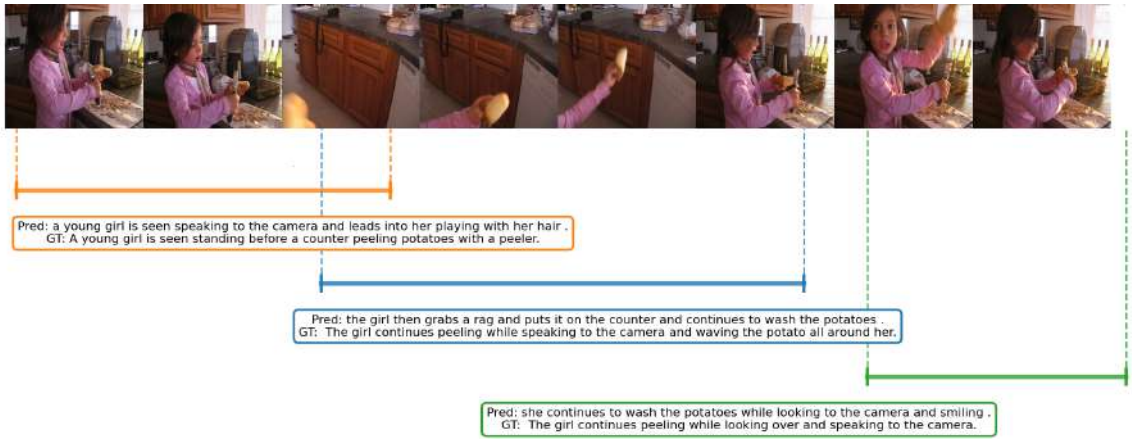
Source: Elaborated by the author

Figure 46 – Comparison between event-predicted results and ground truth for the video v_7NG6UrY2Foo of the ActivityNet dataset.



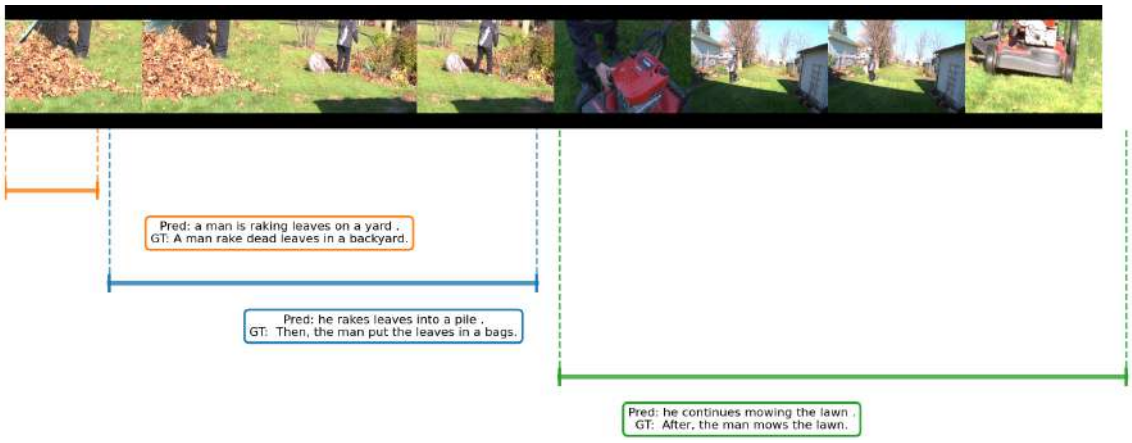
Source: Elaborated by the author

Figure 47 – Comparison between event-predicted results and ground truth for the video v_57buK1yvKPk of the ActivityNet dataset.



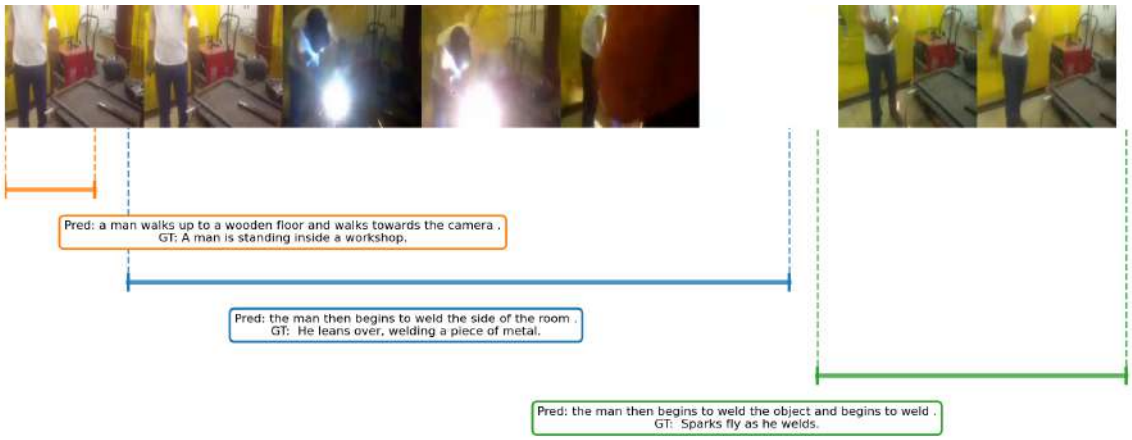
Source: Elaborated by the author

Figure 48 – Comparison between event-predicted results and ground truth for the video v_2Sev8z4P7pE of the ActivityNet dataset.



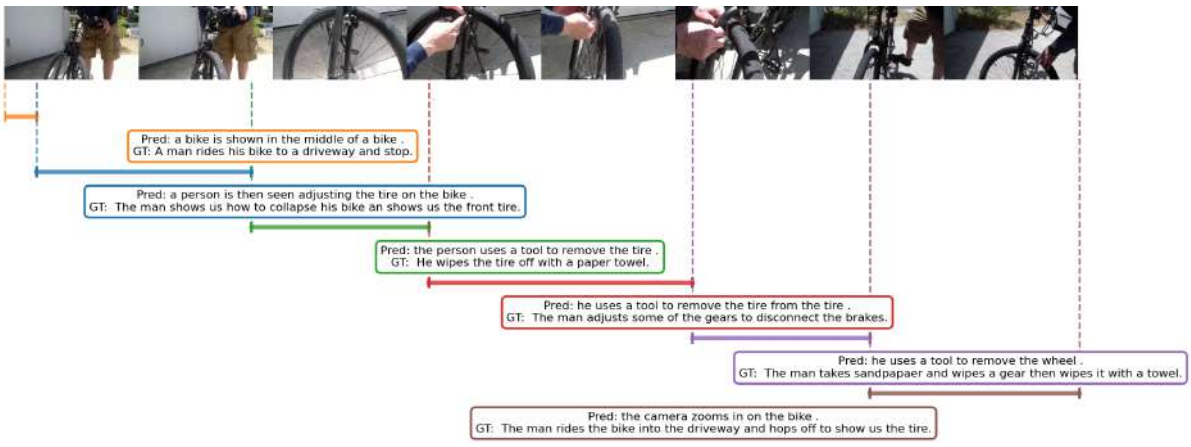
Source: Elaborated by the author

Figure 49 – Comparison between event-predicted results and ground truth for the video v_2VTEseqA5SA of the ActivityNet dataset.



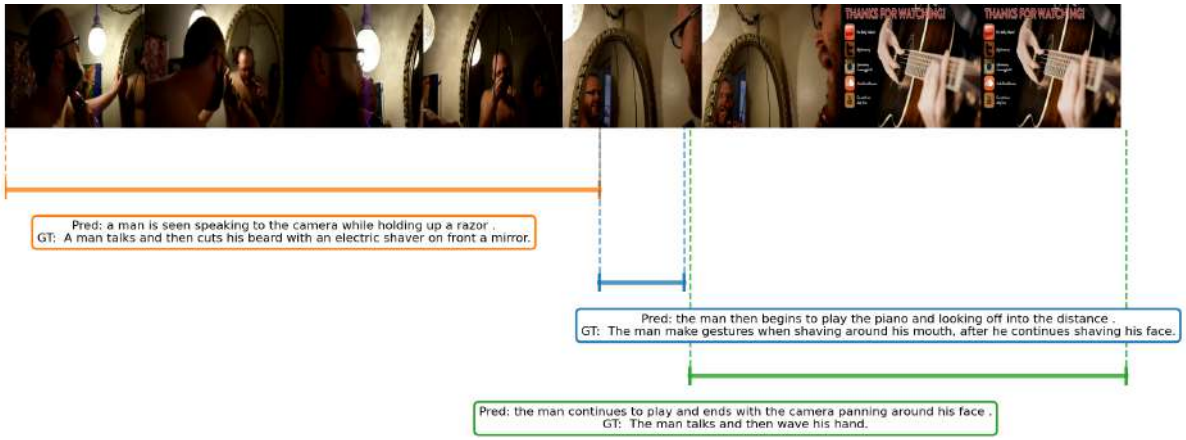
Source: Elaborated by the author

Figure 50 – Comparison between event-predicted results and ground truth for the video v_am4Z43QlUrg of the ActivityNet dataset.



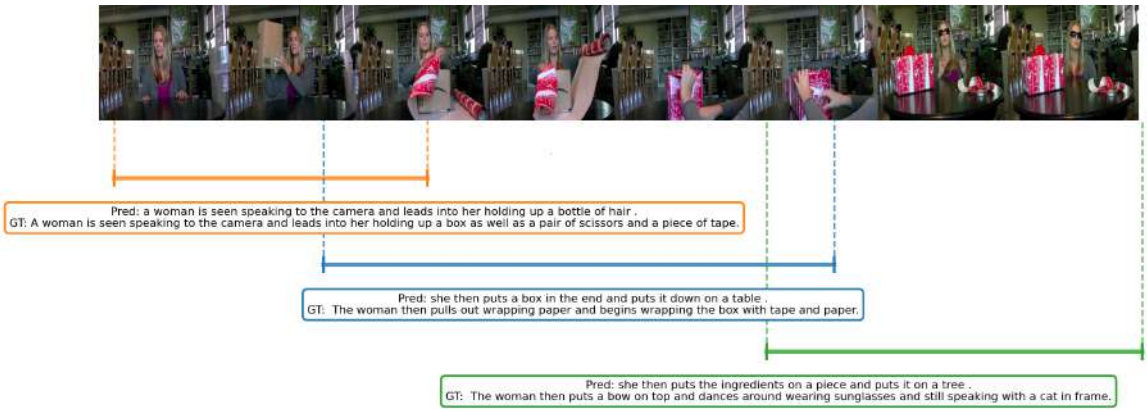
Source: Elaborated by the author

Figure 51 – Comparison between event-predicted results and ground truth for the video v_ChH3zlLeWug of the ActivityNet dataset.



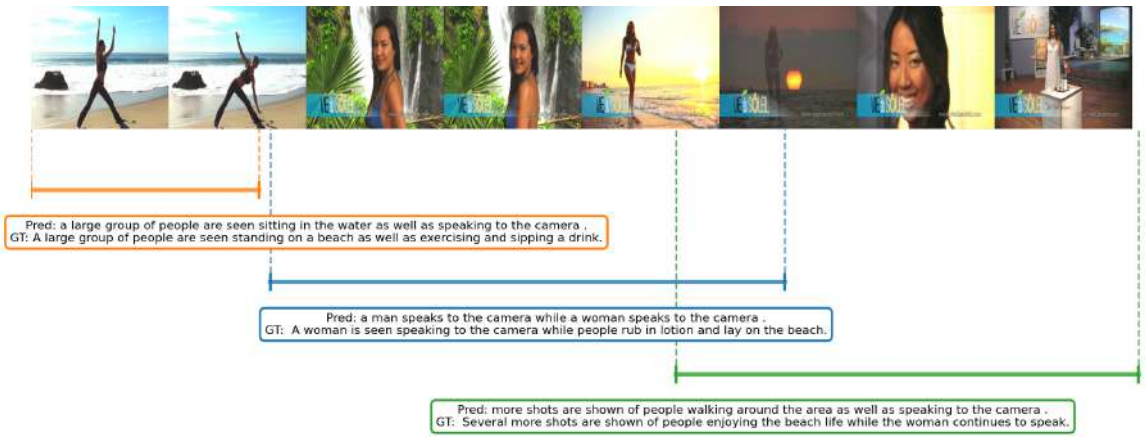
Source: Elaborated by the author

Figure 52 – Comparison between event-predicted results and ground truth for the video v_DTWZhe352y8 of the ActivityNet dataset.



Source: Elaborated by the author

Figure 53 – Comparison between event-predicted results and ground truth for the video v_IQ4SUx8ythk of the ActivityNet dataset.

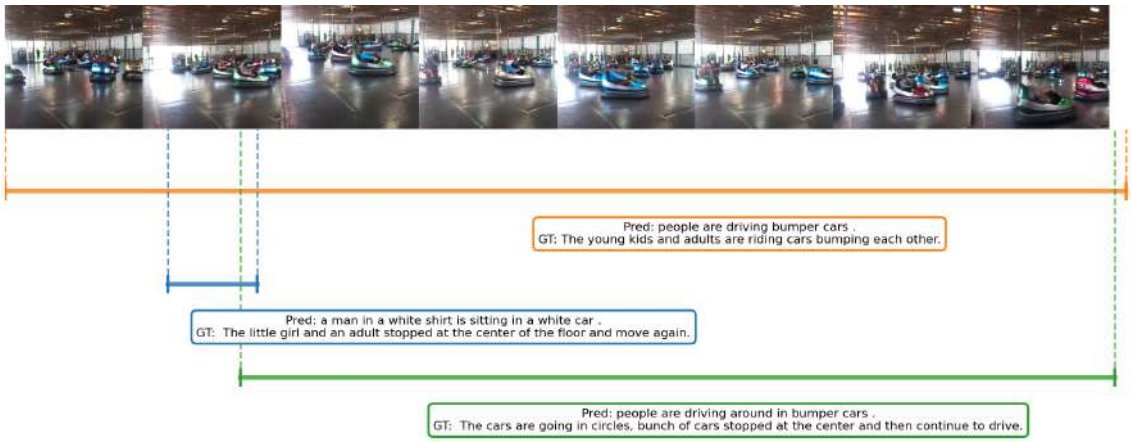


Source: Elaborated by the author

Figure 54 – Comparison between event-predicted results and ground truth for the video v_kkIClKG5xY8 of the ActivityNet dataset.

Source: Elaborated by the author

Figure 55 – Comparison between event-predicted results and ground truth for the video v_NDK0XQnsnmA of the ActivityNet dataset.



Source: Elaborated by the author

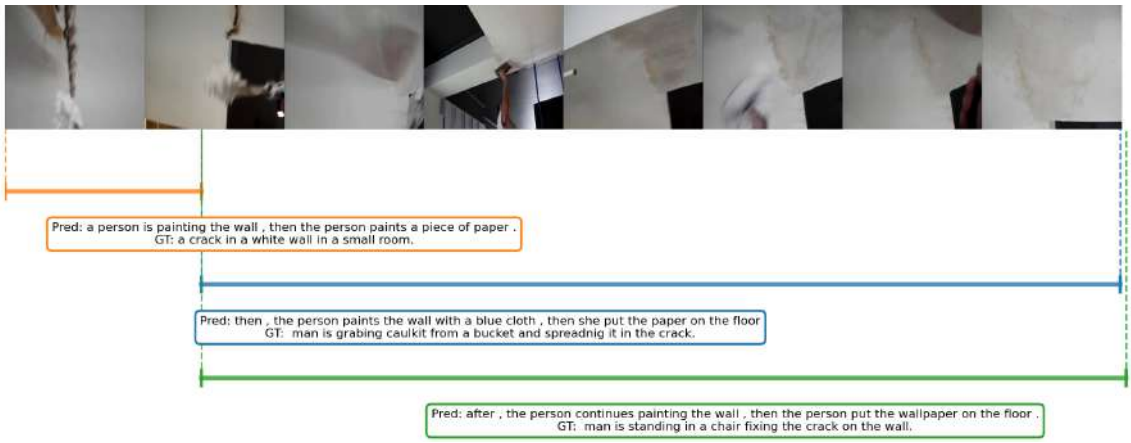
Figure 56 – Comparison between event-predicted results and ground truth for the video v_oA8ZUG1y4Lc of the ActivityNet dataset.

Source: Elaborated by the author

Figure 57 – Comparison between event-predicted results and ground truth for the video v_oobYvNJU5ko of the ActivityNet dataset.

Source: Elaborated by the author

Figure 58 – Comparison between event-predicted results and ground truth for the video v_oW0G_C86fz0 of the ActivityNet dataset.

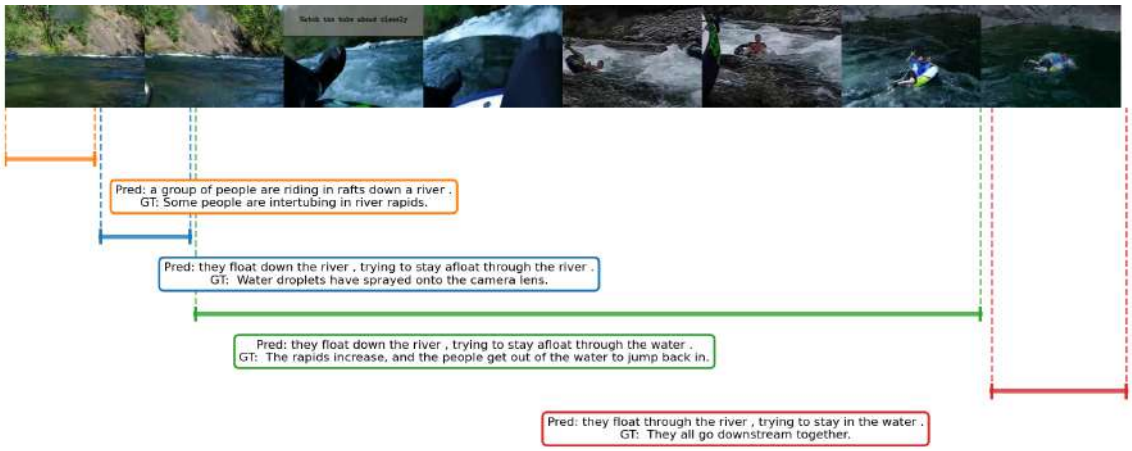


Source: Elaborated by the author

Figure 59 – Comparison between event-predicted results and ground truth for the video v_PUJYZEq8H64 of the ActivityNet dataset.

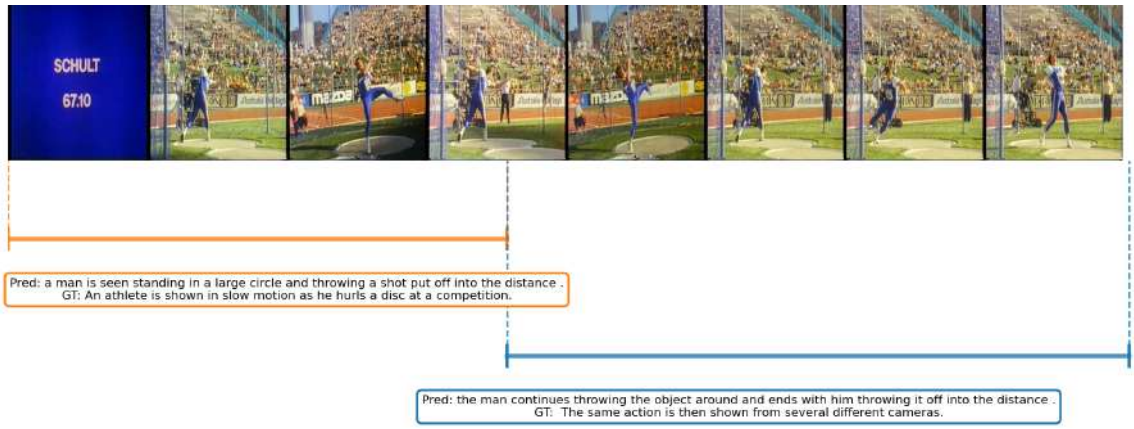
Source: Elaborated by the author

Figure 60 – Comparison between event-predicted results and ground truth for the video v_u-X4YO91V78 of the ActivityNet dataset.



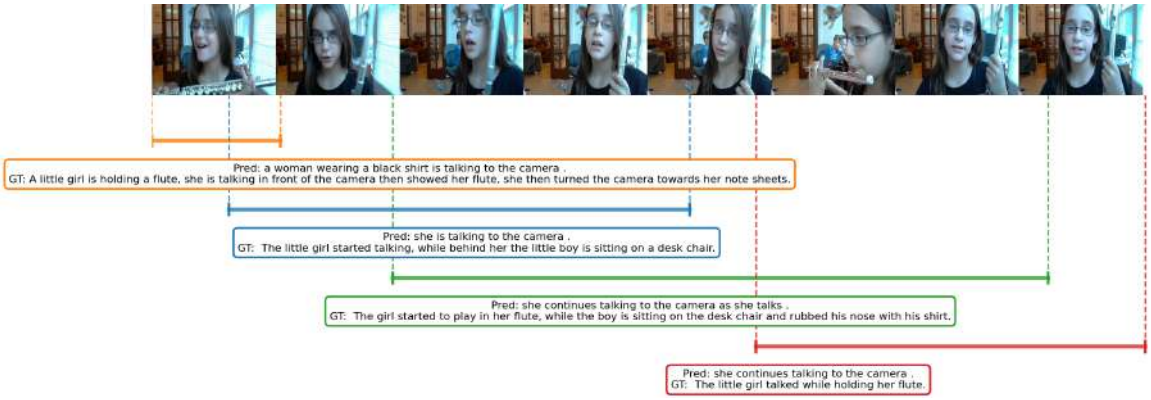
Source: Elaborated by the author

Figure 61 – Comparison between event-predicted results and ground truth for the video v_vrwJEvpeHyM of the ActivityNet dataset.



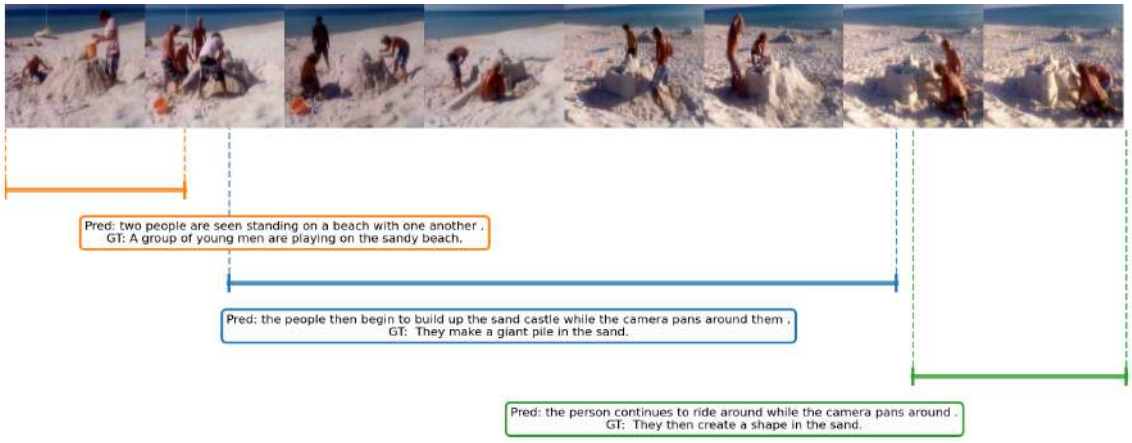
Source: Elaborated by the author

Figure 62 – Comparison between event-predicted results and ground truth for the video v_WGEKoGRIJGk of the ActivityNet dataset.



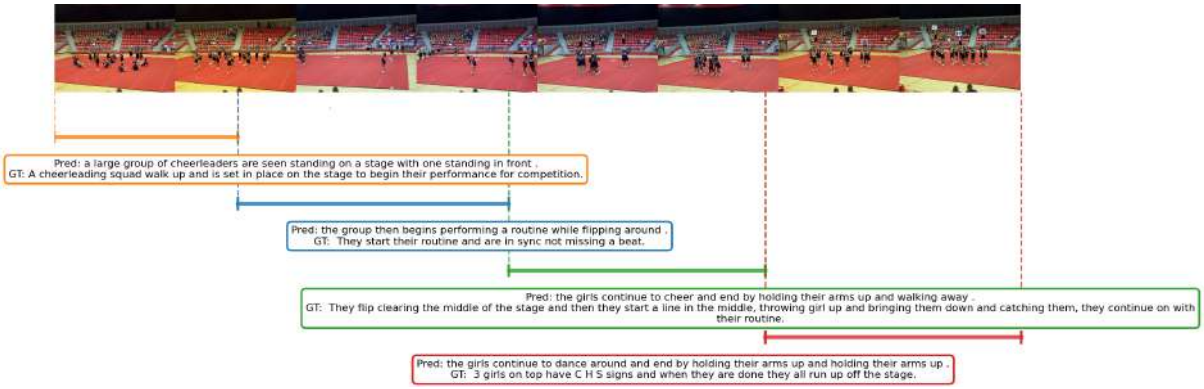
Source: Elaborated by the author

Figure 63 – Comparison between event-predicted results and ground truth for the video v_5asz3rt3QyQ of the ActivityNet dataset.



Source: Elaborated by the author

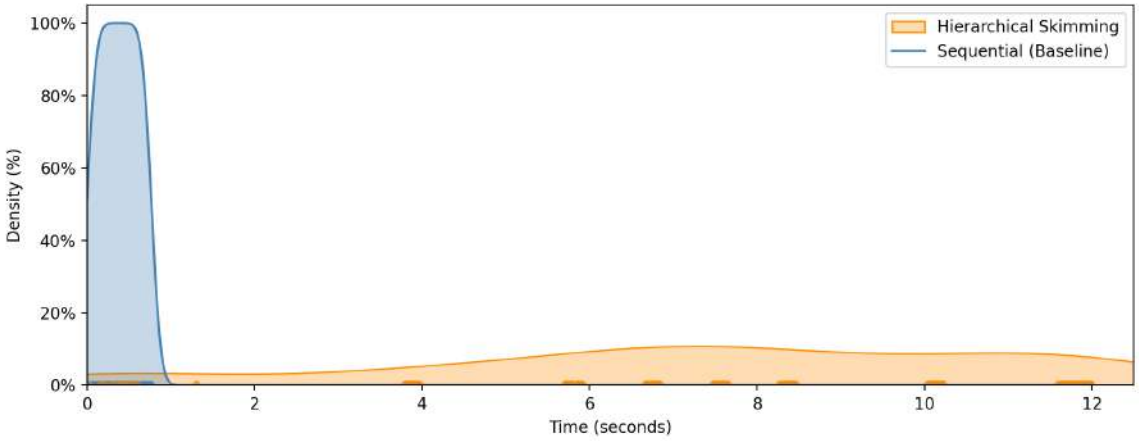
Figure 64 – Comparison between event-predicted results and ground truth for the video v_jRXF5_vNUWE of the ActivityNet dataset.



Source: Elaborated by the author

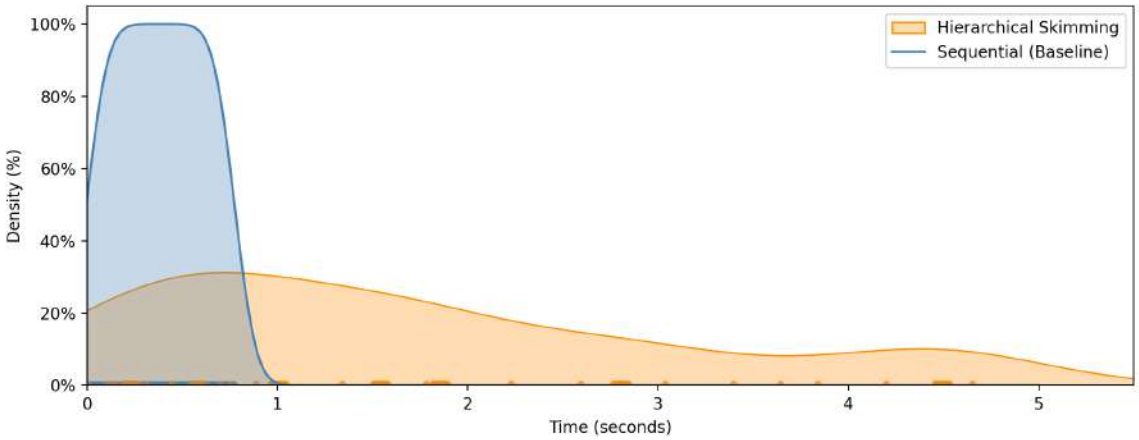
APPENDIX B – ADDITIONAL FRAME DISTRIBUTION DATA WITH KERNEL DENSITY ESTIMATION

Figure 65 – Estimated frame density distribution using Kernel Density Estimation (KDE) for the video v_H-5nHSHwFOk of the ActivityNet dataset.



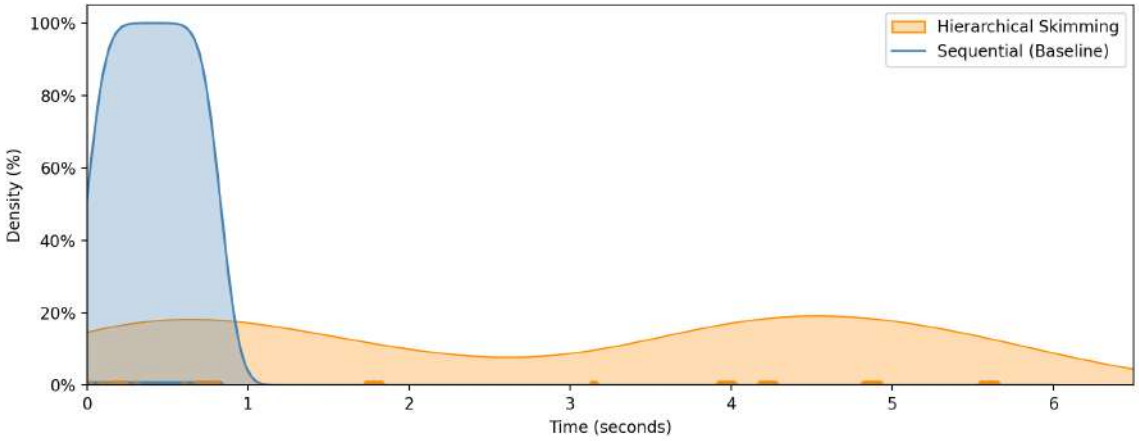
Source: Elaborated by the author

Figure 66 – Estimated frame density distribution using Kernel Density Estimation (KDE) for the video v_L67RSiR2X78 of the ActivityNet dataset.



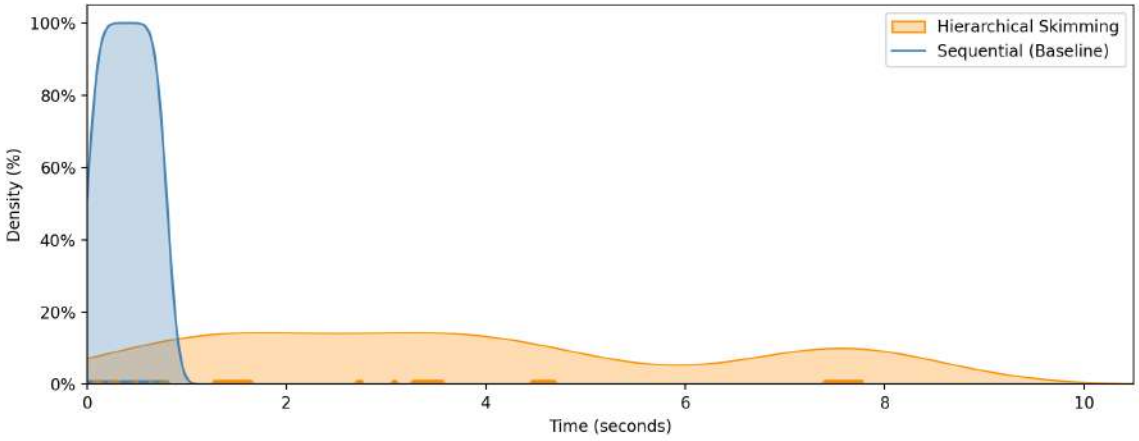
Source: Elaborated by the author

Figure 67 – Estimated frame density distribution using Kernel Density Estimation (KDE) for the video v_lPCl1ZYH2xI of the ActivityNet dataset.



Source: Elaborated by the author

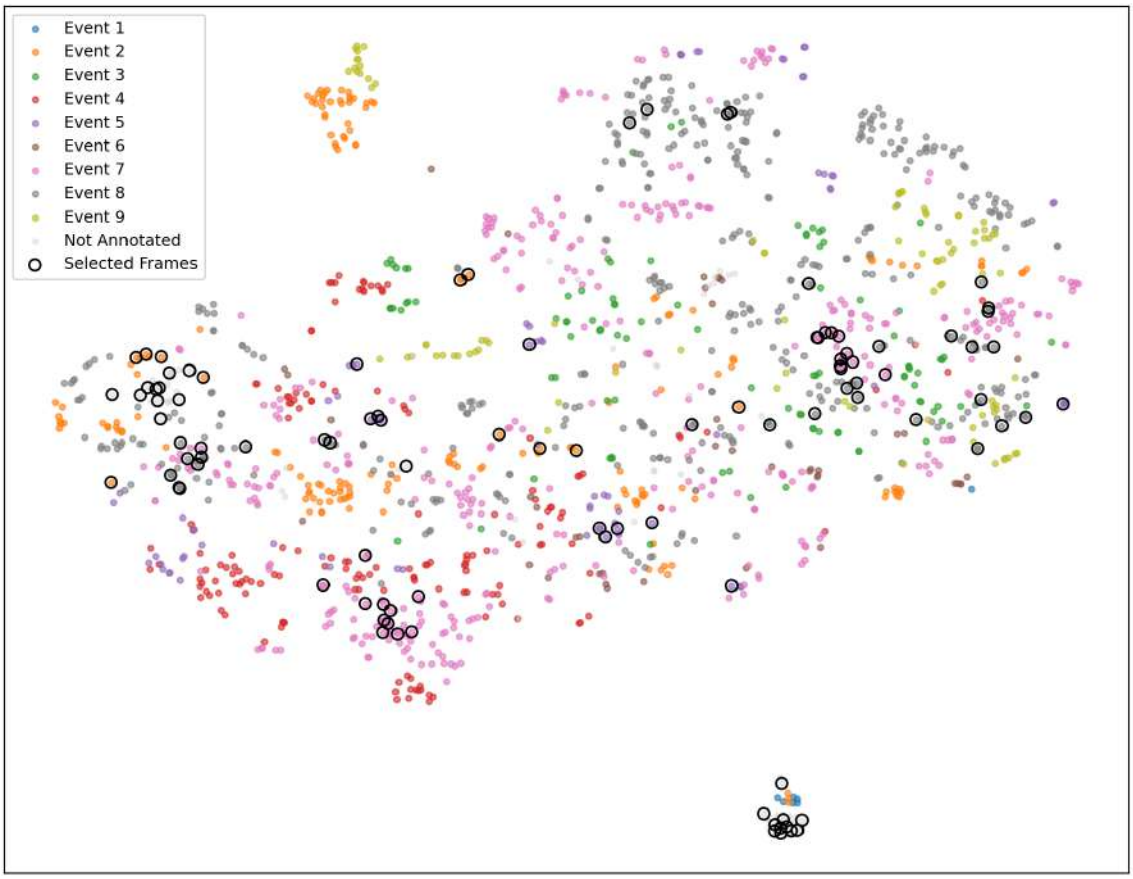
Figure 68 – Estimated frame density distribution using Kernel Density Estimation (KDE) for the video v_pev7rvOE8eM of the ActivityNet dataset.



Source: Elaborated by the author

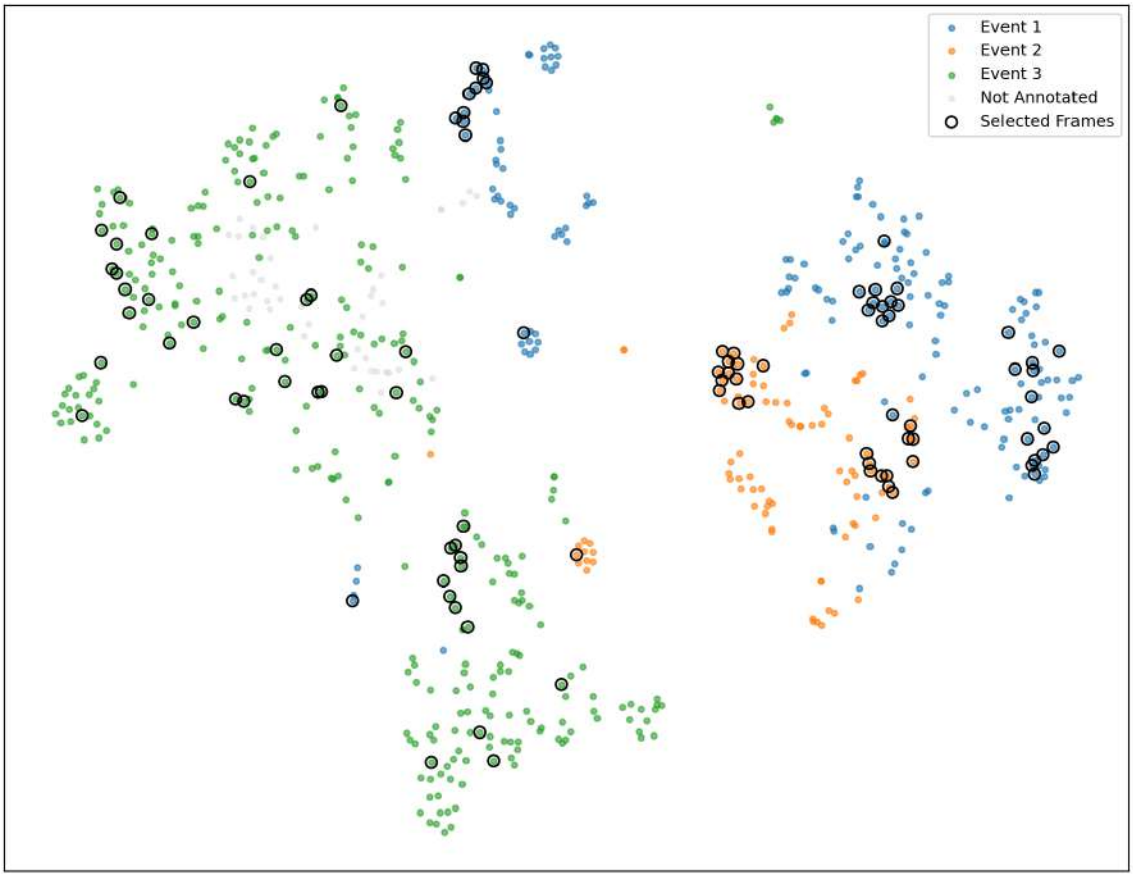
APPENDIX C – ADDITIONAL RESULTS OF T-SNE VISUALIZATION OF FRAME-LEVEL FEATURE EMBEDDINGS

Figure 69 – t-SNE visualization of frame-level feature embeddings for the video v_H-5nHSHwFOk of the ActivityNet dataset.



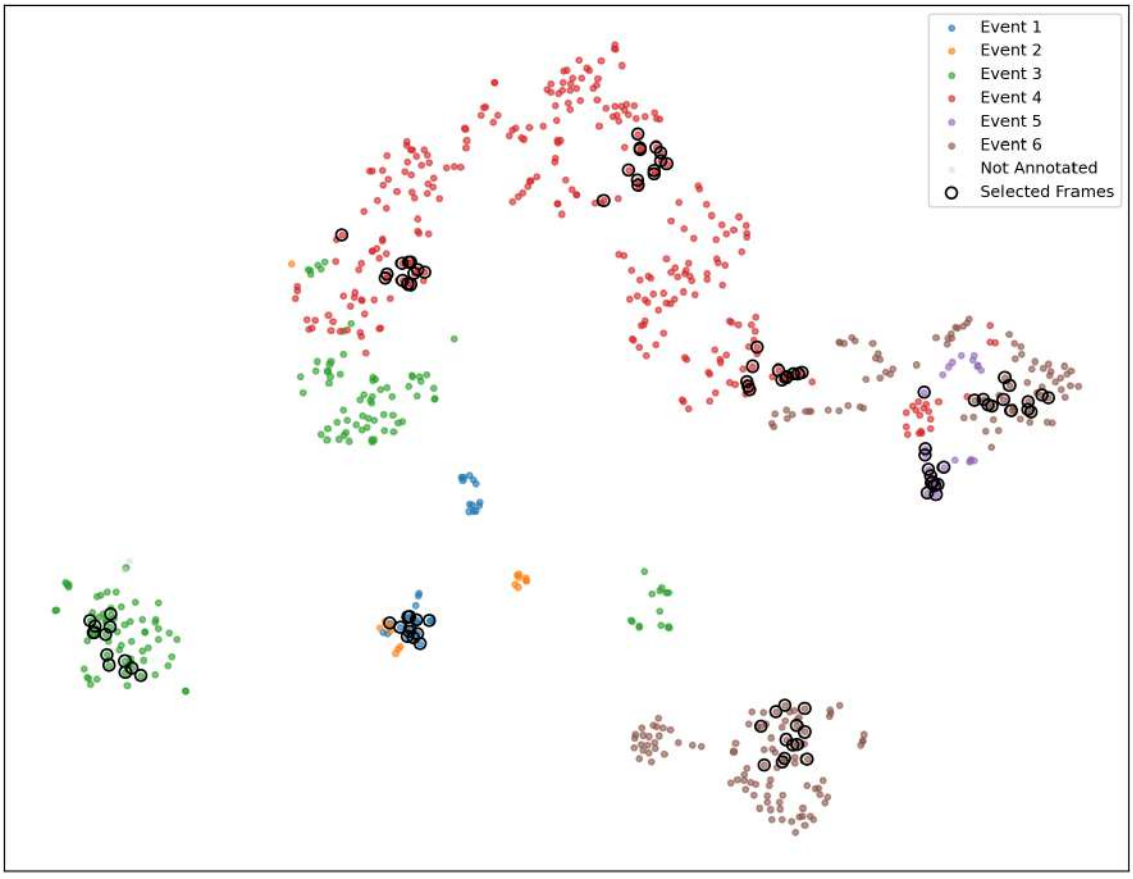
Source: Elaborated by the author

Figure 70 – t-SNE visualization of frame-level feature embeddings for the video v_L67RSiR2X78 of the ActivityNet dataset.



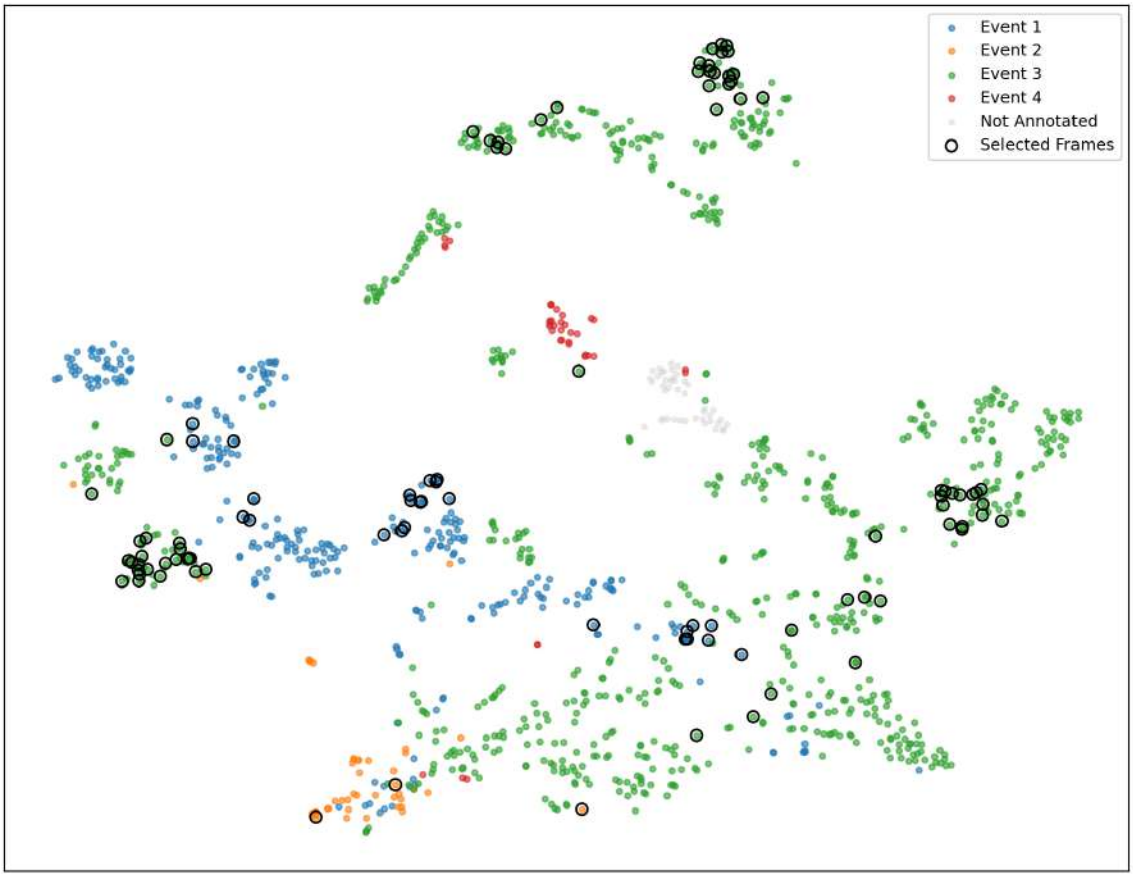
Source: Elaborated by the author

Figure 71 – t-SNE visualization of frame-level feature embeddings for the video v_lPCl1ZYH2xI of the ActivityNet dataset.



Source: Elaborated by the author

Figure 72 – t-SNE visualization of frame-level feature embeddings for the video v_pev7rvOE8eM of the ActivityNet dataset.



Source: Elaborated by the author