UNIVERSIDAD POLITÉCNICA DE MADRID
Escuela Técnica Superior de Ingenieros de Telecomunicación

Automatic Sports Video Summarization with Identity-Aware Highlight Selection

DOCTORAL THESIS

Submitted for the degree of Doctor by:

Marcos Rodrigo Talavera
Master of Science in Telecommunication Engineering

Madrid, 2026

UNIVERSIDAD POLITÉCNICA DE MADRID
Escuela Técnica Superior de Ingenieros de Telecomunicación

Doctoral Degree in Communication Technologies and Systems

Automatic Sports Video Summarization with Identity-Aware Highlight Selection

D O C T O R A L T H E S I S

Submitted for the degree of Doctor by:

Marcos Rodrigo Talavera
Master of Science in Telecommunication Engineering

Under the supervision of:
Dr. Carlos Cuevas Rodríguez
Dr. Narciso García Santos

Madrid, 2026

Title: Automatic Sports Video Summarization with Identity-Aware Highlight Selection
Author: Marcos Rodrigo Talavera
Doctoral Programme: Communication Technologies and Systems

Thesis Supervision:

Dr. Carlos Cuevas Rodríguez, Professor at Universidad Politécnica de Madrid

Dr. Narciso García Santos, Professor at Universidad Politécnica de Madrid

External Reviewers:

Thesis Defense Committee:

Thesis Defense Date:

Agradezco a mis padres y a mi hermana, por estar siempre ahí y por animarme en cada etapa de la tesis.

A Icíar, por su paciencia infinita, por soportar mis verborreas sobre papers y experimentos, y por estar a mi lado durante todo este proceso.

A mis directores, Carlos y Narciso, por revisar siempre con tanta rapidez y cuidado cada borrador que les he enviado.

Al GTI, por todo lo que he aprendido a vuestro lado y por convertir el trabajo diario en un lugar de crecimiento y complicidad.

A mis amigos, por su interés sincero y, sobre todo, por regalarme esas vías de escape que han hecho este camino mucho más liviano.

Y a los que faltan en esta lista, gracias igualmente; si no está vuestro nombre, es porque esto es una tesis, no los créditos de una película.

Abstract

In the modern era, recorded video has grown dramatically, with sports broadcasts, streaming platforms, and social media generating far more footage than any viewer can watch in full. This explosion creates a need for automatic methods that can distill long recordings into concise, informative summaries tailored to different audiences. This thesis addresses the problem of automatically generating sports highlights that are both narratively meaningful and personalized to specific athletes, by combining advances in video summarization and face recognition within a unified framework.

The first strand focuses on automatic sports highlight detection. In the challenging domain of martial-arts tricking, a motion-centric pipeline based on optical-flow descriptors, temporal smoothing and duration-aware selection is introduced, together with the MATDAT dataset and its evaluation protocol. This classical approach achieves very high frame-level accuracy when domain structure is well understood and annotations are scarce. To move beyond a single sport, the thesis then introduces SportCLIP, a text-guided, zero-shot framework that reframes highlight detection as cross-modal retrieval between video frames and short text descriptions of what counts as a highlight. Building on vision–language embeddings, SportCLIP can be prompted with new sports and event types without any additional training. It is evaluated on two new resources, the SportCLIP and Olympic Highlights datasets, showing that promptable, text-guided summarization can approach the performance of tailored motion models while remaining training-free, sport-agnostic, and easily adaptable to different user needs.

The second strand studies face recognition under realistic broadcast conditions, where athletes appear at varying distances, under frequent occlusions, and often only for a few frames. To capture these challenges in a controlled way, the thesis introduces the UPM-GTI-Face dataset as a dedicated benchmark that explicitly factors distance and occlusion, together with an end-to-end recognition baseline tailored to this setting. Building on this resource, a comprehensive comparison between convolutional neural networks and vision transformers as backbone families is carried out, revealing how architectural choices, input resolution, and training strategies interact to determine robustness to degraded faces, long-range shots, and other nuisances common in sports footage but rarely isolated.

The final strand integrates these components into a Personalized Video Summarization (PVS) system that produces identity-aware highlight reels. The system is modular: it can operate with either SportCLIP, a text-guided vision–language model, or QD-DETR, a Transformer-based query-driven detector, as interchangeable summarizers, and with either ArcFace (CNN-based) or TransFace (Transformer-based) as plug-and-play face-recognition backbones. Within this unified pipeline, highlight segments are first selected by the chosen summarizer and then assigned to the most likely athlete according to the selected recognition model. Experiments on the Olympic Highlights dataset show that this design can deliver high-quality, identity-aware sports summaries while exposing clear trade-offs between classical motion features, vision–language models, Transformer-based detectors, and competing face-recognition architectures. The thesis contributes new datasets, models, and evaluation protocols that clarify these trade-offs and provide a solid basis for future work on personalized sports video summarization.

Resumen

En la era actual, el volumen de vídeo registrado ha crecido de forma drástica, y las retransmisiones deportivas, las plataformas de streaming y las redes sociales generan mucho más material del que cualquier espectador puede ver. Esta explosión de contenido crea la necesidad de métodos automáticos capaces de condensar grabaciones largas en resúmenes concisos e informativos, adaptados a distintos públicos. Esta tesis aborda el problema de generar automáticamente resúmenes de jugadas destacadas que sean narrativamente significativos y personalizados a atletas concretos, combinando técnicas de resumen de vídeo y reconocimiento facial en un marco unificado.

La primera línea de trabajo se centra en la detección automática de jugadas destacadas. En el exigente dominio del *martial-arts tricking*, se propone un enfoque clásico centrado en el movimiento, basado en descriptores de flujo óptico, suavizado temporal y selección sensible a la duración, junto con la base de datos MATDAT y su protocolo de evaluación. Este enfoque alcanza una precisión muy alta a nivel de fotograma cuando la estructura del dominio está bien caracterizada y las anotaciones son escasas. Para ir más allá de un único deporte, la tesis introduce SportCLIP, un marco guiado por texto y de tipo *zero-shot* que reformula la detección como recuperación multimodal entre fotogramas y breves descripciones de lo que se considera destacado. Basado en representaciones visión–lenguaje, SportCLIP puede emplearse con nuevos deportes y tipos de eventos sin entrenamiento adicional. Se evalúa sobre las bases de datos SportCLIP y Olympic Highlights, mostrando que un resumen guiado por texto puede aproximar el rendimiento de modelos de movimiento específicos, manteniéndose independiente del deporte y fácilmente adaptable a distintas necesidades.

La segunda línea estudia el reconocimiento facial en condiciones realistas de emisión, donde los atletas aparecen a distintas distancias, bajo oclusiones frecuentes y, a menudo, solo durante unos pocos fotogramas. Para capturar estos retos de forma controlada, la tesis introduce la base de datos UPM-GTI-Face como banco de pruebas que factoriza distancia y oclusión, junto con un sistema de reconocimiento extremo a extremo adaptado a este escenario. A partir de este recurso se realiza una comparación exhaustiva entre redes neuronales convolucionales y *Vision Transformer* como familias de *backbones*, analizando cómo las decisiones arquitectónicas, la resolución de entrada y las estrategias de entrenamiento afectan a la robustez frente a rostros degradados, planos lejanos y otras dificultades típicas del vídeo deportivo.

La última línea integra estos componentes en un sistema de *Personalized Video Summarization* (PVS) que genera resúmenes de jugadas con identificación de los atletas. El sistema es modular: puede operar con SportCLIP o con QD-DETR como resumidores intercambiables, y con ArcFace (basado en CNN) o TransFace (basado en *Transformers*) como *backbones* de reconocimiento facial. Dentro de este flujo de procesamiento unificado, los segmentos destacados se seleccionan con el resumidor elegido y después se asignan al atleta más probable según el modelo de reconocimiento. Los experimentos sobre Olympic Highlights muestran que este diseño produce resúmenes deportivos de alta calidad, con identificación explícita de los protagonistas, y pone de manifiesto los compromisos entre las distintas familias de modelos empleadas. La tesis aporta nuevas bases de datos, modelos y protocolos de evaluación que sirven de base para futuras investigaciones en resumen de vídeo deportivo personalizado.

Table of Contents

List of Figures

List of Tables

Abbreviations and acronyms

Adam	Adaptive Moment Estimation
AP	Average Precision
AUC	Area Under the ROC Curve
CASIA	CASIA-WebFace Dataset
CCTV	Closed-Circuit Television
CLIP	Contrastive Language–Image Pretraining
CNN	Convolutional Neural Network
CPU	Central Processing Unit
DETR	Detection Transformer
DL-VHD	Dual-Learner Video Highlight Detection
DNN	Deep Neural Network
E2E	End-to-End
EER	Equal Error Rate
FN	False Negative
FNR	False Negative Rate
FP	False Positive
FPR	False Positive Rate
FPS	Frames Per Second
GPU	Graphics Processing Unit
HL	Highlight Label
IMFW	Indian Masked Faces in the Wild Dataset
IoU	Intersection over Union
IR	Infrared
LBP	Local Binary Pattern

LFW Labeled Faces in the Wild Dataset

LLM Large Language Model

LSTM Long Short-Term Memory

MATDAT Martial Arts Tricking Dataset

mAP Mean Average Precision

NHL Non-Highlight Label

Olympic Highlights Olympic Highlights Dataset

PCA Principal Component Analysis

PVS Personalized Video Summarization

QD-DETR Query-Dependent Detection Transformer

RGB Red-Green-Blue Color Space

ROC Receiver Operating Characteristic

ROF Real World Occluded Faces Dataset

RNN Recurrent Neural Network

SCface Surveillance Cameras Face Dataset

SCRFD Strongly Constrained Realtime Face Detection

SIFT Scale-Invariant Feature Transform

SoccerNet SoccerNet Dataset

SPOT Structure Preserving Object Tracker

SportCLIP SportCLIP method and dataset

TP True Positive

TPR True Positive Rate

UN Uncertainty Label

UPM Universidad Politécnica de Madrid

UPM-GTI-Face UPM-GTI-Face Dataset

UTK-LRHM University of Tennessee Knoxville Long Range and High Magnification Dataset

VGGFace2 VGGFace2 Dataset

ViT Vision Transformer

VLM Vision-Language Model

WIDER WIDER FACE Dataset

YOLO You Only Look Once

YouTube Highlights YouTube Highlights Dataset

Chapter 1

Introduction

Digital video has become the dominant medium for recording, broadcasting, and consuming events. This growth is especially evident in sports, where professional broadcasts, training sessions, and user-generated content coexist on the same platforms. A single match can generate hours of footage from multiple viewpoints, yet audiences typically engage with only a small fraction of this material. In practice, viewers gravitate toward short highlight packages that concentrate the most informative and emotionally salient moments, while discarding the large amount of routine or redundant content. Producing these summaries, however, still relies heavily on manual editing or bespoke, sport-specific tools, and the resulting outputs are usually generic: the same recap is delivered to all viewers, regardless of their individual preferences or their interest in particular teams or players.

This gap between the abundance of raw video and the limited, homogeneous nature of available summaries motivates the central problem addressed in this thesis: the automatic generation of sports highlights that are both narratively meaningful and personalized. Beyond identifying *what* are the most important actions of a match and *where* did they take place, we seek to extend this to the *who* by enabling the system to isolate the key moments in which a specified player is involved. Achieving this requires bringing together two traditionally separate research strands. On one side, *video summarization* methods must identify which temporal segments are worth preserving. On the other, *face recognition* systems must determine who appears in those segments, under realistic conditions of distance, occlusion, and variable image quality. The overarching goal of the thesis is to design, analyze, and integrate these components into a coherent pipeline for personalized video summarization.

From a methodological perspective, this leads to a composite system in which an upstream summarization module proposes candidate highlight segments, and a downstream face-analysis module filters those segments based on the presence of a target identity. The summarization component is responsible for constructing a relevance signal over time and converting it into a set of temporally coherent clips that preserve the key dynamics of the underlying sport. The face-analysis component detects and crops faces, builds robust descriptors, and compares them to reference images of the user or the desired subject. A high-level overview of this dual-stream pipeline is shown in Figure 1.1. The figure emphasizes the flow from long-form input footage to a compact, identity-conditioned highlight reel, decomposed into a video

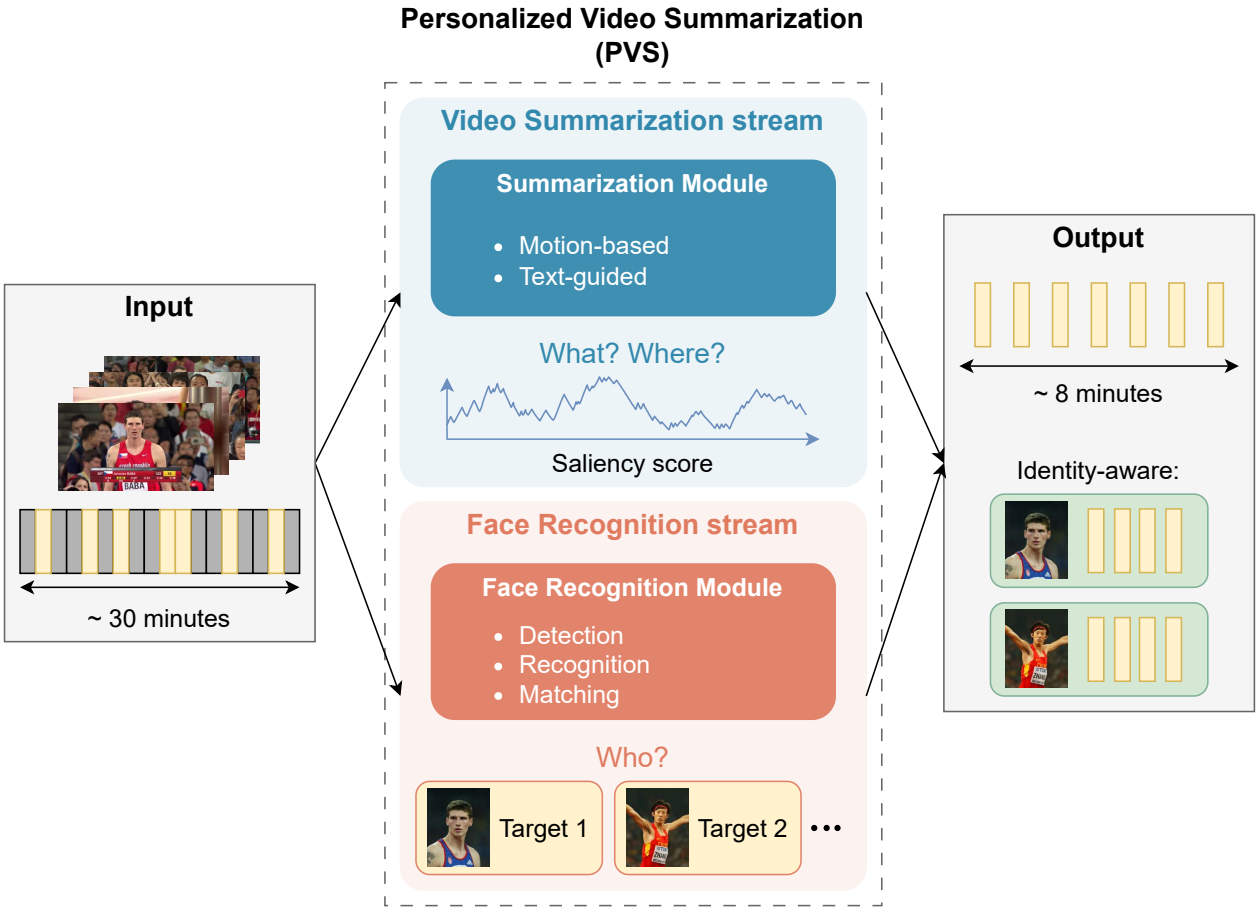


Figure 1.1: High-level overview of the Personalized Video Summarization (PVS) system proposed in this thesis. The video summarization stream estimates a saliency score over time and selects candidate highlight segments (motion-based or text-guided), while the face recognition stream detects, recognizes, and matches one or more target identities within those segments. The final output is a short, identity-conditioned highlight reel tailored to specific players.

summarization stream that answers *what* and *where*, and a face recognition stream that answers *who*. This composition is studied in detail in Chapter 3, where it is formalized as a modular pipeline with clear interfaces between components.

Focusing on this pipeline exposes several challenges on both sides of the problem. In video summarization, sports footage exhibits long, mostly uneventful intervals punctuated by short, high-impact events whose visual appearance depends on the sport, the camera positions, and the broadcast style. Annotated data is often scarce, especially at frame or event level, and existing highlight detectors trained on generic web videos may not transfer cleanly to structured sports scenarios. Moreover, users increasingly expect flexible ways of specifying what constitutes a highlight, ranging from high-level concepts ("attacking plays", "counterattacks") to fine-grained actions ("three-point shots from the corner"). These requirements call for methods that can exploit classical motion cues when supervision is limited, while also leveraging modern deep architectures and vision–language models when

richer information is available.

In the face recognition strand of the thesis, the difficulties are of a different nature. In sports footage, players typically appear with low-resolution faces, non-frontal viewpoints, partial occlusions, and substantial variation in capture distance. Public benchmarks typically emphasize either large-scale diversity, unconstrained conditions, occlusions, or surveillance imagery, but seldom the combined effect of distance and occlusion that is common in real deployments such as sports broadcasts. Furthermore, the rapid evolution from Convolutional Neural Networks (CNNs) to Transformer-based architectures raises practical questions about how architectural choices, loss functions, and training strategies affect recognition robustness under such stressors. Understanding these trade-offs is crucial when selecting or designing the recognition backbones that will operate within a personalized summarization system.

Within this overall objective, the thesis develops three tightly connected strands: (i) automatic highlight detection in long, untrimmed sports videos; (ii) robust face recognition under variations in distance, occlusions, and image quality; and (iii) their integration into a modular Personalized Video Summarization (PVS) system that produces identity-conditioned highlight reels. These strands structure the methodological core presented in Chapter 3 and the empirical analyses reported in the Chapter 4.

On the summarization side, the work progresses from classical, motion-based pipelines to modern architectures that incorporate long-range temporal reasoning and text guidance. A first contribution is a motion-centric framework for automatic highlight detection in videos of martial arts tricking, together with the MATDAT dataset and an evaluation protocol tailored to sparse events in long, untrimmed videos [84]. This framework establishes core concepts—such as temporal smoothing, and relevance modeling of events—that will be reused throughout the thesis.

Building on these foundations, the thesis then explores how vision–language models can serve as a flexible interface for specifying what to summarize. A CLIP-based framework for text-guided sports highlights [86] is proposed to retrieve highlightable segments directly from broadcast footage using natural language queries. The method combines contrastive vision–language embeddings with prompt engineering, negative queries, and temporal calibration to produce robust relevance signals across multiple sports. Crucially, these signals are fed into the same selection machinery introduced in the classical setting, yielding a unified view in which the scoring source may be motion, visual semantics, language, or a combination thereof.

On the face recognition side, the thesis introduces UPM-GTI-Face [87], a dedicated dataset designed to isolate and quantify the impact of distance and mask-type occlusions on detection and recognition performance. The dataset provides controlled captures at multiple distances and mask configurations, enabling systematic experiments that would be difficult to conduct with purely in-the-wild data. Alongside the dataset, a baseline face analysis pipeline is defined, encompassing detection, alignment, and embedding extraction, and serving as the starting point for subsequent comparisons and improvements.

Complementing this dataset contribution, the thesis presents a comprehensive comparison between CNN and Transformer backbones for face recognition [85]. Under a common training and evaluation protocol, multiple architectures are assessed on a suite of benchmarks that

include large-scale web collections, occlusion-focused datasets, surveillance imagery, and the newly introduced UPM-GTI-Face. This analysis clarifies how architectural differences—local receptive fields versus global self-attention—translate into embedding quality, robustness to distance and occlusions, and computational cost. The resulting insights inform the choice of recognition backbones within the personalized summarization pipeline.

The third strand of the thesis is the design and implementation of the PVS system that composes the preceding components into an end-to-end application. The system receives raw sports footage, applies a chosen summarization module (classical, deep, or text-guided) to generate candidate highlights, and then filters and re-ranks them according to the presence of a target identity using the selected face recognition model. The pipeline includes practical components such as face tracking, temporal aggregation of detections, and simple analytics to characterize the involvement of the target player across the summary. Its design is deliberately modular, so that new summarization or recognition models can be integrated with minimal changes, and its behavior can be analyzed through controlled ablations. This system is described in detail in the personalized summarization section of Chapter 3 and serves as the vehicle through which the thesis goal is realized.

Beyond these individual contributions, an important cross-cutting theme of the thesis is the separation between scoring and selection. Whether the relevance signal arises from motion statistics, deep video features, or vision–language embeddings, a common post-processing stage converts saliency curves into discrete highlight clips. This "last mile" is reused across methods, ensuring that advancements in representation learning or query formulation translate into tangible improvements in the final summaries. A similar principle governs the face analysis and personalization stages, where detection, embedding, and matching are specified independently but connected through standardized interfaces.

The remainder of the thesis is organized as follows. Chapter 2 reviews the relevant literature on video summarization and face recognition, with particular emphasis on sports applications, vision–language models, and robustness to occlusions and distance. Chapter 3 presents the methodological core of the work: classical and deep architectures for highlight detection; the text-guided summarization frameworks; the UPM-GTI-Face dataset and the associated recognition pipeline; the comparative study of CNN and Transformer backbones; and the final PVS system. Chapter 4 reports the experimental results obtained on the proposed and public datasets, analyzing performance at both frame and event level for summarization, and across multiple benchmarks for face recognition. Chapter 5 interprets these results in a broader context, highlighting the strengths and limitations of the proposed methods and drawing connections between the video summarization and face recognition strands. Finally, Chapter 6 summarizes the main findings of the thesis, discusses the main open challenges and future research directions, and reflects on how they collectively advance the goal of personalized video summarization and robust face analysis. All datasets and implementations developed in the context of this thesis are publicly at https://www.gti.ssr.upm.es/data.

Chapter 2

State of the art

The work developed in this thesis builds on two main technical pillars: automatic video summarization, with a particular focus on highlight detection in sports, and robust face recognition under challenging conditions such as distance and occlusions. This chapter reviews the most relevant literature in both areas, emphasizing methods and datasets that are most closely related to the contributions presented in Chapters 3 and 4.

2.1 Video Summarization

The exponential growth of online video content has made automatic video summarization a key research topic across multiple domains, from personal media and news to surveillance and sports. In general terms, video summarization aims to condense a long video into a shorter representation—either as a set of keyframes, a sequence of shots, or a highlight reel—that preserves the most informative and appealing parts of the content. Early survey works and more recent reviews provide a broad taxonomy of approaches, covering supervised, unsupervised, and weakly supervised methods, as well as their applications in consumer and professional settings [2], [49], [52], [68], [104].

Within this general landscape, sports video summarization has emerged as a particularly demanding subproblem. Sports broadcasts are long, densely packed with events, and subject to strong domain conventions. At the same time, user-generated sports videos—recorded with handheld cameras or mobile phones—lack any standardized structure or editing rules. The remainder of this section focuses on summarization methods for sports videos, with an emphasis on those most relevant to the highlight detection pipelines developed in this thesis.

2.1.1 Classical approaches and broadcast-oriented methods

A first generation of sports video summarization methods relied heavily on handcrafted features and domain-specific heuristics. These approaches are especially common in *broadcast* sports, where camera work, replays, and on-screen graphics encode high-level editorial semantics that can be exploited by automatic systems. Comprehensive surveys of broadcast sports summarization highlight the use of cues such as shot size, camera motion, replay detection, or

scoreboard overlays to infer important events in sports like soccer, basketball, and cricket [13], [90], [109].

Several classical systems explicitly target broadcast soccer games. Ekin *et al.* [21] proposed one of the earliest automatic soccer summarization pipelines, which detects semantic elements such as the playing field, goal posts, and scoreboards from carefully designed color and geometry features. The system leverages the constrained set of canonical camera angles used in edited matches to detect events like goals or attacks. Pan *et al.* [76] focused on the detection of slow-motion replays, a strong editorial indicator of an important moment in many sports broadcasts. More recently, Yan *et al.* [120] combined YOLO-v3 and OpenPose in a three-level framework to clip sports video streams by detecting players and their poses, before categorizing segments as highlights using heuristic rules.

These methods demonstrated that broadcast-specific cues can achieve high precision in well-structured scenarios. However, they also exhibit clear limitations: they are inherently sport-specific, brittle to changes in production style, and difficult to transfer to user-generated or niche sports videos where there is no standardized camera work, no replays, and no overlay graphics that can be reliably exploited.

2.1.2 User-generated sports and niche disciplines

In contrast to broadcast footage, user-generated sports videos—for example those recorded by athletes or spectators and uploaded to social media platforms—do not follow fixed editing conventions. They often consist of long, untrimmed recordings with significant redundancy, diverse viewpoints, and unstructured camera motion. Summarization strategies in this setting must rely primarily on the visual and, when available, audio content itself.

Early user-generated approaches exploit low-level redundancy cues such as color or motion. Lienhart [61] proposed a dynamic summarization technique that selects keyframes based on color histograms and motion activity to reduce redundancy in home videos. Meng *et al.* [69] extended this idea by using object-centric features to select keyframes based on the presence of significant objects, improving the semantic relevance of the summary.

More recent methods operate closer to the final goal of highlight detection. Tejero *et al.* [103] used deep action recognition features to summarize user-generated sports videos, leveraging pre-trained CNNs to capture high-level motion patterns. Applying this specifically to martial arts (Kendo), the authors proposed a deep neural network-based pipeline to extract two types of action-related features—body joint-based and holistic—to classify video segments as highlights. This work already points toward the difficulties of summarizing sports with complex body motions and non-standard movement patterns.

Other niche or aesthetic sports have also been addressed. Inter-frame similarity measures and keyframe selection have been applied to gymnastics and figure skating [34], while CNN-based pose estimators have been used to identify trampoline skills from a single camera [12]. In rhythmic gymnastics, Marjana *et al.* [111] proposed an end-to-end CNN that directly controls camera motion to follow athletes and capture highlight segments. In taekwondo, Kong *et al.* [55] employed a Structure Preserving Object Tracker (SPOT) to keep the target athlete

in view, feeding those frames into a deep PCANet classifier, followed by an SVM to detect relevant technique sequences. Performance assessment systems have been developed for pommel horse gymnastics [83] and Olympic rhythmic gymnastics [18], using depth sensors and spatiotemporal templates respectively. More recently, the AGF-Olympics dataset [126] introduced challenging artistic gymnastics routines and a discriminative attention module to disentangle complex motion patterns for performance analysis.

Overall, prior work has largely focused on sports with well-established competition formats and clearer scoring schemes. *Tricking*, the focal sport of the first study in this thesis, poses additional challenges: extremely fast acrobatic movements, spontaneous and unstructured sequences, and frequent changes of direction that do not follow predefined routines. As argued in [84], methods designed for broadcast cues, depth cameras, or handcrafted rules do not transfer well to this domain, motivating the development of new, content-driven highlight detection strategies.

2.1.3 Deep learning-based summarization and highlight detection

The advent of deep learning has substantially transformed video summarization. Instead of relying on handcrafted features and rigid heuristics, modern methods learn to predict frame or segment importance directly from data. Comprehensive surveys [2], [52], [68] discuss trends in supervised, unsupervised, and reinforcement learning-based approaches, as well as architectures that jointly consider spatial and temporal information.

In supervised settings, CNNs and recurrent neural networks (RNNs)—long short-term memory networks (LSTMs)— have been widely used to estimate importance scores. In [103], deep action recognition features extracted by CNNs are fed into temporal models to generate summaries of user-generated sports videos. Xu *et al.* [119] proposed a cross-category highlight detection strategy that operates in domains where annotated videos are scarce, showing that suitable feature representations and temporal modeling can generalize across related sports. Attention mechanisms further refine frame selection: Badamdorj *et al.* [3] introduced a bimodal attention network that jointly analyzes audio and visual cues to detect highlight events, demonstrating the benefit of multi-modal modeling in sports broadcasts.

Graph-based and encoder–decoder architectures have also been explored. Wei *et al.* [116] combined 3D CNNs and visual saliency in an encoder–decoder framework to detect interesting segments in sports footage, explicitly learning pixel-level distinctions that correlate with highlightness. Pan *et al.* [77] studied the balance between global diversity and local temporal context in supervised summarization, showing that explicitly optimizing for both leads to more stable frame-level importance predictions. From a computational perspective, Issa *et al.* [50] proposed compressed-domain summarization methods that exploit bitstream features and CNNs to reduce the cost of content-based analysis, which is particularly relevant for consumer devices and embedded systems.

Unsupervised and self-supervised learning strategies alleviate the need for dense annotations. Badamdorj *et al.* [4] introduced a contrastive learning framework for unsupervised highlight detection, in which a video is encoded into a representation where highlight clips help discriminate it from other videos. Li *et al.* [59] proposed a method that augments features via

intra-modality encoding and cross-modality co-occurrence encoding, combined with a hard-pairs guided contrastive loss to improve fine-grained feature discrimination. More recent works explore temporal attention schemes for unsupervised summarization, aesthetic-driven frame selection, and audio–visual contrastive objectives [47], [60], [110], although these methods are not always evaluated on sports content.

Complementary directions include optimization-based summarization frameworks aimed at consumer scenarios. Ghatak *et al.* [27] proposed HSAJAYA, a hybrid Harmony Search–Jaya algorithm that produces collision-free synopses of home-surveillance footage. Priyadharshini and Mahapatra [80] introduced MOHASA, an approach for summarizing 360° videos by jointly optimizing spatial coverage and temporal continuity. While not sports-specific, these works illustrate how multi-objective optimization and heuristic search can be combined with machine-learned scoring functions to generate compact and informative summaries.

2.1.4 Vision–language models for highlight detection

A more recent and rapidly growing line of work uses *vision–language models* (VLMs) to guide video summarization. These models learn a joint embedding space for images (or video frames) and text, enabling semantic matching between visual content and natural-language descriptions. Surveys on VLMs and cross-modal retrieval [37], [108], [127] highlight their role in tasks such as image–text retrieval, video browsing, and interactive search.

In highlight detection, VLMs can be used to encode both video segments and textual queries or descriptions. Badamdorj *et al.* [4] already illustrate how contrastive learning between video clips can be leveraged to identify interesting moments without explicit labels. He *et al.* [39] proposed a dual contrastive loss to better align video and text embeddings, leading to more coherent language-guided summaries.

Query-focused summarization methods build on these ideas to provide user control. Wu *et al.* [118] introduced IntentVizor, which allows users to steer the summarization process with multi-modal queries (e.g. text plus example frames), producing summaries that reflect user intent. Moon *et al.* [72] proposed a query-based highlight detection framework that retrieves segments aligned with a given textual query. Yoon *et al.* [125] explored prototype-based representations for content-based video retrieval, which can serve as a retrieval backbone in VLM-guided summarization systems.

Li *et al.* [57] demonstrated progressive refinement of video summaries by iteratively updating video–text representations, improving summary quality over multiple passes. In the sports domain, Han *et al.* [36] proposed an automatic soccer editing system based on multimodal learning, aligning audio and visual signals with high-level event semantics, while Mujtaba *et al.* [73] addressed client-driven personalization on resource-constrained devices.

Beyond summarization, VLMs have also appeared in consumer imaging and text-conditioned video generation. Zhang *et al.* [128] used CLIP-style embeddings to fuse infrared and visible images under natural-language guidance, and Kim *et al.* [54] proposed an early text-to-video generation framework. These works reinforce the idea that language-guided semantics can drive temporal visual content, an idea that this thesis exploits for multi-sport highlight

detection.

The SportCLIP framework introduced in this thesis (see Chapter 3) falls squarely within this vision–language paradigm: it leverages a CLIP-based backbone, automatically constructs and filters candidate textual prompts, and evaluates highlight detection performance across multiple sports datasets with frame-level annotations, bridging the gap between generic VLM capabilities and the specific structure of sports events.

2.1.5 Discussion and relation to this thesis

In summary, classical sports summarization methods demonstrate the usefulness of broadcast-specific cues but do not transfer well to unstructured or user-generated videos. Deep learning approaches have improved the semantic modeling of sports content, yet many still rely on large annotated datasets or remain tied to a single sport or style of production. Vision–language models offer a promising avenue for generalizing highlight detection across sports by decoupling the description of events from the underlying visual domain, but practical VLM-based systems often lack detailed, multi-sport evaluations and reproducible pipelines.

The video summarization work in this thesis is positioned at the intersection of these trends. On one hand, it introduces a content-driven highlight detector tailored to a niche, highly dynamic sport (martial arts tricking) where classical cues fail. On the other hand, it develops a CLIP-based framework for language-guided sports highlights that is systematically evaluated across multiple datasets and sports. Together, these contributions aim to move from sport-specific heuristics toward flexible, semantically grounded summarization methods that can serve as the highlight detection backbone for the personalized video summarization system described at the end of Chapter 3.

2.2 Face Recognition

Face recognition is the second core component of this thesis. While video summarization identifies *what* happens and *when*, face recognition addresses the complementary question of *who* is present. The field has evolved from early holistic methods to deep convolutional networks, and more recently to vision transformers and hybrid architectures. At the same time, the operating conditions of interest—including long distances, low resolution, and face masks—have become increasingly challenging.

This section reviews the evolution of face recognition methods, key datasets for training and evaluation, and recent comparative studies between CNNs and ViTs, with a particular focus on scenarios that combine distance and occlusion, as studied in this thesis.

2.2.1 From classical methods to deep CNN-based recognition

Early face recognition systems were based on linear subspace methods and local descriptors. Pioneering approaches such as Eigenfaces and Fisherfaces represented faces as projections onto low-dimensional spaces learned via principal component analysis (PCA) or linear discriminant analysis (LDA), followed by simple classifiers or nearest-neighbor matching. In parallel, local

feature descriptors, including Gabor-filter responses and Local Binary Patterns (LBP), were used to capture texture information robust to modest changes in illumination and expression. These methods achieved good performance in controlled scenarios but degraded markedly under large pose, illumination, or occlusion variations [102].

The introduction of deep learning, and CNNs in particular, marked a major turning point. Inspired by the success of CNNs on ImageNet, Taigman *et al.* proposed *DeepFace*, one of the first deep architectures to close the gap between automatic and human performance on the Labeled Faces in the Wild (LFW) benchmark [100]. Around the same time, the DeepID family [97] showed that multiple CNNs, trained on different face patches, could be combined to achieve very high recognition accuracy. These works established deep CNNs as the dominant paradigm for face recognition.

Subsequent advances focused on both architecture and loss design. Generic CNN backbones such as VGG, Inception, ResNet, MobileNet, and EfficientNet were adapted and specialized for face recognition [8], [11], [65], [121], supported by increasingly large training datasets (see Section 2.2.2). FaceNet introduced a triplet-loss framework that directly optimizes an embedding space for face verification and clustering. Later, angular-margin losses such as SphereFace, CosFace, and particularly ArcFace [7], [46] imposed geometric constraints on the embedding, significantly enhancing inter-class separability while keeping intra-class variance low. Surveys [20], [48], [114] provide detailed taxonomies of these architectures and loss functions. Among these methods, ArcFace has become a de facto standard backbone and margin-based loss for deep face recognition, and it is adopted in this thesis as a representative strong CNN-based recognizer in the subsequent experimental chapters.

For deployment on resource-constrained devices, lightweight CNNs such as MobileFaceNets [11] or ShuffleFaceNet [65] use depthwise separable convolutions, bottleneck layers, and pruning techniques to greatly reduce memory and computational cost with minimal accuracy loss. These models are particularly relevant for real-time applications, including mobile authentication or embedded video surveillance, where full-size backbones may be impractical.

2.2.2 Datasets for large-scale and challenging face recognition

The rapid progress of deep face recognition has been tightly coupled with the availability of large and diverse datasets. Early benchmarks such as *Labeled Faces in the Wild* (LFW) [46] provided unconstrained, web-collected images of celebrities, setting a standard protocol for face verification. However, LFW's limited size and its near-saturation by modern methods mean that it is now primarily used for sanity checking new models.

Large-scale training datasets followed. MS-Celeb-1M [32] introduced millions of images for hundreds of thousands of identities, enabling the training of very deep models but also raising concerns about label noise and privacy. VGGFace2 [7] was specifically designed to balance breadth (number of identities) and depth (images per identity), with strong variation in pose, age, and ethnicity. It quickly became a standard training and evaluation resource for deep face recognition, and is extensively used in this thesis.

In parallel, several datasets were created to stress-test specific challenges:

- **In-the-wild recognition and detection:** LFW [46] and YouTube Faces DB [117] evaluate recognition under unconstrained imaging conditions and video sequences, respectively. WIDER FACE [122] focuses on face detection with extreme variation in pose, occlusion, and scale, and has driven the development of modern detectors that are robust to tiny and heavily occluded faces. Context-aggregating models such as Tiny Faces [45], strong anchor-based baselines like TinaFace [134], and more recent lightweight architectures such as SCRFD [31] achieve state-of-the-art detection performance on WIDER FACE and related benchmarks. These families of detectors provide the building blocks for the face detection modules used in the end-to-end systems evaluated in Chapters 3 and 4.
- **Surveillance and distance:** SCface [30] provides images of 130 subjects captured by multiple surveillance cameras at three fixed distances, plus high-quality mugshots, making it a reference benchmark for face recognition at a distance. Other datasets, such as UTK-LRHM [123], the Remote Face Dataset [9], and QUIS-CAMPI [74], extend this idea to longer distances and outdoor surveillance settings, often including multi-modal traits (face, gait, iris) and cross-camera recordings.
- **Masked and occluded faces:** The COVID-19 pandemic stimulated the creation of datasets dedicated to masked face recognition. Indian Masked Faces in the Wild (IMFW) [71] collects masked/unmasked pairs for Indian subjects, while FaceMask [112] and ViDMASK [75] provide images and videos of people with and without masks in indoor and outdoor environments. These datasets target both mask detection and identity recognition under mask occlusion.

Despite this variety, most datasets focus on either distance or facial masks, but not both at once. The UPM-GTI-Face dataset [87], introduced and analyzed in this thesis, addresses this gap by jointly annotating distance and mask conditions for each subject. It is specifically designed to quantify the performance drop of detection and recognition systems when both challenges are present, reflecting realistic scenarios in modern surveillance.

2.2.3 CNN-based face recognition under challenging conditions

CNN-based face recognition has achieved near-perfect performance on unconstrained benchmarks such as LFW, but challenging conditions like large standoff distances and occlusions remain problematic. When faces are small, blurred, or captured by low-quality sensors, the discriminative power of deep features degrades, and the performance gap between laboratory and real-world conditions becomes apparent.

For long-range and low-resolution face recognition, several strategies have been explored. Yao *et al.* [123] showed that image enhancement and super-resolution significantly improve recognition on UTK-LRHM, where faces are captured at distances up to 300 meters. Similarly, many works on SCface combine deep CNNs with super-resolution or deblurring modules to boost performance on the hardest probe sets [102]. Li *et al.* [102] provide an overview of such techniques, illustrating that even modern embeddings require auxiliary processing to cope with very low-resolution faces.

Masked face recognition introduces a different failure mode: the lower part of the face, which carries important identity cues, is occluded. Off-the-shelf CNNs trained on unmasked faces experience a substantial drop in accuracy on masked faces. To address this, some authors have proposed training with synthetic masks overlaid on existing datasets (e.g. on VGGFace2), while others rely on real masked images from IMFW, FaceMask, or ViDMASK [71], [75], [112]. Strategies include focusing on periocular regions, designing occlusion-aware losses, or adding attention modules that suppress masked regions. Empirical studies have also compared the impact of different occlusions, finding that covering the eye region (e.g. sunglasses) can be even more disruptive than surgical masks, underscoring the importance of upper-face features.

The UPM-GTI-Face dataset [87], discussed in detail in Chapter 3, was used in this thesis to systematically evaluate CNNs under combined distance and mask conditions. The results confirm that classical backbones (VGG, ResNet, MobileNet, EfficientNet, Inception) suffer significant performance degradation as distance increases and when masks are present, despite achieving excellent accuracy on standard benchmarks. This motivates the exploration of alternative architectures, such as vision transformers, that may offer better robustness.

2.2.4 Vision transformers and hybrid models for face recognition

The introduction of Vision Transformers (ViTs) [19] has opened a new line of research in face recognition. ViTs dispense with convolutions and instead model images as sequences of patches processed by self-attention layers. Surveys on transformers in vision [35], [53] and comparative studies [5], [82] have shown that ViTs can match or surpass CNNs on generic image classification tasks, provided sufficient data and appropriate regularization.

In the context of face recognition, several works have adapted or extended ViTs. Zhong and Deng [131] proposed a *Face Transformer* architecture, demonstrating competitive performance on standard benchmarks. Sun and Tzimiropoulos [98] developed a part-based transformer that attends to facial regions, improving robustness under pose and occlusion changes. Hybrid models that combine convolutions and attention have also been proposed: George and Marcel [25] introduced EdgeFace, which fuses efficient CNN layers with transformer blocks to create a compact yet accurate model suitable for edge devices, while Li *et al.* [58] proposed MobileFaceFormer, a mobile-friendly network that interleaves MobileNet-style convolutions with lightweight transformer modules.

Beyond individual architectures, there is growing interest in systematic comparisons between CNNs and ViTs for face recognition. Rodrigo *et al.* [23], [67] review comparative analyses across different domains, while Tuli *et al.* [106] show that ViTs often exhibit higher shape bias and error patterns closer to human perception. George and Marcel [26] reported that ViT-based architectures outperform state-of-the-art CNNs in zero-shot anti-spoofing, suggesting superior generalization. Zhou *et al.* [132] studied the transferability of representations learned by CNNs and ViTs, concluding that ViTs often yield more robust features in downstream tasks.

Beyond generic ViT backbones, specialized transformer-based face recognition models have been introduced. A representative example is TransFace [14], which calibrates transformer training for face recognition from a unified perspective by jointly considering architectural

choices, data augmentation, and optimization strategies. TransFace achieves state-of-the-art verification accuracy on a range of benchmarks (including LFW, CFP-FP, AgeDB, CALFW, and CPLFW) while maintaining competitive model size and inference cost, making it particularly attractive for applications that must cope with large pose, quality, and occlusion variations. In this thesis, a TransFace-based model is adopted as the main transformer backbone in the personalized video summarization pipeline, complementing the CNN-based ArcFace baseline described above.

Building on this context, one of the studies compiled in this thesis carries out a comprehensive comparison between a ViT and several widely used CNNs (ResNet, VGG, Inception, MobileNet, EfficientNet) on multiple face recognition datasets, including VGGFace2, LFW, SCface, ROF, and UPM-GTI-Face. The results, detailed in Chapter 4, show that the ViT achieves higher verification and identification accuracy, particularly under occlusions and long-range conditions, with a favorable trade-off between accuracy, model size, and inference time. These findings support the idea that ViTs are a strong alternative to CNNs for face recognition in challenging real-world scenarios.

2.2.5 Discussion and relation to this thesis

The face recognition literature has progressed from handcrafted descriptors and holistic subspace methods to deep CNNs and, more recently, to ViTs and hybrid architectures. Large-scale datasets such as VGGFace2 and MS-Celeb-1M have enabled training highly accurate models, which now attain near-saturated performance on canonical benchmarks like LFW. However, specialized datasets focusing on surveillance, distance, and masks reveal that recognition under realistic, degraded conditions remains far from solved. In particular, long-distance and masked face recognition pose distinct challenges that are only partially addressed by off-the-shelf CNNs.

Within this landscape, the contributions of this thesis are twofold. First, the UPM-GTI-Face dataset [87] provides a publicly available, carefully controlled resource for evaluating the impact of distance and face masks on face detection and recognition systems, filling a gap in existing benchmarks. Second, the extensive comparative analysis between CNNs and ViTs conducted in this thesis offers empirical evidence on the relative strengths of both families of models across multiple datasets and conditions. Together, these contributions inform the design of the face recognition component of the personalized video summarization system, ensuring that the final application can reliably identify individuals in the challenging sports footage considered in subsequent chapters.

2.3 Personalized and identity-aware video summarization

The methods reviewed so far mostly aim at generating a *generic* summary for a given video: all viewers receive essentially the same set of keyframes, shots, or highlight clips. In contrast, *personalized video summarization* seeks to adapt the summary to the interests, preferences, or context of a specific user. A recent survey by Peronikolis and Panagiotakis [79] provides a

comprehensive taxonomy of personalized summarization techniques, including feature-based methods, keyframe and shot selection strategies, trajectory-based approaches, and clustering-based schemes, and highlights that personalization can be driven by user profiles, explicit queries, implicit feedback, or even physiological signals.

Early work already explored personalization in sports. Chen *et al.* proposed an autonomous framework to produce and distribute personalized basketball summaries from a network of cameras and sensors [10]. Their system combines automatic analysis of game events with a resource-allocation model that decides which views and time intervals to include for each user. While conceptually close to the goal of delivering player- or fan-specific summaries, these approaches are tightly coupled to dedicated capture infrastructures and broadcast-style productions, and do not explicitly model the visual identity of individual athletes.

A second line of research focuses on *person-centric* or *identity-aware* video summarization, where the summary is organized around the people that appear in the video. Zhang *et al.* introduced VideoWho, a person-based summarization and retrieval system that detects faces, tracks them over time, and clusters temporal face sequences to build summaries indexed by individual identities [129]. Zhou *et al.* [133] proposed a character-oriented summarization framework that uses visual and textual cues (face appearance, clothing, subtitles) to associate shots with fictional characters and generate character-specific summaries for long-form content such as TV series. In the movie domain, Ul Haq *et al.* [107] exploit deep CNN-based facial expression recognition and user-specified emotion preferences to select those segments where favorite characters exhibit target expressions, producing emotion-aware personalized movie summaries. These works demonstrate how face detection, tracking, and recognition can be integrated into the summarization pipeline, but they typically focus on general consumer or entertainment videos rather than sports, and often optimize for presence or affect rather than explicit event semantics.

More recent methods explicitly combine rich person descriptors with identity-aware summarization. Mirjalili *et al.* [70] propose a human-centric summarization pipeline that integrates body, pose, and face features extracted per person. Identities are assigned offline via density-based clustering, and frames are scored according to presence frequency, interaction density, and temporal coverage to produce concise, interpretable summaries centered on human activity. In soccer, the PlayerTV framework [92] leverages object detection and tracking, team-color analysis, and OCR of jersey numbers to track individual players and automatically generate player-specific clips from broadcast matches. While PlayerTV is primarily evaluated in terms of player and team identification accuracy, it illustrates how identity cues can be used to produce per-player highlight reels in professional sports settings.

Finally, several personalized summarization methods combine event semantics with user preferences without explicitly modeling visual identity. Fei *et al.* [24] learn user interest with an improved triplet deep-ranking framework and web-image priors for topic-related video summarization, while other works build client-driven personalized highlight detectors for long-form content such as movies and documentaries [73]. These approaches are closer in spirit to query-focused or intent-guided summarization, including VLM-based systems (see Section 2.1.4), but they usually assume generic domains and do not provide explicit identity-aware control.

Within this landscape, the Personalized Video Summarization (PVS) system introduced in Chapter 3 can be seen as an identity-aware, highlight-oriented specialization of these ideas for sports. It combines a text-guided highlight detector (SportCLIP or QD-DETR) with a face recognition pipeline (CNN-based ArcFace or Transformer-based TransFace) to produce athlete-specific summaries: the summarizer selects semantically relevant segments, and the face recognition component assigns them to the most likely athlete in a target set. To the best of our knowledge, none of the works discussed above jointly leverage modern vision–language summarization and robust face recognition to generate identity-aware highlight reels in sports: person-centric systems typically target movies or generic videos and rely on face clustering or expression analysis rather than explicit highlight semantics [70], [107], [129], [133], while sports-oriented systems such as PlayerTV focus on player and team identification from jersey and tracking cues rather than text-guided summarization [92]. The PVS system therefore occupies a relatively unexplored niche at the intersection of personalized summarization, sports highlight detection, and face recognition.

Chapter 3

Materials and methods

The methodological core of the thesis spans two complementary domains—video summarization and face recognition—and converges in a final, application-oriented pipeline for personalized video summarization. Across these domains, the guiding principles are: (i) reliable modeling of spatio-temporal dynamics in unconstrained videos; (ii) the use of representations that balance generality (transfer across categories, identities, and environments) with task specificity; (iii) the integration, when appropriate, of multimodal cues (e.g., vision–language alignment for text-guided selection) without conflating method description with empirical findings; and (iv) computational practicality to enable deployment on real-world footage.

In video summarization, the methods considered progress from classical motion and saliency pipelines to modern deep architectures that operate on sequences. Classical techniques (e.g., optical-flow estimation and feature point tracking) provide explicit motion cues and motivate early heuristic formulations such as subset selection and budgeted optimization. Contemporary approaches rely on learned representations—2D/3D CNNs, recurrent models, and, more prominently, Transformer-based architectures and vision–language models—to score or retrieve highlightable segments under diverse supervision regimes. Particular attention is paid to text-guided settings, where alignment between video and language enables query-conditioned summaries, and to design choices that emphasize temporal aggregation, cross-modal attention, and segment-level scoring. When relevant, combinatorial formulations (e.g., knapsack-style selection) are treated purely as methodic machinery to turn per-segment utilities into coherent summaries subject to duration constraints.

In face recognition, the methods surveyed and employed address the well-known imbalance between low inter-class variance and high intra-class variance, along with practical stressors such as occlusions and varying capture distances. The comparative analysis considers both Convolutional Neural Networks and Vision Transformers as backbone families, clarifying architectural differences (local receptive fields vs. global self-attention) and their implications for embedding quality, robustness, and efficiency. The methodological focus includes the construction of face descriptors, the choice of verification/identification losses, and the pre-/post-processing steps that connect detection to recognition in realistic pipelines. Scenarios involving masks and long-range imagery are framed as evaluation conditions that motivate certain design choices but are not interpreted in this section; performance observations are

deferred to the Results and Discussion chapters.

These two strands are ultimately combined in the Personalized Video Summarization (PVS) pipeline. Given a full sports video, a text query describing the events of interest, and one or more target identities specified through reference face images, PVS runs two parallel streams: a summarizer that proposes narratively salient candidate segments, and a face-analysis stream that detects faces, extracts embeddings, and builds temporal resemblance curves for each target. A final fusion stage then assigns highlight segments to the relevant identities according to these resemblance scores, yielding for each person a compact set of identity-aware highlight clips.

The remainder of the chapter is structured in three parts. First, the video summarization strand (Sections 3.1–3.3) traces the evolution from a classical, motion-based baseline for martial arts tricking to text-guided and Transformer-based summarizers, and formalizes a common framework for frame-level scoring and highlight event formation. Second, the face recognition strand (Sections 3.4–3.6) delineates the detection-to-verification pipeline and contrasts CNN and ViT backbones and their training objectives under practical conditions such as occlusion and distance. Third, the personalized video summarization section (Section 3.7) specifies the end-to-end composition that integrates both domains into the PVS pipeline, from full-video face analysis and resemblance curves to identity-aware highlight selection. Each subsection limits itself to methodological formulation, design rationales, and implementation details; all quantitative results and their interpretation are deferred to Chapter 4 and Chapter 5.

3.1 Video Summarization

In this thesis, video summarization is approached as a highlight detection problem: given a long sports video, and optionally a textual description of what the user is interested in, the goal is to return a small set of short segments that jointly capture the key attempts and outcomes while discarding redundant or uninformative footage. Throughout Chapter 3, all summarization methods follow the same basic pattern. First, the video is converted into a sequence of short temporal units (frames or clips) described by motion, visual, or multimodal features. Second, a model assigns a relevance score to each unit, reflecting how likely it is to belong to a highlight under the current task and query. Finally, a lightweight temporal post-processing stage smooths these scores, groups them into contiguous events, and outputs a list of highlight segments. This section outlines how this general recipe is instantiated by the different summarization models developed in this chapter and how their outputs interface with the PVS pipeline.

The first summarization approach focuses on a challenging, user-generated domain: martial arts tricking. Here, the aim is to detect sparse, high-impact tricks in long, untrimmed videos recorded with a single camera. Section 3.2 develops a fully classical, motion-centric pipeline that starts from key point detection and tracking, aggregates motion within regions that roughly correspond to the athlete's workspace, and constructs frame-level saliency curves that reflect how strong and localized the motion is over time. These curves are stabilized with temporal smoothing and then converted into candidate events by grouping neighboring high-saliency frames and assigning each event a relevance score based on its short- and long-

term motion patterns. The result is a transparent, end-to-end strategy that turns low-level key points into well-localized highlight events with an associated measure of importance, and that already introduces many of the temporal design choices—fine frame-level resolution, explicit event grouping, and relevance-based ranking—that will be reused by later methods. Experimentally (Section 4.3), this approach achieves high frame- and event-level F-scores on MATDAT and clearly outperforms a deep learning baseline designed for generic video highlights, showing that, in domains with fine-grained annotations and strong motion cues, carefully engineered classical features can still be highly competitive.

The second approach generalizes from a single tricking discipline to a much broader family of sports by introducing text guidance. In many broadcast settings, what counts as a highlight depends less on the amount of motion and more on semantics: whether a jumper clears the bar, how cleanly a diver enters the water, or whether a javelin lands in bounds. To handle this diversity without training sport-specific models, Section 3.3 presents *SportCLIP*, a CLIP-based framework that scores each frame by its similarity to short sentences describing highlight and non-highlight situations. Frames and sentences are mapped into a shared vision–language embedding space, and the difference between highlight and non-highlight scores yields a semantic relevance curve over time. Because CLIP is sensitive to prompt wording, the method generates multiple candidate sentence pairs and applies a two-stage filtering procedure based on score distributions and mean event area; only sentence pairs that produce discriminative and temporally coherent highlight patterns are retained, and their predictions are averaged into a single, robust curve. Crucially, the resulting relevance curves are fed to the same post-processing stage as in the motion-based method, which smooths scores over time and groups neighboring peaks into final highlight intervals. This allows a clean comparison between "motion-only" and "language-guided" summarization under a shared evaluation protocol. On MATDAT, the SportCLIP framework attains F-scores comparable to those of the motion-based baseline, while on the SportCLIP multi-sport dataset it maintains consistently strong F-scores across sports. In both datasets, it uses a single parameter set and operates in a fully zero-shot manner (Sections 4.4). Compared with the tricking-specific motion baseline, SportCLIP sacrifices a few points of performance on MATDAT but gains the ability to summarize multiple broadcast sports without any additional supervision, matching the thesis objective of designing methods that scale beyond a single discipline.

The third approach investigates Transformer-based, query-dependent architectures. While SportCLIP relies on a fixed CLIP encoder and simple cosine similarities, models such as Query-Dependent DETR (QD-DETR) jointly learn how to align visual and textual information over time. QD-DETR extends DETR with cross-attention between a textual query and video clips, producing a set of query-conditioned proposals that indicate when the described event occurs. In a zero-shot configuration, using off-the-shelf weights trained on generic web video, QD-DETR struggles to transfer to the fine-grained structure of sports highlights and underperforms SportCLIP framework on MATDAT and SportCLIP datasets (Section 4.4). However, when fine-tuned directly on the Olympic Highlights dataset with its dense highlight annotations (Section 4.7), QD-DETR becomes a strong supervised baseline. In the context of PVS, this fine-tuned model acts as an "oracle" summarizer—an approximate upper bound obtained by training and evaluating on the same distribution—and provides a reference point for assessing how close the zero-shot, text-guided methods come to the performance of a fully

adapted Transformer-based architecture.

All three summarizers ultimately feed into a common abstraction that is reused throughout the thesis and is made explicit in the PVS formulation (Section 3.7). Given a video and, when applicable, a textual specification of the desired events, the chosen model—motion-based, CLIP-based, or Transformer-based—assigns a relevance score to each frame. This per-frame sequence of scores is then smoothed, thresholded, and segmented into a set of temporal intervals, each representing a candidate highlight with an associated start time, end time, and duration. In the standalone experiments of Chapter 4, these intervals are compared against manual annotations to compute frame- and event-level metrics under different sports and supervision regimes. In the PVS system of Section 3.7, these highlight segments are passed to the face-analysis stream, which attributes them to specific target athletes.

This unified view clarifies the role of video summarization within the overall thesis. On MATDAT, the motion-centric framework and its evaluation protocol establish a transparent baseline, showing that carefully engineered motion descriptors can deliver very high event-level accuracy when highlights are short and annotations are fine-grained. SportCLIP, evaluated on both the SportCLIP and Olympic Highlights datasets, demonstrates that a single text-guided, zero-shot summarizer can generalize across broadcast sports and approach the performance of tailored motion models while remaining training-free and sport-agnostic. On Olympic Highlights, fine-tuned QD-DETR serves as an oracle summarizer trained and tested on the same distribution, providing an upper bound against which the zero-shot SportCLIP configuration remains surprisingly competitive—both in standalone evaluation and within the full PVS system. Taken together, these methods and datasets show that progress in sports video summarization stems from combining strong spatio-temporal representations with explicit temporal design, language-based control, and a clear separation between scoring and selection, and that this combination is precisely what enables the modular, identity-aware summaries produced by the PVS pipeline in the final part of the thesis.

3.2 Automatic highlight detection in videos of martial arts tricking

In this section we detail the work developed in *Automatic highlight detection in videos of martial arts tricking* [84] and place it within the overarching goal of this thesis: building a PVS system that produces highlights which are both narratively meaningful and tailored to a target identity. Reaching that goal requires two complementary capabilities: a robust mechanism for detecting moments of high salience in long, untrimmed videos (highlight detection), and an identity-aware mechanism that detects the target subject and prioritizes candidate segments accordingly (face recognition). This work is our first step toward the former, delivering a complete solution that clarifies the problem space, exposes practical challenges, and establishes a solid baseline and evaluation protocol for automatic highlight detection. As an initial instantiation, we concentrate on the sport of martial arts tricking, where highlight events are brief, high-velocity, and objectively delineated into passes—properties that make it a rigorous testbed for studying salience under demanding dynamics.

We propose a practical pipeline that begins with coarse motion cues and progressively refines them to yield a ranked set of highlight segments. Using long, untrimmed recordings with fixed viewpoints, we address three key difficulties: highlights are brief and sparse; non-highlight intervals (e.g., approaches, recoveries, near-misses) can closely resemble true events; and temporal localization must be precise. To handle these, the method comprises successive stages for temporal candidate generation, clip representation, and supervised ranking, together with an evaluation protocol suited to long, uncurated videos. Although purpose-built for tricking, several components are explicitly parameterized (e.g., window durations, kinematic thresholds, region definitions), which facilitates adaptation to other sports with minimal adjustments.

Beyond introducing the pipeline, this chapter contributes two elements that persist through the rest of the thesis: a lightweight post-processing stage for saliency score sequences—combining temporal smoothing, non-maximum suppression, and mild calibration to convert framewise scores into stable, well-localized segments—and a standardized evaluation setup for long, untrimmed videos. Subsequent chapters, despite adopting very different models, retain these two components (with minor adaptations), ensuring continuity and fair comparison across the thesis.

The work presented in this section builds directly on classical motion and saliency foundations. Following the blueprint in Section 3.2.1, we operate on low-level motion cues: we extract and track key points (Section 3.2.2); aggregate them into regions and summarize their kinematics (Section 3.2.3); detect temporally coherent events and derive an attention map that selects the most relevant region per frame (Section 3.2.4); and finally perform frame-wise classification with a refinement stage to obtain ranked highlight events (Section 3.2.5). This classical, motion-centric pipeline provides a well-specified baseline and a principled understanding of the summarization problem that frames the more advanced methods presented in the following sections.

3.2.1 System overview

Highlight events in the sport of martial arts tricking consist in players performing passes. Passes can have different duration and incorporate a variety of skills, but they all share that during a pass, a player combines different skills in quick succession. So it is the fast motion of a player performing which makes a highlight event stand out from other events. We propose a strategy able to extract such information to automatically identify highlight events from a given video. The basic outline of our strategy consists in four processing blocks, as shown in Fig. 3.1, each with a well-defined task.

In the key point extraction and tracking block (Section 3.2.2), the most prominent corners, or key points, of a frame are extracted and filtered from well-known background key points. This filtering is based on a probabilistic model dependent on the location history of the extracted set of key points, to focus on the region of interest (i.e., the foreground where players are performing). The resulting foreground key points are tracked from frame to frame to estimate their motion vectors, and both foreground key points along with their motion vectors will serve as low-level features that capture players' motions.

In the region-based analysis block (Section 3.2.3), foreground key points along with their motion vectors are grouped up into regions, which account for the spatial relationship among key points of the same frame, and summarize the information provided by the sparse set of key points that form them in a more compact way.

The event detection block (Section 3.2.4) comes after all frames of the input video have been analyzed for regions, and it is in charge of identifying the events they participate in. These will serve to generate an attention map that indicates, for each frame, the region participating in the most relevant event, under the assumption that the region of a frame participating in the most relevant event suffices to determine whether or not the entire frame can be classified as a highlight or not later on.

Finally, the event classification block (Section 3.2.5) analyzes the information contained in the regions spanned by the attention map to perform an initial binary classification at the frame level, classifying frames as either highlight or not. This initial classification is followed by a refinement stage in which highlight frames close in time are grouped forming highlight events, for which we model their relevance to produce the final result. The final result consists in a set of video sequences extracted from the input video where highlight events have been identified.

Following our proposed method, we are able to establish a correspondence between the initial key points (first system block) and the final events identified. Initial key points exclusively capture spatial information, while additional higher-level semantics are incorporated through the processing performed in the following blocks, including essential temporal information, which is key for the identification of highlights.

3.2.2 Key point extraction and tracking

This processing block extracts a set of key points from a frame (Subsection 3.2.2.1), separating the key points belonging to the foreground of the scene from those that are part of the background (Subsection 3.2.2.2), and tracks the foreground key points from frame to frame (Subsection 3.2.2.3) to estimate their motion vectors.

3.2.2.1 Key point extraction

Key points are the lowest-level features of the proposed strategy and they serve to identify locations of interest in an image. For instance, let I^n be the current image being analyzed at time n, and let (x^n, y^n) be a pixel with column x^n and row y^n. Locations of interest are identified as a set of K^n key points, $Q^n = \{(x_k^n, y_k^n), 1 \leq k \leq K^n\}$, where the pair (x_k^n, y_k^n) represents the coordinates (column and row) of the k-th key point extracted from the current image, I^n. The strategy we propose builds on top of these key points to extrapolate higher-level semantics.

Many relevant key point extraction methods were evaluated to determine the most suitable one for our strategy. Such method should be able to extract enough and well distributed key points from players performing, and these should be stable for tracking purposes. Moreover, this work contemplates video sequences where in addition to the player performing a highlight

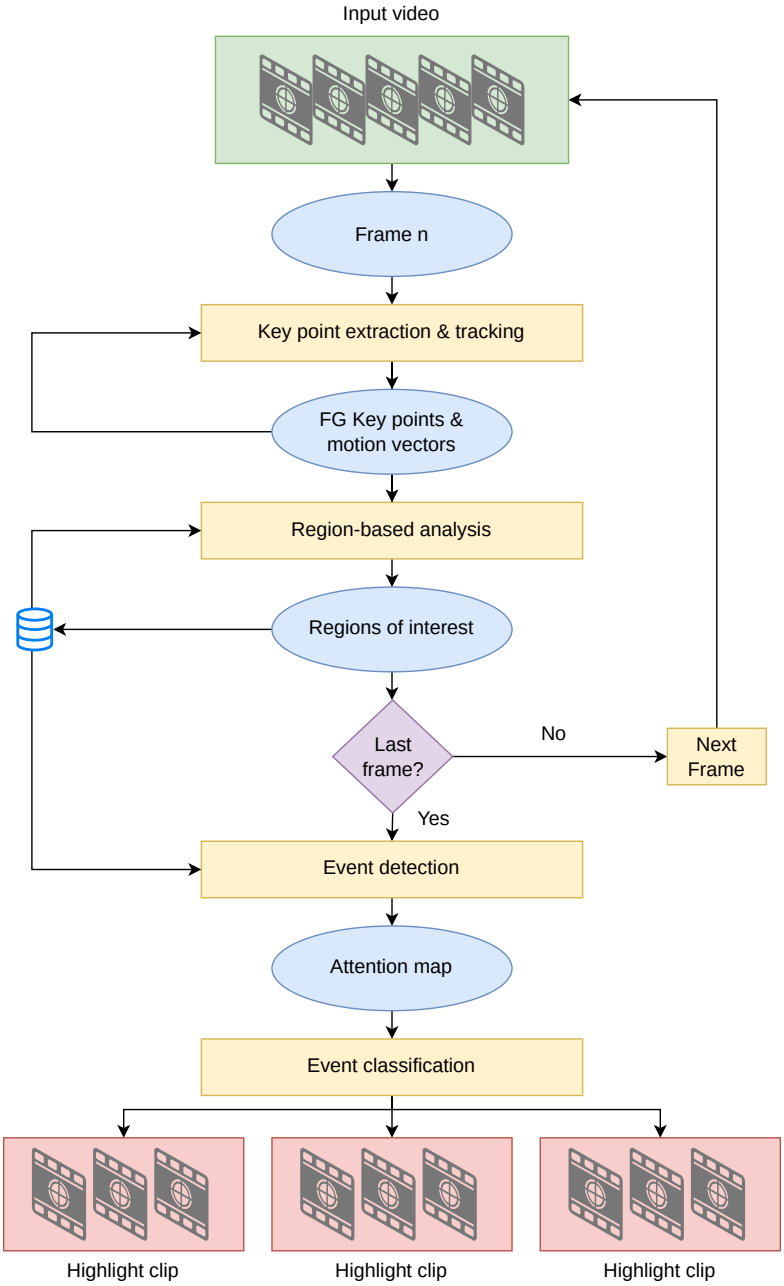
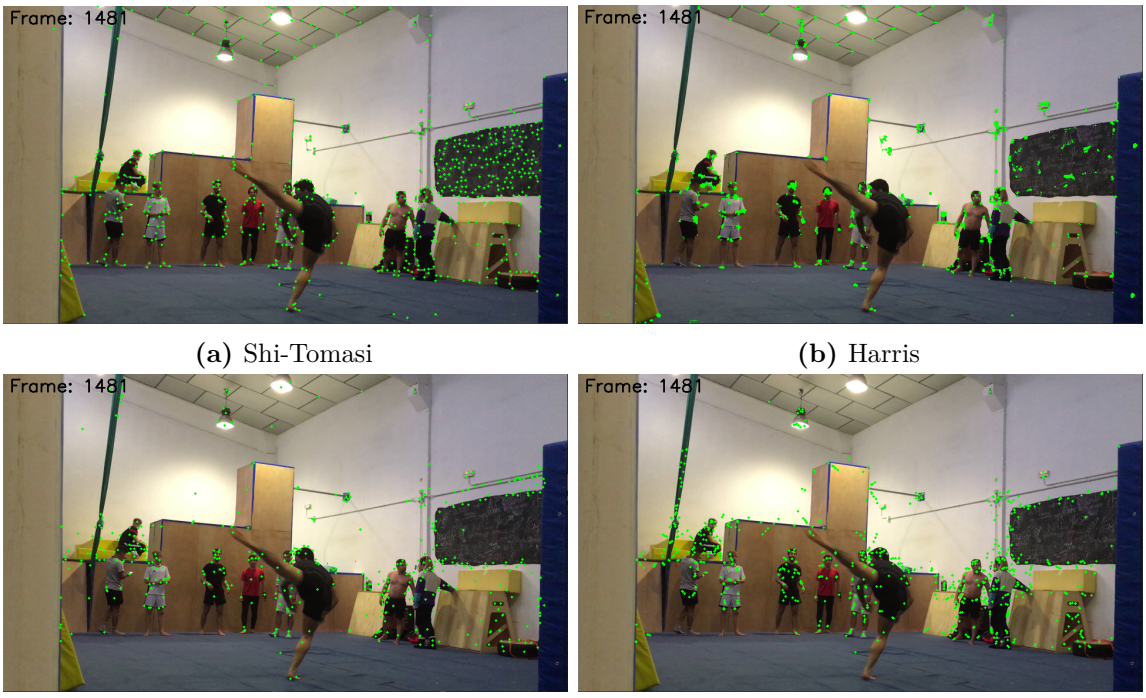


Figure 3.1: Block diagram of the proposed strategy. Rectangular blocks denote processing blocks, round-edge blocks denote data, and diamond blocks denote decision making.



(a) Shi-Tomasi **(b)** Harris

(c) SIFT **(d)** AKAZE

Figure 3.2: Key points detected using different feature extraction methods.

pass, there are other people moving in the foreground and the background, and so, the best fitting method should also be able to extract enough and well distributed key points over the entire frame. This allows to characterize both the players (i.e., foreground) and the environment (i.e., background).

Among others, we performed extensive experiments with Shi-Tomasi [51], Harris [38], SIFT [63], and AKAZE [1] algorithms. Fig. 3.2 illustrates the key points obtained with these methods after tuning them for the most favorable results possible. Shi-Tomasi provides a significant amount of key points well distributed throughout the frame and the players, and these are stable for tracking. Harris provides key points that are well distributed over the frame (although appearing in dense clusters, adding for some redundancy) but not over the players. Additionally, these are not stable for tracking, as they cannot be found on players performing due to their blurriness, which makes corners appear more diffuse. SIFT provides a stable set of key points, but these are ill-distributed over the frame and the players (e.g., ceiling and players' feet show very few key points). Finally, AKAZE provides a stable set of key points well-distributed over the players, but not over the frame, and similarly to SIFT, it also struggles extracting key points from players' feet.

Thus, the Shi-Tomasi algorithm proved to be the most adequate solution for extracting the set of key points Q^n for our strategy, as it identifies a sufficient quantity of key points that are well-distributed over the frame and the players. Additionally, these key points are stable across video frames, which facilitates smooth tracking. The algorithm detects the most prominent corners in a gray-scale representation of the frame by identifying little image patches or windows that generate significant variations in intensity when moved around.

3.2.2.2 Key point filtering

The set of key points Q^n is filtered from well-known background (BG) key points, Q^n_{BG}, to focus on the region of interest (i.e., the players performing). This filtering reduces the data to process and also prevents matching foreground (FG) key points with well-known BG key points when tracking them in the next processing step, described in Subsection 3.2.2.3.

Similar to Sun *et al.* [96], we use a method to update the probability of each pixel of being part of the BG, based on the location history of the extracted set of key points, Q^n. The probability of a pixel (x^n, y^n) at frame I^n being part of the BG is computed as

$$\Pr^n(x^n, y^n) = \begin{cases} \Pr^{n-1}(x^n, y^n)\lambda + (1-\lambda), & (x^n, y^n) \in Q^n \\ \Pr^{n-1}(x^n, y^n)\lambda, & (x^n, y^n) \notin Q^n \end{cases} \tag{3.1}$$

where λ is a learning factor set to 0.95 as suggested in [89]. Following equation (3.1), if a key point is consistently identified in the same location across frames, the probability of such pixel of being part of the BG will increase. Therefore, a set of filtered key points is obtained as

$$Q^n_{\mathrm{f}} = \{(x^n, y^n) \in Q^n \mid \Pr^n(x^n, y^n) < T\} \tag{3.2}$$

where T is a threshold value (set at the empirical value of 0.15) that filters well-known BG key points. As depicted in Fig. 3.3, BG key points (in red) can be found on static objects of the scene as well as on players who stay immobile for a period of time. On the other hand, the remaining set of key points (in green) can be found on moving objects and, due to illumination changes, on some static objects. However, as described in Subsection 3.2.2.3, the latter will be easily removed.

3.2.2.3 Key point tracking

The set of filtered key points, Q^n_f, is matched against the set Q^{n-1}_f, corresponding to the previous frame, to estimate the motion vectors associated to each key point in Q^n_f, by making use of the iterative Lucas-Kanade method with pyramids [6].

Key points presenting very small motion vector magnitudes (e.g., less than 2 pixels) are filtered to mitigate illumination changes previously mentioned in Subsection 3.2.2.2. This filtering prevents static key points from being interpreted as moving due to changes in illumination, and results in the set of FG key points, Q^n_{FG}. Fig. 3.4 shows an example of the results obtained following this method. The interesting region of the scene (i.e., the player performing) is captured by the set of FG key points along with their motion vectors. It can be seen that after filtering key points with little motion associated, all key points that were incorrectly classified as part of the FG in the previous stage have been discarded.

3.2.3 Region-based analysis

FG key points, Q^n_{FG}, along with their associated motion vectors, are grouped up by vicinity forming a set of regions. The underlying idea is that the motion of these regions can provide

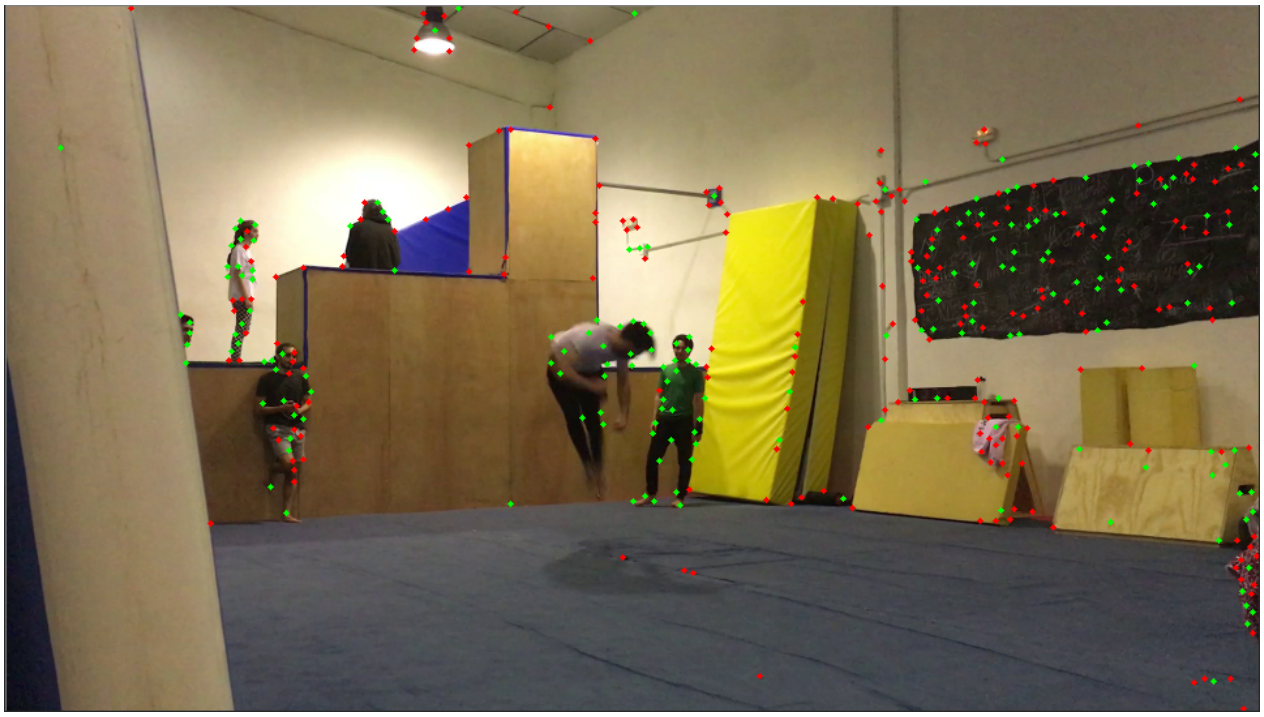


Figure 3.3: Example of well-known *BG* key points filtering. Red represents well-known *BG* key points while green represents the remaining set of key points.

Figure 3.4: Example of motions estimated using a pyramidal implementation of the Lucas-Kanade algorithm. In green the set of motion vectors associated to *FG* key points, in blue those with very small magnitudes, and in red well-known *BG* key points.

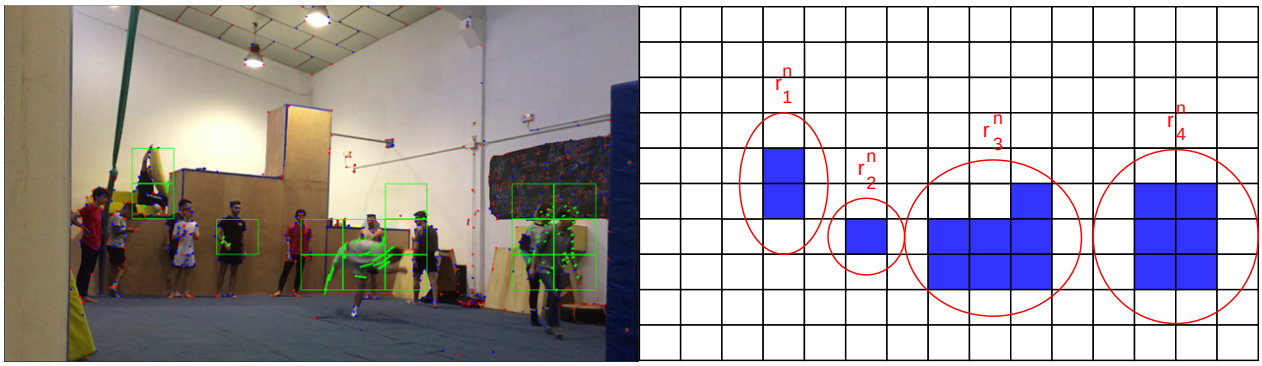


Figure 3.5: Example of identified regions using the proposed method. Foreground key points are mapped to their corresponding cells and four regions of different sizes are identified (enumerated from 1 to 4 from left to right).

more robust and reliable information than that provided by the sparse key points that form them, allowing to identify and characterize interesting regions of the scene. It is worth noting that players are neither rigid nor always present the same orientation. Therefore, it is not feasible to establish durable key point correspondences throughout a sequence, which motivates the proposed region-based approach.

For this purpose I^n is tessellated into a grid of non-overlapped uniform cells $C_{\{W,H\}}$ (e.g., a grid of 15×10 cells), where W and H represent a cell location, column and row, respectively. Cells are sized large enough to represent a region but small enough to provide local information. The set of FG key points, Q^n_{FG}, is mapped onto these cells revealing which ones are active (i.e., cells containing at least 1 FG key point). Using a fixed grid resolution (e.g., 15×10) expresses region size in relative image units rather than pixels, which makes the subsequent filtering thresholds easier to interpret across videos of different resolutions. Nevertheless, the effective physical size represented by a cell depends on the camera distance/zoom: if subjects appear significantly smaller or larger, the grid resolution (and derived size thresholds) should be adjusted accordingly. In practice, this can be done by selecting the grid such that a typical athlete spans several cells (so that relevant motion regions are neither fragmented into many tiny components nor merged into overly coarse blobs).

Active cells in the grid are grouped up by vicinity following a connected-component labelling approach [41], resulting in a set of J^n regions, $R^n = \{r^n_j, 1 \leq j \leq J^n\}$, where r^n_j represents the j-th region of the n-th frame. Regions are characterized by the cells that form them ($C_{\{W,H\}}$), by the number of FG key points that fall within them (N) along with the sum of their associated motion vectors magnitudes (S), and by their normalized motion ($\overline{M} = S/N$). When taking motion into account we do not consider its direction but only its magnitude, as this suffices to represent the overall motion present in a region.

Intuitively, fast-moving objects are commonly blurrier than objects moving slower. Therefore, it is likely that less key points will be extracted from them in the first place (see Subsection 3.2.2.1), and these are likely to present larger motions. The normalized motion of a region, $\overline{M}$, allows to compare the amplitude of the motion between regions irrespective of the number of detected key points.

Table 3.1: Regions description. r_j^n: j-th region of the n-th frame. $C_{\{W,H\}}$: active cells. N: number of key points. S: sum of associated motion vectors magnitudes (pixels). $\overline{M}$: normalized motion (pixels/key point).

r_j^n	$C_{\{W,H\}}$	N	S	$\overline{M}$
r_1^n	$C_{\{3,4\}}$, $C_{\{3,5\}}$	4	22.55	5.64
r_2^n	$C_{\{5,6\}}$	3	11.70	3.90
r_3^n	$C_{\{7,6\}}$, $C_{\{7,7\}}$, $C_{\{8,6\}}$, $C_{\{8,7\}}$, $C_{\{9,5\}}$, $C_{\{9,6\}}$, $C_{\{9,7\}}$	26	273.11	10.50
r_4^n	$C_{\{12,5\}}$, $C_{\{12,6\}}$, $C_{\{12,7\}}$, $C_{\{13,5\}}$, $C_{\{13,6\}}$, $C_{\{13,7\}}$	46	130.40	2.83

In the example of Fig. 3.5 we can distinguish four regions that correspond, from left to right, to a person moving an object, the waving of a hand of another person, a person performing a pass, and two people walking together. The information each region contains is summarized in Table 3.1. Regarding N and S, regions 3 and 4 are much more relevant than the other two regions. This alone manifests the capability of this method to identify foreground regions of interest and characterize them. In addition, $\overline{M}$ shows its usefulness for differentiating regions more relevant in terms of motion. Region 3 having half as many points as region 4, but an overall motion twice as large, presents a normalized motion almost four times greater.

3.2.3.1 Region filtering

Regions can be further filtered at this processing block in order to save some computational cost at later blocks, storing only regions of interest that are good candidates to be a part of a highlight. This is done under two simple assumptions: (1) good candidate regions are at least of a certain size and (2) are likely to be located around the same area than a good candidate region of the previous frame. Note that this block acts as a lightweight pruning heuristic to reduce downstream computation; it does not encode sport-specific semantics and its parameters can be adapted without altering the remaining stages of the pipeline.

The first assumption is easy to interpret. A region of interest must contain at least a minimum number of cells, as these are dimensioned large enough so that they can represent a region but not so large that they can be of interest by themselves. Regions that are too small relative to the chosen grid resolution (e.g., on the order of only a couple of cells for the 15×10 configuration) are discarded, as they are unlikely to correspond to an athlete or a salient moving object and typically arise from sporadic keypoints or background clutter. Importantly, this threshold is expressed in *cell units* and should be interpreted jointly with the grid resolution: if the camera distance/zoom changes (or if adapting the method to other sports with different subject scales), the minimum region size can be re-scaled accordingly.

The second assumption is based on the underlying idea that regions of interest will smoothly evolve along frames, so it is highly unlikely for a region to appear at a location in a frame and in the next one be at a completely different location. This allows removing regions that appear sporadically over frames by seeking neighboring cells with regions of interest identified in the previous frame and discarding those which do not share neighbors.

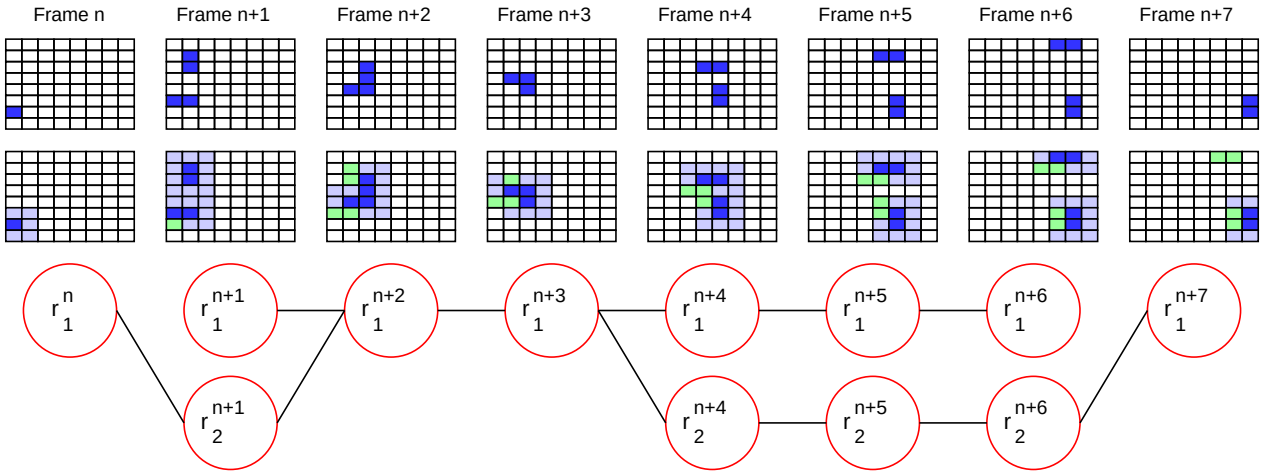


Figure 3.6: Example of a set of linked regions. In blue are represented the active cells that form each region. Purple displays the neighborhood around each region. Green overlays the regions' locations of the previous frame. The bottom graph represents the regions that are linked.

3.2.4 Event detection

This processing block is executed after all frames of the input video have been analyzed for regions (see Section 3.2.3), and it is in charge of identifying the events they participate in. These will serve to generate an attention map that indicates, for each frame, the region participating in the most relevant event. This is done under the assumption that the region of a frame participating in the most relevant event suffices to determine whether or not the entire frame can be considered as part of a highlight (see Section 3.2.5).

The region-based analysis described in Section 3.2.3 summarizes the foreground motion information of each frame in a set of regions uniquely identified. To reveal possible events, regions of contiguous frames which share neighboring cells are first linked as depicted in Fig. 3.6 to analyze their evolution over frames: how they appear, disappear, displace, split, or merge.

All these possible events can be revealed following a directed acyclic graph (DAG[1]) approach. Each node represents a region that is directed to other nodes (regions) of the following frame. Start nodes are those regions that do not have any links with the previous frame, whereas end nodes are those that do not have them with the following one.

Taking into account all start and end nodes we can define all series of unique events E_i as depicted in Fig. 3.7. Events consist in a set of linked regions which represent how a region evolves through frames from its appearance to its disappearance, and are characterized by the motion features contained in the regions they span. Different events can overlap in frames and share one or several regions, but there cannot be two identical events with the same set of linked regions. Additionally, very short events (e.g., less than 1 second) are removed at

[1]A graph consisting of nodes that are directed from one to another such that following those directions will never form a closed loop.

this processing block, similarly to what we did in Subsection 3.2.3.1, as they are not likely to represent a highlight event.

3.2.4.1 Attention map

Under previous assumptions, the best candidate region of a frame to be a part of a highlight event would be that with the largest normalized motion. But temporal information plays a key role when assessing motion and has to be accounted for too. For this reason, we generate an attention map that indicates, for each frame, the region participating in the event that averages the largest normalized motion along the regions it spans. This allows selecting those regions participating in the most significant events, even when these regions do not show the largest normalized motion for a particular frame. This will serve as a cue to detect the start and end of a highlight event as will be further explained in Section 3.2.5.

The output of this processing block can be formulated as an attention map that indicates, at each frame, where the region more likely of being a part of a highlight event is, as depicted in Fig. 3.8. Comparing this figure with previous Fig. 3.7, it can be appreciated that when multiple events concur at a frame, the selected region is that which participates in the event that averages the largest normalized motion. For instance, let the average normalized motion of E_1 be the largest of the four events. For frame $n+1$, where the four events concur, region r_2^{n+1} is selected over r_1^{n+1} as it participates in the most relevant event. In frame $n+7$ only two events concur, E_2 and E_4, and both share the same region, so r_1^{n+7} is selected for that frame.

3.2.5 Event classification

Event classification constitutes the last processing block of the proposed strategy. It performs after each frame of the input video has been assigned a single region (or no region if none was identified) that participates in the most relevant event, as indicated by the attention map obtained in Subsection 3.2.4.1. The motion information these regions contain is used to perform an initial binary classification at the frame level, classifying frames as either highlight or not. Highlight frames that are close in time are grouped together to form highlight events during a subsequent refinement stage, for which we model their relevance to produce the final result. The final result consists in a set of video sequences extracted from the input video where highlight events have been identified.

Fig. 3.9 illustrates the motion information contained in the regions indicated by the attention map, where the three curves represent the values of S, N, and $\overline{M}$ over frames. In blue are represented the instantaneous values at each frame, which correspond to those of the region indicated by the attention map for that frame. It can be appreciated that these values remain zero for many frames, for which no regions of interest have been identified on previous blocks. In red and green are represented the rolling averages of these values for short and medium time windows (1 and 10 seconds) respectively, which will serve to measure how much a short event centered at a particular frame stands out from its surroundings. This choice can be interpreted as a simple local-contrast test on the motion signal: the 1-second window provides a responsive estimate of the current activity level, whereas the 10-second window acts as an adaptive baseline that summarizes the recent context of the broadcast. A frame is

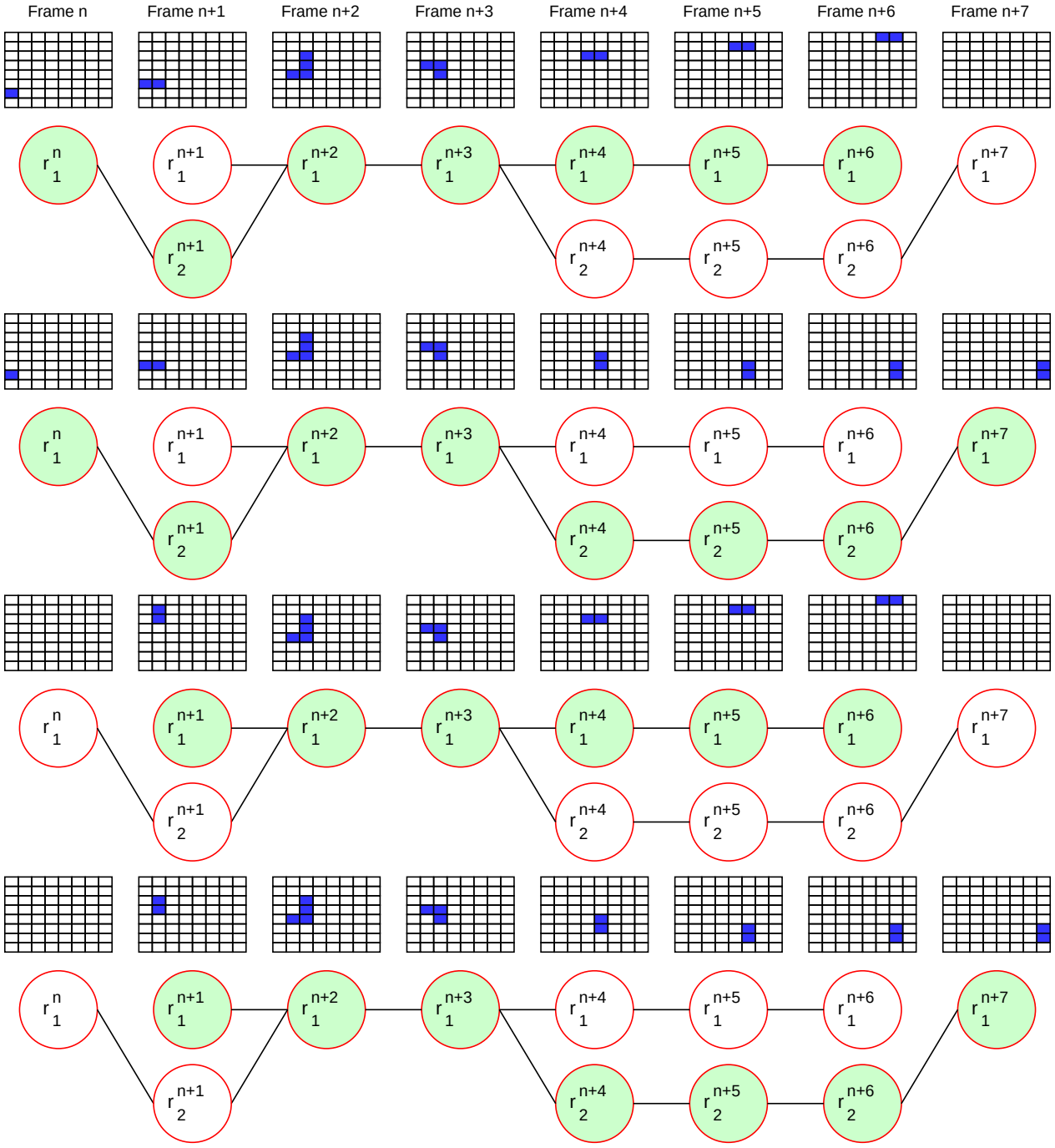


Figure 3.7: All possible events identified among the linked regions of the original sequence of Fig. 3.6 following a DAG approach. Four events are identified, enumerated from E_1 to E_4 from top to bottom. Bottom figures illustrate the paths followed from start to end nodes. Top figures illustrate the regions associated to each of those paths.

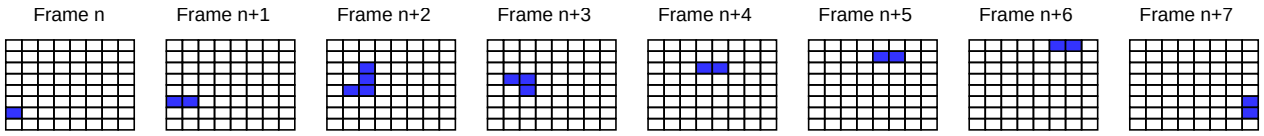


Figure 3.8: Example of generated attention map, which indicates, for each frame, the region participating in the event that averages the largest normalized motion.

considered salient when its short-term motion rises above what is typical in its immediate neighborhood, which is precisely the behavior expected for highlights (brief, high-motion bursts) compared to non-highlight intervals (low motion or sustained, steady motion). Using a longer baseline rather than a global threshold also makes the decision less sensitive to video-dependent factors such as camera motion, replays, or sport-specific pacing, since the reference level is continuously updated from the same temporal segment.

3.2.5.1 Initial classification

An initial binary classification is performed at the frame level, classifying frames where the short-term average normalized motion exceeds the long-term (i.e., red line above green line) as highlight, or as non-highlight otherwise (see Fig. 3.10). Thus, frames are classified as highlights when the short-term motion level significantly exceeds the local baseline, i.e., when a brief action-centered interval stands out from its immediate temporal context.

3.2.5.2 Classification refinement

The initial classification is first refined by extracting the set of identified highlight events from the grouping of highlight frames that are close in time (e.g., less than 1 second apart) and therefore are likely to belong to the same highlight event, as shown in Fig. 3.11.

The relevance of identified highlight events is modeled using the area enclosed between the short-term average normalized motion and the long-term, as depicted in Fig. 3.12. This area serves as a measure of how much identified highlight events stand out from their surrounding, making it possible to characterize their relevance. The relevance modeling of the identified highlight events constitutes an addition to their previous detection, making it possible to compare, order, or filter identified highlight events based on their relevance.

The final refinement step consists in the filtering of identified highlight events that are poor candidates, either because of their short duration (e.g., less than a second[2]) or because of their low relevance. To better illustrate this process, Fig. 3.13 shows the highlight events obtained after filtering poor candidates, along with the ground truth annotations, which will be covered in Subsection 4.1.1.1. The final result produced by the proposed strategy is a set of video sequences comprised of frames from the input video where highlight events have been identified.

Taken together, the blocks described in this section form a coherent, end-to-end strategy

[2]Note that even though events of less than a second were filtered in Section 3.2.4, highlight events of shorter duration can appear in the initial frame classification performed in Subsection 3.2.5.1.

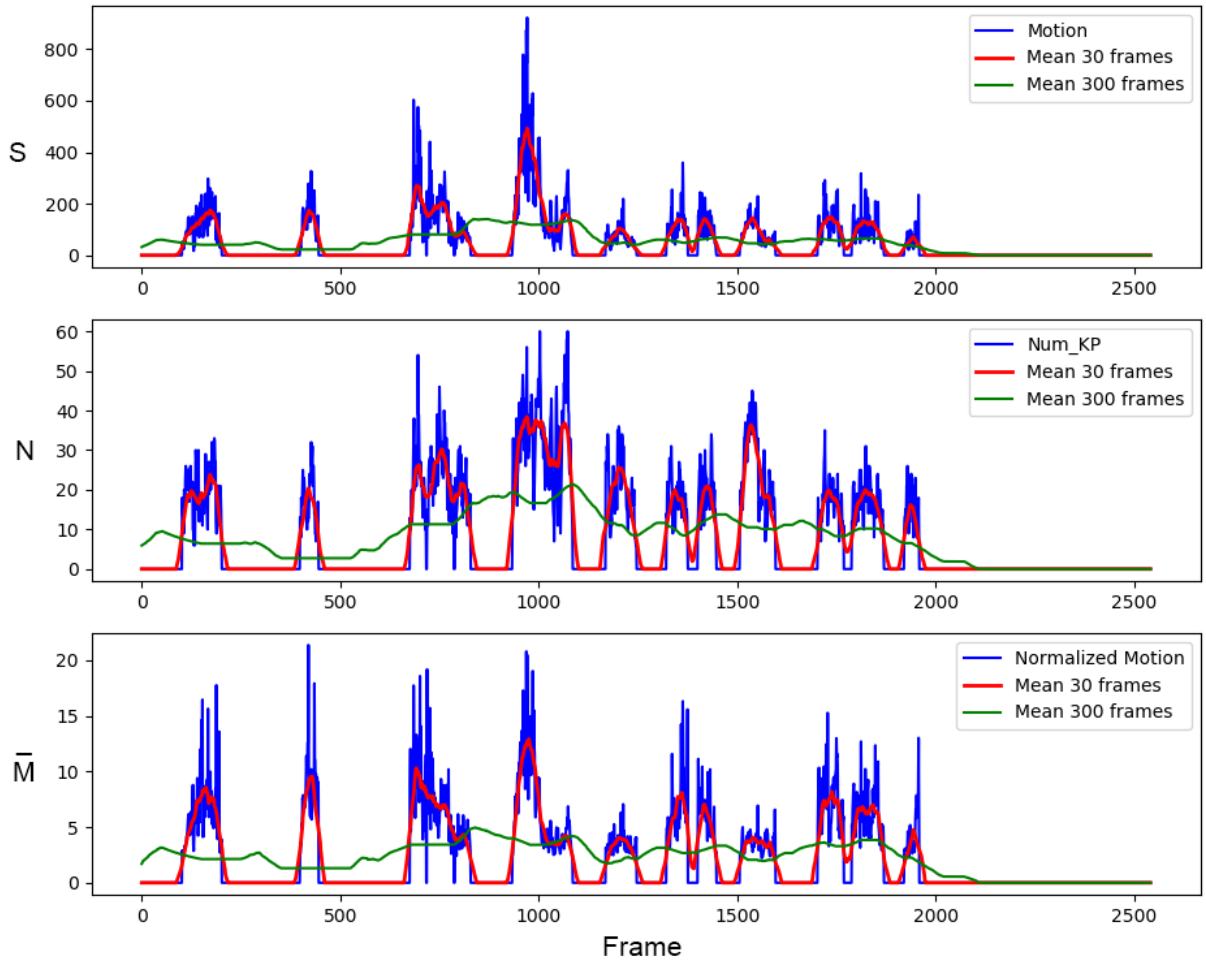


Figure 3.9: Summary of the motion information stored in the regions of the attention map. The three graphics correspond to the variables S, N and $\overline{M}$, respectively. The current values at each frame, which correspond to those of the region selected for that frame, are represented in blue. In red and green their rolling averages for a short and medium time windows, respectively.

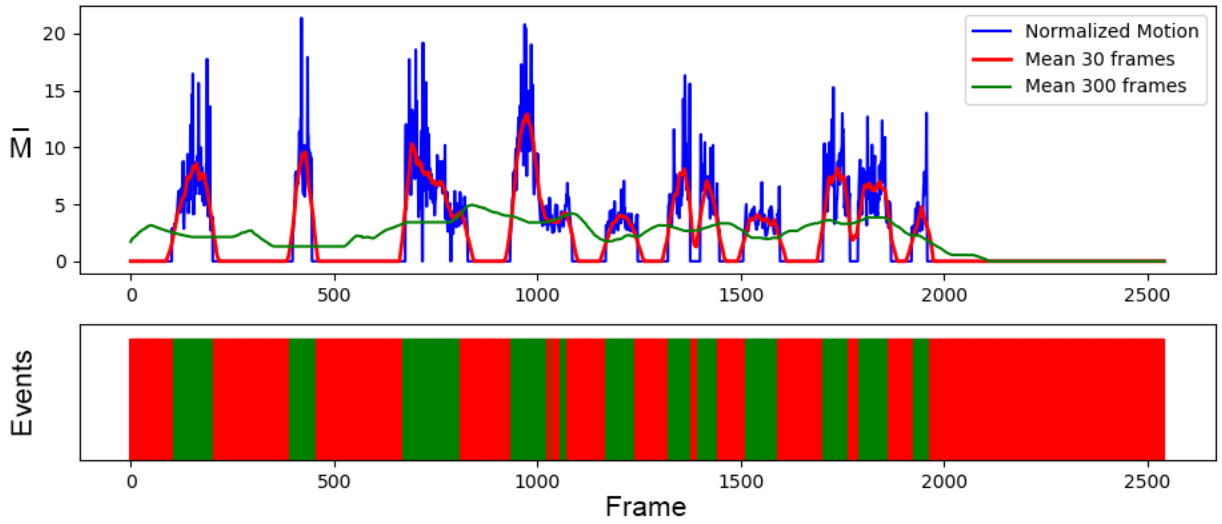


Figure 3.10: Initial binary classification of frames. Frames where the short-term average normalized motion exceeds the long-term (i.e., red line above green line) are initially classified as highlights, represented as green bars. Red bars correspond to frames classified as not a highlight.

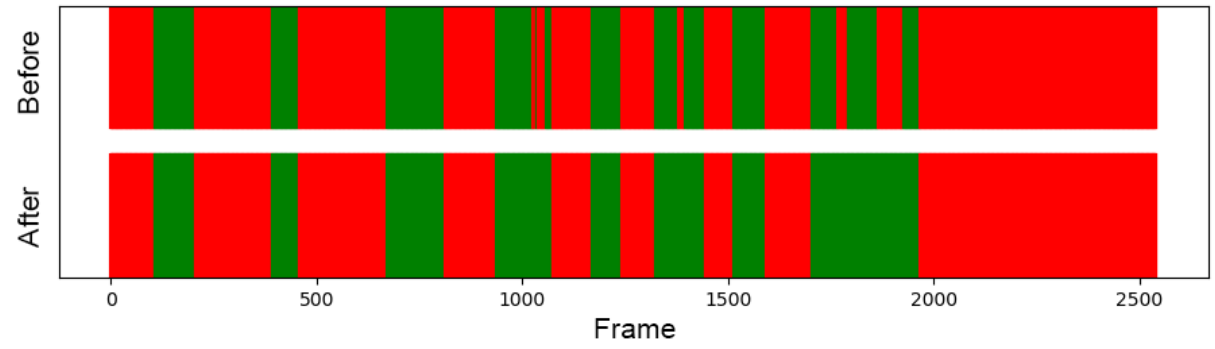


Figure 3.11: Grouping of highlight frames that are close in time, and thus, are likely part of the same highlight event. Top bar represents the classified frames before the closing operation, whereas the bottom bar represents the results after the operation, the highlight events.

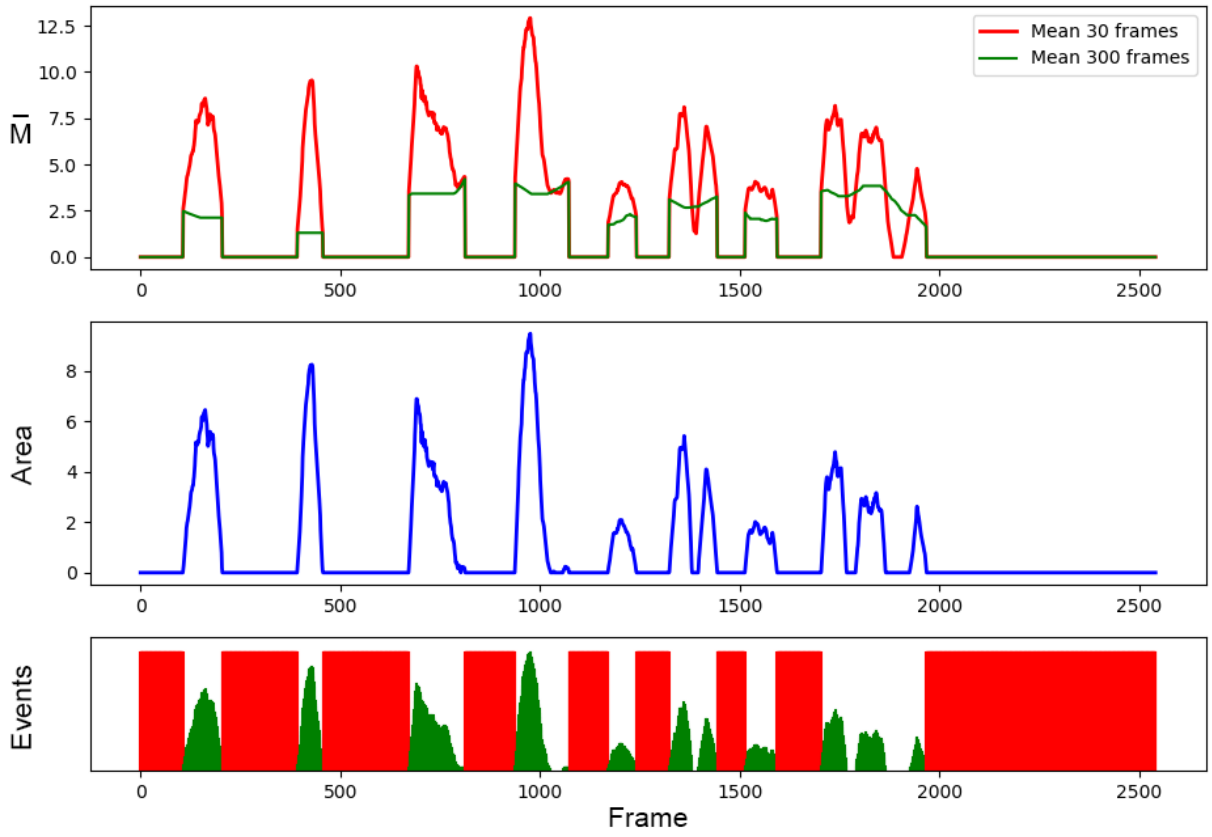


Figure 3.12: a) Short- and long-term average normalized motions for a short and a medium time windows where highlight events have been identified. b) Enclosed area. c) Events after modeling their probability of being a highlight.

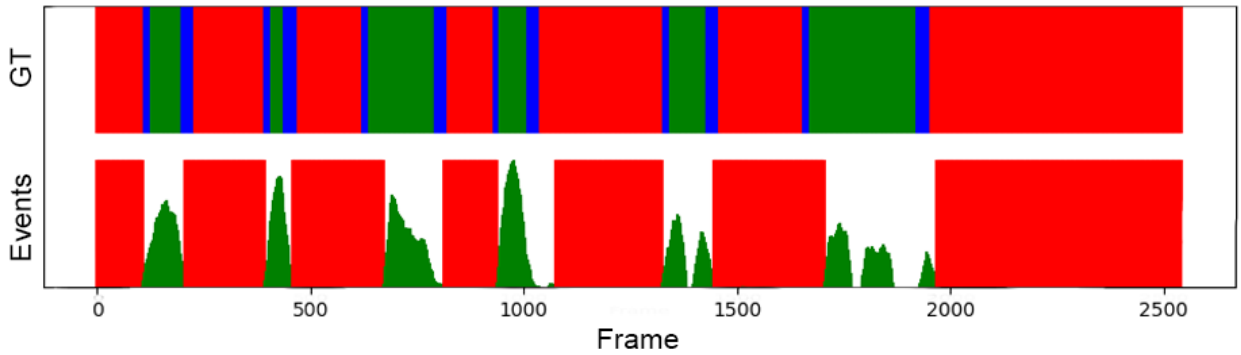


Figure 3.13: Final result of the proposed strategy after filtering poor candidate highlight events. The top bar represents the ground truth events, whereas the bottom bar corresponds to the identified highlight events. Red indicates not a highlight, green indicates a highlight and blue indicates uncertainty.

that transforms low-level key points into well-localized highlight events with an associated measure of relevance. Beyond its conceptual clarity, the method has proved to be both practical and robust when confronted with the challenges of long, untrimmed tricking videos, coping well with sparse highlights, distractor motions, and the need for precise temporal delimitation. Its reliance on transparent, classical components facilitates analysis, debugging, and adaptation to other sports, while the final relevance modeling provides a flexible control over the number and prominence of reported events. As will be shown in Section 4.3, the experimental results confirm the utility and quality of the proposed approach, demonstrating competitive performance and establishing a solid baseline that will serve as a reference point for the more advanced, learning-based methods developed later in this thesis.

3.3 Text-Guided Sports Highlights: A CLIP-Based Framework for Automatic Video Summarization

The previous section presented our foundational work in highlight detection [84], a classical, motion-centric baseline that is effective within the specific domain of martial arts tricking. To summarize a wider range of sports—where "what counts as a highlight" often hinges on semantic cues (e.g., a clean landing, a ball crossing the line, a successful vault) rather than raw motion alone—we move to a more flexible and scalable formulation.

This section introduces *Text-Guided Sports Highlights: A CLIP-Based Framework for Automatic Video Summarization* [86], a text-guided, zero-shot framework that replaces hand-crafted motion rules with learned, multimodal representations from CLIP [81]. Instead of inferring saliency exclusively from motion, this work scores each frame by its semantic similarity to concise textual prompts describing highlight (*HL*) and non-highlight (*NHL*) situations (e.g., "an athlete sticks the landing" vs. "an athlete preparing to jump"). Frames and prompts are embedded into a shared space and compared directly, enabling open-vocabulary operation across different sports without sport-specific training or fine-tuning.

A central challenge in vision–language models is prompt sensitivity. This work addresses this with a simple, robust procedure: we generate multiple *HL*/*NHL* sentence candidates, compute per-frame scores for every sentence pair, and then apply a two-stage filtering-and-aggregation step. First, a distribution-based filter removes pairs whose score histograms are nearly uniform (high entropy) and hence non-discriminative. Second, a mean-event-area filter discards pairs that, after refinement, produce weak or fragmented events. The predictions from the remaining pairs are averaged to form a single, stable highlight score curve.

To ensure temporal coherence and comparability with our prior baseline, the aggregated score curve is passed through the same lightweight post-processing pipeline introduced in Section 3.2 (temporal smoothing, non-maximum suppression, and mild calibration). This preserves continuity with earlier design choices while allowing a clean comparison between a motion-based front end and a language-guided one under a shared post-processing and evaluation setup.

The remainder of this section is organized as follows: Section 3.3.1 sketches the pipeline at a high level and clarifies its interfaces; Section 3.3.2 formalizes the scoring, filtering,

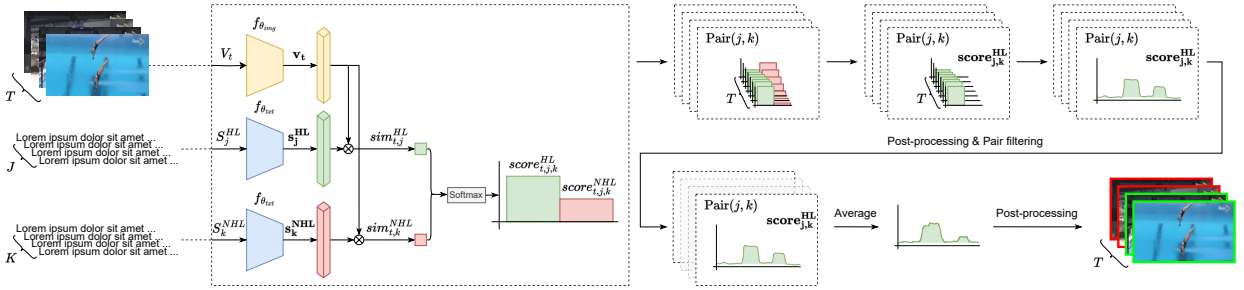


Figure 3.14: Overview of the proposed video summarization solution. The input video is divided into T individual frames ($t = 1, ..., T$), each processed by CLIP's image encoder to obtain frame embeddings. A set of J highlight (HL) sentences ($j = 1, ..., J$) and K non-highlight (NHL) sentences ($k = 1, ..., K$)—generated by a large language model—are fed to CLIP's text encoder to produce text embeddings for every (j, k) pair; their cosine similarities (after softmax) yield frame-level highlight predictions for each sentence pair. These predictions are refined, filtered to remove unsuitable pairs, and then averaged into a single, more robust highlight score curve. A final post-processing step converts this averaged curve into the final video summary, where frames are color-coded (green for highlights, red for non-highlights) to visually indicate the summarized events.

and aggregation procedure, detailing the entropy- and mean-event-area criteria and their parameterization. As in the rest of Chapter 3, all quantitative results, ablations, and cross-sport analyses are deferred to the results (Chapter 4) and discussion (Chapter 5) chapters.

3.3.1 System overview

CLIP (Contrastive Language-Image Pre-training) [81] represents a groundbreaking advancement in the field of artificial intelligence, seamlessly integrating vision and language modalities to facilitate cross-modal understanding. CLIP operates on the principle of contrastive learning, it learns to associate images and corresponding textual descriptions through a process of self-supervised training. At its core, CLIP comprises image and text encoders, $f_{\theta_{img}}$ and $f_{\theta_{txt}}$, that transform visual and textual inputs into high-dimensional embeddings. These encoders encode images and text into a shared latent space, where semantic similarities between image-text pairs are effectively captured. By leveraging a large-scale dataset containing diverse image-text pairs, CLIP learns to generate robust embeddings that encapsulate rich semantic information. Central to CLIP's efficacy is its ability to perform zero-shot and few-shot classification tasks. It can accurately classify images based on textual prompts even without explicit training on specific classes. This capability stems from CLIP's holistic understanding of the underlying semantics present in both images and text.

Figure 3.14 provides a high-level illustration of our video-summarization approach. In this framework, CLIP acts as a binary frame classifier: we extract embeddings from every video frame, capturing its semantic essence, while a large-language model supplies multiple candidate sentences that describe typical HL and NHL situations. For each HL–NHL sentence pair, we compute a similarity score between the frame embedding and both sentence embeddings, producing one set of frame-level highlight probabilities per pair. The resulting collections of

scores quantify how closely each frame aligns with the highlight or non-highlight scenario.

These frame-level highlight predictions undergo the post-processing stage described in [84] while all subsequent scoring, filtering, and aggregation steps are newly developed here. The post-processed predictions are then evaluated with novel robust filtering techniques, discarding only those *HL–NHL* pairs that produce clearly unsuitable results. We average the frame-level predictions from the remaining valid pairs to form a single, more reliable set of frame-level highlight predictions, which in turn are subjected to the same refinement procedure to create the final cohesive video summary. This approach effectively captures the essential moments of the video while preserving semantic coherence across the varied textual descriptions. We defer a full, step-by-step explanation of each component to Section 3.3.2.

3.3.2 Methodology

Let us consider a given video frame $V_t, t \in \{0, ..., T\}$, a highlight sentence $S_j^{HL}, j \in \{0, ..., J\}$ and a non-highlight sentence $S_k^{NHL}, k \in \{0, ..., K\}$. We feed the video frame V_t into the image encoder $f_{\theta_{tmg}}$ and the sentences S_j^{HL} and S_k^{NHL} into the text encoder $f_{\theta_{txt}}$ to obtain a frame embedding $\mathbf{v_t}$ and sentence embeddings $\mathbf{s_j^{HL}}$ and $\mathbf{s_k^{NHL}}$ respectively, where

$$\begin{aligned} \mathbf{v_t} &= f_{\theta_{tmg}}(V_t), \\ \mathbf{s_j^{HL}} &= f_{\theta_{txt}}(S_j^{HL}), \\ \mathbf{s_k^{NHL}} &= f_{\theta_{txt}}(S_k^{NHL}). \end{aligned} \tag{3.3}$$

A cosine similarity function is utilized to compute the similarity between the visual and the textual representations for both the *HL* and *NHL* scenarios:

$$\begin{aligned} sim_{t,j}^{HL} &= \frac{\mathbf{v_t} \cdot \mathbf{s_j^{HL}}}{\|\mathbf{v_t}\| \, \|\mathbf{s_j^{HL}}\|}, \\ sim_{t,k}^{NHL} &= \frac{\mathbf{v_t} \cdot \mathbf{s_k^{NHL}}}{\|\mathbf{v_t}\| \, \|\mathbf{s_k^{NHL}}\|}. \end{aligned} \tag{3.4}$$

Thus, by comparing the embeddings of each video frame to both sentence embeddings, we obtain raw cosine similarity values in the range $[-1, 1]$. These raw similarities are converted into probabilities via a softmax function, mapping them to the score interval $[0, 1]$. A higher probability indicates a stronger alignment with the corresponding scenario:

$$\begin{aligned} score_{t,j,k}^{HL} &= \frac{e^{sim_{t,j}^{HL}}}{e^{sim_{t,j}^{HL}} + e^{sim_{t,k}^{NHL}}}, \\ score_{t,j,k}^{NHL} &= \frac{e^{sim_{t,k}^{NHL}}}{e^{sim_{t,j}^{HL}} + e^{sim_{t,k}^{NHL}}}. \end{aligned} \tag{3.5}$$

The two scores are complementary, satisfying

$$score^{HL}_{t,j,k} + score^{NHL}_{t,j,k} = 1, \forall t, (j,k). \tag{3.6}$$

For each sentence combination (j,k), we obtain one set of frame-level predictions for the *HL* and *NHL* scenarios:

$$\begin{aligned} \mathbf{score^{HL}_{j,k}} &= \{score^{HL}_{t,j,k}\}^{T}_{t=0}, \\ \mathbf{score^{NHL}_{j,k}} &= \{score^{NHL}_{t,j,k}\}^{T}_{t=0}, \end{aligned} \tag{3.7}$$

which capture how each frame t aligns with the *HL* or *NHL* scenario, respectively.[3]

After computing the frame-level highlight predictions $\mathbf{score^{HL}_{j,k}}$ for each sentence pair (j,k), we apply the post-processing steps described in [84]. These steps merge adjacent highlight frames into contiguous events—maximal stretches of consecutive frames marked as highlights—so that segment-level properties can be analyzed. The resulting event set for pair (j,k) is

$$\mathbf{E_{j,k}} = \{e_1, e_2, ...\}. \tag{3.8}$$

At this stage, to discard pairs that fail to differentiate highlights from non-highlights, we introduce a new filtering strategy that consists of the application of the two filters described below, which behavior is controlled by a set of parameters whose values and interpretation will be further detailed later in the results section (Subsection 4.4.3).

Distribution-based filter. Let $\{p_b\}^{B}_{b=1}$ be the empirical probabilities obtained by partitioning the score range $[0,1]$ into B equal-width bins and tallying the highlight scores $\mathbf{score}^{HL}_{j,k}$. The Shannon entropy for sentence pair (j,k) is then

$$\Lambda_{j,k} = -\sum_{b=1}^{B} p_b \log_2 p_b. \tag{3.9}$$

A perfectly uniform histogram—where every bin is equally likely—reaches the maximum entropy $\Lambda_{j,k} = \log_2 B$. Such a flat distribution means the highlight scores do not favor any particular region of the interval, and thus the pair offers no discriminative power. Conversely, lower entropy indicates that the scores concentrate in specific bins, signaling a stronger separation between highlights and non-highlights. We therefore discard any pair whose normalized entropy satisfies $\frac{\Lambda_{j,k}}{\log_2 B} > \tau_\Lambda$, where τ_Λ is the entropy threshold.

We compute the histogram independently for each sentence pair (j,k), using all frame-level scores $\mathbf{score}^{HL}_{j,k}$ over time. Intuitively, an informative pair yields a non-uniform score distribution—typically with many frames concentrated in lower-score bins (non-highlights) and a smaller subset reaching higher scores (highlights), although other structured, non-uniform patterns may also occur. This non-uniformity leads to lower entropy, indicating that the pair

[3] In the remainder of this thesis we report only $\mathbf{score^{HL}_{j,k}}$; the non-highlight curve is simply its complement, $\mathbf{score^{NHL}_{j,k}} = 1 - \mathbf{score^{HL}_{j,k}}$, and adds no extra information.

carries discriminative information; nearly-uniform score distributions suggest the pair does not separate highlights from non-highlights and is therefore discarded.

Mean event area filter. For each event $e \in \mathbf{E_{j,k}}$, let

$$area(e) = \sum_{t \in e} score^{HL}_{t,j,k}, \tag{3.10}$$

where t indexes the video frames within the temporal extent of event e. The quantity $area(e)$ represents the event's total highlight score [84]. We then compute the mean event area for the (j, k) pair by averaging over all such events:

$$\overline{area}_{j,k} \;=\; \frac{1}{|\mathbf{E_{j,k}}|} \sum_{e \in \mathbf{E_{j,k}}} area(e), \tag{3.11}$$

where $|\mathbf{E_{j,k}}|$ denotes the total number of events in $\mathbf{E_{j,k}}$. This mean value serves as a concise measure of how strongly each pair of sentences (*HL* and *NHL*) distinguishes between highlight and non-highlight frames.

Next, we collect the mean event areas from all (j, k) pairs into a single set, $\mathcal{A} = \{\, \overline{area}_{j,k} \mid \forall (j, k)\}$. Let $\mathcal{A}_{\min} = \min(\mathcal{A})$ and $\mathcal{A}_{\max} = \max(\mathcal{A})$. We then define an area threshold, τ_{area}, as

$$\tau_{\text{area}} = \mathcal{A}_{\min} + \frac{\mathcal{A}_{\max} - \mathcal{A}_{\min}}{D}, \tag{3.12}$$

where D is a histogram division factor. Any pair whose mean event area is below τ_{area} is discarded. Increasing D lowers the threshold, removing fewer pairs, whereas decreasing D raises the threshold and filters out more pairs. This reflects the intuition that pairs producing higher mean event areas more effectively distinguish highlights from non-highlights.

Figure 3.15 provides example outcomes for several *HL-NHL* sentence pairs, demonstrating how the distribution-based and mean-area filters operate in two different sports.

Once sentence pairs failing either filter have been discarded, we average the valid frame-level predictions:

$$\overline{\mathbf{score}}^{HL}_i = \frac{1}{|\mathcal{V}|} \sum_{(j,k) \in \mathcal{V}} score^{HL}_{t,j,k}, \tag{3.13}$$

where $\mathcal{V}$ is the set of valid pairs (j, k) passing both filters, and $|\mathcal{V}|$ is the total number of such pairs. This averaging step yields a single, more robust set of frame-level predictions. We then reapply the post-processing from [84] on $\overline{\mathbf{score}}^{HL}_i$ to obtain the final highlight events.

We average across the filtered set of valid pairs to reduce sensitivity to prompt wording and to obtain a more stable estimate. A max aggregation can over-emphasize spurious high responses from a single pair (i.e., prompt noise), whereas averaging after discarding low-quality pairs tends to be more robust.

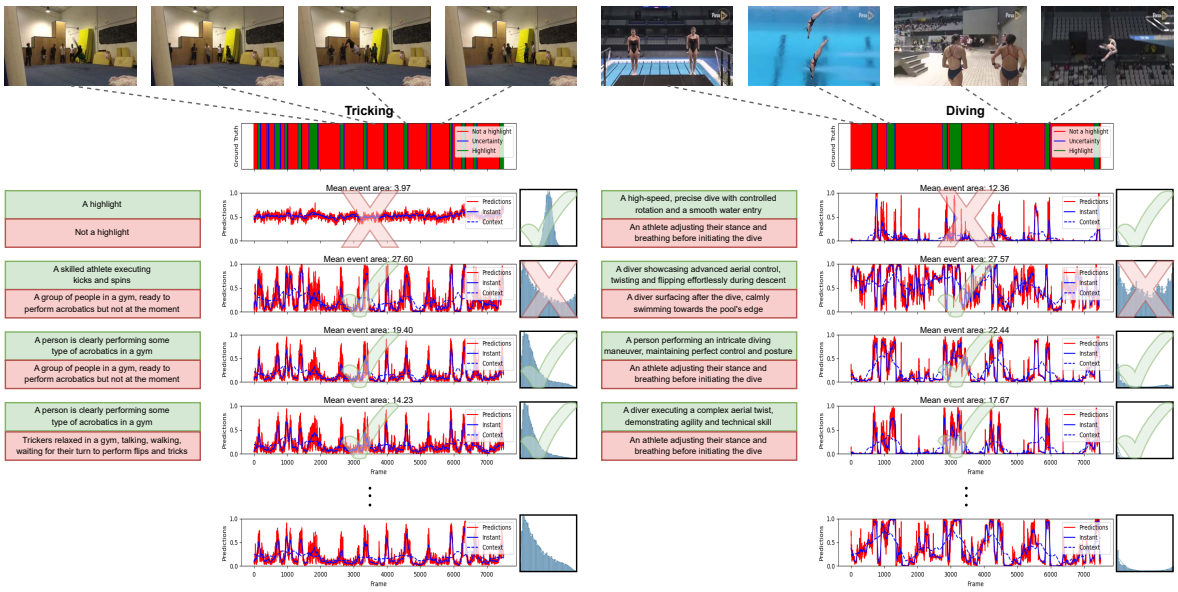


Figure 3.15: Illustration of the two-stage sentence-pair filtering approach across two different sports. Each row corresponds to a distinct *HL–NHL* sentence pair and displays (1) the frame-level highlight prediction curve (only $\mathbf{score}^{\mathbf{HL}}_{\mathbf{j,k}}$ is shown for clarity, since $\mathbf{score}^{\mathbf{NHL}}_{\mathbf{j,k}}$ is its complementary) alongside the mean event area derived from those predictions after post-processing, and (2) the probability distribution of highlight scores. A red cross or a green tick indicates whether the pair fails or passes the distribution-based and mean-area filters described in the text. Sentence pairs that fail either criterion are removed, while those that pass both are used in the final averaging step, thereby producing a more robust summarization result. In each case, the final set of predictions corresponds to the averaged (i.e., aggregated) $\overline{\mathbf{score}}^{HL}_{i}$ after filtering, which forms the basis for the final highlight summaries (depicted in the last row of the figure).

Additionally, events spanning less than half a second (that is, 15 frames at 30 fps) are removed, as such short segments rarely represent meaningful highlights. Finally, we reject any remaining events whose area is below the same threshold used in the second filter (i.e., the lower half of the mean-area histogram). This final pruning ensures that only events with sufficiently large total highlight scores survive, thereby improving overall summarization quality.

Taken together, the components described above define a text-guided, zero-shot framework that casts highlight detection as a simple, binary decision in CLIP's joint image–text space, controllable through compact *HL/NHL* prompts rather than sport-specific training. By combining a strong off-the-shelf vision–language backbone with the proposed two-stage filtering and aggregation strategy, the method produces stable, semantically meaningful highlight curves that are substantially less sensitive to prompt choices and remain applicable across very different sports. Reusing the post-processing and evaluation protocol from Section 3.2 allows us to compare this language-guided formulation directly with the classical, motion-based baseline, isolating the benefits of multimodal representations and open-vocabulary operation under a shared experimental setup. As will be shown in Section 4.4, this framework yields high-quality summaries across multiple datasets while preserving interpretability and ease of deployment, and it serves as a central building block for the more advanced, personalized video summarization systems developed later in this thesis.

3.4 Face Recognition

Face recognition is the second core component of this thesis. While video summarization identifies *what* happens and *where*, face recognition addresses the complementary question of *who* is present. In the context of this thesis, this must be done under demanding conditions: subjects are often far from the camera, captured at low resolution, partially occluded, and recorded under changing illumination. Within the PVS pipeline, these two views meet: the summarizer proposes candidate highlight segments, and the recognition stream decides which of those segments actually contain a given target identity.

Conceptually, all face recognition systems used in this thesis follow the same basic pipeline. A detector first localizes faces in each frame, with particular emphasis on robustness to tiny, distant faces and to partial occlusions. Detected regions are then normalized through light-weight alignment so that the recognition model can focus on identity-bearing cues rather than pose or framing. A backbone network—either a convolutional neural network (CNN) or a Vision Transformer (ViT)—maps each aligned crop to a compact embedding, typically normalized so that simple similarity measures reflect identity proximity. Finally, these embeddings are compared against a gallery of reference images: in an identification setting the goal is to retrieve the correct identity from a closed set, whereas in verification the goal is to decide whether two faces belong to the same person. The precise metrics and operating points used for these tasks are defined in Chapter 4, but throughout the experimental work the emphasis is on achieving reliable decisions when distance and occlusion make the problem most challenging.

The first contribution in this strand is UPM-GTI-Face, a dataset and end-to-end system designed to isolate exactly those stressors. As described in Section 3.5, the dataset factors

capture distance and occlusions (via mask usage) as controlled variables under realistic, surveillance-like acquisition, and pairs this design with a complete detection–recognition pipeline. This combination provides both a practical baseline and a diagnostic tool: it quantifies how performance degrades as subjects move farther from the camera or become occluded, and it offers a controlled yet realistic benchmark for studying recognition models under these conditions.

Building on this foundation, Section 3.6 investigates which backbone architectures are best suited for the scenario defined by UPM-GTI-Face. CNNs and ViTs are trained under a unified protocol and evaluated across a range of public datasets and on UPM-GTI-Face, considering both identification and verification scenarios. This comparative study clarifies how local convolutional features and global self-attention behave when faces are small, occluded, or captured at a distance, and which trade-offs arise in terms of robustness and efficiency. The resulting insights directly inform the design of the face-analysis stream in the PVS system: they determine which detectors and recognizers are adopted, how galleries are constructed, and how conservative the decision thresholds must be to avoid identity contamination in highlight reels.

Taken together, these elements define the operating conditions under which face recognition must function in the remainder of the thesis and provide the tools to meet them. The next sections introduce the UPM-GTI-Face dataset and baseline system, develop the CNN–ViT comparison in more detail, and finally connect these components to the PVS pipeline, where they turn generic highlight proposals into identity-aware, athlete-specific summaries.

3.5 UPM-GTI-Face A dataset for the evaluation of the impact of distance and masks in face detection and recognition systems

The previous sections have developed the first core component of this thesis—highlight detection, culminating in the flexible, language-guided framework [86]—and have outlined the role, operating conditions, and generic pipeline of the second core component, face recognition (Section 3.4). We now move from this high-level description to our first concrete contribution in the face-recognition strand: *UPM-GTI-Face: A dataset for the evaluation of the impact of distance and masks in face detection and recognition systems* [87].

To achieve the final goal of personalized video summarization, the system must not only find what is interesting but also who is present. This is uniquely challenging in sports contexts, where athletes are often distant from the camera, in constant motion, and partially occluded by equipment. This work directly confronts the practical stressors of occlusions and varying capture distances that arise in real-world sports footage. Before we could build or compare robust recognition models (which is the topic of Section 3.6), we first needed a specialized benchmark to rigorously measure the problem. The primary contribution of this paper is the creation of the *UPM-GTI-Face dataset* [87], the first public dataset to systematically and jointly evaluate the impact of distance and face masks (as a proxy for general occlusion) on recognition performance.

The work detailed here thus serves two critical purposes in the thesis. First, it provides the novel dataset and evaluation protocol used for our subsequent research. Second, it establishes a crucial performance baseline by detailing a complete End-to-End (E2E) system (Section 3.5.2) and testing it against this new, challenging benchmark.

The remainder of this section is organized as follows: Section 3.5.1 describes the acquisition methodology and structure of the UPM-GTI-Face dataset. Section 3.5.2 details the full detection and recognition pipeline used to establish our baseline results.

3.5.1 Dataset

UPM-GTI-Face is a new public[4] dataset, which will allow the study of the impact of distance and the use of masks on the results of face detection and recognition strategies. Despite the existence of other datasets similar in spirit (see Section 2.2), ours represent the first dataset that addresses the joint impact of distance and face masks. Additionally, we provide distance annotations in a rigorous manner, as opposed to other datasets that report distances qualitatively, with distance categories such as "far", or "close".

UPM-GTI-Face is composed of $4K$ images from 11 different subjects (8 men and 3 women), under 2 environments (indoors and outdoors), and 2 face mask conditions (subjects with and without face masks). For each combination of subject, environment, and face mask condition, the database includes a mugshot image at a distance of 1 meter, and 10 probe images acquired from 3 to 30 meters (at 3 meter intervals). As a result, UPM-GTI-Face contains 484 face images (22 indoor gallery images, 22 outdoor gallery images, 220 indoor probes and 220 outdoor probes). Figure 3.16 shows a sample of the images for one of the subjects in the dataset.

In an attempt to imitate surveillance scenarios, images were acquired using a Logitech BRIO Webcam camera placed on a tripod, at a height of 2.1 meters (see Fig. 3.17). People were asked to walk towards the camera, starting from a distance of 30 meters, stopping for a couple of seconds at several floor marks indicating intermediate distances, without any further cooperation from them (e.g., pose or expression). A continuous video was recorded for these sequences, while the supervisor voiced aloud the moments at which users stopped at the distance marks. This facilitated the process of extracting the frames corresponding to each distance using audio as a cue.

3.5.2 E2E Face Recognition System

The proposed E2E system is composed of both face detection and face recognition algorithms, and follows the architecture shown in Fig. 3.18.

With respect to face detection, the Tiny Faces network [45] has been used, since after exploring several options [29] it has shown to be the most robust face detection algorithm across distance and low-resolution faces. The Tiny Faces network is based on a pyramidal structure, where face estimation is done at different scales, ranging from very small ones such as 0.1 (useful to detect faces at short distances or higher resolutions) to large ones such as 1.4 or 2.0 (to

[4]The UPM-GTI-Face dataset is available at www.gti.ssr.upm.es/data.

Figure 3.16: UPM-GTI-Face dataset. 11 different subjects were captured under different environments and conditions. Environments: indoor and outdoor. Conditions: mask and no mask. For every combination of environment, condition, and subject in the dataset, there is 1 close-up shot for gallery, and 10 probe images that correspond to each distance mark.

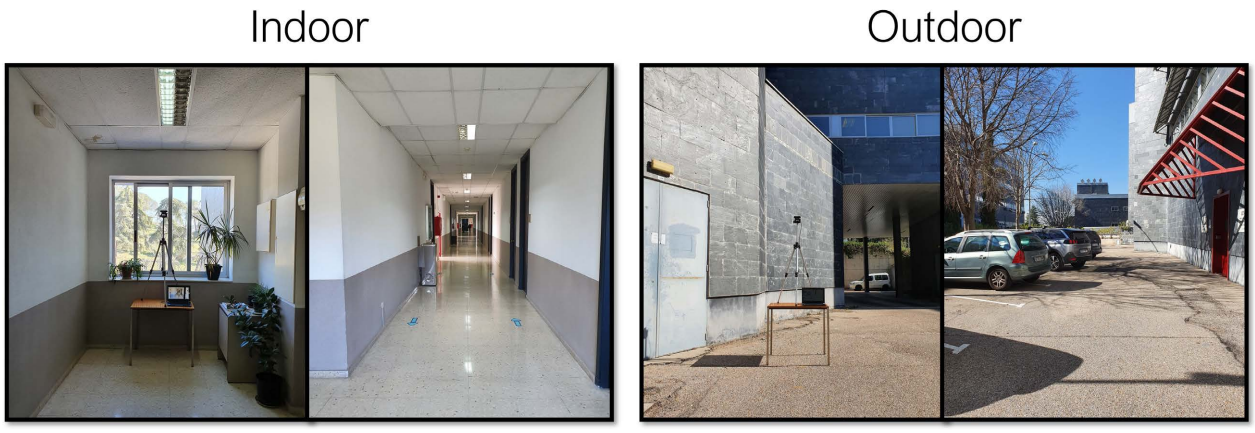


Figure 3.17: UPM-GTI-Face dataset capturing setup for both indoor and outdoor environments.

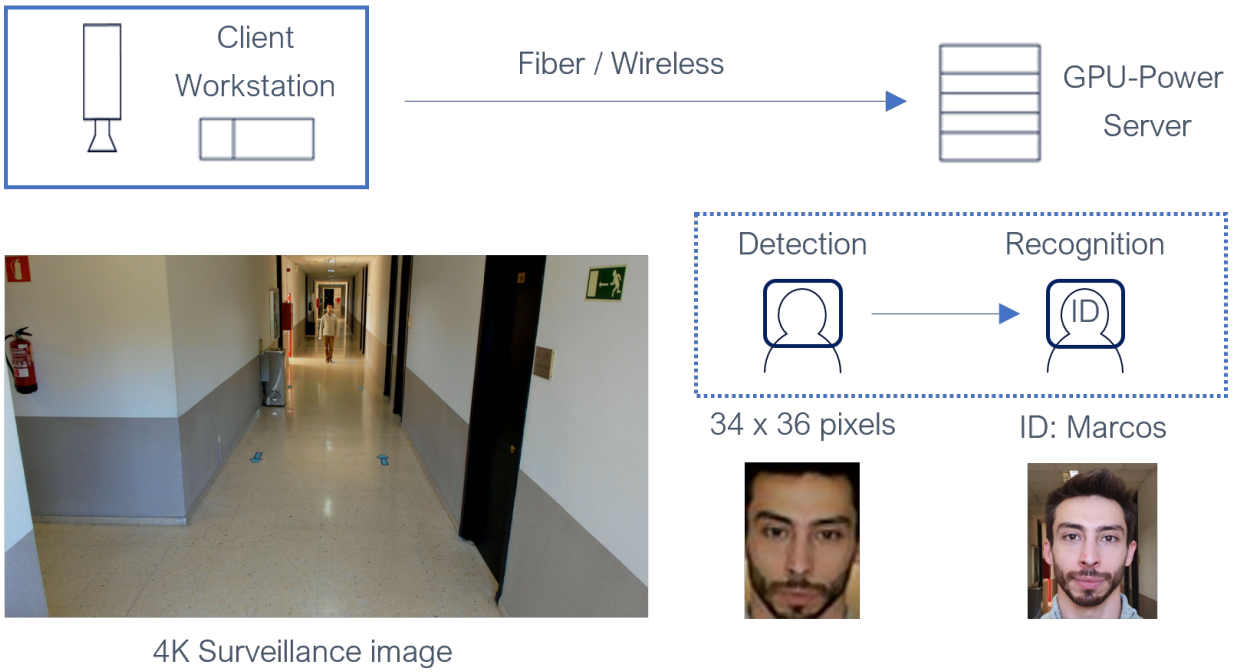


Figure 3.18: E2E system composed of a client workstation in charge of video capturing and a server in charge of face detection and face recognition.

detect very small faces). This network demonstrates that both large context and scale-variant representations are crucial. For context, it defines templates that make use of a massively large receptive field which can be effectively encoded as a descriptor that captures both coarse context (necessary for detecting small objects) and high-resolution image features (helpful for localizing small objects).

Face Recognition is based on the original VGG Face network [78]. The UPM-GTI-Face dataset aims to mimic a surveillance scenario, where normally there are very few samples from the subject to be recognized. Therefore, we decided to use a face recognition network from which highly robust face embeddings from surveillance and gallery images could be extracted. This way, we have explored the following backbones: *i*) VGG16, *ii*) ResNet-50 and *iii*) SeNet50 (from Squeeze-and-Excitation network, winner of the ISLVR 2017 Classification competition [43]). Notice that in all cases, the different backbones where trained using one of the largest public face datasets, VGG-face dataset, composed of 2662 subjects, with 375 images each, making a total of almost 1 million images. Face recognition scores have been obtained by comparing face embeddings from gallery images and probe images using the cosine distance as a metric [121].

The proposed system has been developed following a client-server architecture with TCP-based communication using Flask library. At the side of the client, an Intel Core i7-10710U laptop is in charge of capturing videos from the Logitech Brio Webcam able to acquire $4K$ videos (16MP, 30fps, 90° FOV). At the server side, both face detection and recognition are executed in a Intel(R) Core(TM) i9-10900K CPU @ 3.70GHz and a NVIDIA GeForce RTX 3090 workstation. Both algorithms have been implemented in Python, using Tensorflow and Keras

deep learning libraries.

Together, the UPM-GTI-Face dataset and the accompanying E2E system provide a concrete, application-driven foundation for the face recognition strand of this thesis. By explicitly modeling distance and mask usage under surveillance-like capture conditions, the dataset exposes precisely the factors that most undermine recognition performance in realistic sports scenarios, while the baseline pipeline—combining Tiny Faces detection with VGG-based embeddings and a practical client–server deployment—clarifies how these factors propagate through a complete system. This combination of a rigorously annotated benchmark and a transparent, end-to-end architecture not only supports fair comparison between alternative backbones and configurations, but also offers a reliable testbed for stress-testing robustness in the presence of occlusions, low resolution, and limited gallery data. As will be shown in the subsequent Section 3.6 and in the results section (Section 4.5), the insights derived from this setup are instrumental in guiding the choice of modern architectures for the final personalized video summarization pipeline, ensuring that identity information remains dependable even under the demanding viewing conditions typical of sports footage.

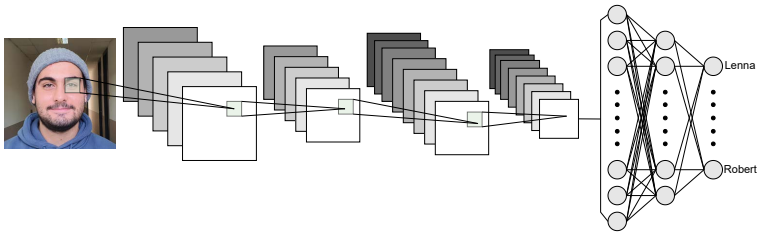
3.6 Comprehensive comparison between vision transformers and convolutional neural networks for face recognition tasks

The previous section initiated our exploration of the face recognition domain, a critical component of the final PVS system. That work [87] identified the core practical challenges—namely varying capture distances and occlusions—and provided a rigorous benchmark, the *UPM-GTI-Face dataset*, along with a baseline End-to-End (E2E) system to test against it.

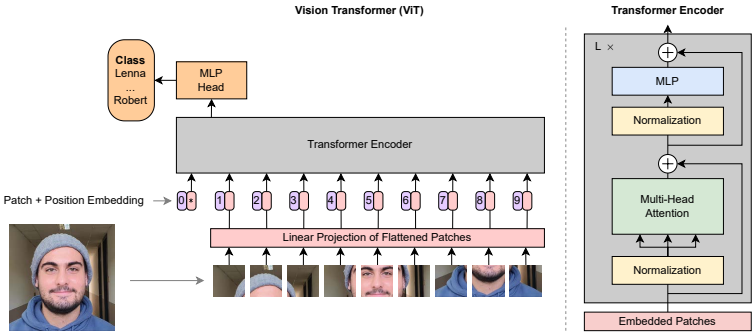
With a clear benchmark for these practical stressors in place, the next logical step is to determine the optimal backbone architecture for the task. This section details our comparative study, *Comprehensive comparison between vision transformers and convolutional neural networks for face recognition tasks* [85], which directly addresses the methodological goal of comparing Convolutional Neural Networks and Vision Transformers as backbone families.

The core of this investigation is to understand how the fundamental architectural differences between CNNs (built on local receptive fields) and ViTs (built on global self-attention) impact embedding quality, robustness to occlusion, and resilience to variance in capture distance. This aligns with our guiding principle of finding representations that balance generality with task-specific robustness. To perform this comparison, we train and evaluate multiple architectures from both families under a unified framework, testing them on a wide range of public datasets, including LFW [46], SCface [30], ROF [22], and our own *UPM-GTI-Face* [87].

The remainder of this section is structured as follows. Section 3.6.1 first introduces the core concepts of face identification and verification. Following this, Section 3.6.2 provides a detailed technical comparison of the operational mechanisms of ViT and CNN architectures, setting the stage for the experimental results presented in Chapter 4.



(a) Common CNN architecture



(b) Vision Transformer architecture

Figure 3.19: Visual depiction of the fundamental differences between Vision Transformers (a) and Convolutional Neural Networks (b) architectures and operational mechanisms. Original image from [66].

3.6.1 ViT for Face recognition

The field of face recognition can be broadly classified into two sub-tasks: face identification and face verification [20]. Face identification involves the determination of whether a particular face image corresponds to a person in a given dataset and can be treated as a one-to-many problem. In contrast, face verification involves determining the similarity between two face images, and is a one-to-one problem. Face identification differs from face verification in that the former can be formulated as a classification problem and so, has prior knowledge of the number of classes, whereas the latter does not. This influences the selection of appropriate loss functions for each task.

For both tasks, face embeddings, also known as face descriptors, are high-level representations of a face image obtained in the final layers of a Deep Neural Network (DNN). Ideally, these embeddings should have low intra-class variance and high inter-class variance, meaning that they are similar for images of the same subject and distinct for images of different subjects. In the case of face identification, this is typically achieved implicitly through a softmax loss function, while in face verification, it is usually achieved explicitly through the use of a different loss function (e.g., Triplet-loss [88], ArcFace [17], CosFace [113], or SphereFace [62].

3.6.2 Differences between ViTs and CNNs

The operational mechanisms of ViTs diverges significantly from that of CNNs, as illustrated in Fig. 3.19. CNNs operate through a series of layers, each of which detects different features in an image by applying various filters, and sends them to the subsequent layer. These filters

can initially begin detecting very simple features, and increase in complexity to detect features that uniquely identify a specific object or category.

In contrast, ViTs break an image into patches, which are flattened and turned into a series of tokens through a linear projection to enable them to be fed into the network (originally developed for natural language processing applications, and thus designed to accept word-like inputs). These tokens are passed through a series of transformer encoder layers, which exhibit a remarkable capacity for understanding how different parts of the picture relate to each other. This is achieved by means of a multi-head self-attention mechanism. Each head learns diverse dependencies by generating three representations (query, key, and value) for each input token, which are then processed to obtain an output representation. These output representations are subsequently transmitted to the next transformer encoder layer. These representations are similar to the activation maps outputted by the filters of a CNN in that they serve to detect features that uniquely define a specific object or category, but there are some significant differences.

One of the major differences between ViTs and CNNs lies in the large field of view of the initial ViTs' layers. CNNs employ a fixed-size kernel field of view, gradually extending it by repeatedly convolving the information around the kernel layer by layer. In contrast, ViTs utilize a self-attention mechanism that enables the model to have a whole field of view, even at the lowest layer. Hence, ViTs obtain global representations from the beginning, while CNNs need to propagate layers to obtain global representations. This can also be a disadvantage for ViTs, which require large amounts of data to obtain local representations in the lowest layers. However, this can be addressed through a well-designed pre-training strategy.

ViTs and CNNs also differ significantly in terms of their memory footprint [64]. During training, CNNs must save all intermediate activation maps resulting from the performed convolutions. These activation maps have a significant impact on the memory footprint during training, and can quickly restrict the maximum number of samples in a batch, which in turn can affect the ability of the model to converge. ViTs, on the other hand, are much less affected by this issue, as the initial step divides the image into tokens, which have a much smaller memory footprint than activation maps during training.

Taken together, this study positions the ViT–CNN comparison as the methodological bridge between the controlled benchmark introduced in UPM-GTI-Face and the practical needs of the final personalized video summarization pipeline. By analyzing identification and verification performance across both classical (LFW, SCface, ROF) and stress-test datasets (UPM-GTI-Face), under a unified training and evaluation protocol, it becomes possible to disentangle how architectural choices—local convolutions versus global self-attention—translate into concrete gains or failures under distance, occlusion, and low-resolution conditions. The resulting picture is not limited to headline accuracy numbers: it clarifies the trade-offs in memory footprint, data requirements, and robustness that must be considered when deploying a face recognizer in realistic sports-like scenarios with scarce gallery data. As will be detailed in Section 4.6, these insights lead to a principled selection of backbones for subsequent experiments, ensuring that the identity component of the PVS system is grounded in architectures whose strengths have been systematically validated rather than adopted purely on the basis of recency or popularity.

3.7 Personalized Video Summarization

The preceding sections established the two methodological strands underpinning this thesis: video summarization (the *what* and *where*) and face recognition (the *who*). We first developed robust highlight detectors, progressing from a classical, motion-based approach [84] to a flexible, text-guided framework [86]. In parallel, we characterized the core challenges of face recognition in sports—distance and occlusion—by introducing a new benchmark [87] and conducting a comparative study of architectural families [85].

This section integrates both strands into a single end-to-end system: the Personalized Video Summarization (PVS) pipeline. PVS takes as input a full sports video and one or more target individuals, specified through face images, and produces as output a set of identity-aware highlight clips for each person. Conceptually, the system decouples event discovery from identity analysis. A video summarizer model proposes candidate segments that are narratively salient, while a separate face-analysis pipeline runs over the full video to determine who appears at each point in time. The two streams are then fused in time to obtain personalized highlight selections.

Figure 3.20 summarizes the overall architecture of the PVS pipeline. Starting from the input video and a set of target faces, the system splits into two main processing branches. In the upper branch, a face-detection module locates and crops faces, a face-recognition module converts these crops into embeddings, and a resemblance-computation block turns the embeddings into frame-wise similarity scores and temporally smoothed resemblance curves; an optional target-update block can refine the reference embeddings and crops using high-confidence in-video detections. In the lower branch, a video summarization model processes the same input video and produces a set of candidate highlight segments. In the final stage, a clip-selection and results block combines the highlight segments with the resemblance measurements, assigns segments to the relevant targets according to a similarity criterion, and outputs the personalized highlight clips together with their associated statistics.

The remainder of this section is structured as follows. Section 3.7.1 formalizes the personalized video summarization problem, introduces the main design goals, and describes the overall architecture at a conceptual level. Section 3.7.2 then details the face-analysis stream, from full-video detection to embedding extraction, target refinement, and the construction of temporal resemblance curves. Section 3.7.3 describes the video-summarization stream and its text-guided scoring mechanism. Finally, Section 3.7.4 defines the clip-selection rule that fuses both streams into identity-aware highlight sets. Quantitative and qualitative evaluations of this system are presented later in Section 4.7.

3.7.1 Formulation and overall design

We consider a full-length sports video represented as a sequence of T frames

$$V = \{V_t\}_{t=1}^{T}, \tag{3.14}$$

where V_t denotes the frame at index t. We also assume a user-provided textual description q (or a small set of descriptions) specifying the types of events of interest (e.g., “high jump attempts” or “vault landings”).

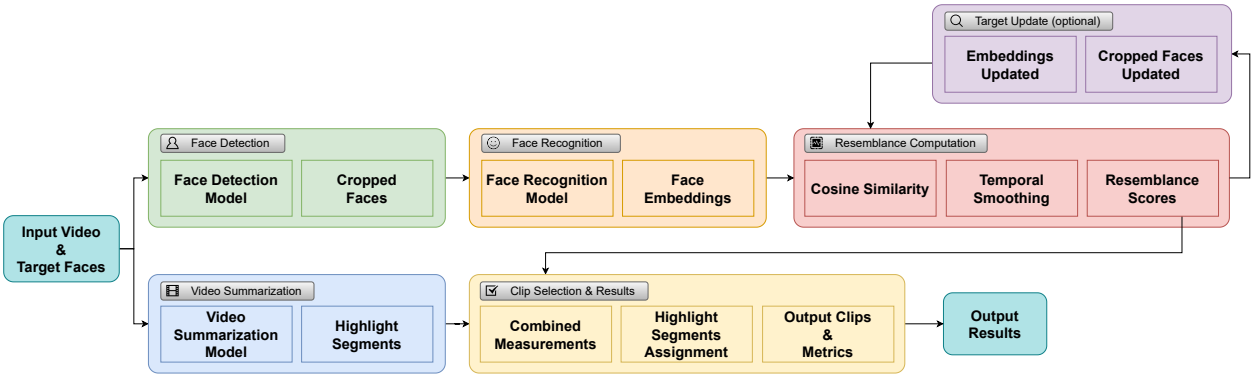


Figure 3.20: Block diagram of the Personalized Video Summarization (PVS) pipeline. From the input video and target faces, a face-analysis stream (green, orange, red, and optional purple blocks) performs face detection, face recognition, resemblance computation, and optional target updating, producing temporal resemblance scores for each target. In parallel, a video summarization stream (blue block) generates generic highlight segments. A final clip-selection and results module (yellow block) fuses both sources of information to decide which segments belong to each target and yields the personalized highlight clips and quantitative metrics.

Personalization is defined with respect to a set of target individuals

$$\mathcal{P} = \{p_1, \ldots, p_M\}, \tag{3.15}$$

where each target p_m is represented with a single reference image R_m. The goal of PVS is to automatically produce, for each p_m, a compact set of highlight clips in which (i) the depicted content is narratively relevant according to q, and (ii) the target appears clearly and reliably.

In this thesis, we call a segment "narratively relevant" when it is (i) semantically aligned with the user query q according to the summarization model's scoring, and (ii) forms a self-contained, broadcast-meaningful unit (e.g., a single attempt, action execution, or outcome) rather than transitional footage.

We denote by $\mathcal{H}(V, q)$ the set of candidate highlight segments returned by a generic, identity-agnostic summarizer, and by $\mathcal{S}(V, p_m)$ the identity evidence extracted from the video for target p_m. The PVS pipeline aims to construct, for each p_m,

$$\mathcal{H}^{\star}_{p_m}(V, q) \;\subseteq\; \mathcal{H}(V, q), \tag{3.16}$$

such that every segment $h \in \mathcal{H}^{\star}_{p_m}(V, q)$ is both relevant to q and supported by sufficient identity evidence for p_m. The way in which relevance and identity evidence are quantified is made explicit in the following subsections.

Several design goals guided the construction of this system.

- **Separation of concerns.** The system maintains a clear separation between event discovery and identity analysis. A text-guided video summarization model is responsible solely for proposing narratively salient segments, while a separate face-analysis stream models who appears in the video. This separation allows us to reuse and compare different summarizers without altering the face-analysis components, and vice versa.

- **Modularity and reusability.** Each major component—summarizer, detector, embedder, target handler, resemblance computation, and clip selector—exposes a simple, file-based interface: it reads standardized inputs and writes standardized outputs. This design makes it possible to swap models, change hyperparameters, or extend the system to new sports or backbones without rewriting the pipeline.
- **Full-video identity grounding.** Athletes often appear outside of highlight segments, for instance during preparation, celebrations, or transitions. To capture this, PVS analyzes faces in the full video rather than only inside candidate clips. Identity information thus becomes independent of any particular summarizer, and the same face-analysis stream can be reused across multiple highlight generators and textual queries.
- **Multi-identity, multi-backbone support.** The system is designed to handle multiple target individuals and multiple recognition backbones. This is crucial for the experimental goals of the thesis: comparing CNN-based and Transformer-based embeddings under realistic sports conditions, and studying how different choices of target representation (user-provided images versus adapted in-video references) affect resemblance scores and downstream selection.
- **Interpretability.** Finally, the system is built to provide interpretable intermediate representations, not just a list of selected clips. Temporal resemblance curves summarize, in a single plot per target and model, when the system believes the athlete is present and with what degree of confidence.

At a conceptual level, these goals lead naturally to the architecture sketched in Figure 3.20. The video-summarization stream takes (V, q) and produces a set of highlight segments

$$\begin{aligned} \mathcal{H}(V, q) &= \{h_1, \ldots, h_N\}, \\ h_i &= [t_i^{\text{start}}, t_i^{\text{end}}], \end{aligned} \tag{3.17}$$

where t_i^{start} and t_i^{end} denote the start and end frame indices of the segment (and thus its temporal extent), and each h_i is associated with a clip extracted from V.

In parallel, the face-analysis stream takes $(V, \mathcal{P})$ and, for each target p_m, produces a temporal resemblance signal $s_{p_m}(t)$ defined over frame indices $t = 1, \ldots, T$, which estimates how likely it is that p_m appears in the video at time t. The final clip-selection stage fuses these two sources of information by deciding, for each pair (h_i, p_m), whether there exists sufficient evidence in $s_{p_m}(t)$ over the interval $[t_i^{\text{start}}, t_i^{\text{end}}]$ to assign segment h_i to target p_m. The resulting subsets $\mathcal{H}_{p_m}^{\star}(V, q)$ constitute the identity-aware highlight sets that PVS returns.

Scope of applicability. Although the proposed methods are presented under the general umbrella of sports video summarization, the thesis primarily targets *broadcast* videos of sports that can be decomposed into *discrete, temporally localized attempts or actions* (e.g., an athlete's individual run/attempt/sequence), for which highlight segments are well-defined and can be associated with a specific identity. This setting naturally covers many individual Olympic-style disciplines and similar sports where highlights correspond to short, salient events separated by non-highlight context. Extensions to domains with fundamentally different structure—such as team sports with overlapping actions, endurance races with long continuous activity, or combat sports without a consistent pre-attempt "ritual"—are not the main focus

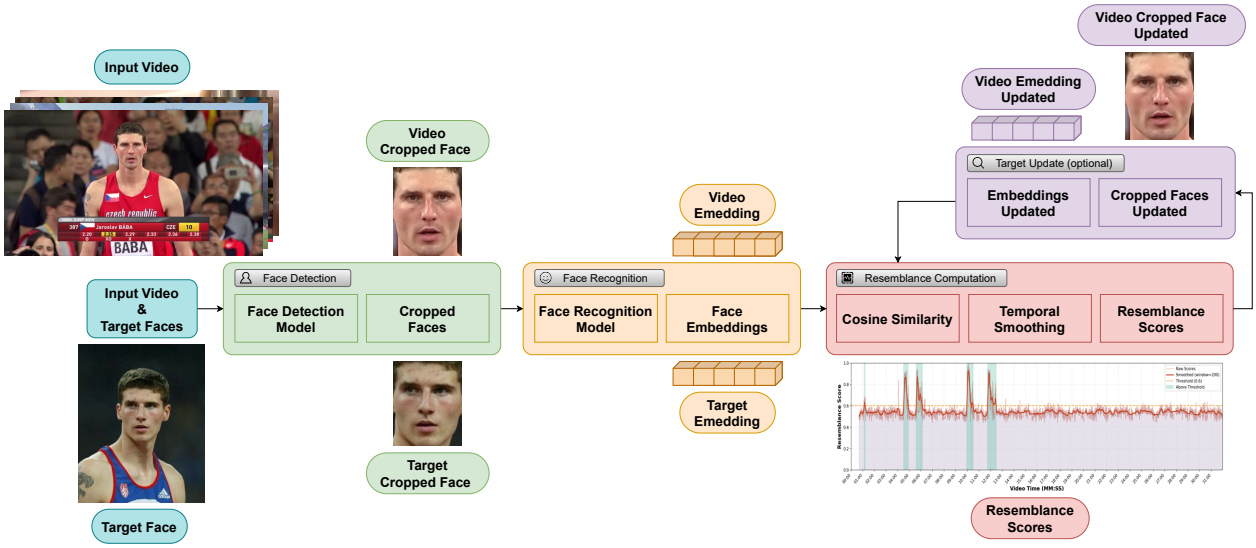


Figure 3.21: Detailed view of the face-analysis stream in the PVS pipeline, illustrated with example inputs and outputs. Starting from an input broadcast video and a reference target face (left), the pipeline (i) detects and crops all visible faces, (ii) maps every crop and the target image to high-dimensional embeddings with a recognition backbone, and (iii) computes frame-level resemblance scores via cosine similarity, applies temporal smoothing, and produces the resemblance score curve for the target (bottom right). An optional fourth step updates the target crop and embedding using the most confident in-video match (top right); when this refinement is enabled, step (iii) is run again to obtain updated resemblance scores.

of this thesis and would require additional modeling of event structure (e.g., defining what constitutes an event/highlight in that sport) and, in some cases, different cues beyond localized motion/appearance (e.g., game-state, temporal context over longer horizons, or multi-actor interactions).

3.7.2 Face-analysis stream

The face-analysis stream is responsible for turning raw video frames and reference images into temporal identity evidence. It comprises four main components: face detection, face recognition, resemblance computation, and an optional target-update mechanism. Together, these stages produce, for each target and backbone, a temporal resemblance curve that quantifies how strongly the video supports the presence of that target over time. A more detailed view of this branch is shown in Figure 3.21.

3.7.2.1 Face detection

For each frame V_t in the video, a face detector maps the image to a set of face hypotheses. Let ϕ^{SCRFD} denote the detection function implemented by the SCRFD model [31]; then

$$\mathcal{D}_t = \phi^{\text{SCRFD}}(V_t) \;=\; \{(B_{t,k}, \ell_{t,k}, c_{t,k})\}_{k=1}^{K_t}, \tag{3.18}$$

where $B_{t,k}$ is a bounding box, $\ell_{t,k}$ denotes the predicted facial landmarks (five 2D keypoints: left eye, right eye, nose tip, left mouth corner, and right mouth corner), $c_{t,k} \in [0,1]$ is a detection confidence, and K_t is the number of detected faces in frame t. In practice, we adopt the SCRFD detector, a modern one-stage method that is designed to redistribute training samples and computation, achieving a strong accuracy–efficiency trade-off while requiring significantly fewer FLOPs than prior detectors such as TinaFace [134]. Compared to the Tiny Faces detector [45] used in the E2E system of Section 3.5.2, SCRFD offers comparable robustness to small, low-resolution faces but with a lighter, GPU-friendly architecture and built-in landmark prediction, making it better suited for processing every frame of long sports broadcasts in the PVS pipeline.

Each element of $\mathcal{D}_t$ yields a cropped face image

$$C_{t,k} = V_t\big|_{B_{t,k}}, \tag{3.19}$$

which is stored on disk together with its metadata $(t, k, B_{t,k}, \ell_{t,k}, c_{t,k})$. This operation is applied to every frame, independently of the highlight segments, producing a dense map of all visible faces in V.

3.7.2.2 Face recognition

To model identity, PVS maps each cropped face to one or more points in an embedding space. Building on the comparative study of CNN-based and Transformer-based backbones in Section 3.6, we select one strong, representative model from each family: ArcFace [17] as a margin-based convolutional baseline, and TransFace [14] as a modern Transformer-based alternative. Let ϕ^{Arc} and ϕ^{Trans} denote the embedding functions implemented by these two backbones, respectively. For every crop $C_{t,k}$ the system computes

$$\begin{aligned} \mathbf{e}_{t,k}^{\text{Arc}} &= \phi^{\text{Arc}}(C_{t,k}) \in \mathbb{R}^{512}, \\ \mathbf{e}_{t,k}^{\text{Trans}} &= \phi^{\text{Trans}}(C_{t,k}) \in \mathbb{R}^{512}. \end{aligned} \tag{3.20}$$

These embeddings are stored in indexed containers, grouped by video and backbone, so that they can be retrieved efficiently when computing resemblance scores. In this implementation, we directly embed the detected crops without an additional landmark-based alignment step; the predicted keypoints remain available for future extensions (Section 6).

Using both backbones allows us to instantiate, within the same pipeline, the two architectural families analyzed in Section 3.6. This, in turn, makes it possible to directly compare CNN-based (ArcFace) and Transformer-based (TransFace) embeddings under identical conditions in the PVS setting and to study how each choice of backbone affects the resulting resemblance curves and personalized summaries.

3.7.2.3 Target update (optional)

Target individuals are specified by a single reference image R_m for each p_m. These images are first processed by the same SCRFD detector, yielding a set of face hypotheses

$$\mathcal{D}_m^{\text{ref}} = \phi^{\text{SCRFD}}(R_m) \;=\; \{(B_{m,j}^{\text{ref}}, \ell_{m,j}^{\text{ref}}, c_{m,j}^{\text{ref}})\}_{j=1}^{J_m}, \tag{3.21}$$

where J_m is the number of detected faces in the reference image R_m, and $B^{\text{ref}}_{m,j}$, $\ell^{\text{ref}}_{m,j}$ and $c^{\text{ref}}_{m,j}$ denote, respectively, the bounding box, landmarks and confidence of detection j.

Among these hypotheses, we select the face with the largest bounding-box area,

$$j^\star = \arg\max_{1\le j\le J_m} \operatorname{area}\Big(B^{\text{ref}}_{m,j}\Big), \tag{3.22}$$

and define the corresponding crop as

$$C^{\text{ref}}_m = R_m\Big|_{B^{\text{ref}}_{m,j^\star}}. \tag{3.23}$$

This crop is then passed through the embedding functions to obtain the *original* target embeddings:

$$\begin{aligned} \mathbf{e}^{\text{Arc,orig}}_m &= \phi^{\text{Arc}}(C^{\text{ref}}_m), \\ \mathbf{e}^{\text{Trans,orig}}_m &= \phi^{\text{Trans}}(C^{\text{ref}}_m). \end{aligned} \tag{3.24}$$

These embeddings approximate the realistic scenario in which a user uploads a single photograph and expects the system to find that person in the video without additional adaptation.

User-provided images, however, often differ substantially from broadcast footage in terms of resolution, lighting, viewpoint, and appearance. To explore the effect of better domain alignment, PVS also supports an *updated target* configuration. Starting from $\mathbf{e}^{\text{Arc,orig}}_m$, the system searches over all video embeddings $\{\mathbf{e}^{\text{Arc}}_{t,k}\}$ and finds the most similar face according to cosine similarity:

$$(t^\star, k^\star) = \arg\max_{(t,k)} \frac{\mathbf{e}^{\text{Arc,orig}}_m \cdot \mathbf{e}^{\text{Arc}}_{t,k}}{\|\mathbf{e}^{\text{Arc,orig}}_m\| \, \|\mathbf{e}^{\text{Arc}}_{t,k}\|}. \tag{3.25}$$

The corresponding crop $C_{t^\star,k^\star}$ is treated as an updated target image, and its embedding

$$\mathbf{e}^{\text{Arc,upd}}_m = \mathbf{e}^{\text{Arc}}_{t^\star,k^\star} \tag{3.26}$$

is used as a refined reference. The same procedure is performed independently for the TransFace backbone, yielding $\mathbf{e}^{\text{Trans,upd}}_m$. These updated embeddings are fully aligned with the video domain and approximate an upper bound on recognition performance given an ideal in-video reference.

In subsequent steps we simply choose, for each target and backbone, whether to use the original embedding or its updated counterpart. For notational simplicity, we denote this chosen embedding generically as $\mathbf{e}_{p_m}$. In the results section (Section 4.7), we will explicitly compare ArcFace and TransFace under both configurations (original vs. updated targets) to quantify how backbone choice and target adaptation jointly affect the resulting resemblance curves and personalized summaries.

3.7.2.4 Resemblance computation

The embeddings computed above are ultimately used to produce temporal resemblance curves, which quantify, for each target and backbone, how similar the video looks to that target over time.

Consider a fixed video V, a target p_m, and one of the recognition backbones. For each frame t, the face-analysis stream provides zero or more embeddings $\{\mathbf{e}_{t,1}, \ldots, \mathbf{e}_{t,K_t}\}$ (suppressing the backbone superscript for brevity). PVS first computes the cosine similarity between $\mathbf{e}_{p_m}$ and each face embedding in frame t:

$$s_{t,k}^{(m)} \;=\; \frac{\mathbf{e}_{p_m}^{\top} \cdot \mathbf{e}_{t,k}}{\|\mathbf{e}_{p_m}\| \, \|\mathbf{e}_{t,k}\|}. \tag{3.27}$$

These similarities are then aggregated into a single frame-level resemblance score

$$s_t^{(m)} \;=\; \begin{cases} \max_{k \in \{1,\ldots,K_t\}} s_{t,k}^{(m)}, & K_t > 0, \\ 0, & K_t = 0, \end{cases} \tag{3.28}$$

which measures how similar the most plausible face in frame t is to target p_m.

The sequence of frame-level scores

$$\{s_t^{(m)}\}_{t=1}^{T} \tag{3.29}$$

defines the *raw* resemblance curve for target p_m under the chosen backbone, indexed over video frames.

To make its temporal structure easier to interpret, PVS also computes a smoothed version by averaging scores in a short window $W(t)$ around each frame:

$$\bar{s}_t^{(m)} = \frac{1}{|W(t)|} \sum_{u \in W(t)} s_u^{(m)}, \tag{3.30}$$

where $W(t)$ contains the indices of neighboring frames (e.g., those satisfying $|u - t| \le w$ for a small window radius w). The smoothed curve $\bar{s}_t^{(m)}$ highlights sustained visibility (for example, during celebrations), while the raw curve $s_t^{(m)}$ preserves sharp peaks corresponding to brief but confident detections. An illustrative example of resemblance curves is shown in Figure 3.22.

When multiple targets $\mathcal{P} = \{p_1, \ldots, p_M\}$ are considered, the same procedure is applied independently to each one, yielding separate curves $s_t^{(m)}$ and $\bar{s}_t^{(m)}$ per target and backbone. Combined visualizations can overlay several curves on the same axes, using different colors, to show how the system's confidence shifts between athletes across time, as illustrated in Figure 3.23.

For later reference, we collect the smoothed scores into the identity-evidence representation

$$\mathcal{S}(V, p_m) = \{(t, \bar{s}_t^{(m)})\}_{t=1}^{T}, \tag{3.31}$$

which serves as the compact identity signal for target p_m used by the clip-selection stage. In practice, relying on the temporally smoothed scores $\bar{s}_t^{(m)}$—rather than the raw values $s_t^{(m)}$—makes threshold-based decisions more stable, reduces the impact of frame-level noise, and yields more consistent personalized summaries.

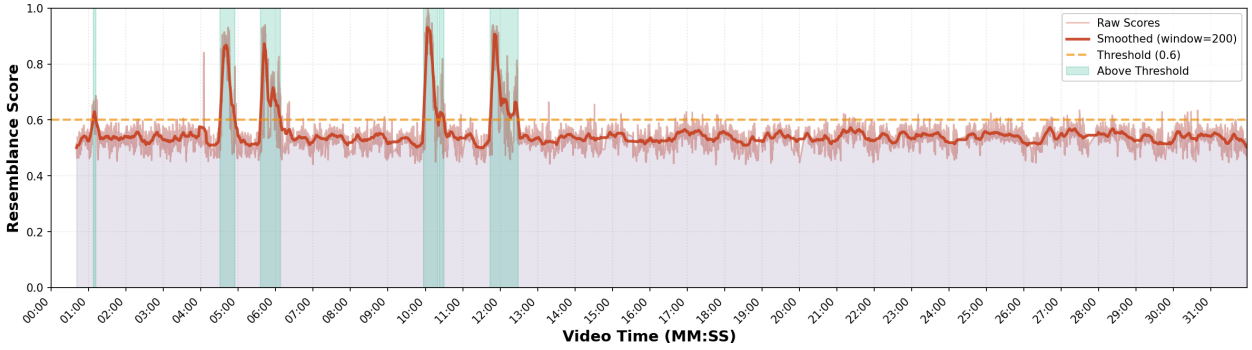


Figure 3.22: Illustrative example of temporal resemblance curves and highlight segments for a single target. The raw resemblance scores (thin line) capture frame-level similarity, while a smoothed curve (thick line) reveals broader patterns. Highlight segments are shown as shaded intervals. These curves are later combined with a similarity threshold to decide which segments are assigned to the target.

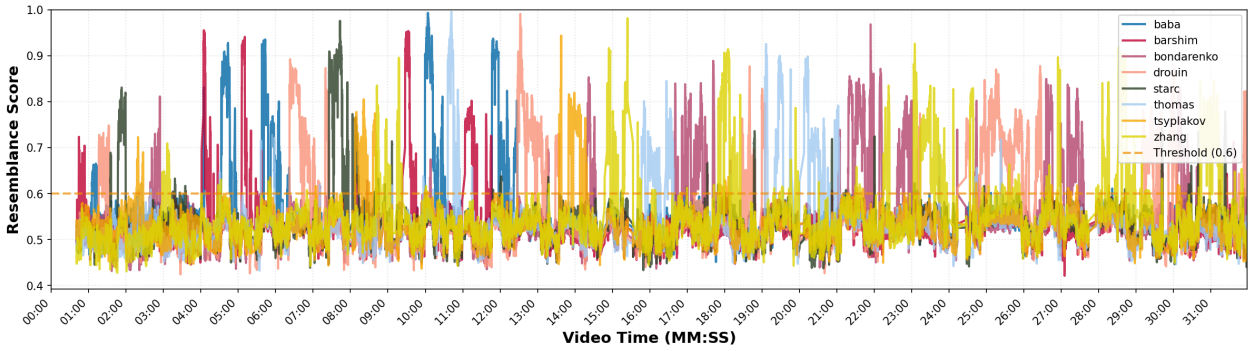


Figure 3.23: Example of combined temporal resemblance curves for multiple athletes in a single video. Each colored curve corresponds to one target, and the horizontal dashed line marks a fixed similarity threshold. Peaks above the threshold indicate time intervals where the system has strong evidence that the corresponding athlete is visible in the video.

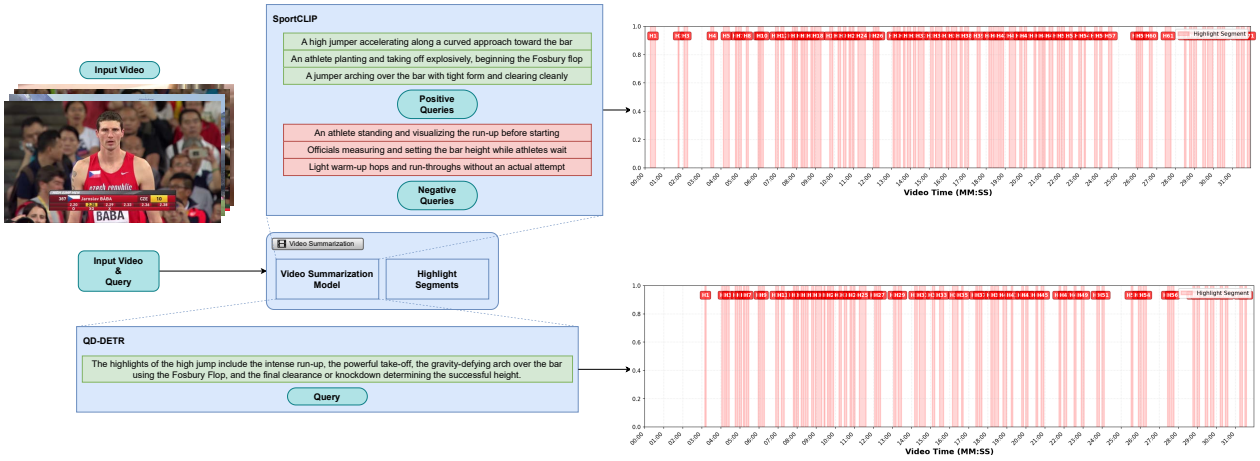


Figure 3.24: Detailed view of the video-summarization stream. The same input video is summarized with two alternative text-guided models. **Top:** SportCLIP [86] uses several positive (highlight) and negative (non-highlight) sentences, all collectively denoted by q, to describe which situations should be included or excluded; these prompts are processed to produce a highlight score curve that is discretized into highlight segments. **Bottom:** QD-DETR [72] takes a single natural-language query q describing the desired moments and outputs a saliency curve over time, which is post-processed in the same way. In both cases, the resulting highlight intervals and their corresponding clips are exported in a common metadata format for downstream identity analysis.

3.7.3 Video-summarization stream

The video-summarization stream is responsible for identifying candidate segments where interesting events occur, independently of who is involved. In the PVS pipeline, this stream is treated as a modular component with a fixed input/output interface: given a video V and a user-provided textual description q of the desired events, it returns a set of highlight segments $\mathcal{H}(V, q)$ expressed in terms of frame indices. A more detailed view is shown in Figure 3.24.

In the formulation of Subsection 3.7.1, q was introduced as a textual description specifying the types of events of interest. In practice, its exact structure depends on the chosen summarization model, but for notational simplicity we continue to denote it generically by q:

- For SportCLIP [86], q stands for a *set* of highlight and non-highlight sentences, typically written as $\{Q_j^H\}_j \cup \{Q_k^N\}_k$, that describe what we want the summary to contain (positive queries) and what we prefer to exclude (negative queries).
- For QD-DETR [72], q is a *single* natural-language sentence describing the desired moments (e.g., "the athlete attempts to clear the bar").

In both cases, the model ultimately produces a saliency score curve over the video that measures how well each frame matches the intent expressed by q.

For SportCLIP, this curve is obtained using CLIP embeddings as detailed in Section 3.3. Each frame is mapped to a visual embedding, and the sentences in q are mapped to text embeddings in the same CLIP space. For every highlight/non-highlight sentence pair, CLIP yields frame-level highlight probabilities via cosine similarity and a softmax over the two

prompts. These curves are refined through the entropy- and mean-event-area filtering strategy, and the remaining ones are averaged to obtain a single, robust highlight relevance sequence

$$\{r_t^{\mathrm{SC}}(q)\}_{t=1}^{T}, \tag{3.32}$$

where $r_t^{\mathrm{SC}}(q)$ denotes the SportCLIP highlight score at frame t.

For QD-DETR, the input consists of the video and the single query q. The model builds a query-dependent video representation via cross-attention between the textual query and video clips, and predicts a saliency score for each clip that reflects how well it matches q. These clip-level scores are then projected onto frame indices (assigning the clip score to all frames within that clip), yielding a saliency sequence

$$\{r_t^{\mathrm{QD}}(q)\}_{t=1}^{T}, \tag{3.33}$$

defined over the entire video.

From the perspective of PVS, both models are handled through a common abstraction: a per-frame relevance sequence

$$\{r_t(q)\}_{t=1}^{T}, \tag{3.34}$$

where $r_t(q)$ denotes either $r_t^{\mathrm{SC}}(q)$ or $r_t^{\mathrm{QD}}(q)$ depending on the chosen summarizer. This relevance curve is then passed through the same temporal post-processing pipeline used in Section 3.2: smoothing, thresholding, and merging adjacent highlight frames into contiguous events. The resulting events define the set of highlight intervals

$$\begin{aligned} \mathcal{H}(V,q) &= \{h_1, \ldots, h_N\}, \\ h_i &= [t_i^{\mathrm{start}}, t_i^{\mathrm{end}}], \end{aligned} \tag{3.35}$$

where t_i^{start} and t_i^{end} are start and end frame indices in V.

These segments are stored, together with their frame indices, timestamps and durations, in a standardized metadata file. This common representation allows the same downstream face-analysis and clip-selection components to operate seamlessly on highlights generated by either SportCLIP or QD-DETR, as well as any future summarization model (text-guided or otherwise) that exposes a compatible per-frame score output.

3.7.4 Clip selection

Clip selection is the final stage where identity evidence encoded in the resemblance curves is combined with candidate highlight segments produced by the video-summarization stream. The result is, for each target p_m, a personalized subset $\mathcal{H}_{p_m}^{\star}(V,q) \subseteq \mathcal{H}(V,q)$ of segments that are both narratively relevant and supported by sufficient identity evidence.

The central challenge in this stage is the temporal misalignment between identity detection and athletic performance. In broadcast sports videos, athletes are often most clearly visible outside of their actual performance moments—during preparation, immediately before or after the event, or in reaction shots. During the performance itself, faces may be small, distant, occluded, or viewed from extreme angles, making reliable face detection difficult or

impossible. This temporal-spatial misalignment necessitates algorithms that can associate identity evidence observed at one group of frames with highlight segments centered on different frame intervals.

We address this challenge with two complementary assignment algorithms, each making different assumptions about the temporal relationship between identity observations and highlight segments. Both algorithms operate independently for each recognition backbone (ArcFace [17] and TransFace [14]), enabling model-specific performance analysis and direct comparison of CNN-based versus Transformer-based approaches under identical conditions.

3.7.4.1 Common framework

Both assignment methods share a common mathematical foundation. Let

$$h_i = [t_i^{\text{start}}, t_i^{\text{end}}] \in \mathcal{H}(V, q) \tag{3.36}$$

denote a highlight segment with start and end frame indices in V. From the face-analysis stream (Section 3.7.2) we obtain, for each target p_m, a smoothed resemblance curve $\{\bar{s}_t^{(m)}\}_{t=1}^{T}$. For clarity of notation in this subsection, we define the *identity-evidence sequence*

$$g_t^{(m)} = \bar{s}_t^{(m)}, \tag{3.37}$$

and perform all assignments in terms of $g_t^{(m)}$ rather than directly using $\bar{s}_t^{(m)}$.

For any frame interval $[a, b]$ with $1 \le a \le b \le T$, we define the maximum identity evidence over that interval as

$$g_{\max}^{(m)}([a, b]) = \max_{t \in [a,b]} g_t^{(m)}. \tag{3.38}$$

Given a resemblance threshold $\tau_{\text{sim}} \in [0, 1]$, both algorithms decide whether to assign segment h_i to target p_m based on whether the maximum identity evidence in some interval related to h_i exceeds τ_{sim}. The key difference lies in how each method defines this interval and how it handles temporal context across multiple segments.

3.7.4.2 Method 1: Sequential assignment

The sequential assignment algorithm models the common broadcast pattern where an athlete is shown in close-up immediately before their performance, and the camera then follows their action. This pattern creates a natural temporal ordering: identity evidence appears first, followed by the corresponding highlight segment.

The algorithm maintains a state variable representing the currently active athlete and propagates this assignment forward in time until new evidence indicates a change. Specifically, it scans the timeline left-to-right, processing highlights $h_1, h_2, \ldots, h_N$ in order of increasing t_i^{start}. For each segment h_i, it searches a lookback window

$$W_i^{\text{look}} = \left[\max(1,\, t_i^{\text{start}} - \Delta L_{\text{look}}),\; t_i^{\text{start}}\right], \tag{3.39}$$

where ΔL_{look} is a lookback length expressed in frames. Within this window, the algorithm identifies which targets (if any) crossed the threshold τ_{sim}:

$$\mathcal{C}_i = \left\{ p_m \;\middle|\; g_{\max}^{(m)}(W_i^{\text{look}}) \geq \tau_{\text{sim}} \right\}. \tag{3.40}$$

If multiple targets crossed the threshold in W_i^{look}, the algorithm selects the one whose most recent crossing occurred closest to t_i^{start}. Let

$$t_{\text{last}}^{(m)} = \max \left\{ t \in W_i^{\text{look}} \;\middle|\; g_t^{(m)} \geq \tau_{\text{sim}} \right\} \tag{3.41}$$

denote the latest frame index in W_i^{look} where target p_m exceeds the threshold (if no such frame exists, $p_m \notin \mathcal{C}_i$). The assignment rule is then

$$p_i^\star = \begin{cases} \arg\max_{p_m \in \mathcal{C}_i} t_{\text{last}}^{(m)}, & \text{if } \mathcal{C}_i \neq \emptyset, \\ p_{i-1}^\star, & \text{otherwise,} \end{cases} \tag{3.42}$$

where $p_i^\star$ denotes the athlete assigned to segment h_i, and $p_{i-1}^\star$ is the assignment of the previous segment (initialized to null for the first segment). This propagation rule allows the system to continue assigning highlights to the same athlete even when no clear face is visible, as long as no new evidence contradicts that assignment.

The sequential method is particularly effective when broadcasts follow a predictable structure, such as showing an athlete's preparation immediately before their attempt. However, it can propagate errors: if an early segment is misassigned, all subsequent segments may inherit that error until new evidence corrects the state. In addition, because assignments are made by propagating the last confidently recognized identity, this method is only meaningful when the gallery contains representative images for all athletes of interest; otherwise, highlights tend to be systematically attributed to the subset of registered athletes, biasing the resulting personalized summaries toward those identities.

3.7.4.3 Method 2: Instant assignment with temporal expansion

The instant assignment algorithm evaluates each highlight segment independently, without maintaining state across segments. This design makes it robust to sequential errors and to cases where athletes are only visible after their performance (e.g., celebration or reaction shots).

For each segment h_i, the algorithm defines an expanded frame window

$$W_i^{\text{exp}} = \left[\max(1,\, t_i^{\text{start}} - \Delta L_{\text{exp}}),\; \min(T,\, t_i^{\text{end}} + \Delta L_{\text{exp}}) \right], \tag{3.43}$$

where ΔL_{exp} is an expansion margin in frames. This expansion compensates for the temporal misalignment between face visibility and performance moments, allowing the system to capture identity evidence from preparation or reaction shots that occur near, but not necessarily within, the highlight segment itself.

Within W_i^{exp}, the algorithm computes for each target the set of frames where identity evidence exceeds the threshold:

$$\mathcal{U}_i^{(m)} = \left\{ t \in W_i^{\text{exp}} \;\middle|\; g_t^{(m)} \geq \tau_{\text{sim}} \right\}. \tag{3.44}$$

If $\mathcal{U}_i^{(m)} \neq \emptyset$, target p_m is considered a candidate for segment h_i. To handle cases where multiple candidates appear in the same window, the algorithm computes an average identity-evidence score over the frames above threshold:

$$\bar{g}_i^{(m)} = \begin{cases} \dfrac{1}{|\mathcal{U}_i^{(m)}|} \displaystyle\sum_{t \in \mathcal{U}_i^{(m)}} g_t^{(m)}, & \text{if } \mathcal{U}_i^{(m)} \neq \emptyset, \\ 0, & \text{otherwise.} \end{cases} \tag{3.45}$$

The segment is then assigned to the target with the highest average score above threshold:

$$p_i^\star = \begin{cases} \arg\max_{p_m : \mathcal{U}_i^{(m)} \neq \emptyset} \bar{g}_i^{(m)}, & \text{if at least one candidate exists,} \\ \text{null}, & \text{otherwise.} \end{cases} \tag{3.46}$$

This independent evaluation strategy prevents error propagation and is particularly robust when highlights are interspersed (e.g., alternating athletes in a competition). However, it may be less effective when identity evidence is sparse or when the temporal gap between visibility and performance exceeds the expansion margin ΔL_{exp}.

3.7.4.4 Output and interpretation

Both methods produce assignment dictionaries

$$\begin{aligned} \mathcal{A}^{\text{seq}} &: \{0, 1, \ldots, N-1\} \to \mathcal{P} \cup \{\text{null}\}, \\ \mathcal{A}^{\text{inst}} &: \{0, 1, \ldots, N-1\} \to \mathcal{P} \cup \{\text{null}\}, \end{aligned} \tag{3.47}$$

that map each segment index to an assigned target (or null if unassigned). For each target p_m, the personalized highlight set is then

$$\mathcal{H}_{p_m}^\star(V, q) = \left\{ h_i \;\middle|\; \mathcal{A}(i) = p_m \right\}, \tag{3.48}$$

where $\mathcal{A}$ denotes either $\mathcal{A}^{\text{seq}}$ or $\mathcal{A}^{\text{inst}}$ depending on the chosen method.

Figure 3.25 provides a visual comparison of how the two assignment strategies behave on a single full-length video. To make their behavior easier to interpret, PVS visualizes the final segment-to-athlete mapping as a timeline of colored bars: each bar corresponds to one highlight segment and is colored according to the assigned athlete, with labels (H1, H2, ...) indicating the segment identifiers and light gray bars marking unassigned segments.

These visualizations, combined with the per-target resemblance curves from Section 3.7.2, provide a complete interpretable representation of the system's decision-making process. By comparing $\mathcal{A}^{\text{seq}}$ and $\mathcal{A}^{\text{inst}}$ side-by-side, one can identify scenarios where temporal context

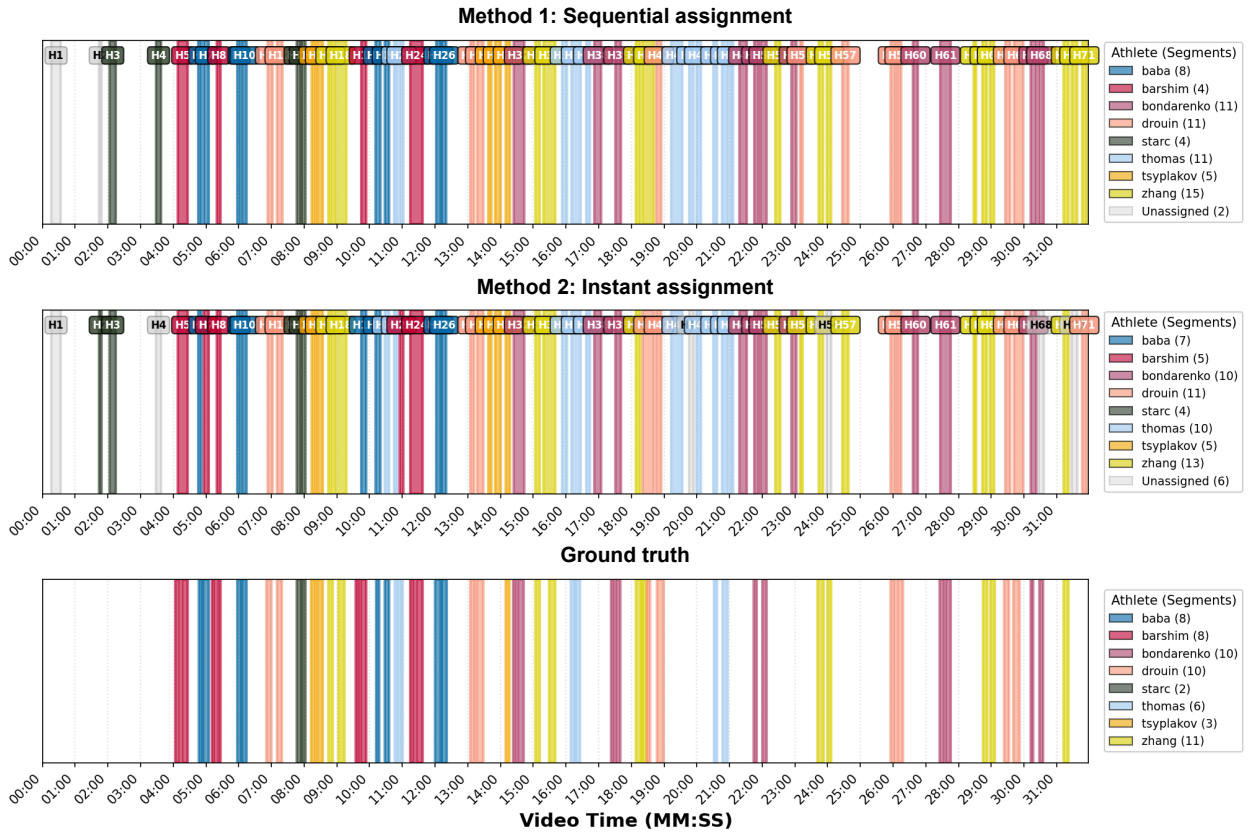


Figure 3.25: Comparison of highlight assignments for a single full-length video. **Top:** Method 1 (sequential assignment), where segments inherit the identity of the most recent athlete whose resemblance curve crossed the similarity threshold. **Middle:** Method 2 (instant assignment with temporal expansion), where each segment is evaluated independently within an expanded window around its temporal extent. **Bottom:** Ground-truth segment-to-athlete annotations. In all panels, colored bars indicate segments assigned to a particular athlete, light gray bars denote unassigned segments, labels (H1, H2, etc.) identify individual highlights, and the horizontal axis shows video time (MM:SS).

helps (segments correctly assigned by sequential but not instant) versus scenarios where independence is beneficial (segments correctly assigned by instant but not sequential).

From a methodological perspective, the availability of two complementary assignment strategies closes the loop between event detection and identity analysis. The sequential method leverages the structured, narrative flow typical of broadcast sports, while the instant method provides robustness and independence. Quantitative evaluation of both methods against ground-truth annotations, presented in Section 4.7, reveals their relative strengths and the contexts in which each excels.

Chapter 4

Results

Chapter 3 established the methodological framework for this thesis, detailing the progression from classical, motion-centric pipelines to modern, multimodal deep learning architectures. We defined the core components for two complementary domains: video summarization (the *what* and *where*) and face recognition (the *who*), which together enable the final goal of personalized video summarization.

This chapter provides the empirical validation for the methods, datasets, and systems introduced. Here, we move from theoretical design to practical application, presenting the quantitative and qualitative results that measure the performance, robustness, and real-world viability of our contributions.

To maintain clarity and avoid redundancy, Section 4.1 first introduces the key datasets that serve as benchmarks for our evaluations, centralizing the description of resources like MATDAT, SportCLIP, and the UPM-GTI-Face dataset. Section 4.2 then summarizes the evaluation metrics used throughout the chapter, providing a unified interpretation framework for both video summarization and face recognition experiments. Building on this common ground, the remainder of the chapter is structured to mirror the methodological journey of the thesis. Section 4.3 presents the results for our foundational work, the *automatic highlight detector for martial arts tricking* [84], analyzing the performance of the classical, motion-centric pipeline on the specialized MATDAT dataset. Next, Section 4.4 evaluates the *text-guided CLIP-based framework* [86], assessing its flexible, zero-shot summarization capabilities across diverse sports and comparing it against baselines on the MATDAT and SportCLIP benchmarks. The chapter then transitions to the face recognition strand in Section 4.5, which presents the baseline detection and recognition performance on the *UPM-GTI-Face dataset* [87], validating its challenges and quantifying the severe impact of distance and occlusions. Section 4.6 details the findings of our *comprehensive comparison between ViTs and CNNs* [85], using the UPM-GTI-Face dataset and other public benchmarks to identify the most robust backbones for in-the-wild recognition. Finally, Section 4.7 demonstrates the complete Personalized Video Summarization (PVS) system, integrating the previously validated components to showcase the end-to-end workflow and its effectiveness in generating identity-centric highlights.

Collectively, these sections provide a complete performance picture, validating each component

of the proposed PVS system and demonstrating its effectiveness through rigorous empirical testing.

4.1 Datasets

This section presents the datasets used to empirically validate the methods proposed in this thesis across its two core domains: sports video summarization and face recognition. For video summarization (Section 4.1.1), we rely on a family of complementary benchmarks specifically tailored to highlight detection: three sports-oriented datasets introduced in this thesis—MATDAT [84], SportCLIP [86], and Olympic Highlights—and the established YouTube Highlights corpus [95]. Together, they support the experimental analyses reported in Section 4.3 and Section 4.4.

For face recognition (Section 4.1.2), we combine large-scale training data (VGGFace2 [7]) with a set of complementary evaluation benchmarks: LFW [46] for unconstrained verification, ROF [22] for occlusion robustness, SCface [30] for surveillance imagery, and the UPM-GTI-Face dataset [87] proposed in this thesis to jointly study the impact of distance and occlusions. These resources underpin the experimental results discussed in Section 4.5 and Section 4.6, providing a coherent empirical basis for assessing the proposed methods under realistic conditions.

4.1.1 Video Summarization

In the context of highlight detection, it is essential to have fine-grained annotations indicating which parts of a video constitute true highlights. However, many existing benchmarks only partially satisfy this requirement. General-purpose video summarization datasets such as SumMe [33] and TVSum [93] comprise at most a few dozen videos and span heterogeneous, mostly non-sport domains (e.g., egocentric, documentary, vlogs). They typically provide user summaries or shot-level importance scores rather than exhaustive frame-level highlight vs. non-highlight labels, and their limited scale can lead to unstable evaluations. Sports-focused resources like SoccerNet [28] and SoccerNet-v2 [15], while large-scale, are primarily designed for event spotting and related tasks—goals, cards, substitutions—anchored to sparse timestamps in full-match soccer broadcasts, rather than continuous highlight detection across the whole game. More recent large-scale benchmarks such as Mr. HiSum [94] provide frame-importance scores for tens of thousands of web videos, but rely on aggregated "Most Replayed" statistics from many users instead of manual, domain-specific annotation, and are not specialized for structured sports scenarios. Similarly, QVHighlights [56] targets query-based moment retrieval and highlight detection conditioned on natural-language descriptions over diverse YouTube content, so although it is a strong resource for general language-guided video retrieval, it does not match the sports-specific, frame-level highlight benchmarks that we focus on in this thesis.

To address these limitations, the video summarization strand of this thesis relies on a set of complementary benchmarks specifically tailored to sports highlight detection. We constructed three sports-oriented datasets—MATDAT [84], SportCLIP [86], and Olympic

Table 4.1: Statistical summary of ground truth events across the MATDAT [84], SportCLIP [86], and Olympic Highlights datasets. The table details the temporal distribution of Highlight (HL), Non-Highlight (NHL), and Uncertainty (UN) segments, reporting the average event duration (Avg), total cumulative duration (Tot), and the percentage of the total video length occupied by each category (%).

		Highlight (HL)			**Non-Highlight (NHL)**			**Uncertainty (UN)**		
Dataset	**Video**	**Avg(s)**	**Tot(m)**	**%**	**Avg(s)**	**Tot(m)**	**%**	**Avg(s)**	**Tot(m)**	**%**
MATDAT	Tricking V1	2.93	3.12	15.0	14.88	16.13	77.4	0.75	1.60	7.7
	Tricking V2	2.69	2.29	10.4	21.16	18.34	83.7	0.75	1.28	5.8
	Tricking V3	3.02	2.31	23.1	8.55	6.56	65.4	0.75	1.15	11.5
SportCLIP	Diving	6.16	0.82	16.8	25.72	3.86	79.1	0.75	0.20	4.1
	Long Jump	7.89	1.18	50.3	6.32	0.95	40.4	0.77	0.22	9.3
	Pole Vault	8.40	0.98	37.5	12.54	1.46	55.9	0.80	0.17	6.6
	Tumbling	9.98	3.33	32.3	21.63	6.49	63.0	0.79	0.49	4.7
Olympic Highlights	High Jump V1	8.08	7.81	24.2	23.53	22.75	70.5	0.90	1.73	5.4
	High Jump V2	5.41	5.32	16.3	25.46	25.46	78.2	0.90	1.77	5.4
	High Jump V3	7.43	6.93	19.4	30.01	28.00	78.3	0.45	0.84	2.3
	High Jump V4	5.79	3.09	15.3	29.26	16.09	79.9	0.90	0.96	4.8
	High Jump V5	6.13	5.83	18.7	24.40	23.59	75.8	0.90	1.71	5.5
	Javelin V1	12.11	7.06	26.0	31.82	19.09	70.2	0.90	1.05	3.9
	Javelin V2	11.77	7.26	30.1	24.88	15.76	65.3	0.90	1.11	4.6
	Javelin V3	11.41	6.46	28.2	26.47	15.44	67.4	0.90	1.02	4.5
	Javelin V4	8.90	4.60	24.9	24.24	12.93	70.1	0.90	0.93	5.0
	Javelin V5	13.31	9.54	31.3	27.69	20.31	66.6	0.45	0.65	2.1
	Long Jump V1	10.81	8.65	29.1	24.01	19.61	66.0	0.90	1.44	4.9
	Long Jump V2	10.99	14.84	40.8	14.13	19.07	52.5	0.90	2.42	6.7
	Long Jump V3	13.45	11.21	31.4	27.03	22.98	64.4	0.90	1.50	4.2
	Long Jump V4	12.95	12.31	26.1	34.28	33.14	70.3	0.90	1.71	3.6
	Long Jump V5	15.18	8.86	31.1	30.85	18.51	65.0	0.94	1.09	3.9
	Pole Vault V1	10.97	7.86	30.4	22.84	16.75	64.7	0.90	1.29	5.0
	Pole Vault V2	11.29	7.71	24.6	31.96	22.37	71.4	0.90	1.23	3.9
	Pole Vault V3	12.03	9.22	26.0	32.66	25.58	72.1	0.45	0.69	1.9
	Pole Vault V4	12.54	9.82	26.2	33.62	26.33	70.1	0.90	1.40	3.7
	Pole Vault V5	10.40	8.32	27.9	25.48	20.81	69.7	0.45	0.72	2.4

Highlights[1]—with explicit frame-level annotations that distinguish between highlight (HL), non-highlight (NHL), and uncertainty (UN) segments in controlled training sessions, compact multi-sport clips, and long-form broadcast-style coverage, respectively. Only highlight events of at least one second are labeled as HL, so that very short actions are not treated as meaningful highlights, and each highlight is padded with an uncertainty margin of 0.5 seconds before its start and 1 second after its end to reflect the inherent ambiguity in temporal boundaries. We use a larger margin after the nominal end than before the start because highlight endings are typically more ambiguous than onsets: the end often includes follow-through, landing/aftermath, brief reactions, or editing/transitions, whereas the onset is usually tied to a clearer action initiation cue. This asymmetric padding therefore reduces sensitivity to small boundary shifts while keeping the highlight core cleanly labeled as HL. The detailed breakdown of these ground-truth events for each video, including the average duration, total cumulative duration, and relative proportion of each category, is provided in Table 4.1. These datasets are complemented by the established YouTube Highlights corpus [95], which provides large-scale, web-sourced editorial supervision through pairs of raw and edited videos in several action-centric domains. Together, these resources cover a progression from controlled, single-sport recordings to diverse, long-form multi-sport footage and in-the-wild user-generated content, providing a coherent testbed to analyze both the effectiveness and the generalization capability of the summarization methods proposed in this thesis.

4.1.1.1 MATDAT

MATDAT [84] offers a compact yet challenging test case for the overarching goal of this thesis—automatic sports highlight detection. Existing publicly available sports datasets for highlight detection are limited in both scope and annotation granularity, which motivated the construction of a dedicated benchmark in the discipline of martial arts tricking. Tricking consists of fast acrobatic sequences, frequent transitions between skills, and substantial variability in difficulty and execution quality, all of which naturally give rise to dense highlight candidates. To isolate these factors from production cues, our recordings were captured in controlled training environments, with a fixed camera and no broadcast overlays (e.g., logos, scoreboards, replay transitions), ensuring that methods must identify highlights directly from the visual and motion content. The resulting corpus consists of three video sequences selected to represent the most common scenarios found in tricking, annotated at the frame level according to the highlight (HL), non-highlight (NHL), and uncertainty (UN) protocol described in Section 4.1.1 and summarized in Table 4.1.

The three videos were chosen to span a spectrum of typical training and performance conditions. The first two depict regular training sessions with a limited number of participants: in the first scenario, highlights are mainly composed of isolated skills executed at a relatively slow pace (as in focused technical practice), whereas in the second scenario they generally form longer, more continuous passes with faster motion. The third scenario corresponds to a more dynamic gathering, with additional participants and athletes performing their best combinations of skills in quick succession, resulting in a busier and more visually complex

[1]The datasets introduced in this thesis and their ground-truth annotations are available at www.gti.ssr.upm.es/data.

Figure 4.1: Example frames from the MATDAT dataset [84] in the three tricking videos. Each image is outlined according to its ground-truth annotation: highlight (green), non-highlight (red), and uncertainty (blue), illustrating the dense and fine-grained labeling used for training and evaluation.

scene. As summarized in Table 4.1, highlight events account for only about 10–23% of the total duration, illustrating a strong imbalance between highlight and non-highlight content. This sparsity mirrors the real-world conditions encountered in the other sports datasets used in this thesis and provides a compact yet diverse testbed for evaluating the robustness of highlight detection methods. Figure 4.1 illustrates representative frames from the three videos, with the ground-truth highlight, non-highlight, and uncertainty labels overlaid to show the dense and fine-grained annotation protocol.

4.1.1.2 SportCLIP

SportCLIP [86] is a benchmark we developed to extend the analysis of sports highlight detection beyond a single discipline and into a multi-sport setting. Whereas MATDAT [84] focuses on a single acrobatic sport in a controlled environment, SportCLIP is designed to probe how well summarization methods generalize across different types of athletic motion and highlight structure. The dataset contains four categories—diving, long jump, pole vault, and tumbling—selected for their diverse action styles and temporal dynamics. Although the total durations vary (roughly 5 minutes for diving, 2.5 minutes each for long jump and pole vault, and 10 minutes for tumbling), the key advantage of SportCLIP lies in how each category exhibits a distinct proportion and temporal arrangement of highlight frames. As summarized in Table 4.1, the fraction of time labeled as highlight (HL) ranges from under 20% in diving to around one half of the video in long jump, with pole vault and tumbling occupying intermediate regimes. Highlight segments themselves are of moderate duration (on the order of 6–10 seconds on average), while uncertainty (UN) intervals remain below 10% of the total length in all categories, preserving a clear imbalance between highlight and non-highlight (NHL) content.

All videos are annotated at the frame level according to the highlight (HL), non-highlight

Figure 4.2: Example frames from the SportCLIP dataset [86] across its four sports (diving, long jump, pole vault, and tumbling). As in MATDAT, frames are outlined according to their ground-truth annotation: highlight (green), non-highlight (red), and uncertainty (blue), illustrating the multi-sport, fine-grained labeling used to evaluate generalization of highlight detection methods.

(NHL), and uncertainty (UN) protocol described in Section 4.1.1, enabling rigorous, fine-grained evaluation. Figure 4.2 shows representative frames from each sport with their corresponding HL/NHL/UN labels overlaid, illustrating the visual diversity and annotation granularity of the benchmark. In the context of this thesis, SportCLIP thus plays a complementary role to MATDAT: together, they provide a progression from single-sport, tricking-focused footage to a compact yet heterogeneous multi-sport testbed for measuring the robustness and generalization capabilities of the proposed summarization methods.

4.1.1.3 Olympic Highlights

While MATDAT [84] and SportCLIP [86] focus on relatively short clips and compact sessions, the Olympic Highlights dataset extends the evaluation to long-form, broadcast-style coverage. It is a large-scale benchmark created in this thesis to study how summarization models generalize across a broad range of athletic disciplines and over extended temporal horizons. The dataset consists of 20 long-form videos, sourced from YouTube, totaling over 10 hours of footage. These videos are evenly divided among four athletics disciplines—high jump, javelin, long jump, and pole vault—with five videos per sport and an average duration of approximately 30 minutes per video. In addition to frame-level highlight labels, each highlight interval is annotated with the identity of the athlete performing the attempt, linking event-centric information to the corresponding individual and enabling identity-aware evaluation.

As summarized in Table 4.1, the temporal distribution of highlight content varies substantially across these four disciplines. Figure 4.3 shows representative frames from each sport with their HL/NHL/UN labels overlaid, illustrating the visual diversity and fine-grained annotation of long-form, broadcast-style coverage. Highlight (HL) segments typically account for between roughly 15% and 40% of each video, with high jump sequences exhibiting the sparsest highlight

Figure 4.3: Example frames from the Olympic Highlights dataset across its four sports (high jump, javelin, long jump, and pole vault). Frames are outlined according to their ground-truth annotation: highlight (green), non-highlight (red), and uncertainty (blue), illustrating the fine-grained labeling applied to long-form, broadcast-style athletics coverage.

coverage and long jump videos tending towards the upper end of this range, while javelin and pole vault occupy intermediate regimes. Individual highlight events are of moderate duration (on the order of 6–15 seconds on average, depending on the sport), and are interleaved with extended non-highlight (NHL) intervals that still dominate the total running time in all videos. Uncertainty (UN) segments consistently represent only a small fraction of the timeline (around 2–7%), preserving both a strong imbalance between highlight and non-highlight content and a localized band of temporal ambiguity around event boundaries. The underlying annotations follow the highlight (HL), non-highlight (NHL), and uncertainty (UN) protocol introduced in Section 4.1.1. This design enables both rigorous quantitative evaluation and qualitative inspection of summarization methods on long-form, broadcast-style sports footage and further supports the thesis objective of robust and generalizable sports highlight detection, including scenarios where the identity of the athlete is explicitly taken into account.

4.1.1.4 YouTube Highlights

The YouTube Highlights dataset [95] provides a large-scale, web-sourced benchmark for learning a generic notion of "highlightness" from user-edited videos. It consists of pairs of raw, user-generated YouTube videos and their corresponding edited (highlight) versions across several action-centric domains, such as skating, surfing, skiing, gymnastics, parkour, and dog-related activities. Each pair implicitly specifies which segments of the raw footage were deemed interesting enough to keep in the final edit, yielding rich ranking constraints between highlight and non-highlight segments. Overall, the dataset contains hundreds of videos, totaling nearly 700 minutes of content, and was specifically designed to support learning from noisy, in-the-wild footage without requiring frame-level manual annotation. An illustrative example of how the dataset supports learning a ranking of moments by "highlightness" within a given domain (surfing) is shown in Figure 4.4.

Figure 4.4: Illustrative example from the YouTube Highlights dataset [95] in the surfing domain. Each clip (represented by a sampled frame) is ordered from left to right according to its predicted "highlightness", ranging from high to low. This ranking-based supervision, derived from edited videos on YouTube, enables learning domain-specific highlight detectors without explicit frame-level labels.

In contrast to the sports-specific datasets introduced above (MATDAT [84], SportCLIP [86], and Olympic Highlights), which provide explicit frame-level labels in well-defined sports scenarios, YouTube Highlights encodes supervision indirectly through these editorial preferences. This makes it particularly suitable for training latent ranking models such as the dual-learner video highlight detection (DL-VHD) framework of Xu *et al.* [95]. In this thesis, we use YouTube Highlights exclusively as the source-domain training data for DL-VHD, following the original protocol, and then evaluate the resulting model on our sports-focused benchmarks. This setup allows us to assess how a generic, web-trained highlight detector transfers to structured sports footage and to quantify the benefits of the sport-oriented, frame-level modeling strategies developed in this work.

4.1.2 Face Recognition

Progress in face recognition has been tightly coupled to the availability of large-scale training corpora and specialized evaluation benchmarks. Identity-labeled datasets such as CASIA-WebFace [124] and VGGFace2 [7] provide millions of images spanning thousands of subjects, enabling deep models to learn pose, age, and illumination-invariant representations from web-sourced imagery. More recent collections like WebFace42M/WebFace260M [135] further push the scale of training data to tens of millions of faces. In parallel, benchmarks such as Labeled Faces in the Wild (LFW) [46] have become de facto standards for measuring unconstrained face verification performance, while other datasets target specific stressors, including occlusions (e.g., Real-World Occluded Faces, ROF [22]) and low-quality surveillance imagery (e.g., SCface [30]). Each of these resources emphasizes a different aspect of the problem—scale, unconstrained conditions, occlusions, or surveillance settings—but none alone fully covers the joint impact of distance, occlusions, and image quality that motivates the face recognition strand of this thesis.

In this work, we therefore adopt a combination of public benchmarks and a dedicated in-house dataset to match our research goals. VGGFace2 [7] is used as the primary large-scale training corpus, as it offers a favorable balance between dataset size, label cleanliness, and diversity in pose and age without requiring the computational and storage demands of million-scale

alternatives such as WebFace42M/WebFace260M [135]. For evaluation, we employ LFW [46] to assess performance in standard unconstrained verification, ROF [22] to probe robustness under real-world upper and lower-face occlusions, and SCface [30] to test recognition under degraded, surveillance-like conditions. Complementing these public datasets, the UPM-GTI-Face dataset [87] was specifically created in this thesis to study the joint effect of capture distance and occlusions under controlled acquisition. Together, these resources provide a coherent set of benchmarks that support the thesis objective of systematically analyzing and comparing modern face recognition architectures under realistic variations in occlusion, distance, and image quality.

4.1.2.1 UPM-GTI-Face

UPM-GTI-Face [87] is the dataset we proposed in Section 3.5 as the starting point of the face recognition strand of this thesis. There, it was introduced in more detail to better fit the narrative of the chapter; here we briefly summarize its main characteristics for completeness of the dataset overview. The dataset is specifically designed to study the joint impact of capture distance and occlusions (via face masks) on face detection and recognition under surveillance-like conditions, addressing a gap left by existing benchmarks that typically consider either distance or occlusions in isolation, and often rely on only qualitative distance labels (e.g., "near", "far").

UPM-GTI-Face is composed of 4K images from 11 subjects (8 men and 3 women), captured under two environments (indoors and outdoors) and two face mask conditions (with and without a mask). For each combination of subject, environment, and mask condition, the dataset includes one high-resolution gallery mugshot acquired at 1 meter and 10 probe images acquired at distances ranging from 3 to 30 meters, in 3-meter increments. This yields a total of 484 images (22 indoor gallery, 22 outdoor gallery, 220 indoor probes, and 220 outdoor probes). Images were recorded with a camera mounted on a tripod at a height of 2.1 meters, while subjects walked towards the camera from 30 meters away and briefly stopped at marked distances, mimicking a surveillance scenario without strict control over pose or expression. Distances are annotated in a rigorous, metric fashion, enabling precise analysis of performance degradation as a function of range. Representative examples of one subject across all combinations of environment, mask condition, and distance are shown in Figure 3.16.

In the context of this thesis, UPM-GTI-Face serves as a targeted evaluation benchmark for the second methodological component: robust face recognition under realistic constraints. It is used to quantify how modern architectures degrade with distance and occlusion and to complement the more generic evaluation on LFW [46], ROF [22], and SCface [30]. The dataset, along with its annotations, underpins the empirical analysis of face recognition robustness presented in the subsequent sections.

4.1.2.2 VGGFace2

VGGFace2 [7] serves as the primary large-scale training corpus for the face recognition models evaluated in this thesis. It was specifically designed to provide both breadth (in terms of number of identities) and depth (in terms of images per identity), with an emphasis on

Figure 4.5: Example face images from the VGGFace2 dataset [7], illustrating its diversity in identities, poses, ages, illuminations, and backgrounds, which makes it well suited for training robust face recognition models.

variation in pose and age. The dataset contains 3.31 million images of 9,131 subjects, with between 80 and 843 images per identity (362.6 on average), collected from Google Image Search. These images span a wide range of ethnicities, ages, poses (from frontal to extreme profile), illuminations, and backgrounds, and are curated through a combination of automated filtering and manual verification to keep label noise low. Representative samples illustrating this diversity are shown in Figure 4.5.

This balance of scale, diversity, and label quality makes VGGFace2 particularly well suited to the goals of this thesis. It provides the rich intra-class variation needed to train models that remain reliable under changes in pose, age, and appearance—conditions that are critical when later testing robustness to occlusions, distance, and low-quality imagery on ROF [22], SCface [30], and UPM-GTI-Face [87]. In our experiments, we follow a standard split of 90% of the images for training, 5% for validation, and 5% for evaluating face identification performance. The resulting models provide a strong, common feature representation for the comparative study of CNN and Transformer-based architectures in this thesis and for interpreting their behavior across the different evaluation benchmarks.

4.1.2.3 Labeled Faces in the Wild

Labeled Faces in the Wild (LFW) [46] is a widely adopted benchmark for studying face recognition in unconstrained conditions. It contains 13,233 images of 5,749 individuals collected from web news sources, exhibiting large natural variability in pose, illumination, expression, background, occlusions, and image quality. Unlike many earlier datasets acquired under controlled conditions, LFW was explicitly designed to approximate the complexity of everyday imagery, making it a canonical testbed for unconstrained face verification and pair matching. Representative sample images are shown in Figure 4.6.

In this thesis, we use LFW as a standard reference benchmark to assess how well the models

Figure 4.6: Example face images from the LFW dataset [46], illustrating its unconstrained, in-the-wild nature with substantial variation in pose, expression, illumination, and background.

trained on VGGFace2 [7] generalize to real-world, in-the-wild conditions. Following the conventional evaluation protocol, we consider the verification setting with 6,000 image pairs (3,000 genuine and 3,000 impostor) and report performance over this split. This provides a direct point of comparison with prior work on unconstrained verification and complements the more targeted robustness evaluations on ROF [22], SCface [30], and UPM-GTI-Face [87], which focus on specific stressors such as occlusions, surveillance-quality imagery, and capture distance.

4.1.2.4 Real-World Occluded Faces

Real-World Occluded Faces (ROF) [22] is a benchmark specifically designed to study the impact of real-life occlusions on face recognition systems. Unlike datasets that simulate occlusions synthetically (e.g., by overlaying black boxes or artificial masks), ROF contains in-the-wild face images collected from web sources, where occlusions arise naturally from everyday usage of sunglasses (upper-face occlusion) and protective masks (lower-face occlusion). The dataset includes thousands of neutral, sunglasses, and mask images, spanning a diverse set of public figures and exhibiting substantial variation in pose, illumination, and background. Example neutral, masked, and sunglasses images for the same subjects are shown in Figure 4.7.

In this thesis, ROF is used as an evaluation dataset to assess the robustness of models trained on VGGFace2 [7] when confronted with genuine upper and lower-face occlusions. By comparing performance on neutral versus occluded images, we can quantify the degradation caused by real-world sunglasses and masks and contrast it with the behavior observed on other benchmarks such as LFW [46], SCface [30], and UPM-GTI-Face [87]. ROF therefore plays a complementary role in the experimental design: it isolates occlusion as a primary stressor and helps evaluate whether the architectures and training strategies considered in this thesis remain reliable when key facial regions are partially hidden in unconstrained conditions.

Figure 4.7: Example images from the ROF dataset [22] showing, for the same subjects, neutral faces (top row), faces occluded by protective masks (middle row), and faces occluded by sunglasses (bottom row), illustrating real-world lower and upper-face occlusions.

4.1.2.5 SCface

SCface [30] is a surveillance-oriented face database specifically designed to test recognition algorithms under challenging, real-world CCTV conditions. It contains 4,160 static images of 130 subjects, acquired with five commercially available video surveillance cameras of varying quality, plus additional infrared (IR) devices. For each subject, high-quality frontal mugshot images are captured with a professional photo camera, while surveillance images are recorded indoors at three discrete distances (4.2, 2.6, and 1.0 meters) under uncontrolled illumination. The cameras are mounted slightly above eye level, mimicking typical deployment in security systems and resulting in non-frontal viewpoints, variable resolution, and noticeable degradation in image quality compared to the mugshots. An example image set for a single subject, including the frontal mugshot and corresponding surveillance views, is shown in Figure 4.8.

Beyond the visible-spectrum surveillance images, SCface also includes IR night-vision mugshots and IR surveillance frames, enabling experiments that span different sensing modalities. In this thesis, we focus on the visible-light surveillance subset and follow the identification-oriented protocol proposed in the original work, where frontal mugshots serve as gallery images and low-quality surveillance captures at multiple distances act as probes. This setting provides a stringent test of how well the face recognition architectures studied in Section 4.6 cope with real surveillance artifacts, complementing the controlled, distance-annotated UPM-GTI-Face [87] dataset and the occlusion-focused ROF [22] benchmark. Together, these datasets allow us to analyze recognition robustness not only under synthetic or controlled degradations, but also in genuinely noisy, heterogeneous surveillance imagery.

Figure 4.8: Example image set for a single subject in the SCface dataset [30], including a high-resolution frontal mugshot (right) and multiple low-quality surveillance images captured indoors by different cameras at varying distances.

4.2 Evaluation metrics

The quantitative analyses reported in this chapter rely on a set of standard quality metrics, applied in slightly different ways depending on whether the task involves video summarization or face recognition. This section summarizes the metrics used throughout the experimental sections and clarifies how they should be interpreted, as well as the few task-specific nuances that arise in their computation.

4.2.1 Video summarization

In the video summarization experiments, the goal is to distinguish highlight (HL) from non-highlight (NHL) content and, in some settings, to correctly localize and assign highlight events to specific athletes. This naturally leads to a binary classification problem (HL vs. NHL) at different temporal granularities. Most results are reported in terms of **recall** (R), **precision** (P), and **F-score** (F), occasionally complemented with **mean Average Precision** (mAP). All these metrics are computed from the usual counts of **true positives** (TP), **false positives** (FP), **false negatives** (FN), and, when relevant, **true negatives** (TN). Unless otherwise stated, values are reported as percentages, and higher values always indicate better performance.

Formally, recall, precision, and F-score are defined as

$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \qquad P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \qquad F = \frac{2PR}{P + R}. \tag{4.1}$$

Recall measures how many of the ground-truth highlights are recovered, whereas precision

measures how many of the predicted highlights are actually correct. The F-score combines both into a single value that is high only when both recall and precision are high. In the context of video summarization, high recall indicates that few highlight moments are missed, while high precision indicates that the summary contains few irrelevant or noisy segments.

Frame-level versus event-level metrics. Throughout the chapter, these metrics are computed at two different granularities:

- **Frame-level metrics.** Here each frame is treated as an individual sample to be classified as HL or NHL. Frame-level recall, precision, and F-score therefore measure how accurately a method labels individual frames. This is the primary view adopted in the highlight-detection experiments on MATDAT, SportCLIP, and Olympic Highlights (Sections 4.3, 4.4, and 4.7), where the outputs of the classical motion-based pipeline, the CLIP-based SportCLIP framework, and the Transformer-based QD-DETR personalized summarization system are all converted to frame-wise HL/NHL labels to allow a consistent comparison across methods and datasets. This viewpoint provides a dense, fine-grained picture of performance over the entire timeline of each video.
- **Event-level metrics.** In many applications the basic unit of interest is not an individual frame but a *highlight event*, represented by a contiguous temporal interval. Event-level TP, FP, and FN counts are then obtained by matching predicted highlight segments against annotated ground-truth events based on their temporal overlap. A predicted segment is considered a true positive when its temporal intersection-over-union (IoU) with a ground-truth event exceeds a minimum overlap requirement; predicted segments that do not match any ground-truth highlight are counted as false positives; and ground-truth highlights without any matching prediction are counted as false negatives. In the personalized video summarization experiments (Section 4.7), events are further defined at the level of *athlete-specific* highlight segments, so a predicted event is only counted as a true positive when both the temporal overlap and the assigned identity are correct. Event-level recall, precision, and F-score therefore focus on how well the method detects and localizes the events themselves and, in the personalized setting, how reliably it attributes them to the correct athlete.

Both views are complementary. Frame-level metrics emphasize how often the model gets the instantaneous HL/NHL decision right across the full video, while event-level metrics directly reflect the quality of the resulting summary as a set of complete highlight segments. When interpreting the tables in this chapter, it is therefore important to keep in mind whether a given F-score refers to frame-level classification, to generic event-level detection, or to identity-aware events in the personalized setting.

For comparisons with methods originally evaluated on segment-based annotations (such as Dual-Learner Video Highlight Detection [119]), we additionally report **mean Average Precision** (mAP). mAP summarizes how well the model ranks true highlights ahead of non-highlights by averaging the area under the precision–recall curve across classes or categories. Intuitively, if we sort all candidate segments by their predicted relevance, a high mAP indicates that true highlight segments tend to appear early in this ranking. Its values range from 0 to 1, with higher scores indicating better ranking performance.

Formally, let $p_c(r)$ denote the precision as a function of recall $r \in [0, 1]$ for a given class or category c (e.g. a particular type of highlight). The **Average Precision** (AP) for class c is defined as the area under its precision–recall curve,

$$\mathrm{AP}_c = \int_0^1 p_c(r)\, dr \;\approx\; \sum_{k=1}^{K} \Big(r_k - r_{k-1}\Big)\, p_c(r_k), \tag{4.2}$$

where $\{(r_k, p_c(r_k))\}_{k=1}^{K}$ are the discrete recall–precision points obtained as we traverse the ranked list of predicted segments. The mean Average Precision over a set $\mathcal{C}$ of highlight classes or categories is then

$$\mathrm{mAP} = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \mathrm{AP}_c. \tag{4.3}$$

In the experiments of this thesis, $\mathcal{C}$ typically reduces to the single "highlight" class, so mAP coincides with the AP of that class, but we retain the general definition to allow comparisons with methods evaluated on multiple highlight types.

4.2.2 Face recognition

The face recognition experiments encompass three related tasks: face detection, face identification, and face verification. Each task uses metrics tailored to its specific formulation, while sharing a common vocabulary of rates and errors. These metrics underpin the analyses in Sections 4.5 and 4.6, where the UPM-GTI-Face experiments and the backbone comparison study are reported.

Face detection. In the UPM-GTI-Face experiments, the detection stage is evaluated in terms of **true detection rate** (TDR), defined as the proportion of annotated faces for which the detector returns at least one valid bounding box. TDR is reported as a percentage, broken down by distance, environment (indoor/outdoor), and mask usage. Intuitively, TDR measures how often the detector finds a face when there should be one: higher TDR values mean that most faces are successfully localized under the corresponding conditions, whereas missed detections directly reduce TDR.

Face identification. Face identification is treated as a multi-class classification problem over the set of known identities, as in the identification experiments reported in Sections 4.5 and 4.6. For this task, we report:

- **Accuracy**, the fraction of test images whose predicted identity matches the ground truth. Higher accuracy indicates that the model makes fewer identity mistakes overall.
- **Top-5 accuracy**, the fraction of images for which the correct identity appears among the five most likely predictions. This metric is more permissive: it reflects how often the correct identity is at least considered by the model, even if it is not ranked first.

In both cases, larger values indicate better identification performance. An ideal identifier would achieve 100% accuracy and top-5 accuracy, meaning that it always places the correct identity at the top of its predictions (and therefore also within the top five).

Face verification. In face verification, the system must decide whether a pair of face images belongs to the same person (genuine pair) or to different people (impostor pair). This is a binary decision problem controlled by a similarity threshold: pairs with similarity above the threshold are accepted as genuine, and those below are rejected as impostors. Varying this threshold yields different trade-offs between correctly accepting genuine pairs and incorrectly accepting impostors.

To characterize this trade-off, we use **receiver operating characteristic** (ROC) curves, which plot the **true positive rate** (TPR) against the **false positive rate** (FPR) as the decision threshold is swept. Here

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \qquad \mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}. \tag{4.4}$$

A curve closer to the top-left corner corresponds to a better verifier, as it combines high TPR (most genuine pairs accepted) with low FPR (few impostor pairs incorrectly accepted). Two scalar summaries of the ROC curve are used extensively in this chapter:

- The **area under the ROC curve** (AUC), which measures how well the verifier separates genuine from impostor pairs across all possible thresholds. AUC ranges from 0 to 1, with 0.5 corresponding to random guessing and values closer to 1 indicating better separability.
- The **equal error rate** (EER), defined as the point on the ROC curve where the false positive rate equals the false negative rate (FNR). Lower EER values are preferable, as they indicate that both types of error can be kept simultaneously low at some operating point.

Figure 4.9 provides an illustrative example of these concepts, comparing ROC curves for models with different AUC and EER values and highlighting typical operating points along a single curve (high-security, high-convenience, and EER). For ROC-based metrics, better verifiers are characterized by higher AUC and lower EER.

Threshold-dependent verification metrics. While ROC, AUC, and EER summarize performance over all possible thresholds, some analyses in this chapter also report **recall**, **precision**, and **F-score** for face verification, computed for a *single* operating point. In particular, the threshold is chosen to maximize F-score for each model, and the resulting (R, P, F) values provide a more concrete view of performance under an optimized but fixed decision rule. As in the video summarization setting, recall here measures how many genuine pairs are correctly accepted, precision measures the reliability of accepted pairs (how many of them are truly genuine), and F-score captures their balance. For these verification metrics, larger values again indicate better performance.

In summary, the metrics used in this chapter are chosen to reflect the practical goals of each task: recovering as many true highlights as possible without flooding the summary with irrelevant content, and recognizing faces reliably under challenging conditions such as long distances, surveillance imagery, and occlusions. In practical terms, better models are associated with higher values of recall, precision, F-score, AUC, accuracy, and TDR, and with lower values of EER in the verification setting.

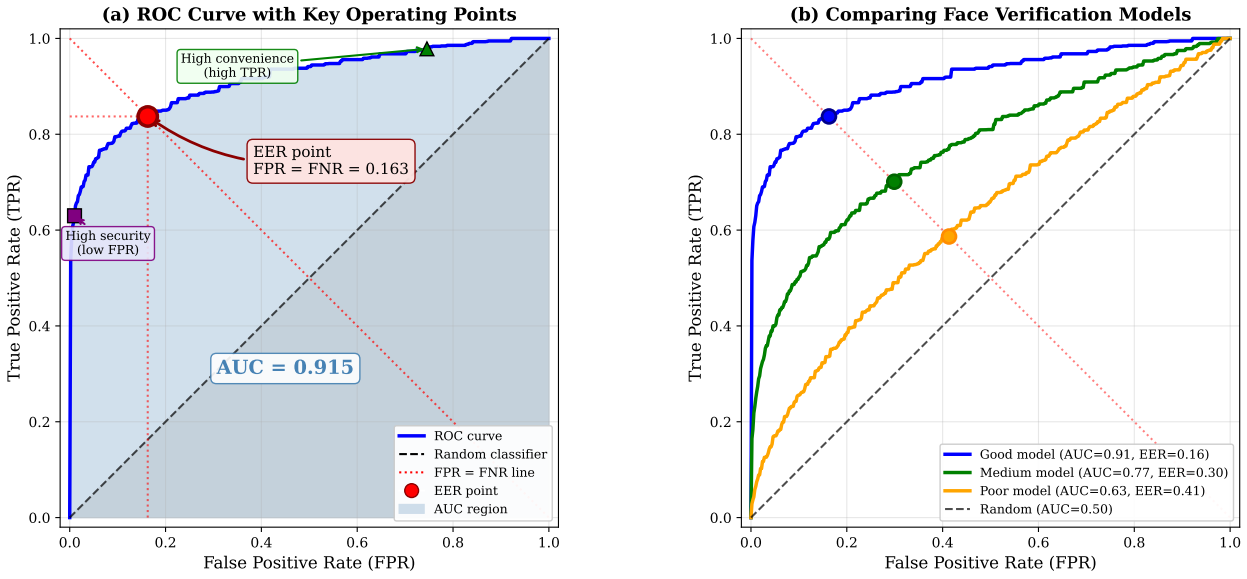


Figure 4.9: Illustrative ROC curves for face verification. (a) ROC curve of a single model, indicating the area under the curve (AUC), the equal error rate (EER) operating point where FPR = FNR, and example operating points emphasizing either higher security (lower FPR) or higher convenience (higher TPR). (b) Comparison of three models with different AUC and EER values against a random classifier.

4.3 Automatic highlight detection in videos of martial arts tricking

This section presents the experimental evaluation of the classical, motion-centric pipeline for automatic highlight detection in martial arts tricking videos introduced in Section 3.2. Using the MATDAT [84] dataset as a compact yet challenging testbed, we first quantify performance at both the frame and event levels, in terms of recall, precision, and F-score as defined in Section 4.2, examining how accurately the method localizes highlight segments and how the different recording scenarios influence these metrics. Next, we exploit the relevance modeling of detected events to study how performance evolves when only the most salient highlights are retained, thereby characterizing the ranking behavior of the system. Finally, we compare our approach with the Dual-Learner Video Highlight Detection (DL-VHD) framework [119], trained on related categories from the YouTube Highlights dataset [95], to situate our results within existing cross-category highlight detection strategies.

4.3.1 Experiments

The results obtained from the final classification described in Section 3.2.5.2 are reported at two granularities: frame level and event level, following the definitions in Section 4.2. At the frame level, recall, precision, and F-score are computed by comparing the HL/NHL label of each frame with the corresponding ground truth, providing an overall picture of how often the method assigns the correct label across the video. Event-level results, in contrast, are based on matching predicted highlight segments to ground-truth events using their temporal

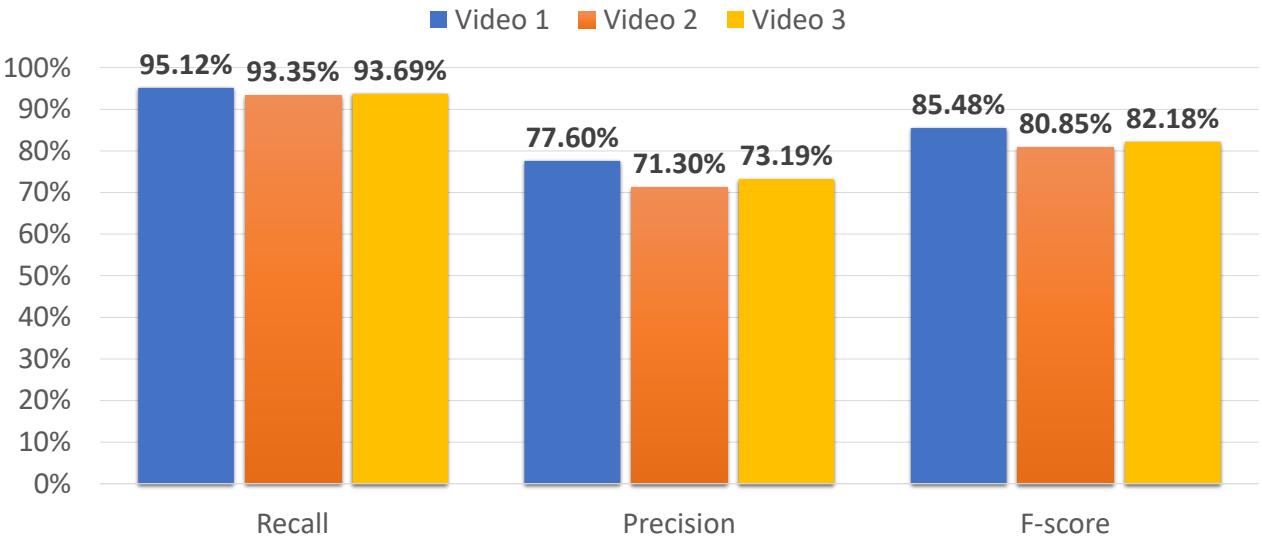


Figure 4.10: Frame-level results consisting in recall, precision, and F-score obtained at frame level for the three different videos.

overlap, and therefore reflect how many ground-truth highlight events are correctly detected and how many spurious events are produced.

Fig. 4.10 summarizes frame-level results obtained for the three videos in the dataset. Their average values are: recall of 94.05%, precision of 74.03%, and F-score of 82.84%. Video 1 gives the best results (F-score of 85.48%), as it displays less movement and highlight events stand out more. Videos 2 and 3 present similar results (F-score above 80%), showing slightly less precision as a result of the increase in movement displayed in these videos, which makes more difficult to detect the start and end of events.

Event-level results obtained for the three videos in the dataset are summarized in Table 4.2. The total number of ground truth highlight events is 161. From the 171 highlight events identified, 158 are true positives (TPs), and 13 correspond to false positives (FPs). Additionally, there are only 3 false negatives (FNs). These results yield and average F-score of 95.18%. Out of the 13 FPs, 3 correspond to people crossing in front of the camera, 6 correspond to people crossing the scene at a fast pace after a long period of time without any highlight events, 3 correspond to players feinting a skill, and 1 corresponds to a celebration of a player's performance by the other players. The 3 FNs correspond to 2 single-skill performances that were very short and occurred around much more relevant highlight events, and to 1 attempt to start a performance by a player that was interrupted to be executed again (the second time being correctly identified as a highlight event).

Additionally, the relevance modeling of highlight events described in section 3.2.5.2 allows us to analyze the results obtained when sorting the identified highlight events by order of relevance[2]. Table 4.3 summarizes some of the event-level results obtained for different sets of the most relevant events identified. The top 100% corresponds to the results obtained

[2]Although ground truth highlight events are not characterized by a relevance value (as it would be highly subjective), we can still sort the identified highlight events by order of relevance to analyze if the events that were assigned a higher value are indeed more likely to correspond to a ground truth highlight.

Table 4.2: Event-level results obtained for the three different videos.

Video	GT highlights	Detected highlights	TP	FP	FN	F-score
V1	64	67	64	3	0	97.71 %
V2	51	54	50	4	1	95.24 %
V3	46	50	44	6	2	91.67 %
Total	161	171	158	13	3	95.18 %

Table 4.3: Event-level results for different relevance ranges.

Top % of the most relevant events identified	Detected highlights	TP	FP	FN	F-score
80 %	137	135	2	26	90.60 %
90 %	154	149	5	12	94.60 %
95 %	163	157	6	4	96.91 %
99 %	170	158	12	3	95.47 %
100 %	171	158	13	3	95.18 %

when taking all identified events into consideration, and match those of Table 4.2. The lower percentages correspond to the results obtained for smaller sets of identified events, after discarding the less relevant ones. It can be appreciated that most FPs correspond to less relevant events, as for the 80% most relevant events identified only 2 correspond to FP events. The best F-score achieved is of 96.91% for the 95% most relevant events identified, which indicates that removing the 5% least relevant events identified could lead to better results, as this is where most erroneous detections occur.

The same results of Table 4.3 are illustrated in Fig. 4.11, where it is easier to appreciate that, as we expected, smaller sets of more relevant highlight events identified show higher precision but lower recall. Recall steadily increases as more identified highlight events are taken into consideration. However, precision slightly drops in the 84-100% range, as most FPs correspond to less relevant highlight events identified. As stated earlier, the best F-score is achieved for the 95% most relevant events, and past this point it drops as a consequence of the FPs introduced by the least relevant highlight events identified.

Fig. 4.12 illustrates some images corresponding to highlight events that have been correctly detected (TPs), while Fig. 4.13 and Fig. 4.14 illustrate some images that correspond to FP and FN events, respectively.

4.3.2 Comparison with other strategies

As stated in section 2.1, no prior research has been conducted with the specific objective of detecting highlights in martial arts tricking. However, there are some strategies that focus on the detection of highlights in similar sports. Among these strategies, the recently proposed

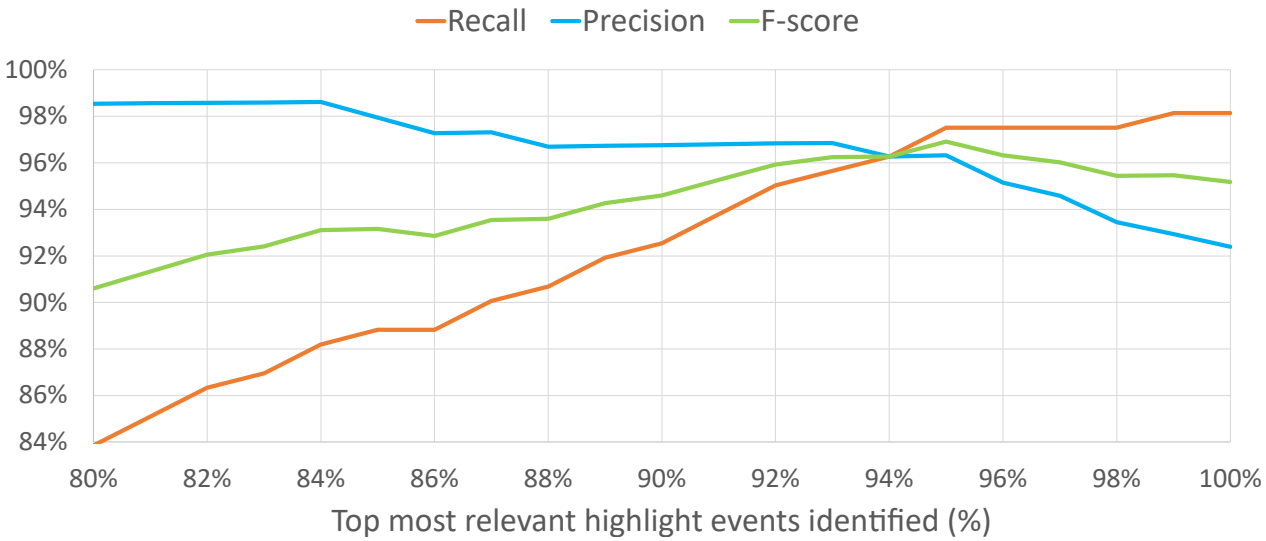


Figure 4.11: Event-level results obtained for different sets of the most relevant events identified.

Figure 4.12: Example images corresponding to TP highlight events identified.

Figure 4.13: Example images corresponding to FP highlight events identified.

Figure 4.14: Example images corresponding to FN highlight events.

Dual-Learner-based Video Highlight Detection (DL-VHD) strategy [119] is the only one that can be applied to the detection of highlights in martial arts tricking, since it allows extracting highlights from a target video category by transferring the highlight knowledge acquired from a source category. This avoids the need to have a large amount of annotated videos of the same type as those to be analyzed.

One of the primary discrepancies between our work and the DL-VHD strategy is in the annotation methodology. Whereas we provide frame-level annotations, the videos in the database used in DL-VHD have been annotated on a segment level [95]. Specifically, each video segment contains 100 frames and overlaps the previous segment by 50%. Using this overlap poses difficulties for the generation of a final video summary, since a significant number of frames can potentially belong to two video segments classified differently (i.e., one as HL and the other as NHL).

In an effort to compare our results with those obtained with the DL-VHD strategy, we have converted our frame-level annotations to the segment-level annotations (similar to those

Table 4.4: Results provided by the strategy in [119] for different source and target categories.

Source category	Target Category	mAP
gymnastics	parkour	0.660
parkour	gymnastics	0.704
gymnastics	tricking	0.256
parkour	tricking	0.305

in [95]) required by the neural network architecture proposed in [119]: each segment has been assigned the label (HL, NHL, or UN) that appears most frequently within the 100 frames that constitute it.

In [119], the dataset used for the experiments is composed of different categories of videos manually annotated [95]. Among these categories, we have selected the two that share the most characteristics with martial arts tricking: gymnastics and parkour. Table 4.4 summarizes the mean Average Precision (mAP) scores (see Section 4.2), which are the main evaluation metric used in the DL-VHD strategy, for different combinations of source and target categories. Here, the *source category* denotes the sport category used to train the model in [119], while the *target category* denotes the category on which the trained model is evaluated (i.e., cross-category generalization when source $\neq$ target). The upper part of the table shows that when the combined categories are gymnastics and parkour, the obtained mAP values are around 0.7. However, the results at the bottom of the table show that when the target category is tricking, the mAP values are much lower. This is mainly due to typical challenges in martial arts tricking videos that are not present in other sports, and also due to the highlight detection based on video segments, which is very dependent on the length of such highlights. It should be noted that the results corresponding to the martial arts tricking videos have been obtained only on the video V1, since the videos V2 and V3 have been used for training the neural network (along with the corresponding gymnastics and parkour categories). This choice intentionally maximizes training diversity by using the faster, more continuous passes in V2 and the busier multi-participant setting in V3, while evaluating on an unseen scenario with different temporal structure (slower, more isolated skills in V1). Moreover, since consecutive frames and nearby segments within the same recording are highly correlated and share the same camera viewpoint, background, and participants, we avoid mixing segments from the same video across train and test to prevent overly optimistic estimates. Although a segment-level random split could be used to measure within-video performance, the adopted protocol provides a stricter assessment of cross-video generalization under different recording conditions.

Fig. 4.15 details the result of the classification provided by the DL-VHD strategy for the first 2500 frames of the sequence V1 (same range of frames illustrated in Fig. 3.13). These results have been obtained manually selecting the threshold that yields the highest F-Score. The top of this figure shows the original ground truth at the frame level (Fig. 4.15.a) and the ground truth at the level of the partially overlapping segments (Fig. 4.15.b). In Fig. 4.15.c and Fig. 4.15.e the results of the classification provided by the DL-VHD strategy have been represented when the gymnastics and parkour categories are used as a source, respectively. In addition, the

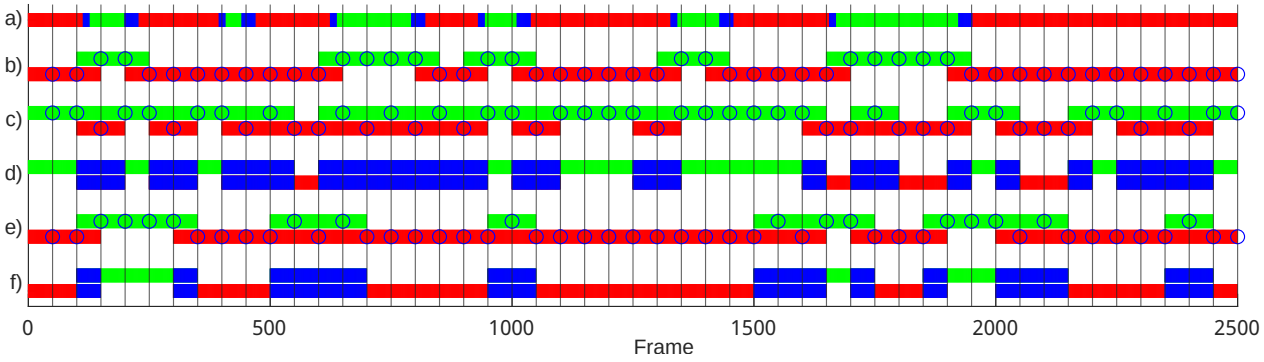


Figure 4.15: Summary of the comparison process with [119]. a) Original ground truth at frame level. b) Ground truth adapted to a video segment level. c) DL-VHD results when extrapolating highlights from gymnastics to tricking. d) Same results as c) but at the frame level. e) DL-VHD results when extrapolating highlights from parkour to tricking. f) Same results as e) but at the frame level. Blue circles and vertical lines represent, respectively, the center and the limits of video segments. Green, red, and blue slices represent, respectively, HL, NHL, and UN events.

Table 4.5: Results obtained in V1 with the strategy in [119] and with the proposed strategy.

Method	Source category	Target Category	Recall	Precision	F_score
DL-VHD	gymnastics	tricking	0.722	0.273	0.396
DL-VHD	parkour	tricking	0.319	0.519	0.398
Ours	—	tricking	**0.951**	**0.776**	**0.855**

result of converting these segment-level classifications to frame-level classifications is also illustrated (Fig. 4.15.d and Fig. 4.15.f). For this, all the frames with more than one label have been classified as UN (i.e., those classified as HL in one segment, but as NHL in another). This detailed representation shows that classification at the level of partially overlapping video segments is not suitable for detecting the highlights in this video.

Finally, Table 4.5 summarizes the recall, precision, and F-score frame-level values for sequence V1. These results show that the proposed strategy clearly outperforms the strategy in [119]. Furthermore, it is important to mention that unlike the strategy in [119], ours is capable of prioritizing the highlights detected by their relevance values.

4.4 Text-Guided Sports Highlights: A CLIP-Based Framework for Automatic Video Summarization

This section empirically evaluates the CLIP-based, text-guided sports highlight summarization framework introduced in Section 3.3. We begin by detailing the implementation choices and parameter settings, and then present quantitative and qualitative results on both MATDAT [84] and SportCLIP [86], analyzing how the method adapts to different sports and highlight structures. Next, we compare our approach against the Dual-Learner Video Highlight

Detection (DL-VHD) framework [119] (trained on YouTube Highlights [95]) and the QD-DETR model [72]: the former is explicitly designed for cross-category highlight detection, whereas the latter is a generic vision–language model for query-dependent moment retrieval. To enable a fair, side-by-side evaluation, we convert their predictions to frame-level recall, precision, and F-score as defined in Section 4.2. Our earlier tricking-specific baseline [84] is discussed separately below, since its sport- and scenario-specific assumptions make a direct multi-sport comparison impractical. Finally, we study the sensitivity of our framework to parameter and prompt variations, assess its computational cost, and discuss practical usage aspects, showing that it achieves robust, training-free performance while remaining suitable for deployment in real-world summarization scenarios.

4.4.1 Experimental results

In this subsection, we assess the performance of our proposed strategy on two datasets—MATDAT and SportCLIP—and compare it with the DL-VHD framework proposed by Xu *et al.* [119]. We begin by detailing the implementation and parameter choices of our method, followed by both quantitative and qualitative results. Finally, we show how DL-VHD generalizes to SportCLIP, reporting mean Average Precision (mAP) as well as frame-level recall, precision, and F-score, using the metric definitions introduced in Section 4.2.

Parameter Choices. Our video summarization pipeline relies on several parameters whose values directly affect the generation and filtering of highlight predictions. Table 4.6 lists these parameters and their default values. First, the context window size, W, originates from the post-processing strategy in [84], where it determines how many neighboring frames are considered around the current frame when generating rolling-average predictions. A larger W smooths out abrupt changes, which can be beneficial for sports featuring longer or more continuous actions (e.g., tumbling), but may slightly dilute sharp highlight boundaries in activities characterized by short bursts of movement (e.g., tricking).

Next, the entropy threshold, τ_Λ, governs the distribution-based filter by removing sentence pairs whose highlight-score distributions are too close to uniform. Lower values of τ_Λ enforce stricter filtering, reducing potentially redundant or near-flat (high-entropy) curves but risking the loss of some informative pairs. Higher values relax the filter, allowing more candidates to pass, potentially capturing more nuanced highlights at the cost of including some noisy scores.

The histogram division factor, D, controls how strictly the second filtering stage removes pairs with small mean event areas. Recall from Section 3.3.2 that τ_{area} is calculated by interpolating between $\mathcal{A}_{\min}$ and $\mathcal{A}_{\max}$ in the global set $\mathcal{A}$. Increasing D lowers τ_{area} and therefore removes fewer candidate pairs, effectively being more lenient with borderline highlights. Conversely, decreasing D tightens τ_{area} and aggressively prunes pairs whose mean event areas fail to exceed the raised threshold.

Finally, beyond these parameters, we also generate multiple highlight (J) and non-highlight (K) sentences—set to eight each in our experiments—to capture varied text descriptions. Relying on a single sentence pair would make performance overly sensitive to that particular wording, whereas an excessively large pool would increase computational cost without commensurate gains. We therefore adopt the compromise $J{=}K{=}8$, yielding 64 *HL–NHL* pairs—enough for

Table 4.6: Parameter Settings for our method.

Parameter	Symbol	Default Value
Context window size	W	600
Entropy threshold	τ_Λ	0.4
Histogram division factor	D	2
Number of H sentences	J	8
Number of NH sentences	K	8

Table 4.7: Frame-level recall, precision, and F-score achieved by our approach on tricking (averaged over all MATDAT dataset videos), diving, long jump, pole vault, and tumbling, along with the overall average across these sports.

	Recall (%)	**Precision (%)**	**F-score (%)**
Diving	97.16	64.00	77.16
Long jump	81.16	96.05	87.98
Pole vault	93.46	88.75	91.05
Tumbling	93.04	82.03	87.19
Tricking	98.68	65.83	77.00
Average	92.70	79.33	84.08

several pairs to survive the subsequent filters and drive stable performance. All sentences were initially produced using a large-scale Transformer-based language model (OpenAI GPT family) prompted with concise sport-specific templates such as "Describe a [sport] highlight moment" and "Describe a non-highlight or preparation moment." Generation used default decoding settings to ensure natural yet diverse phrasing. These sentences were then reviewed and lightly edited to (i) fix grammar, (ii) remove near-duplicates, (iii) ensure a balanced variety of motion-centric vs. context-centric descriptions, and (iv) eliminate prompts that accidentally describe post-event scenes (e.g., celebrations). When multiple formulations expressed similar semantics, the most visually neutral wording was retained to avoid introducing stylistic bias. We release the exact final sentence banks per sport for full reproducibility.

Quantitative Results. Using the frame-level metrics defined in Section 4.2, we evaluated the proposed approach on the MATDAT dataset (three martial arts tricking videos) and the SportCLIP dataset (diving, long jump, pole vault, and tumbling). Figure 4.16 illustrates the recall, precision, and F-score for each individual video, while Table 4.7 compiles the aggregated frame-level results for each sport category plus an overall average. In the table, "Tricking" represents a single mean value computed over the three videos in MATDAT. For that domain, we achieve 98.68% recall, 65.83% precision, and 77.00% F-score. Diving yields 97.16% recall and 77.16% F-score, while maintaining a precision of 64.00%. Long jump attains 87.98% F-score with nearly balanced recall (81.16%) and precision (96.05%). Pole vault records 93.46% recall and 88.75% precision for a 91.05% F-score, and tumbling reaches 93.04% recall, 82.03% precision, and 87.19% F-score. Averaged across all sports, we observe 92.70% recall, 79.33% precision, and 84.08% F-score, indicating that our pipeline adapts effectively to different movement patterns and highlight distributions and consistently identifies key highlight frames while minimizing false positives.

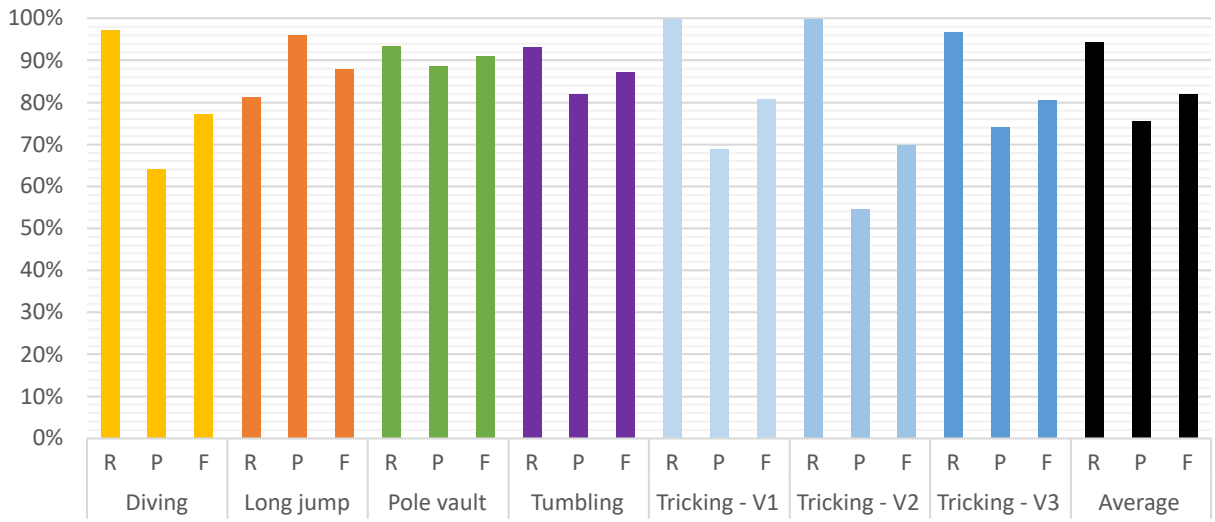


Figure 4.16: Bar plot of frame-level recall, precision, and F-score for each video in both the MATDAT and SportCLIP datasets. The right-hand columns show the average results across all videos.

Qualitative Results. Figure 4.17 compares the final predictions on the first frames of two example videos—tricking V3 (from MATDAT dataset) and tumbling (from SportCLIP dataset)—after averaging the frame-level highlight predictions of all valid *HL-NHL* sentence pairs. The figure also illustrates the sequential post-processing steps described in [84]: raw frame-level predictions, consolidation into events using rolling averages (instant vs. context), merging of closely spaced events, calculation of the area enclosed between the instant and context curves, refined masking based on those areas, and removal of events with areas below the threshold τ_{area}. By the final step, the detected highlights (shown in green) closely match the ground truth annotations, while non-highlight intervals (in red) are accurately separated. These observations show that the pipeline successfully discards noise and retains legitimate highlights, resulting in final predictions that closely align with the ground truth events and thereby complement the quantitative findings presented earlier.

Notably, the proposed framework effectively captures fast-paced or abrupt highlight actions such as flips, vaults, or short acrobatic bursts. These events often span only a few dozen frames, causing sharp peaks in the raw highlight probability curves. The context-window smoothing inherited from [84] preserves these localized peaks while suppressing noise, allowing short but intense highlight segments to survive the subsequent filtering stages. In practice, sports such as tricking and diving—characterized by rapid motion and brief highlight durations—yield stable detections without fragmentation, as the temporal aggregation and event-merging stages consolidate adjacent high-confidence frames into coherent highlight clips.

4.4.2 Comparative analysis

Next, we contrast our approach with the dual-learner-based video highlight detection (DL-VHD) method of Xu *et al.* [119]. To our knowledge, it is the only prior solution that focuses

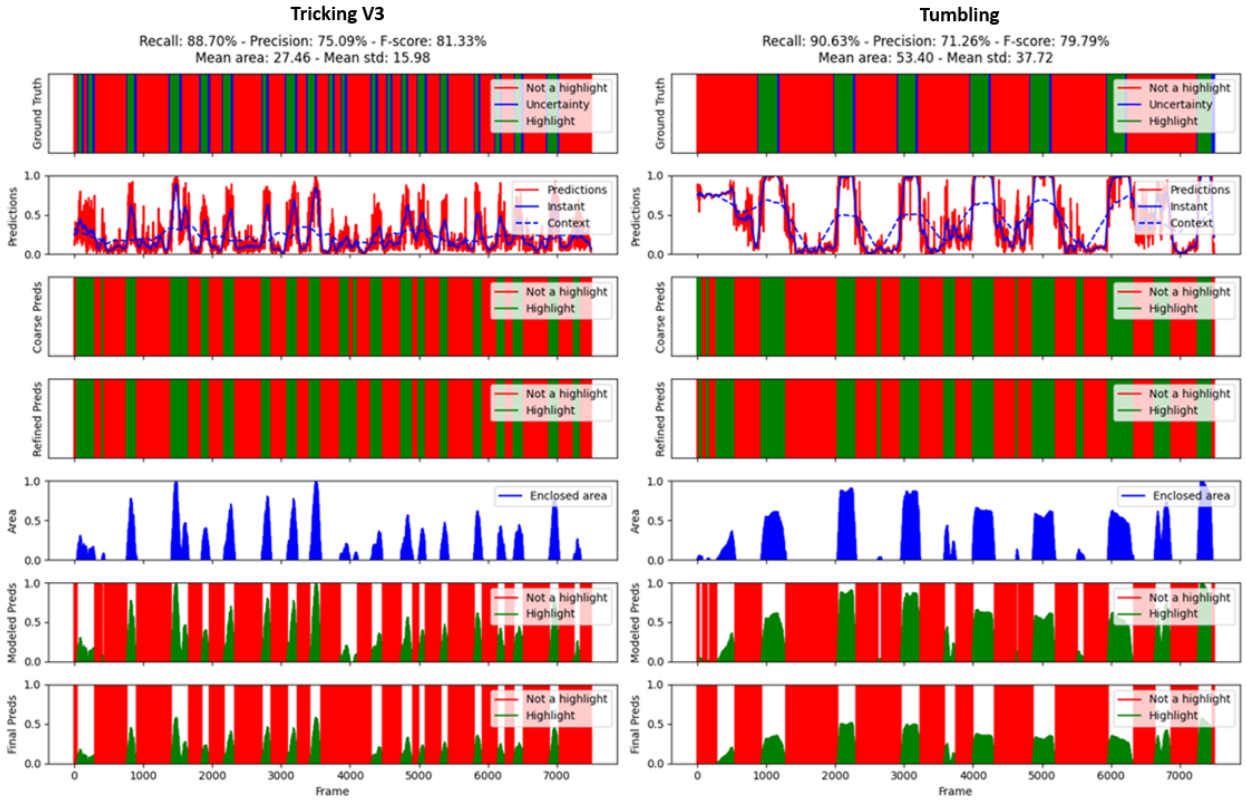


Figure 4.17: Illustration of the post-processing stages described in [84] for two example videos: tricking V3 (from MATDAT dataset) and tumbling (from SportCLIP dataset). Each row shows a different stage (raw frame-level highlight predictions, event consolidation, area computation, and final thresholding), after averaging the predictions from valid *HL-NHL* sentence pairs. Frames marked in green indicate highlights, red indicates non-highlight events, and blue denotes uncertainty. The final outputs closely match the ground truth annotations, underscoring the pipeline's effectiveness in refining noisy predictions and preserving genuinely important segments.

Table 4.8: Cross-category highlight detection mAP scores for DL-VHD, using each row's *Source* category and transferring to the columns' *Target* category. The best mAP per column is shown in bold, and the second best is underlined.

Source	Target (mAP %)				
	Diving	**Long jump**	**Pole vault**	**Tumbling**	**Tricking**
Gymnastics	<u>23.47</u>	72.40	49.10	36.63	25.65
Parkour	**52.88**	<u>77.25</u>	**65.60**	<u>67.31</u>	<u>30.49</u>
Skating	17.90	**82.46**	32.43	66.40	17.72
Skiing	20.58	73.55	<u>52.63</u>	**74.44**	**33.74**
Surfing	22.34	63.06	49.42	49.70	27.94

on cross-category (i.e., multi-sport) highlight detection *and* provides public code, thereby enabling a direct comparison. While other recent methods (e.g., Zhang *et al.* [130], Li *et al.* [60]) also address multi-sport scenarios, they do not release open-source implementations or models, making it impractical to replicate their exact settings on our data. Consequently, we follow the procedure outlined in [119]—training on the YouTube Highlights categories (skating, parkour, etc.) and then testing on the SportCLIP sports (diving, long jump, pole vault, tumbling)—to evaluate how well DL-VHD generalizes across these diverse disciplines.

Regarding our earlier highlight detection method for martial-arts tricking [84], a direct numerical comparison is unfortunately not meaningful in the present multi-sport setting. The method in [84] was engineered exclusively for tricking, presumes a static camera, and demands scenario-specific parameter tuning, which makes cross-domain evaluation impractical. In contrast, the method proposed here is sport-agnostic, camera-agnostic, and uses a single parameter set for all experiments. While [84] attains a slightly higher F-score on the MATDAT tricking videos (an averaged F-score across the three videos of 82.83%), our current multi-sport method obtains 77.00% on MATDAT (Table 4.7) using a single parameter set across sports, trading a marginal drop for substantially increased flexibility and generality.

Table 4.8 shows the mean Average Precision (mAP) scores (see Section 4.2) obtained by DL-VHD on SportCLIP under cross-category evaluation. Although their approach is designed around segment-level classification, we adapt the official implementation to the SportCLIP videos and compute standard mAP for the detected highlight segments.

We further convert the segment-based results to frame-level recall, precision, and F-score, following the definitions in Section 4.2. Table 4.9 shows the resulting values after applying our conversion procedure. Specifically, we follow the annotation-to-segment conversion protocol described in [84]: each video is divided into 100-frame segments with 50% overlap, assigning to each segment the label (highlight, non-highlight, or uncertainty) that appears most frequently within its constituent frames. The resulting segment-level predictions from DL-VHD are then re-expanded to frame level by assigning the segment label to all its frames and resolving overlaps by majority voting. This process ensures metric comparability between DL-VHD's segment-based outputs and our frame-level predictions.

To further reinforce the comparative analysis, we benchmarked an additional vision–language model originally designed for highlight detection, the Query-Dependent DETR (QD-DETR) [72].

Table 4.9: Frame-level recall, precision, and F-scores for cross-category highlight detection, where each row represents the *source* domain and each column the *target* sport. Results correspond to the baseline DL-VHD [119] method, used for comparison against our proposed approach. The best F-score per column is shown in bold, and the second best is underlined.

Source	Target														
	Diving			Long jump			Pole vault			Tumbling			Tricking		
	R (%)	P (%)	F (%)	R (%)	P (%)	F (%)	R (%)	P (%)	F (%)	R (%)	P (%)	F (%)	R (%)	P (%)	F (%)
Gymnastics	66.07	35.05	**45.80**	92.45	66.39	<u>77.29</u>	96.58	48.54	<u>64.61</u>	88.95	57.99	<u>70.21</u>	77.20	27.26	39.58
Parkour	34.34	58.11	<u>43.17</u>	70.64	77.24	73.80	87.43	60.22	**71.32**	63.04	80.16	**70.58**	31.86	51.90	39.48
Skating	83.81	23.82	37.10	100.00	59.65	74.73	100.00	44.09	61.20	46.25	67.20	54.79	100.00	23.01	37.41
Skiing	62.62	23.76	34.45	98.68	64.55	**78.05**	67.61	52.57	59.15	81.17	54.73	65.38	50.89	48.41	**49.62**
Surfing	75.90	20.55	32.34	80.65	71.29	75.69	73.49	57.03	64.22	97.40	38.70	55.39	34.63	46.49	<u>39.69</u>

This Transformer-based approach extends DETR to learn fine-grained temporal associations between visual features and textual queries. We evaluated it in a zero-shot configuration using its official pretrained weights, without any fine-tuning on either SportCLIP or MATDAT. For a fair comparison, its output scores were post-processed using the same smoothing and thresholding strategy employed in our framework and subsequently converted into frame-level predictions.

As summarized in Table 4.10, QD-DETR exhibits limited transferability to sports-specific highlight detection, with F-scores ranging from 19.13% to 48.97%. This outcome is expected, as its pretrained parameters were optimized for generic video-moment retrieval rather than the fine-grained temporal dynamics typical of structured sports. DL-VHD [119], whose per-sport results correspond to the best-performing source domain identified in Table 4.9, achieves notably stronger performance (e.g., 78.05% on long jump and 71.32% on pole vault) owing to its cross-category learning design. However, it still requires supervised retraining and lacks the frame-level precision needed for accurate highlight boundary delineation. By contrast, our proposed method consistently achieves the highest performance across all sports, with F-scores above 77% in every case and reaching over 91% for pole vault. These results highlight that a text-guided, training-free approach can surpass both generic and cross-category supervised baselines while maintaining robustness across distinct motion patterns and highlight durations.

Overall, the results in Table 4.10 demonstrate that generic vision–language detectors such as QD-DETR struggle to generalize effectively to the specialized domain of sports highlight summarization, while cross-category methods like DL-VHD partially bridge this gap but remain constrained by their reliance on labeled data and retraining. In contrast, our CLIP-based framework capitalizes on text guidance and robust sentence-pair filtering to achieve superior frame-level precision and recall without any domain-specific adaptation or supervision.

4.4.3 Parameter sensitivity

We further assessed our method's robustness by varying three key parameters—(i) the context window size used for rolling averages, (ii) the entropy threshold for the distribution-based filter, and (iii) the histogram division factor governing the area-based filter—and tracking their impact on frame-level F-scores across both MATDAT (V1, V2, V3) and SportCLIP (diving, long jump, pole vault, tumbling). As shown in Fig. 4.18, the first subplot illustrates

Table 4.10: Frame-level recall, precision, and F-score per sport for QD-DETR, DL-VHD, and the proposed method. Best values are shown in bold, and the second best is underlined. This table provides a unified summary comparison of all methods, highlighting the performance gap between our approach and prior state-of-the-art baselines.

Method	Recall (%)	Precision (%)	F-score (%)
Diving			
QD-DETR [72]	<u>69.28</u>	28.59	40.47
DL-VHD [119]	66.07	<u>35.05</u>	<u>45.80</u>
Proposed (ours)	**97.16**	**64.00**	**77.16**
Long jump			
QD-DETR [72]	31.80	<u>89.08</u>	46.87
DL-VHD [119]	**98.69**	64.55	<u>78.05</u>
Proposed (ours)	<u>81.16</u>	**96.05**	**87.98**
Pole vault			
QD-DETR [72]	11.33	<u>61.35</u>	19.13
DL-VHD [119]	<u>87.43</u>	60.22	<u>71.32</u>
Proposed (ours)	**93.46**	**88.75**	**91.05**
Tumbling			
QD-DETR [72]	40.37	62.24	48.97
DL-VHD [119]	<u>63.04</u>	<u>80.16</u>	<u>70.58</u>
Proposed (ours)	**93.04**	**82.03**	**87.19**
Tricking			
QD-DETR [72]	28.35	16.27	20.46
DL-VHD [119]	<u>50.89</u>	<u>48.41</u>	<u>49.62</u>
Proposed (ours)	**98.68**	**65.83**	**77.00**
Average			
QD-DETR [72]	36.27	51.51	35.18
DL-VHD [119]	<u>73.22</u>	<u>57.68</u>	<u>63.07</u>
Proposed (ours)	**92.70**	**79.33**	**84.08**

how F-scores evolve with changes to the number of neighboring frames. Notably, a larger window particularly benefits sports such as long jump and pole vault, where slightly lengthier highlight actions gain from the added temporal context—resulting in smoother and more accurate highlight detection. In contrast, tricking sequences (V1–V3) and diving, which may feature shorter, more abrupt motions, show minimal or mixed sensitivity to window size.

The second subplot shows F-scores against a range of entropy thresholds (τ_Λ), indicating that our ensemble-based strategy remains robust even under aggressive filtering of sentence pairs with near-uniform highlight predictions. This resilience suggests that the multi-sentence approach, paired with the subsequent averaging step, effectively mitigates the influence of sentences that fail to discriminate distinct highlights.

Lastly, the third subplot compares frame-level F-scores across different values of D. As expected, a larger D produces a smaller τ_{area}, thus filtering out fewer events. By contrast, a smaller D raises τ_{area} and discards additional borderline segments, occasionally causing mild F-score dips for videos with many short highlights (e.g., V3 or diving). Nonetheless, overall performance remains around 80% across a broad range of D values, illustrating that the framework tolerates varying degrees of strictness while still preserving truly meaningful highlights.

Collectively, these observations confirm that our pipeline reliably captures essential highlights across a range of parameter values, sustaining high-quality detection and underscoring the adaptability of the proposed framework in handling varied highlight durations and motion patterns. Concretely, tricking V3 and diving exhibit small F-score dips when we tighten the area-based filter (smaller D), consistent with their many short highlights; stricter thresholds prune borderline segments that still correspond to valid micro-events. By contrast, long jump and pole vault benefit from larger context windows, reflecting their longer, more cohesive highlight spans. Occasional false positives arise in post-event contexts (e.g., swimmers surfacing or celebrations) that resemble motion endpoints, and false negatives appear under occlusion or motion blur in fast tumbling/tricking passes—effects that also explain the sensitivity to aggressive area thresholds.

Overall, these trends support a simple operating point: a single, sport-agnostic configuration with a moderate context window, a conservative entropy threshold, and a non-aggressive area filter. We adopt this setting throughout the thesis, yielding strong and stable performance across datasets.

4.4.4 Text Sensitivity to Prompt Wording

To evaluate the sensitivity of the proposed CLIP-based framework to variations in textual phrasing, we conducted an experiment using five independently written prompt sets for each sport. Each set included eight *Highlight* and eight *Non-Highlight* sentences, resulting in a total of 64 pairs. These sentences describe analogous situations with slight wording differences, allowing us to analyze whether the model maintains consistent performance when exposed to distinct yet semantically equivalent textual inputs. Because these descriptions are generated from high-level sport templates rather than handcrafted phrases, the resulting linguistic variability resembles realistic user-provided prompts, allowing the analysis in this section to

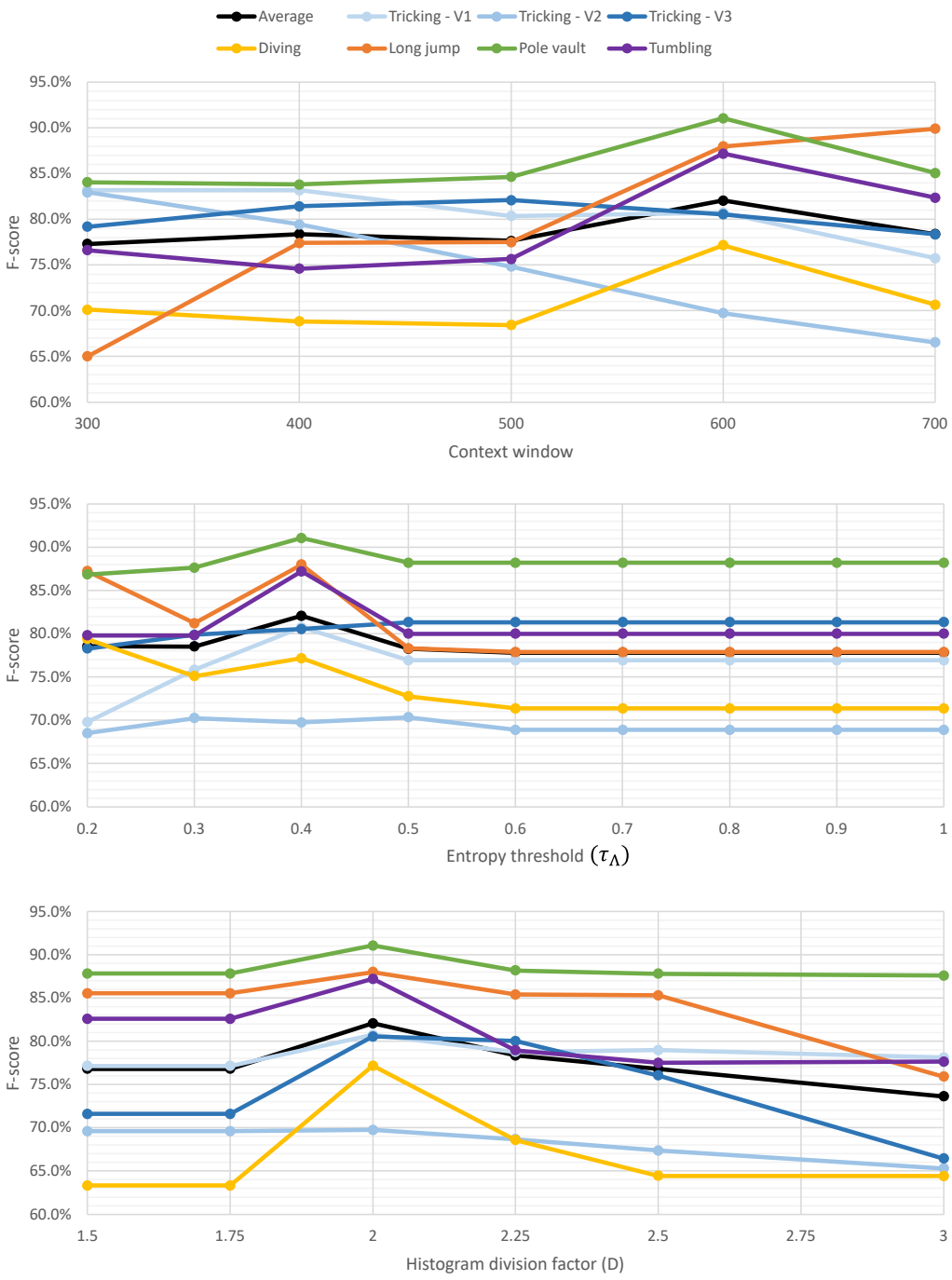


Figure 4.18: Frame-level F-scores under different parameter configurations across MATDAT (V1, V2, V3) and SportCLIP (diving, long jump, pole vault, tumbling). Each subplot isolates one of the three primary settings: (top) context window size, (middle) entropy threshold, and (bottom) histogram division factor for the area-based filter. Although certain videos show local performance variations, the overall F-scores remain consistently strong (typically 75–85% or higher), highlighting the method's resilience to moderate parameter changes.

Table 4.11: Frame-level recall, precision, and F-scores (mean ± std) obtained across five independently written prompt sets for each sport. Each set comprises eight *Highlight* and eight *Non-Highlight* sentences, as defined by the parameters in Table 4.6. Results reflect the stability of the proposed framework under semantically consistent variations in textual input.

Sport	**Recall (%)**	**Precision (%)**	**F-score (%)**
Diving	93.67 ± 7.05	55.64 ± 7.79	69.34 ± 5.16
Long jump	79.71 ± 5.01	87.04 ± 6.73	83.00 ± 3.27
Pole vault	93.15 ± 3.08	87.20 ± 3.95	89.97 ± 1.28
Tumbling	89.71 ± 3.01	82.16 ± 1.62	85.73 ± 0.95
Tricking	96.88 ± 1.21	59.92 ± 4.78	72.63 ± 3.63
Average	90.62 ± 2.14	74.39 ± 3.97	80.13 ± 2.80

reflect practical usage scenarios.

All experiments were conducted using the same parameters reported in Table 4.6 and the same post-processing procedure described in the *Qualitative Results* subsection of Section 4.4, which is also illustrated in Figure 4.17. Table 4.11 summarizes the mean and standard deviation of frame-level recall, precision, and F-score across the five prompt sets for each sport.

As shown, the results remain consistent across prompt formulations. Within each sport, the F-score typically varies by less than 10 percentage points, confirming the robustness of the method to reasonable textual modifications. Across sports, average F-scores range from approximately 70% to 90%, with an overall mean of 80.13%, indicating that CLIP effectively generalizes across distinct linguistic expressions as long as the prompts preserve semantic relevance to the visual content.

While the framework demonstrates robustness to typical wording variations, it remains sensitive to descriptions that deviate excessively from the actual visual content. As discussed in Section 3.3.2 and illustrated in Figure 3.15, prompts that exaggerate or misrepresent the scene (e.g., attributing implausible actions to the athlete) tend to produce unstable saliency curves and are automatically filtered out by the distribution-based and mean-area criteria introduced in our sentence selection strategy. This behavior aligns with our design: the model does not generate meaningful highlight predictions when the text fails to match the video semantics, effectively suppressing incoherent or exaggerated descriptions.

Overall, these results confirm that the proposed text-guided summarization approach is robust to normal variations in sentence formulation, provided that the textual prompts remain semantically consistent with the video content. This validates the method's generalization capability across diverse user-specified descriptions while preventing spurious highlight detections from off-topic or exaggerated text.

Although our framework is text-guided, the balance between visual and textual cues is inherently maintained by CLIP's dual-encoder design. The image encoder captures detailed spatial and appearance information from each frame, while the text encoder provides high-level semantic anchors describing the expected actions or contexts. Because similarity is computed

jointly between both embeddings, accurate alignment requires mutual consistency—frames that visually match the textual description yield high scores, whereas exaggerations or mismatched wording lead to uniformly low responses. In practice, our sentence-pair filtering and temporal smoothing stages automatically discard such incoherent pairs, ensuring that the resulting highlights remain visually grounded even when some textual descriptions are suboptimal.

4.4.5 Computational Cost

To evaluate the efficiency of the proposed framework, we analyzed the computational cost of each component in the pipeline using several SportCLIP and MATDAT videos processed at 30 fps and 1920×1080 resolution. All experiments were conducted on a workstation running Ubuntu 24.04 LTS with an Intel Core i9-13900K, 128 GB RAM, and a single NVIDIA GeForce RTX 4090 GPU.

The computational analysis follows the pipeline illustrated in Figure 3.15, which consists of five stages: sentence embedding computation, frame embedding extraction, similarity evaluation, sentence filtering, and post-processing. Each stage operates at a different frequency: sentence embeddings are computed once per sentence set, frame embeddings are computed once per frame, similarities are evaluated per frame and per sentence pair, while filtering and post-processing are applied once per pair of sentences. To ensure comparability, the processing time of each component was normalized by its natural unit of work (per sentence set, per frame, or per sentence pair).

Table 4.12 summarizes the average time required by each component, using the same parameter configuration as in Table 4.6 (8 *Highlight* and 8 *Non-Highlight* sentences, i.e., 64 pairs). The computation of CLIP text embeddings is performed only once per run and requires approximately 1.26 s to process the entire sentence set, representing the largest fixed cost in the pipeline. The extraction of CLIP frame embeddings is also highly efficient, averaging 0.216 ms per frame. This efficiency stems from the use of large batch sizes during inference: although the GPU could accommodate over 8,000 frames simultaneously, we adopted a more conservative batch size of around 7,000 frames to avoid memory spikes. The similarity computation, which evaluates frame–text alignment scores across all frames and sentence pairs, is similarly lightweight and scales linearly with both video duration and sentence count, requiring about 0.125 ms per frame and pair. The sentence-filtering and post-processing stages introduce negligible additional cost, with average runtimes of 10.60 ms and 0.003 ms per pair, respectively.

As shown in Table 4.12, the total runtime increases linearly with video duration, primarily due to the frame embedding and similarity computation stages, both of which operate at frame-level granularity. For a one-minute video, the complete pipeline requires approximately 16.8 s to process. Scaling this configuration to longer sequences yields total runtimes of about 1.2 min for a five-minute video, 2.5 min for a ten-minute video, and 14.8 min for a one-hour video. The cost of computing the sentence embeddings remains constant across runs, while the remaining components scale predictably with either video length or the number of sentence pairs, depending on their operating frequency.

Table 4.12: Average processing time of each pipeline component, normalized by its natural unit of work. Values are reported as mean ± std over all videos.

Component	Unit of Work	Time
Sentence Embeddings	per sentence set	1.263 ± 0.045 s
Frame Embeddings	per frame	0.216 ± 0.003 ms
Similarity computation	per frame × pair	0.125 ± 0.290 ms
Sentence filtering	per pair	10.600 ± 0.054 ms
Post-processing	per pair	0.003 ± 0.005 ms
Total (64 pairs, 30 fps)	per 1-min video	≈ 16.8 s
Total (64 pairs, 30 fps)	per 5-min video	≈ 1.2 min
Total (64 pairs, 30 fps)	per 10-min video	≈ 2.5 min
Total (64 pairs, 30 fps)	per 1-hour video	≈ 14.8 min

Overall, the majority of the computational effort arises from the frame embedding extraction and the subsequent similarity computation. The filtering and post-processing steps contribute negligible overhead relative to these dominant components. Considering that sentence embeddings are computed once and reused across all videos, and that both frame embedding extraction and similarity evaluation can be parallelized or batched, the framework maintains an excellent balance between accuracy and efficiency. The near-linear scaling behavior demonstrates that the proposed system can process long-form sports recordings within practical time limits, and could approach near–real-time performance on a single GPU when optimized for streaming or deployment scenarios.

4.4.6 Practical Usage

Our publicly released code can be obtained from the project repository[3]; an end-user need only install the standard requirements and supply a video together with at least one pair of *HL-NHL* sentences—optionally several pairs for increased robustness. Our method will automatically extract CLIP features, apply the proposed filtering and aggregation strategy, and return both per-frame highlight scores and a summary clip that can be played on any device.

Because the pipeline is prompt-driven and requires no further training or parameter tuning, it can be integrated into a wide range of consumer-electronics workflows—from desktop media libraries to cloud streaming back-ends—without specialized hardware or bespoke engineering. Modifying the textual prompts is sufficient to tailor the output to new sports, languages, or user preferences, making the method a practical, drop-in tool for automatic highlight generation.

[3]Source code is available at https://www.gti.ssr.upm.es/data.

4.5 UPM-GTI-Face A dataset for the evaluation of the impact of distance and masks in face detection and recognition systems

This section presents the empirical analysis of the UPM-GTI-Face dataset [87] introduced in Section 3.5, focusing on the joint impact of distance, environment, and face masks on face detection and recognition. We first study face detection performance across all probe images, reporting true detection rate (TDR; see Section 4.2.2) and average face sizes at each distance for indoor and outdoor scenarios, with and without masks, using a Tiny Faces detector [45] configured to maximize the effective range. We then evaluate recognition performance by feeding detected face crops to three backbone architectures (VGG16 [91], ResNet50 [40], and SeNet50 [44]), analyzing receiver operating characteristic (ROC) curves, area under the ROC curve (AUC), and equal error rate (EER), following the definitions in Section 4.2.2, at representative distances and for both matched and crossed mask conditions. Together, these results quantify how modern detection and recognition pipelines degrade under surveillance-like constraints and highlight the specific challenges posed by long distances and mask usage.

4.5.1 Face detection performance

Table 4.13 reports the true detection rate (TDR) for the different scenarios (environments and face-mask conditions) at each measured distance, as well as the average face size in pixels. Following Section 4.2, TDR is computed as the proportion of annotated faces for which the detector returns at least one valid bounding box. To understand the operational limits of the face detection algorithm, the detector has been applied to all 440 probe images, which were captured with the highest possible resolution from the camera (4K). Tiny Faces is applied on an image pyramid: each 4K probe image is resized by a set of scale factors before detection. In our experiments we use four scales, $[0.1, 0.5, 1.0, 1.4]$, meaning that the detector processes versions of the image at 10%, 50%, 100% and 140% of the original resolution, so that its scale-specific templates can cope with faces that appear at very different pixel sizes. We found that adding larger scales beyond 1.4 did not lead to a noticeable improvement in detection performance.

From these results, it can be appreciated that the detection network manages to detect faces more consistently in the outdoor environment. This is mainly due to a more homogeneous illumination when compared to that of the indoor environment, where walls and windows produce more abrupt changes in illumination at different distances (i.e., masked subjects at the 24 m indoor mark). Regarding face masks, it is clear the performance drop for those cases where subjects were wearing them. Detection results show that unmasked subjects are substantially easier to detect, especially at longer distances. Given that the detection network was not trained with masked face images this is to be expected. However, we report this degradation numerically, which value should not be underestimated, as it serves as an indicator of the degradation one should expect when employing a state-of-the-art detection algorithm trained using only unmasked subjects. Note that True Detection Rate does not monotonically

Table 4.13: True Detection Rate at different distances and for Indoor / Outdoor environments, and No Mask / Mask conditions. Face size represents the average size in pixels of the detected bounding boxes.

Distance	Indoor			Outdoor		
	No Mask	Mask	Face size (px)	No Mask	Mask	Face size (px)
3 m	100%	100%	149×192	100%	100%	125×164
6 m	100%	100%	77×97	90%	90%	68×92
9 m	100%	100%	49×63	100%	100%	46×62
12 m	100%	100%	33×46	100%	100%	33×46
15 m	100%	100%	31×47	100%	81%	30×36
18 m	100%	81%	26×33	100%	90%	23×32
21 m	100%	90%	23×30	81%	63%	20×28
24 m	45%	9%	16×20	81%	63%	19×27
27 m	36%	36%	20×28	81%	54%	15×20
30 m	18%	9%	22×31	81%	45%	14×17

decrease with distance, as there are other factors that affect detection performance, such as the mentioned illumination conditions.

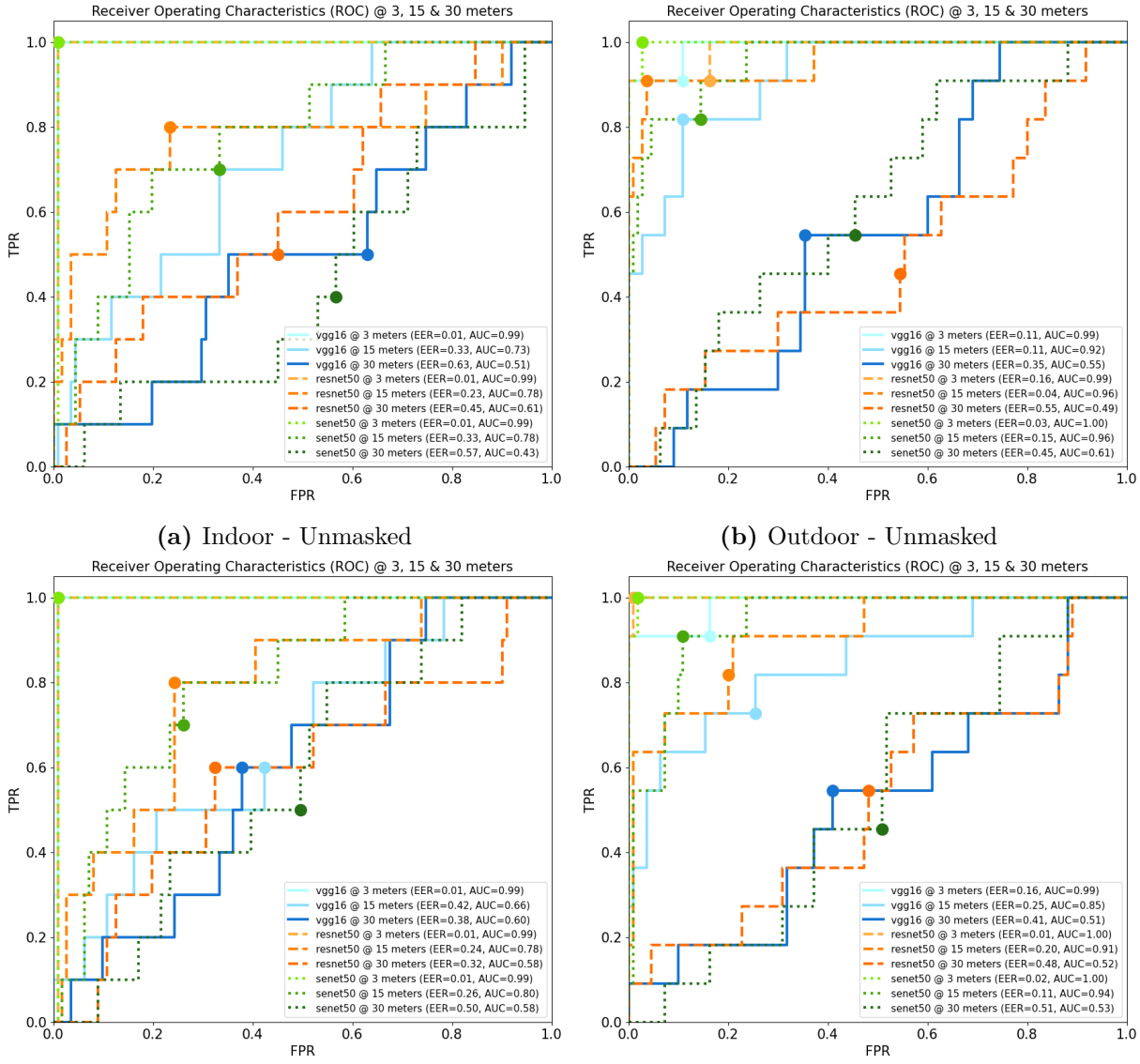
4.5.2 Face recognition performance

To assess the impact of distance and face masks on face recognition performance, we propose different evaluation environments (indoors and outdoors) and conditions (subjects with and without face masks) on which we apply the face recognition algorithm at each of the 10 considered distances. Instead of feeding entire frames into the face recognition network, only face crop images detected by the detection algorithm are used as input. Additionally, we report the results obtained when using three different backbone networks for the face recognition algorithm, namely VGG16 [91], ResNet50 [40], and SeNet50 [44].

All results are summarized in Fig. 4.19, where receiver operating characteristic (ROC) curves are illustrated for all scenarios at three representative distances: 3, 15, and 30 meters. Each subfigure corresponds to a particular environment and mask condition and summarizes the recognition results obtained for the probe face images when contrasted against the face images of the gallery subjects, both under the same conditions. The corresponding AUC and EER values for each curve, defined in Section 4.2, are also reported in the figure. For instance, Fig. 4.19a describes the results obtained for the probe face images of unmasked subjects in the indoor environment when contrasted against the indoor gallery of face images of unmasked subjects.

4.5.2.1 The effect of distance

The effect of distance on face recognition can be easily noticed in Fig. 4.19. For all scenarios and backbones, recognition rates quickly decreased for distances greater than 15 meters. At 30 meters all ROC curves settle around the diagonal, where the false positive rate (FPR) increases linearly with the true positive rate (TPR), indicating random behavior of the recognition



(a) Indoor - Unmasked

(b) Outdoor - Unmasked

(c) Indoor - Masked

(d) Outdoor - Masked

Figure 4.19: ROC curves obtained for different scenarios when contrasting gallery and probe images, both under the same conditions. Additionally, the equal error rate (EER), represented with circles, and the area under the curve (AUC) are provided for every curve.

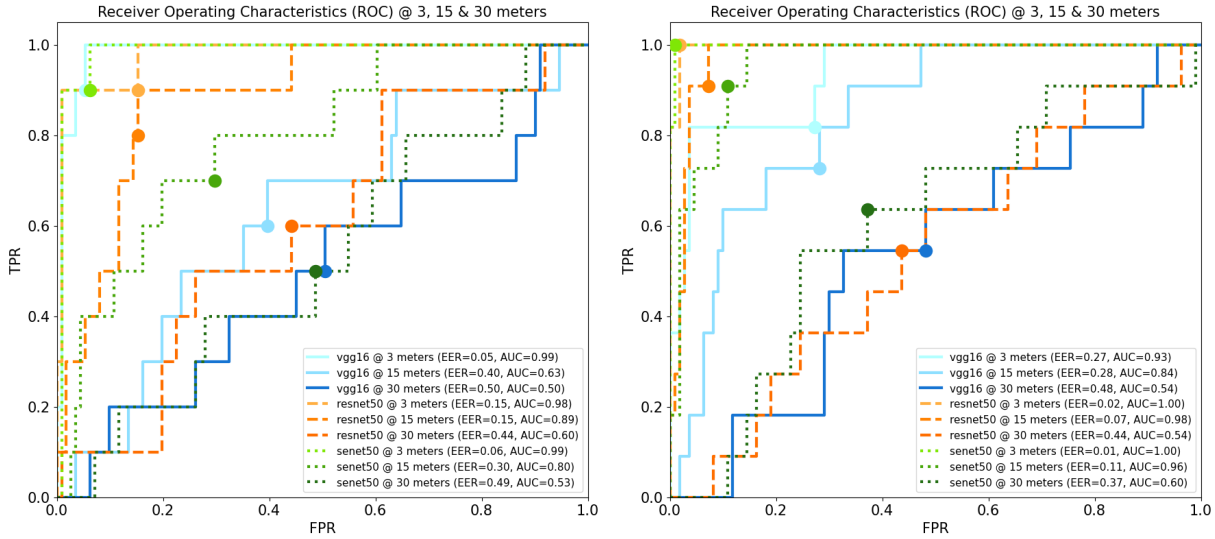
algorithm. However, outdoor environments proved to be less affected by the effect of distance. For instance, ROC curves in Fig. 4.19a and 4.19c dropped significantly lower from 15 to 30 meters than those of Fig. 4.19b and 4.19d. This is mainly due to a more homogeneous illumination than in the indoors environment, where walls and sources of light caused more abrupt changes.

Regarding backbones, they all performed similarly at different ranges up to 30 meters, where all backbones started behaving randomly. However, SeNet50 proved to be more robust to the impact of distance, achieving the highest AUC in almost every combination of scenario and distance, while at the same time presenting some of the lowest Equal Error Rate (EER).

4.5.2.2 The effect of face masks

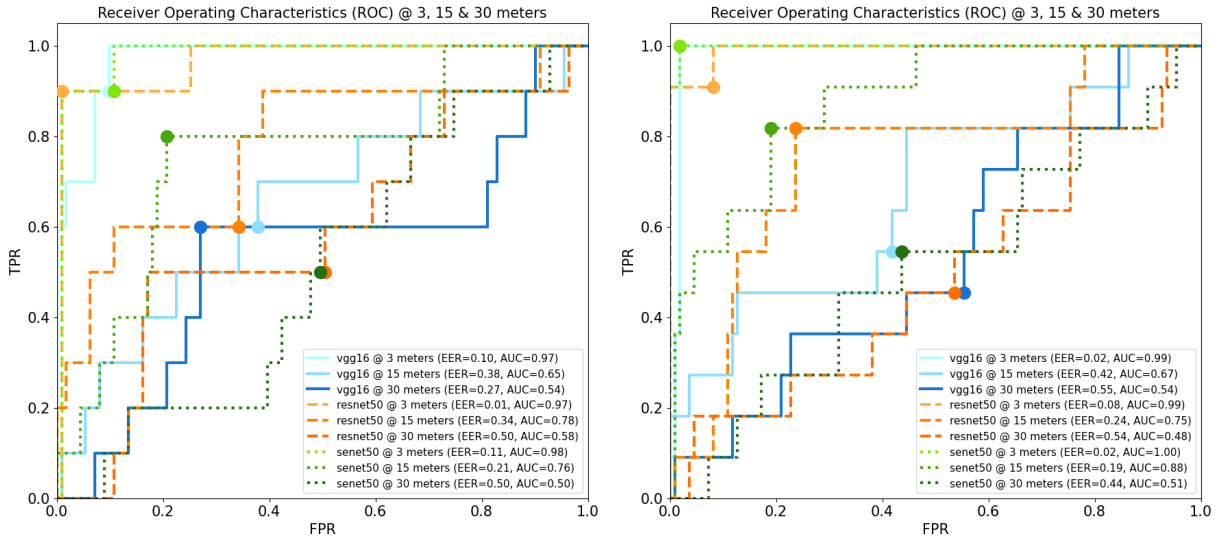
The effect of face masks proved not to be much of a problem at shorter distances, where the rest of the subject's face is still visible. This might also be due to the fact that previous studies indicate that eyes are among the most discriminative facial regions [105]. However, when comparing Fig. 4.19a and 4.19b with Fig. 4.19c and 4.19d, it is easy to notice that recognition rate dropped much quicker with distance when subjects were wearing face masks, especially indoors. Regarding backbones, it is interesting to note how each backbone is affected slightly different by the effect of face masks and distance, with VGG16 being affected significantly more (i.e., VGG16 at 15 meters performed much better than at 30 in Fig. 4.19a, while in Fig. 4.19c it performed equally bad).

In addition to previous Fig. 4.19, which shows the recognition results obtained when contrasting gallery and probe images under the same scenarios (environments and face mask conditions), Fig. 4.20 shows the crossed results obtained when contrasting gallery and probe images captured in the same environment but with different face mask condition. These results are useful for determining the reliability of a face recognition algorithm trying to recognize a masked subject in a probe image when only an unmasked gallery image of that subject is available, and vice-versa, which is a very common occurrence in surveillance scenarios. It is very clear that ROC curves in Fig. 4.20 are distributed more unevenly than in previous Fig. 4.19, which indicates that the recognition results are less reliable, and overall these curves present lower AUC scores. Interestingly enough, results were better when the gallery subjects were masked and the probe subjects were unmasked. This is probably because gallery images were captured at a close distance, so relevant face embeddings could still be extracted from the eyes region, and probe images were unmasked, which made them more differentiable even at a distance away from the camera. The other way around, with gallery subjects unmasked and probe subjects masked, even though more robust face embeddings could be extracted from the gallery subjects, probe images were less differentiable as masks occluded a high percentage of the face area (especially notable at longer distances). Regarding backbones, once again, VGG16 was the most negatively impacted in this scenarios, while ResNet50 and SeNet50 proved to be able to extract more relevant face embeddings as their results were not so negatively affected.



(a) Indoor - Gallery masked, Probe unmasked

(b) Outdoor - Gallery masked, Probe unmasked



(c) Indoor - Gallery unmasked, Probe masked

(d) Outdoor - Gallery unmasked, Probe masked

Figure 4.20: ROC curves obtained for different scenarios when contrasting gallery and probe images, both in the same environment but under different face mask conditions. Additionally, the equal error rate (EER), represented with circles, and the area under the curve (AUC) are provided for every curve.

4.6 Comprehensive comparison between vision transformers and convolutional neural networks for face recognition tasks

This section reports the empirical study of Vision Transformers versus Convolutional Neural Networks for face recognition tasks introduced in Section 3.6. We conduct a comparative analysis of a ViT_B32 model (ViT base architecture with a patch size of 32 pixels) and five widely used CNN families—ResNet [40], VGG [91], Inception [99], MobileNet [42], and EfficientNet [101]—under a unified training protocol on VGGFace2 [7], examining both training behavior and identification accuracy on a held-out subset. We then evaluate all six models on a suite of verification benchmarks—LFW [46] for unconstrained faces, SCface [30] for surveillance imagery, ROF [22] for real-world occlusions, and UPM-GTI-Face [87] for distance and mask robustness—using the ROC-based metrics introduced in Section 4.2.2, namely the area under the ROC curve (AUC) and the equal error rate (EER), together with threshold-dependent recall, precision, and F-score. Finally, we relate these results to model complexity and inference time, providing a comprehensive view of the trade-offs between ViT and CNN architectures in realistic face recognition scenarios.

More specifically, we compare ViT_B32 with ResNet_50, VGG_16, Inception_V3, MobileNet_V2, and EfficientNet_B0, although for simplicity we will refer to them as ViT, ResNet, VGG, Inception, MobileNet, and EfficientNet. To ensure a rigorous and objective comparison, all six networks are trained under a common set of hyperparameters frequently used in the literature: input image size of 224×224 pixels, batch size of 256, 25 epochs, Adam optimizer, and a learning rate of 0.0001. This uniformity in training conditions simplifies the subsequent comparative analysis, allowing us to make more meaningful and objective comparisons among ViTs and CNNs. It is important to emphasize that while this approach streamlines the comparison process, we remain aware that, in practice, networks might indeed perform optimally with distinct hyperparameter settings tailored to their specific characteristics. We also fix the random seeds across runs and datasets to improve the robustness and reproducibility of the comparison.

All networks are implemented in Python using the TensorFlow framework and trained on a workstation equipped with an Intel i9-13900K CPU, two NVIDIA RTX 4090 GPUs, and 128 GB of RAM running Ubuntu 22.04. We adopt a data-parallel strategy to split each batch across both GPUs and report inference times with the models in evaluation mode. The complete implementation, including configuration files and training logs, is publicly available (see Chapter 1), so that our experimental protocol can be inspected and replicated in detail.

4.6.1 Evaluation metrics

The evaluation of the six backbones follows the common framework described in Section 4.2. For the *face identification* experiments on VGGFace2, we report accuracy and top-5 accuracy, measuring respectively the fraction of test images whose predicted identity matches the ground truth and the fraction for which the correct identity appears among the five most likely predictions (see Section 4.2.2). Higher values in both cases indicate better identification

Table 4.14: Training summary on the VGG Face 2 dataset. The accuracy corresponds to the highest values obtained on the training and validation sets during training for the face identification task.

Network	Training accuracy (%)	Validation accuracy (%)
ViT_B32	98.86	**99.81**
ResNet_50	**99.66**	99.55
VGG_16	97.59	98.21
Inception_V3	99.46	99.54
MobileNet_V2	99.18	98.90
EfficientNet_B0	99.27	95.90

performance.

For the *face verification* benchmarks (LFW, SCface, ROF, and UPM-GTI-Face), we use receiver operating characteristic (ROC) curves together with their scalar summaries, the area under the ROC curve (AUC) and the equal error rate (EER), as introduced in Section 4.2.2. ROC curves trace the trade-off between true positive rate and false positive rate as the decision threshold varies; higher AUC and lower EER correspond to better verification behavior. In addition, some figures report recall, precision, and F-score at a single operating point, obtained by selecting for each model the threshold that maximizes F-score. These threshold-dependent metrics provide a more concrete view of performance when the system is operated with a fixed decision rule.

4.6.2 Training

To speed up convergence during training, networks were pre-trained using the Imagenet [16] large-scale dataset. Subsequently, we conducted training on the VGG Face 2 dataset, utilizing both its training and validation subsets. The outcomes of this training endeavor are summarized in Table 4.14, which provides the highest accuracy achieved on both the training and validation sets. Remarkably, all six networks demonstrated exceptional performance, with accuracy levels approaching 100% on both the training and validation sets by the conclusion of the training process. Notably, ViT stood out not only by achieving the highest validation results but also by accomplishing this feat in fewer epochs than its CNN counterparts (for further details, please refer to the additional material in our publicly available implementation, accessible through the link provided in Chapter 1). Furthermore, upon completing training, ViT results on the validation set were still superior to those of the training set by a large margin. Specifically, ViT's accuracy rose from 98.86% to 99.81%, reflecting a remarkable 83.33% loss reduction. This indicates that overfitting [115] has not yet occurred and as such, it is possible that the results could be further improved if the training continued for a few more epochs. On the other hand, CNNs started to exhibit some overfitting signs as the training concluded, raising concerns about their ability to perform as good on datasets that differ from the one they were trained on.

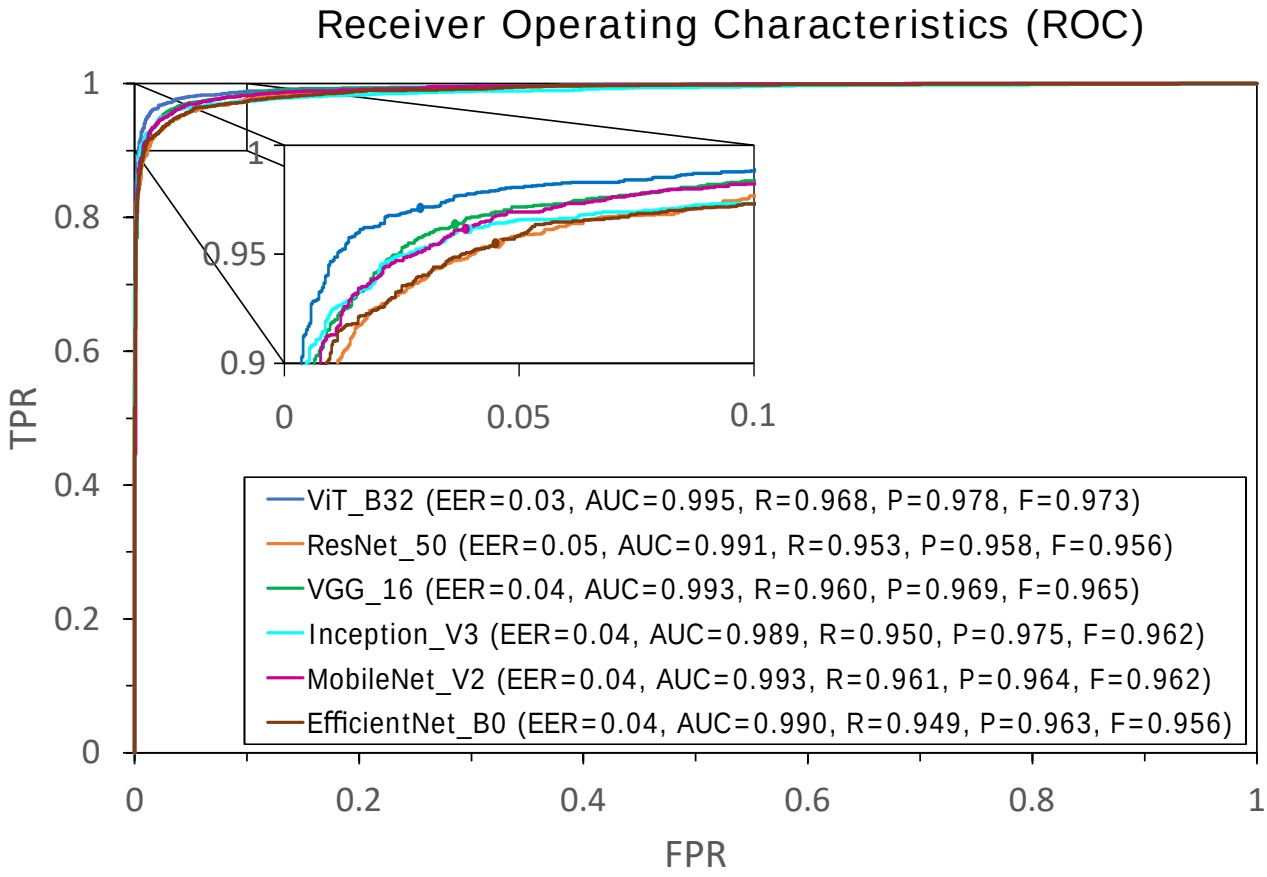


Figure 4.21: ROC curves obtained for the pairs of images proposed in the LFW dataset. EER, AUC, R, P, and F scores are displayed in the legend for each network.

Table 4.15: Face identification results obtained on the evaluation set of the VGG Face 2 dataset. We report test accuracy and top-5 accuracy, as well as the number of parameters of each model and the inference time per batch of 256 images.

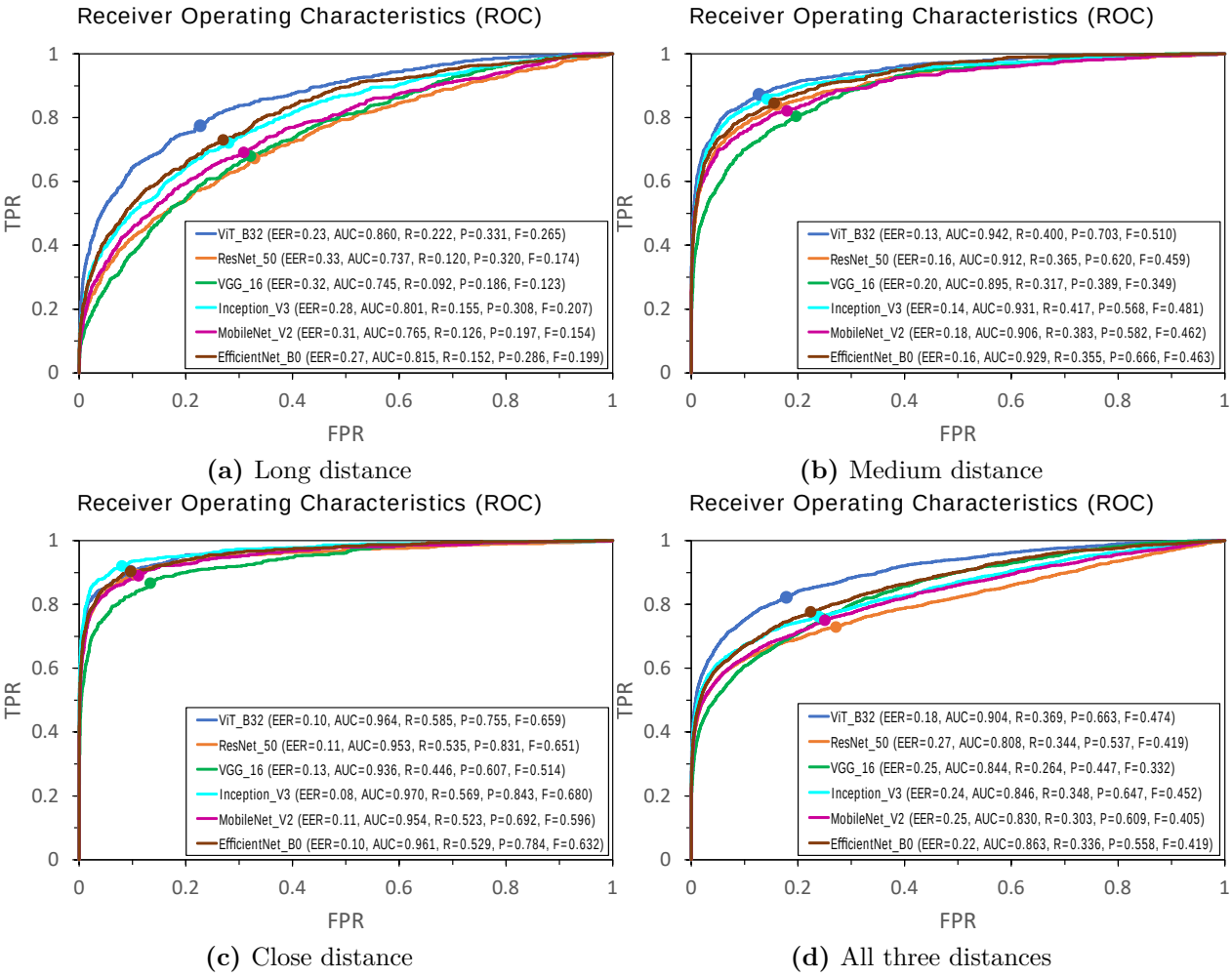
Network	Test accuracy (%)	Top-5 accuracy (%)	Parameters (M)	Inference time (ms / batch)
ViT_B32	**99.80**	**100.00**	94	78
ResNet_50	99.57	99.99	41	105
VGG_16	98.17	99.91	170	141
Inception_V3	99.58	99.99	39	64
MobileNet_V2	98.92	99.96	13	**63**
EfficientNet_B0	95.90	99.42	15	91

4.6.3 Evaluation

The 157,000 images from the VGG Face 2 dataset that were excluded from the training process served as a valuable resource for evaluating the face identification capabilities of the six networks under consideration. In Table 4.15, we present the accuracy results for each network when classifying these images as belonging to one of the subjects within the dataset, alongside the corresponding top-5 accuracy scores. Additionally, we provide information regarding the number of parameters and the inference time per batch. The latter denotes the time required for processing a batch of 256 images when the networks are not in training mode. The findings from this evaluation clearly underscore the superiority of ViTs in terms of accuracy. ViT not only scores the highest accuracy but also distinguishes itself as the only network to attain a flawless 100% top-5 accuracy. Furthermore, ViT's inference speed is notably competitive, aligning with some of the fastest CNNs, with MobileNet being the fastest one. Remarkably, ViT inference speed is only 23.81% slower than that of MobileNet, despite having more than 7 times the number of parameters.

The ROC curves presented in Figure 4.21 offer a summary of the outcomes derived from the face verification task conducted on the LFW dataset by the six networks under scrutiny. While all six networks exhibit commendable performance, ViT stands out with a slightly superior performance, positioning closer to the top-left corner of the graph. Notably, ViT achieves the highest AUC value and the lowest EER. It is worth noting that the LFW dataset, being an older dataset, does not pose a substantial challenge, and all six networks demonstrate exceptional performance on it. Nevertheless, this dataset serves a crucial purpose by showcasing that all six networks are capable of extracting high-quality facial embeddings suitable for face verification tasks. The forthcoming datasets, however, present more formidable challenges.

The outcomes of the face verification task conducted on the SCface dataset are visually depicted in Fig. 4.22. Within this dataset, each subject is represented by a high-quality mugshot gallery image, alongside several images captured by five different cameras at three distinct distances. The ROC curves showcased in Figure 4.22 result from comprehensive comparisons, encompassing all gallery images against all probe images. Specifically, Figure 4.22a presents the outcomes of comparing all mugshot gallery images against the probe images captured at the long distance by every camera. Likewise, Fig. 4.22b and Fig. 4.22c offer parallel



(a) Long distance **(b)** Medium distance

(c) Close distance **(d)** All three distances

Figure 4.22: ROC curves obtained for SCface dataset. This figure encompasses the results obtained for long (a), medium (b), and close (c) distances, as well as the results obtained when synthesizing all results into a single representation (d).

insights for the medium and close distances, respectively. To consolidate these findings, Figure 4.22d synthesizes all the results into a single representation. From these results, it becomes evident that while all six networks perform reasonably well at close distance ViT significantly outperforms its CNN counterparts at medium and long distances. This observation suggests that ViT's face embeddings demonstrate greater robustness to variations in distance in the context of face recognition tasks.

The results obtained for the face verification task in the ROF dataset are presented in Fig. 4.23. In this figure, we present a histogram illustrating the dataset's distribution, revealing the number of images each subject in the dataset has for each category, namely: neutral, masked, and sunglasses. The bottom figures delve into the outcomes obtained when comparing neutral images of every subject against images with masks, sunglasses, and both categories combined, respectively. These visualizations distinctly showcase ViT's superior performance in handling occlusions compared to CNNs. ViT excels in deriving global face embeddings, whereas CNNs

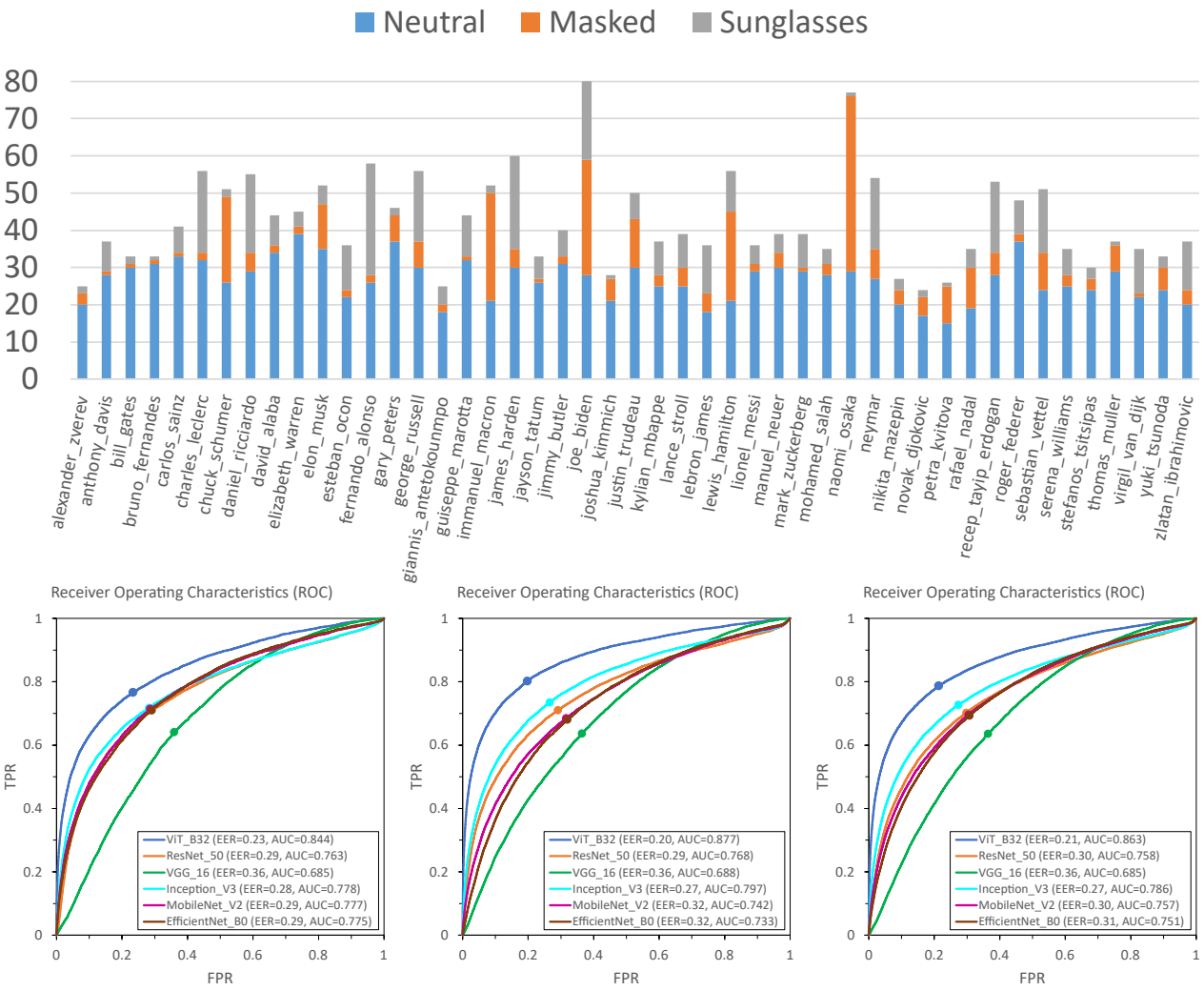
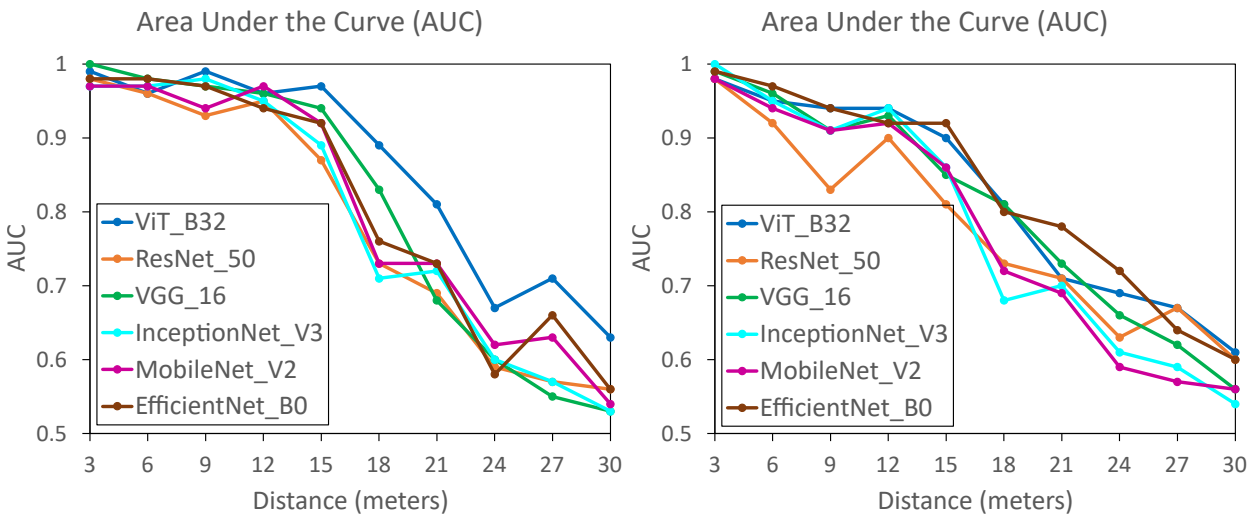


Figure 4.23: Histogram displaying the distribution of the ROF dataset (top), as well as the ROC curves obtained for the categories of mask (bottom left), sunglasses (bottom center), and both combined (bottom right).

demonstrate proficiency in extracting local representations. This is likely the reason why ViT presents itself as a more robust approach against occlusion in face recognition tasks, as it is less affected by local occlusions, whether they occur in the lower or upper facial regions.

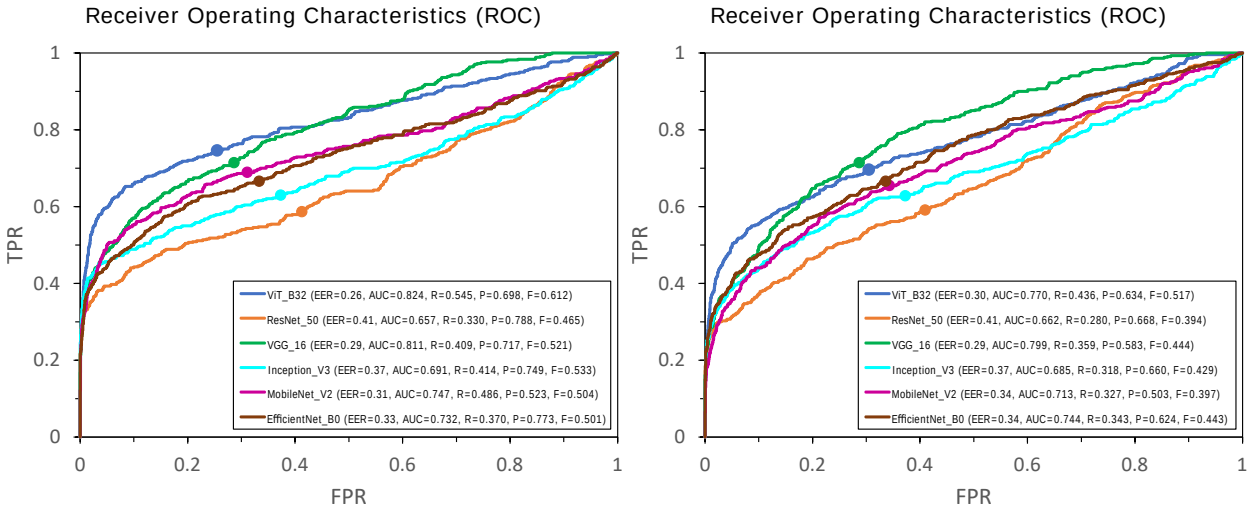
The results obtained for the face verification task in the UPM-GTI-Face dataset are presented in Fig. 4.24. This figure provides a summary of the information contained in the ROC curves computed for every intermediate distance, using the AUC values. In the absence of masks, as depicted in Figure 4.24a, the performance of the six networks is similar for distances ranging from 3 to 12 meters, while for distances greater than 15 meters, the performance of all networks experiences a significant decline. Notably, for distances from 6 meters onwards, ViT consistently outperforms CNNs, as evidenced by its superior AUC scores. This suggests that ViT's face embeddings exhibit greater discrimination and resilience against the influence of distance. This distinction becomes particularly evident at a distance of 30 meters, where



(a) Without mask **(b)** With mask

Figure 4.24: AUC curves derived from the ROC curves obtained at every intermediate distance in the UPM-GTI-Face dataset for the unmasked (a) and masked (b) scenarios.

CNNs' AUC scores fluctuate around 0.5, indicating random behavior, while ViT maintains an AUC score of 0.63. In the scenario involving masks, portrayed in Figure 4.24b, all six networks perform similarly, with ViT continuing to rank among the top performers. The complexity inherent in real-world scenarios, a characteristic captured by this dataset, contributes to the differences observed in the results at different distances, which do not decrease linearly with distance. Similarly, the mask and unmasked scenarios do not follow the exact same patterns. To gain a comprehensive understanding of both masked and unmasked scenarios across various distances, we conducted an additional experiment in which we compared all gallery images with all probe images, combining all distances. These results are illustrated in Figure 4.25, offering a more generalized perspective on this dataset. For the scenario without masks, results align with our earlier observations, with ViT consistently emerging as the top-performing model. Regarding the scenario with masks, ViT maintains a strong performance but ranks second to VGG, which surpasses it by a small margin, boasting a 4% increase in AUC. This occurrence is an exception; VGG has learned to detect a specific set of features that, in general, perform less effectively than ViT. However, for this particular scenario involving a small dataset, very small and occluded images due to masks, it performs better. This incident is not replicated across the other datasets used for evaluation, indicating it as an isolated case that seems to be a non-reproducible anomaly. VGG's success in this specific context can be attributed to its ability to leverage certain characteristics that may be advantageous for these particular conditions. It's noteworthy that ViT, in its overall performance across multiple datasets, consistently exhibits superior discrimination and resilience, as observed in our broader evaluation. This isolated instance highlights the complexity of real-world scenarios and the potential influence of dataset characteristics on individual model performance, emphasizing the need for robust and diverse evaluations.



(a) Without mask **(b)** With mask

Figure 4.25: ROC curves obtained from combining all intermediate distances (from 3 to 30 meters) in the UPM-GTI-Face dataset for the unmasked (a) and masked (b) scenarios.

4.7 Personalized Video Summarization

This section presents the experimental evaluation of the Personalized Video Summarization (PVS) pipeline introduced in Section 3.7. Building on the components developed throughout the thesis—classical, motion-based highlight detection [84], text-guided video summarization with SportCLIP [86], and the two face recognition works UPM-GTI-Face [87] and our comparative study of CNN- and Transformer-based backbones [85]—we now quantify how well the full system can generate identity-aware highlight sets for real broadcast sports videos, using the identity-aware event-level metrics introduced in Section 4.2.1.

Our evaluation is conducted on the Olympic Highlights dataset introduced in Section 4.1.1.3, which consists of twenty broadcast videos with frame-level highlight labels and identity annotations specifying which athlete is responsible for each attempt. On top of this, we explore a grid of PVS configurations, described in detail in Section 3.7, combining two video summarization backbones (SportCLIP [86] and QD-DETR [72]), two face recognition models (ArcFace [17] and TransFace [14]), two target representations (original vs. updated target images), and two clip-assignment strategies (sequential vs. instant). This grid yields sixteen distinct PVS configurations, all of which are evaluated and discussed in the remainder of this section.

A key design choice in this section is the inclusion of QD-DETR [72] as a reference summarization backbone. In earlier experiments on the MATDAT and SportCLIP datasets (Section 4.4, Table 4.10), we already observed that QD-DETR, when used with its original pre-trained weights, performed poorly compared to SportCLIP in the sports domain. The same behavior was qualitatively observed on Olympic Highlights: the off-the-shelf model frequently missed attempts or over-reacted to replays and crowd shots, making it an unreliable baseline for PVS. To obtain a meaningful comparison point, and given the lack of any other dataset with fine-grained annotations of both highlights and athlete identities, we therefore fine-tuned QD-

DETR directly on the Olympic Highlights videos. As a result, the PVS configurations built on QD-DETR should be interpreted as optimistic, upper-bound baselines: they benefit from training on the same distribution as the test videos, whereas SportCLIP-based configurations do not. Despite this advantage, we will see that PVS variants driven by SportCLIP achieve performance that is competitive with QD-DETR, but without resorting to any task-specific training.

To disentangle the contributions of the different components, we complement the end-to-end PVS experiments with two ablation studies. First, we evaluate SportCLIP and QD-DETR purely as video summarizers, ignoring identity, by comparing their predicted highlight segments directly against the ground-truth annotations. Second, we evaluate the face recognition branch in isolation by assuming that the highlight segments are perfect (i.e., using ground-truth events as input) and measuring how accurately ArcFace and TransFace assign each segment to the correct athlete. Together, these analyses clarify how much of the final PVS performance is attributable to the summarization model versus the identity backbone and the choice of target representation.

The remainder of this section is organized as follows. Section 4.7.1 describes the evaluation protocol and the configuration grid used in the experiments. Section 4.7.2 presents the end-to-end PVS results across all sixteen configurations, highlighting the impact of the summarization backbone, face recognition model, and clip-assignment strategy. Section 4.7.3 reports the ablation studies on video summarization and face recognition, providing independent evidence for the strengths and limitations of each component.

4.7.1 Experimental setup

4.7.1.1 Data and annotations

We evaluate PVS on the Olympic Highlights dataset introduced in Section 4.1.1.3. As described there, the dataset comprises twenty broadcast videos across four Olympic sports, with frame-level labels distinguishing highlight (HL), non-highlight (NHL), and uncertainty (UN) regions. Each event is additionally annotated with the identity of the athlete responsible for the attempt. The resulting per-athlete event sets serve as the ground-truth identity-aware summaries.

Given a full video V and the set of athletes $\mathcal{P} = \{p_1, \ldots, p_M\}$ appearing in it, the PVS system (Section 3.7) produces, for each target p_m, a set of personalized highlight segments $\mathcal{H}^{\star}_{p_m}(V, q)$, where q denotes the text query or prompt set specifying the type of event of interest. To evaluate the system, we compare these predicted segments against the ground-truth highlights annotated for each athlete using an event-level matching protocol based on temporal intersection-over-union (IoU). For a given athlete, both predicted segments and ground-truth highlights are treated as temporal intervals on the video timeline, and the IoU between a prediction and a ground-truth event is defined as the duration of their intersection divided by the duration of their union (in frames). A predicted segment is counted as a true positive if there exists a ground-truth highlight for the *same athlete* with which it attains a temporal IoU of at least 0.3, and that ground-truth event has not already been matched to another prediction.

We adopt an IoU threshold of 0.3 to be tolerant to boundary ambiguity in highlight annotations and to the inherent imprecision of predicted start/end times. Empirically, we observed that predictions with IoU around 0.3 often still cover the core portion of the highlight (typically missing only a small prefix or suffix), which is the most relevant part in practice. A stricter threshold would over-penalize near-miss boundaries and increase sensitivity to subjective annotation choices. Moreover, small temporal misalignments can be compensated in downstream post-processing (e.g., extending predicted segments slightly), whereas missing the highlight entirely cannot. Predicted segments that do not reach IoU ≥ 0.3 with any ground-truth event are counted as false positives, and ground-truth events that remain unmatched after this one-to-one assignment are counted as false negatives.

From these counts of true positives (TP), false positives (FP), and false negatives (FN), we compute the identity-aware event-level recall, precision, and F-score described in Section 4.2.1. In this setting, a prediction is only considered correct when both the temporal overlap and the athlete identity match. Metrics are first computed at the video level, aggregating over all athletes in that video, and then averaged per sport and over the full set of twenty videos.

4.7.1.2 Configuration grid

To systematically probe how the main design choices of PVS affect performance, we evaluate a grid of configurations obtained by varying four key factors: the video summarization backbone, the face recognition model, the target representation, and the clip-assignment strategy.

- **Video summarization model.** The summarization stream uses either SportCLIP [86], operating in a zero-shot fashion with the CLIP-based scoring and filtering pipeline described in Section 3.3, or QD-DETR [72], a query-dependent Transformer-based summarizer fine-tuned on the Olympic Highlights videos as detailed in Section 4.7.3.1.
- **Face recognition backbone.** The face-analysis stream employs either ArcFace [17], representing convolutional architectures, or TransFace [14], representing Transformer-based architectures, following the comparative study in Section 3.6.
- **Target representation.** For each backbone, we consider two ways of representing each target athlete: an *original target*, obtained directly from a single user-provided reference image, and an *updated target*, obtained by replacing the original embedding with the highest-scoring in-video detection, as described in the optional target-update step in Section 3.7.2.
- **Clip-assignment method.** Finally, we evaluate both assignment strategies introduced in Section 3.7.4: *Method 1 (Sequential)*, which propagates identity decisions forward in time based on the most recent athlete whose resemblance curve crossed the similarity threshold, and *Method 2 (Instant)*, which evaluates each highlight segment independently within an expanded temporal window around its extent.

The total number of distinct PVS configurations is therefore

$$N_{\text{conf}} = 2 \text{ (summarizers)} \times 2 \text{ (face models)} \times 2 \text{ (targets)} \times 2 \text{ (assignment methods)} = 16. \quad (4.5)$$

For each sport, we report the average recall, precision, and F-score across all five videos and all

athletes, resulting in four tables with sixteen rows each (one per configuration). To facilitate cross-sport comparison, we additionally summarize the results in a bar-plot figure where each bar corresponds to one of the sixty-four (4 sports $\times$ 16 configurations) sport-specific scores, and a second set of aggregated bars overlays, for each configuration, its mean performance across all sports.

4.7.1.3 Training and evaluation of QD-DETR

Unlike SportCLIP, which is applied in a fully zero-shot manner, QD-DETR requires supervised training. In preliminary experiments, we observed that the off-the-shelf model, trained on generic web video datasets, struggled with the specific broadcast style and event structure of Olympic Highlights: it frequently missed key attempts or over-reacted to replays and crowd shots, making it an unreliable baseline for PVS. Combined with the poor off-the-shelf results previously reported on related sports data (Section 4.4), this motivated us to adapt QD-DETR to our setting.

Given the lack of any other dataset with fine-grained annotations of both highlights and athlete identities, the only viable option was to fine-tune QD-DETR directly on the Olympic Highlights videos, using the highlight intervals described in Section 4.1.1.3 as supervision. In this sense, QD-DETR plays the role of an "oracle" summarizer in the context of this thesis: its training and test distributions coincide, and the model effectively has access, during training, to the same type of temporal supervision later used to evaluate it.

In contrast, SportCLIP is never fine-tuned on Olympic Highlights and operates purely via CLIP's pre-trained vision–language representations, whereas QD-DETR benefits from domain-specific training on the same videos used for evaluation. Within this asymmetric setting, the comparison serves a dual purpose. First, it shows that zero-shot SportCLIP remains highly competitive even against a state-of-the-art summarizer trained directly on the target data, underscoring the strength of its zero-shot capabilities in this domain. Second, the integration of QD-DETR highlights the modular design of PVS: new video summarization backbones—whether zero-shot or fully supervised—can be plugged into the pipeline with minimal changes, providing a flexible framework for future extensions.

4.7.2 End-to-end PVS performance

Tables 4.16–4.19 summarize the end-to-end PVS results for high jump, javelin, long jump, and pole vault, respectively. Each table reports event-level recall, precision, and F-score for the sixteen configurations obtained by combining the two summarizers, two face recognition backbones, two target representations, and two clip-assignment strategies. Figure 4.26 aggregates these results into a compact visualization: colored bars show the per-sport F-scores for each configuration, while an overlaid set of grey bars indicates the overall F-score across the four sports.

Several consistent trends emerge across sports and configurations.

Effect of the video summarization model. As anticipated in Section 4.7.1, configurations built on QD-DETR tend to occupy the upper end of the F-score range in Figure 4.26. This

Table 4.16: Event-level PVS performance on the high jump subset of the Olympic Highlights dataset. Each row corresponds to one of the 16 configurations obtained by combining video summarizer, face recognition backbone, clip-assignment strategy, and target representation.

Sport	VS model	FR model	Method	Target	Recall (%)	Precision (%)	F-score (%)
High jump	SportCLIP	ArcFace	SEQ	Orig	56.9	41.3	47.8
	SportCLIP	ArcFace	SEQ	Upd	50.0	35.8	41.7
	SportCLIP	ArcFace	INS	Orig	51.1	46.5	48.7
	SportCLIP	ArcFace	INS	Upd	55.0	40.3	46.5
	SportCLIP	TransFace	SEQ	Orig	84.4	61.7	71.3
	SportCLIP	TransFace	SEQ	Upd	89.3	64.6	75.0
	SportCLIP	TransFace	INS	Orig	64.5	60.1	62.2
	SportCLIP	TransFace	INS	Upd	72.9	60.1	65.9
	QD-DETR	ArcFace	SEQ	Orig	63.0	53.1	57.6
	QD-DETR	ArcFace	SEQ	Upd	54.2	45.1	49.2
	QD-DETR	ArcFace	INS	Orig	54.2	56.8	55.5
	QD-DETR	ArcFace	INS	Upd	59.2	50.5	54.5
	QD-DETR	TransFace	SEQ	Orig	88.2	75.5	81.3
	QD-DETR	TransFace	SEQ	Upd	92.0	77.0	83.8
	QD-DETR	TransFace	INS	Orig	69.5	75.5	72.4
	QD-DETR	TransFace	INS	Upd	79.4	74.8	77.0

Table 4.17: Event-level PVS performance on the javelin subset of the Olympic Highlights dataset. Each row corresponds to one of the 16 configurations obtained by combining video summarizer, face recognition backbone, clip-assignment strategy, and target representation.

Sport	VS model	FR model	Method	Target	Recall (%)	Precision (%)	F-score (%)
Javelin	SportCLIP	ArcFace	SEQ	Orig	56.7	32.6	41.4
	SportCLIP	ArcFace	SEQ	Upd	50.6	29.1	36.9
	SportCLIP	ArcFace	INS	Orig	48.9	40.2	44.1
	SportCLIP	ArcFace	INS	Upd	53.9	35.0	42.5
	SportCLIP	TransFace	SEQ	Orig	76.7	44.1	56.0
	SportCLIP	TransFace	SEQ	Upd	85.0	48.9	62.1
	SportCLIP	TransFace	INS	Orig	68.3	53.2	59.9
	SportCLIP	TransFace	INS	Upd	77.8	53.0	63.1
	QD-DETR	ArcFace	SEQ	Orig	58.3	39.2	46.9
	QD-DETR	ArcFace	SEQ	Upd	51.7	34.7	41.5
	QD-DETR	ArcFace	INS	Orig	52.2	52.2	52.2
	QD-DETR	ArcFace	INS	Upd	60.6	45.8	52.2
	QD-DETR	TransFace	SEQ	Orig	90.0	60.7	72.5
	QD-DETR	TransFace	SEQ	Upd	96.7	65.2	77.9
	QD-DETR	TransFace	INS	Orig	74.4	70.9	72.6
	QD-DETR	TransFace	INS	Upd	88.3	71.3	78.9

Table 4.18: Event-level PVS performance on the long jump subset of the Olympic Highlights dataset. Each row corresponds to one of the 16 configurations obtained by combining video summarizer, face recognition backbone, clip-assignment strategy, and target representation.

Sport	VS model	FR model	Method	Target	Recall (%)	Precision (%)	F-score (%)
Long jump	SportCLIP	ArcFace	SEQ	Orig	54.2	39.1	45.4
	SportCLIP	ArcFace	SEQ	Upd	53.1	38.3	44.5
	SportCLIP	ArcFace	INS	Orig	35.4	27.7	31.1
	SportCLIP	ArcFace	INS	Upd	39.1	29.2	33.4
	SportCLIP	TransFace	SEQ	Orig	84.1	61.8	71.3
	SportCLIP	TransFace	SEQ	Upd	91.5	66.5	77.0
	SportCLIP	TransFace	INS	Orig	69.4	71.2	70.3
	SportCLIP	TransFace	INS	Upd	81.9	70.0	75.5
	QD-DETR	ArcFace	SEQ	Orig	52.4	42.1	46.7
	QD-DETR	ArcFace	SEQ	Upd	52.4	42.1	46.7
	QD-DETR	ArcFace	INS	Orig	35.4	31.2	33.2
	QD-DETR	ArcFace	INS	Upd	39.9	33.3	36.3
	QD-DETR	TransFace	SEQ	Orig	82.7	67.9	74.5
	QD-DETR	TransFace	SEQ	Upd	88.6	71.2	78.9
	QD-DETR	TransFace	INS	Orig	70.8	76.5	73.6
	QD-DETR	TransFace	INS	Upd	84.1	74.3	78.9

Table 4.19: Event-level PVS performance on the pole vault subset of the Olympic Highlights dataset. Each row corresponds to one of the 16 configurations obtained by combining video summarizer, face recognition backbone, clip-assignment strategy, and target representation.

Sport	VS model	FR model	Method	Target	Recall (%)	Precision (%)	F-score (%)
Pole vault	SportCLIP	ArcFace	SEQ	Orig	47.6	26.6	34.1
	SportCLIP	ArcFace	SEQ	Upd	48.9	27.3	35.0
	SportCLIP	ArcFace	INS	Orig	44.9	39.3	41.9
	SportCLIP	ArcFace	INS	Upd	57.3	36.2	44.4
	SportCLIP	TransFace	SEQ	Orig	77.3	43.7	55.9
	SportCLIP	TransFace	SEQ	Upd	82.7	46.7	59.7
	SportCLIP	TransFace	INS	Orig	64.9	50.3	56.7
	SportCLIP	TransFace	INS	Upd	72.4	50.3	59.4
	QD-DETR	ArcFace	SEQ	Orig	52.0	32.5	40.0
	QD-DETR	ArcFace	SEQ	Upd	54.7	33.9	41.8
	QD-DETR	ArcFace	INS	Orig	45.8	49.5	47.6
	QD-DETR	ArcFace	INS	Upd	57.3	42.3	48.7
	QD-DETR	TransFace	SEQ	Orig	79.1	49.6	61.0
	QD-DETR	TransFace	SEQ	Upd	86.7	54.2	66.7
	QD-DETR	TransFace	INS	Orig	66.2	61.1	63.5
	QD-DETR	TransFace	INS	Upd	75.6	61.2	67.6

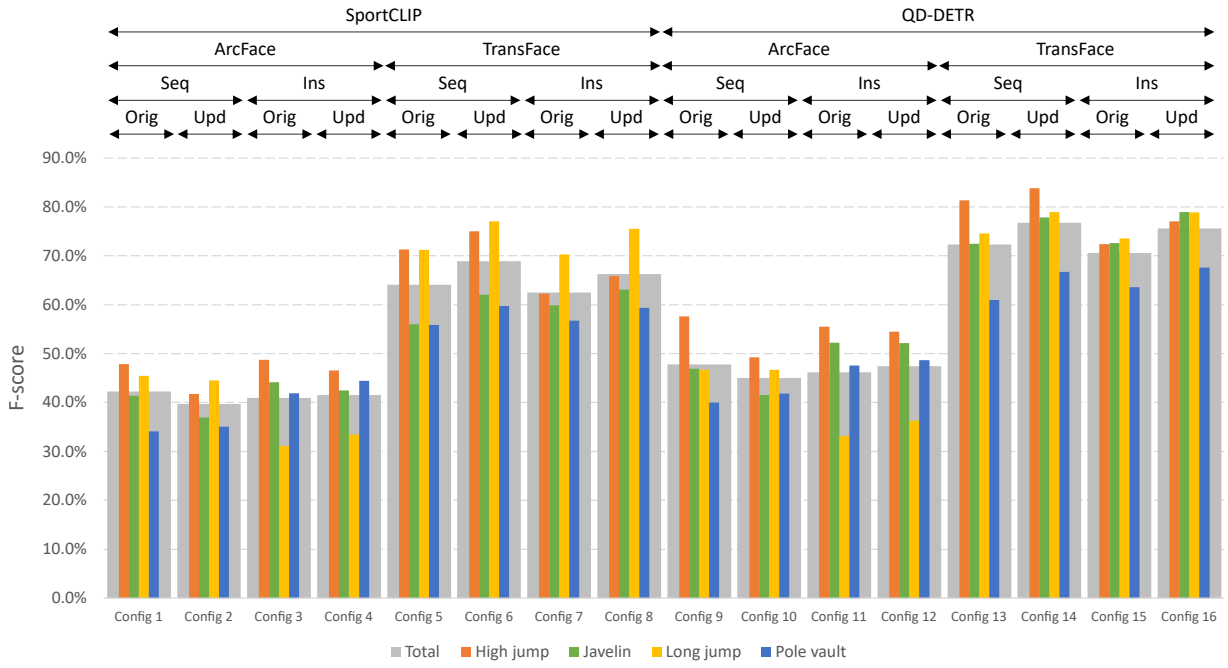


Figure 4.26: Aggregated event-level F-scores for all PVS configurations on the Olympic Highlights dataset. Thin bars correspond to the per-sport F-scores of the sixteen configurations across the four sports (high jump, javelin, long jump, and pole vault), yielding a total of 64 values. Thicker overlaid grey bars show, for each configuration, the total F-score across all four sports. Configurations are grouped by video summarizer (SportCLIP vs. QD-DETR), face recognition backbone (ArcFace vs. TransFace), assignment method (sequential vs. instant), and target representation (original vs. updated), making it possible to compare both per-sport behavior and overall trends at a glance.

behavior is expected: QD-DETR is fine-tuned directly on the Olympic Highlights videos and thus acts as an oracle summarizer, trained and evaluated on the same distribution. What is more remarkable is how close the zero-shot SportCLIP configurations come to this oracle baseline. Across sports and identity settings, SportCLIP-based PVS variants consistently trail QD-DETR by only a modest margin, despite never seeing Olympic Highlights during training. This confirms that SportCLIP offers strong zero-shot capabilities for sports highlight detection and that, within the PVS framework, it constitutes a highly competitive alternative to a fully supervised, domain-specific transformer.

Effect of the face recognition backbone. Across all summarization and clip-assignment settings, the choice of face recognition backbone has a clear and systematic effect on performance. For any fixed combination of video summarizer, target representation, and clip-assignment strategy, configurations that use TransFace consistently achieve higher F-scores than those based on ArcFace. This behavior is fully aligned with the conclusions of our backbone comparison in Section 4.6, where Transformer-based models were shown to produce more robust and discriminative face embeddings than CNNs across multiple datasets and viewing conditions. In the context of PVS, this translates into TransFace-based variants occupying the upper tier of configurations within each summarization family, while ArcFace-based variants provide a natural lower bound.

Original vs. updated targets. The effect of replacing the user-provided reference image with an in-video updated target depends strongly on the quality of the underlying identity signal. When TransFace is used as the backbone, updated targets consistently lead to higher F-scores across all sports and configurations: adapting the embedding to a broadcast frame reduces domain shift with respect to the test clips and sharpens resemblance curves. For ArcFace, the picture is more nuanced. In sports where faces are often small, oblique, or partially occluded, the in-video crops used for updating can be noisier than the original gallery photo, and performance sometimes degrades slightly when the original embedding is replaced. In contrast, in settings with more reliable detections, the updated target still provides modest gains even for ArcFace. Overall, these trends suggest a practical guideline: target updating is most beneficial when a strong Transformer-based backbone is available and the broadcast offers clean in-video views of the athletes, whereas retaining the original reference image can be safer when identity evidence in the video is scarce or of lower quality.

Sequential vs. instant assignment. The two clip-assignment strategies offer complementary trade-offs. Across most sports, sequential assignment tends to achieve higher recall by propagating identity decisions across neighboring segments, which helps recover attempts where the athlete is briefly hard to detect. However, because it propagates the last confidently recognized identity forward in time, this method is only meaningful when the gallery contains representative images for all athletes of interest; otherwise, highlights tend to be biased toward the subset of registered identities. Instant assignment, by evaluating each segment independently within an expanded temporal window and relying only on local evidence, does not impose this requirement and is more conservative, typically yielding higher precision. In the aggregated bar plot, this appears as pairs of configurations that achieve similar overall

F-scores but with different recall–precision balances, giving practitioners a degree of control over whether missing events or including extra segments is more acceptable.

Differences across sports. The relative ranking of configurations is broadly stable across sports, but the absolute difficulty varies. Long jump and high jump generally yield higher F-scores, consistent with their regular competition structure and frequent close-up shots of the athletes, which provide abundant identity evidence. Javelin and pole vault involve wider views, faster camera switches, and more cutaways, which make both highlight detection and face recognition more challenging and compress the spread of F-scores across configurations. Even in these harder settings, however, the best PVS configurations maintain strong performance, indicating that the pipeline generalizes well across different field events.

Overall, the combination of a fine-tuned QD-DETR summarizer, TransFace backbone, sequential assignment, and updated targets yields the strongest end-to-end PVS performance across all four sports. Equally important, the best zero-shot configuration based on SportCLIP—using the same recognition backbone, assignment strategy, and target setup—remains very close to this oracle baseline, trailing it by only a modest margin in F-score. Despite never being trained on Olympic Highlights, SportCLIP thus delivers personalized summaries that are competitive with those of a state-of-the-art transformer trained directly on the evaluation videos. Taken together, these findings indicate that zero-shot text-guided summarization, when combined with a strong identity branch and appropriate clip-assignment mechanisms, can serve as a robust and practically attractive alternative to fully supervised, domain-specific models for personalized highlight generation.

4.7.3 Ablation studies

To better understand the contribution of each component to the final PVS performance, we now analyze the video summarization and face recognition branches in isolation. In the first ablation, we ignore identity entirely and evaluate SportCLIP and QD-DETR solely as highlight detectors. In the second ablation, we assume perfect highlight segmentation and evaluate how accurately ArcFace and TransFace assign those segments to the correct athlete under the two assignment methods and target configurations.

4.7.3.1 Video summarization stream

In this experiment, we isolate the summarization branch and evaluate SportCLIP and QD-DETR purely as highlight detectors, ignoring identity altogether. Unlike the IoU-based event-level protocol used for the personalized PVS evaluation in Section 4.7.1, here we adopt a *frame-level* evaluation scheme so that the results are directly comparable to those reported in Sections 4.3 and 4.4. Concretely, both summarizers are run exactly as described at the end of Section 3.3: saliency scores are first computed at frame level and then converted into binary highlight predictions through the post-processing pipeline introduced in Section 3.2. Frame-level recall, precision, and F-score are computed per video from the resulting TP/FP/FN counts, then averaged per sport and over all twenty videos.

Table 4.20 summarizes the resulting performance. As expected for a zero-shot method,

Table 4.20: Frame-level video summarization performance on the Olympic Highlights dataset when identity is ignored. For each model and sport, we report the mean recall, precision, and F-score across the five videos of that sport, computed from frame-level TP/FP/FN. The last row for each model shows the overall averages across all twenty videos.

Model	Sport	Recall (%)	Precision (%)	F-score (%)
SportCLIP	High jump	93.43	58.85	72.10
	Javelin	79.33	58.39	67.20
	Long jump	81.24	77.23	78.88
	Pole vault	85.86	63.05	72.62
	All sports	**84.97**	**64.38**	**72.70**
QD-DETR	High jump	95.49	75.12	83.93
	Javelin	92.41	81.21	86.42
	Long jump	92.23	87.67	89.71
	Pole vault	90.33	77.89	83.52
	All sports	**92.62**	**80.47**	**85.89**

SportCLIP tends to trade precision for recall: across all four sports, it recovers a large fraction of the ground-truth highlight frames but it also activates on semantically related yet non-highlight content (e.g., extended run-ups, slow replays, or generic build-up shots), which inflates the number of false positives and lowers precision. Even so, the overall frame-level F-score remains strong considering that SportCLIP is never trained on Olympic Highlights and relies solely on CLIP's generic vision–language representations.

QD-DETR, in contrast, is fine-tuned directly on the Olympic Highlights videos using the ground-truth highlight intervals as supervision, and this domain adaptation translates into a clear advantage under the same frame-level protocol. Across sports, QD-DETR achieves higher recall and substantially higher precision than SportCLIP, leading to consistently stronger F-scores. In this sense, QD-DETR should be viewed as an oracle summarizer: it is trained and evaluated on the very same set of broadcast videos and thus represents an optimistic upper bound on what a state-of-the-art Transformer-based model can achieve in this domain. This stands in sharp contrast to the comparative analysis in Subsection 4.4.2, where the off-the-shelf QD-DETR model performed poorly on related sports data. Once adapted to the specific broadcast style and event structure of Olympic Highlights, however, its frame-level decisions become much more reliable, making it a useful reference point against which to gauge the zero-shot performance of SportCLIP.

Taken together, the ablation paints a complementary picture of the two summarizers. Once fine-tuned on Olympic Highlights, QD-DETR delivers the strongest frame-level scores and can be regarded as an optimistic upper bound for a fully supervised, domain-trained approach. SportCLIP, in turn, attains solid F-scores across all four sports despite operating in a strictly zero-shot regime, showing that CLIP-based text guidance alone already captures much of the underlying highlight structure. In practice, this positions SportCLIP as an attractive option when annotated highlights are scarce or when portability across domains is a priority, while QD-DETR provides a useful reference point for the gains that become attainable once rich supervision on the target domain is available.

4.7.3.2 Face analysis stream

The second ablation isolates the face-analysis branch by assuming perfect knowledge of which temporal segments are highlights. Instead of using segments proposed by a summarizer, we take the ground-truth highlight events as the candidate clips and apply the face detection, embedding, resemblance computation, and clip-assignment stages exactly as in the full PVS pipeline. This yields a setting in which any errors can be attributed solely to the face recognition backbone, the target representation, or the assignment strategy. Tables 4.21 and 4.22 summarize the resulting event-level recall, precision, and F-score, aggregated across all videos and broken down by sport, respectively.

First, the choice of backbone has a dominant and consistent effect on performance. Across all rows of Table 4.21, TransFace yields substantially higher F-scores than ArcFace under every combination of assignment strategy and target configuration, with improvements that amount to several tens of percentage points. The per-sport breakdown in Table 4.22 shows that this gap is stable across high jump, javelin, long jump, and pole vault: for each sport, TransFace configurations occupy the upper part of the performance range, while the best ArcFace variants form a lower tier. This mirrors the conclusions of Section 4.6, confirming that Transformer-based embeddings provide a markedly stronger identity signal than CNN-based descriptors under broadcast-like conditions.

Second, the sequential and instant assignment strategies express a clear recall–precision trade-off that is largely consistent across backbones and sports. Comparing SEQ and INS rows within each block in Table 4.21, sequential assignment tends to favor higher recall (often leading to the best overall F-scores for TransFace), as identity decisions are propagated across temporally adjacent highlight segments. Instant assignment, by contrast, is more conservative: for a given backbone and target setup it typically increases precision at the cost of reduced recall, reflecting the fact that each segment is treated more independently. Table 4.22 shows that the same pattern appears in all four events, making the assignment strategy a natural knob for trading off missed athlete-specific highlights against the risk of incorrect attributions.

Third, the effect of target updating depends strongly on the underlying backbone and assignment scheme. For TransFace, comparing original and updated rows in Tables 4.21 and 4.22 reveals a consistent gain: replacing the user-provided reference image with the most confident in-video detection improves recall and precision under both SEQ and INS, across all sports. This indicates that adapting the embedding to a broadcast frame is an effective way to reduce domain shift and sharpen resemblance curves when the backbone is already strong. For ArcFace, the picture is more nuanced: updated targets often degrade performance under sequential assignment, while they provide modest benefits under instant assignment. This suggests that, for a weaker and more brittle backbone, noisy in-video crops can be problematic when identity decisions are propagated over time, but still helpful when assignments are made on a clip-by-clip basis.

Finally, the per-sport averages in Table 4.22 highlight how the difficulty of identity assignment varies across events. High jump and javelin consistently exhibit the strongest scores, particularly for TransFace, reflecting the relatively favorable viewing conditions (frequent close-ups and less cluttered scenes). Long jump and pole vault are more challenging: ArcFace

Table 4.21: Event-level face recognition performance on the Olympic Highlights dataset when ground-truth highlight segments are provided as input. For each combination of backbone (ArcFace vs. TransFace), assignment method (sequential vs. instant), and target representation (original vs. updated), we report the mean recall, precision, and F-score across all twenty videos and all athletes; differences therefore reflect only the behaviour of the face-analysis stream.

FR model	Method	Target	Recall (%)	Precision (%)	F-score (%)
ArcFace	SEQ	Original	56.69	56.99	56.84
ArcFace	SEQ	Updated	52.63	52.83	52.73
ArcFace	INS	Original	47.11	63.61	53.25
ArcFace	INS	Updated	54.06	59.69	56.54
TransFace	SEQ	Original	84.43	84.94	84.68
TransFace	SEQ	Updated	89.74	89.81	89.78
TransFace	INS	Original	70.89	90.33	78.97
TransFace	INS	Updated	82.02	91.69	86.46

performance drops noticeably, especially under instant assignment, and even TransFace shows slightly lower F-scores than in the easier sports, likely due to wider shots, more occlusions, and more frequent camera changes. Importantly, however, the relative ranking of configurations remains stable across sports: TransFace with updated targets stays at the top, followed by TransFace with original targets and then the ArcFace variants, indicating that the design choices validated on the aggregate metrics carry over robustly to each individual field event.

4.7.4 Computational cost

Beyond accuracy, the practicality of a personalized video summarization system depends critically on its computational footprint. In this subsection we characterize the cost of the proposed PVS pipeline, distinguishing between the main branches (video summarization and face analysis) and a set of shared components. All experiments were conducted on the same workstation used in Section 4.4.5, running Ubuntu 24.04 LTS on an Intel Core i9-13900K CPU with 128 GB RAM and a single NVIDIA GeForce RTX 4090 GPU. Unless otherwise stated, videos were processed at 30 fps and 1920×1080 resolution, and the reported runtimes are derived from these per-frame and per-face measurements for the corresponding video durations.

Table 4.23 summarizes the measured cost of the main components of the pipeline. For each module we report the average processing time normalized by its natural unit of work. The lower part of the table aggregates these figures into end-to-end PVS runtimes for four representative configurations (SportCLIP+ArcFace, SportCLIP+TransFace, QD-DETR+ArcFace, and QD-DETR+TransFace) and for different video lengths, under a straightforward but deliberately unoptimized evaluation regime: all frames are processed, and all stages are executed sequentially.

Several conclusions can be drawn from Table 4.23. First, the two video summarizers differ markedly in speed: SportCLIP requires about 8.2 ms per frame, whereas QD-DETR is substantially faster at 1.8 ms per frame (roughly a 4–5× speed-up). Nonetheless, both are dominated by the cost of the face-analysis branch. Face detection alone takes roughly 84 ms per

Table 4.22: Event-level face recognition performance on the Olympic Highlights dataset when ground-truth highlight segments are provided as input, broken down by sport. For each sport, backbone (ArcFace vs. TransFace), assignment method (sequential vs. instant), and target representation (original vs. updated), we report the mean recall, precision, and F-score across the five videos of that sport and all athletes, isolating the contribution of the face-analysis stream.

Sport	FR model	Method	Target	Recall (%)	Precision (%)	F-score (%)
High jump	ArcFace	SEQ	Orig	68.18	68.22	68.19
	ArcFace	SEQ	Upd	54.58	54.58	54.58
	ArcFace	INS	Orig	48.24	72.38	57.81
	ArcFace	INS	Upd	50.56	69.58	58.75
	TransFace	SEQ	Orig	92.87	93.39	93.13
	TransFace	SEQ	Upd	93.77	93.77	93.77
	TransFace	INS	Orig	78.26	92.56	84.58
	TransFace	INS	Upd	88.41	93.82	90.99
Javelin	ArcFace	SEQ	Orig	55.67	55.67	55.67
	ArcFace	SEQ	Upd	42.39	42.39	42.39
	ArcFace	INS	Orig	53.39	72.11	61.96
	ArcFace	INS	Upd	60.09	73.16	66.92
	TransFace	SEQ	Orig	87.09	87.09	87.09
	TransFace	SEQ	Upd	93.09	93.09	93.09
	TransFace	INS	Orig	75.98	96.78	85.01
	TransFace	INS	Upd	86.79	97.48	91.79
Long jump	ArcFace	SEQ	Orig	50.35	50.35	50.35
	ArcFace	SEQ	Upd	49.47	49.47	49.47
	ArcFace	INS	Orig	25.77	28.34	26.98
	ArcFace	INS	Upd	30.59	33.71	32.07
	TransFace	SEQ	Orig	83.28	83.86	83.53
	TransFace	SEQ	Upd	89.85	89.85	89.85
	TransFace	INS	Orig	71.93	92.12	80.84
	TransFace	INS	Upd	84.28	93.91	88.76
Pole vault	ArcFace	SEQ	Orig	48.94	49.85	49.38
	ArcFace	SEQ	Upd	50.68	51.46	51.06
	ArcFace	INS	Orig	41.47	67.10	49.56
	ArcFace	INS	Upd	56.51	63.95	59.69
	TransFace	SEQ	Orig	78.16	78.75	78.45
	TransFace	SEQ	Upd	85.51	85.51	85.51
	TransFace	INS	Orig	65.82	82.79	73.00
	TransFace	INS	Upd	74.25	85.42	79.32

Table 4.23: Computational cost of the main components of the PVS pipeline. Per-component times are averaged over Olympic Highlights videos and normalized by their unit of work. Total runtimes assume dense processing of all frames at 30 fps and fully sequential execution of all stages.

Component	Unit of work	Time
Video summarization		
SportCLIP	per frame	8.22 ms
QD-DETR	per frame	1.79 ms
Face recognition		
ArcFace (face embeddings)	per detected face	35.95 ms
TransFace (face embeddings)	per detected face	10.96 ms
Shared components		
Face detection	per frame	83.89 ms
Resemblance-curve computation	per frame × athlete	≈ 0.0004 ms
Target update (optional)	per frame × athlete	≈ 0.0004 ms
Clip assignment and selection	per highlight segment	≈ 0.002 ms
Total PVS runtime (per 1-min video)		
SportCLIP + ArcFace	per 1-min video	20.96 min
SportCLIP + TransFace	per 1-min video	8.30 min
QD-DETR + ArcFace	per 1-min video	20.78 min
QD-DETR + TransFace	per 1-min video	8.12 min
Total PVS runtime (per 5-min video)		
SportCLIP + ArcFace	per 5-min video	104.80 min
SportCLIP + TransFace	per 5-min video	41.52 min
QD-DETR + ArcFace	per 5-min video	103.88 min
QD-DETR + TransFace	per 5-min video	40.60 min
Total PVS runtime (per 10-min video)		
SportCLIP + ArcFace	per 10-min video	209.60 min
SportCLIP + TransFace	per 10-min video	83.03 min
QD-DETR + ArcFace	per 10-min video	207.76 min
QD-DETR + TransFace	per 10-min video	81.19 min
Total PVS runtime (per 1-hour video)		
SportCLIP + ArcFace	per 1-hour video	20.96 h
SportCLIP + TransFace	per 1-hour video	8.30 h
QD-DETR + ArcFace	per 1-hour video	20.78 h
QD-DETR + TransFace	per 1-hour video	8.12 h

frame, and, on the Olympic Highlights videos, an average of 17 faces are processed per frame after applying a minimum size threshold to discard very small crops; this filtering reduces the number of faces to analyze and largely removes background spectators, while retaining the athletes and other prominent participants. Under these conditions, the embedding stage is by far the main bottleneck: about 611 ms per frame for ArcFace and 186 ms per frame for TransFace. In contrast, resemblance-curve computation, target update, and clip assignment are effectively free at the scale of the rest of the pipeline, with runtimes well below 0.01 ms per unit in all cases.

The lower block of Table 4.23 propagates these per-component costs to full videos. Because all stages are executed sequentially and all frames are processed, the resulting runtimes are relatively high and should be interpreted as upper bounds for a prototype implementation focused on modularity rather than efficiency. Even for the most efficient configuration (QD-DETR + TransFace), processing 1 minute of video takes approximately 8 minutes, while the combination SportCLIP + ArcFace requires about 21 minutes. Extrapolating linearly, a 1-hour broadcast would take between roughly 8 hours (QD-DETR + TransFace) and 21 hours (SportCLIP + ArcFace) to process. These figures reflect a deliberately straightforward design: video summarization and face analysis are run one after another, and all per-frame operations are evaluated at the full 30 fps.

Projected impact of simple optimizations. The structure of PVS, however, exposes several straightforward opportunities for acceleration. Two particularly simple ones are:

- **Parallel branches.** The video summarization and face-analysis streams are largely independent: the summarizer only needs the raw frames, while the face branch only requires the same frames plus the target identities. Running these two branches in parallel allows the GPU to process frames for summarization and faces for recognition concurrently, so that the end-to-end runtime is dominated by the slower of the two streams rather than by their sum.
- **Subsampled face analysis.** In broadcast sports footage, faces remain visible for many consecutive frames. It is therefore unnecessary to run face detection and embedding extraction at the full frame rate. Evaluating the face branch every kth frame (with simple interpolation between them) reduces the number of detection and embedding calls by a factor of k, with limited impact on the quality of the resemblance curves.

To quantify the potential benefit of these two modifications, we used the per-unit measurements of Table 4.23 to simulate an optimized scenario in which (i) video summarization and the face pipeline run fully in parallel, and (ii) face detection and embeddings are computed every 10th frame (i.e., at 3 fps instead of 30 fps). The resulting projected runtimes are reported in Table 4.24. These values are not the result of a new implementation, but rather an extrapolation from the measured costs under the assumptions above.

Under this simple configuration, the worst-case combination (SportCLIP + ArcFace) would process a 10-minute video in about 21 min, compared to approximately 210 min in the current sequential implementation, i.e., a speed-up of roughly one order of magnitude. The best configuration (QD-DETR + TransFace or SportCLIP + TransFace) would process the same

Table 4.24: Projected PVS runtime under a simple optimization scenario: video summarization and the face-analysis stream run in parallel, and face detection / embeddings are computed every 10th frame. Values are extrapolated from the per-frame and per-face measurements in Table 4.23.

Configuration	1-min video	5-min video	10-min video	1-hour video
SportCLIP + ArcFace	124.32 s	10.36 min	20.72 min	124.32 min
SportCLIP + TransFace	48.40 s	4.03 min	8.07 min	48.40 min
QD-DETR + ArcFace	124.32 s	10.36 min	20.72 min	124.32 min
QD-DETR + TransFace	48.40 s	4.03 min	8.07 min	48.40 min

10-minute sequence in about 8 min, which corresponds to roughly 0.8 minutes of processing per minute of video. For shorter clips the projected runtimes are correspondingly smaller: between 124 s and 48 s for a 1-minute video, depending on the chosen backbone combination.

Overall, these results show that the current runtimes primarily reflect an implementation geared towards clarity and functionality rather than aggressive optimization. The prototype evaluates all frames, runs the main branches sequentially, and uses off-the-shelf components without additional engineering for speed. The analysis above suggests that, once these aspects are revisited—for example, by subsampling the face-analysis stream, batching operations on the GPU, and running the video and face branches in parallel—the same PVS design can operate much closer to real time, while preserving its identity-aware capabilities and modularity for future extensions.

Chapter 5

Discussion

This chapter reflects on the experimental results presented in Chapter 4, connecting them to the methodological developments of Chapter 3 and to the broader objectives laid out in Chapter 1. The discussion is organized around the three main strands of the thesis: (i) automatic sports highlight detection, from classical motion-based methods to text-guided and Transformer-based models; (ii) robust face recognition under distance and occlusion stressors; and (iii) the integration of both strands into a modular Personalized Video Summarization (PVS) pipeline. For each strand, we highlight the strengths and limitations of the proposed approaches, how they complement or improve over existing work, and how they contribute to the overarching goal of generating identity-aware sports highlights.

5.1 Video summarization: from motion-centric to text-guided and Transformer-based models

5.1.1 Classical highlight detection in martial arts tricking

The first strand of the thesis addresses automatic highlight detection in a highly challenging, user-generated sports domain: martial arts tricking. The motion-centric framework developed in Section 3.2 builds on carefully designed motion descriptors and temporal post-processing to detect sparse, high-impact events in long, untrimmed videos. The MATDAT dataset and the associated evaluation protocol provide a controlled test bed for this problem.

The results in Chapter 4 show that, under this setting, classical motion-based analysis remains extremely competitive when supervision is scarce and domain structure is well understood. At frame level, the method achieves an average recall of about 94%, precision of about 74%, and an F-score of 83% across the three MATDAT videos. Event-level evaluation is even more revealing: out of 161 ground-truth highlight events, 158 are correctly detected, yielding overall event-level F-scores above 95%. These confirm that the combination of carefully tuned motion descriptors, temporal smoothing, and duration-aware selection can be highly effective in domains where highlights are strongly correlated with sudden changes in motion.

At the same time, the comparison with DL-VHD in Section 4.3 underlines the limitations of

segment-level neural approaches when faced with sparse, irregular events. DL-VHD, trained on segment-level annotations with 50% overlap, reaches mAP values around 26–30% when transferring from gymnastics or parkour to tricking, and the conversion back to frame level reveals substantial misalignments between predicted segments and true event boundaries. The proposed motion-based method clearly outperforms DL-VHD on MATDAT, especially when results are measured at frame or event level rather than at a coarse segment resolution. This highlights one of the recurring themes of the thesis: when annotation is fine-grained and highlights are short and irregular, temporal resolution and post-processing design are as crucial as the underlying representation.

However, the same results also expose the limited generality of purely motion-based solutions. The method is tuned to a single domain (martial arts tricking) with relatively homogeneous camera setups and visual patterns. Extending this approach to broadcast sports with richer semantics (e.g., ball trajectories, landing quality, or rule-specific events) quickly becomes impractical, as the motion cues alone are insufficient to separate highlights from context. This motivates the move, in later sections, toward text-guided and Transformer-based architectures that can ingest semantic cues and leverage large-scale pre-training.

5.1.2 Text-guided sports highlights with CLIP

The SportCLIP framework described in Section 3.3 generalizes the motion-centric ideas to a much broader class of sports. Instead of relying on handcrafted motion rules, frames are scored by their semantic similarity to highlight and non-highlight sentences, and the resulting saliency curves are passed through essentially the same post-processing and selection machinery introduced in the classical pipeline. This preserves the separation between scoring and selection emphasized in Chapter 1: only the origin of the relevance signal changes (motion vs. language-guided semantics), while the final summarization logic remains stable.

Quantitatively, the SportCLIP method demonstrates strong performance both on the single-sport tricking benchmark and in multi-sport scenarios. On MATDAT (martial arts tricking), it reaches 99% recall, 66% precision, and 77% F-score while already using the parameter configuration that is kept fixed across all subsequent evaluations. On the multi-sport SportCLIP dataset (diving, long jump, pole vault, tumbling), F-scores range from about 77% to over 91% per sport, with an overall average of 93% recall, 80% precision, and 84% F-score. Taken together, these results indicate that, with a single configuration, the pipeline adapts effectively to very different movement patterns and broadcast conventions, while keeping the same post-processing and selection stages.

The multi-sport formulation, however, comes with a trade-off. Compared to the tricking-specific motion-based method, which attains an F-score above 82% on MATDAT, the CLIP-based framework sacrifices a few points of performance (77% on MATDAT) in exchange for broad generality and the ability to operate on sports for which no motion heuristics or domain-specific annotations exist. This trade-off is consistent with the thesis objective of designing methods that can scale beyond a single discipline.

Several analyses in Chapter 4 further characterize the behavior of the SportCLIP approach. The parameter-sensitivity study shows that frame-level F-scores remain in a high range

(typically 75–85% or above) when varying the context window, entropy threshold, and area-based filtering parameters. This robustness is important in practice, as it suggests that extensive per-sport tuning is not necessary. The text-sensitivity experiments indicate that, as long as prompts remain semantically faithful to the underlying action, performance fluctuations across independently written prompt sets stay within about 10 percentage points. Nonetheless, CLIP is sensitive to prompts that exaggerate or misdescribe the scene; the distribution- and area-based filters proposed in Section 3.3 mitigate this by discarding flat or incoherent sentence pairs. Together, these studies show that CLIP offers a powerful text interface for sports summarization, but that careful prompt design and filtering remain essential to avoid brittle behavior.

The comparative analysis in Section 4.4 further reinforces these conclusions by benchmarking SportCLIP against both DL-VHD and an additional vision–language detector, QD-DETR. While DL-VHD attains reasonable cross-category mAP values when transferring between similar sports, its segment-based design and lack of textual conditioning limit its flexibility, and the conversion back to frame level reveals imprecise highlight boundaries. QD-DETR, evaluated in a zero-shot setting using its official pretrained weights and the same post-processing as SportCLIP, also struggles to generalize to the specialized domain of sports highlights: its frame-level F-scores on SportCLIP remain in a relatively low range (roughly 20–50%), well below those of SportCLIP. This is consistent with its original design and training objective, which target generic web videos rather than structured sports broadcasts.

By contrast, the SportCLIP framework handles multiple sports with a unified parameter set, can be steered via natural-language queries without re-training, and consistently outperforms both DL-VHD and off-the-shelf QD-DETR under the same evaluation protocol. This positions text-guided summarization with CLIP as a practical solution when access to labeled data is limited but generic vision–language models are available. In later sections (notably the Personalized Video Summarization experiments), QD-DETR will be revisited in a fine-tuned configuration on Olympic Highlights, providing an upper-bound, supervised reference that complements the training-free but broadly generalizable behavior of SportCLIP.

5.2 Face recognition under distance and occlusion

5.2.1 UPM-GTI-Face and baseline performance

The second strand of the thesis focuses on face recognition under stressors that are particularly relevant in sports and surveillance contexts: varying capture distance and occlusions. The UPM-GTI-Face dataset introduced in Section 3.5 provides a controlled, publicly available benchmark where both factors can be systematically studied. The dataset includes images at multiple distances (from 3 m to 30 m), with and without masks, and in different environments, enabling experiments that would be difficult to replicate with purely in-the-wild data.

The baseline end-to-end pipeline defined in Section 3.5 (face detection with Tiny Faces, alignment, and embedding extraction with CNN backbones) exposes several key bottlenecks, as detailed in Section 4.5. First, even with a detector configured to maximize range, reliable detection becomes challenging as distance increases and face size shrinks: the true detection

rate remains high at short and medium distances but drops sharply once the face occupies only a few dozen pixels, particularly in masked scenarios. Second, recognition performance deteriorates markedly with distance and masks: ROC curves at 3 m show clear separation between genuine and impostor pairs, whereas at 30 m—and especially when masks are present—AUC values approach random chance for some configurations. Crossed mask conditions (gallery unmasked vs. probe masked and vice versa) further reduce reliability, mirroring realistic situations in which reference images and operational data differ in mask usage.

These findings underscore the need for dedicated datasets like UPM-GTI-Face and confirm that face recognition performance reported on standard benchmarks does not automatically transfer to long-distance, partially occluded sports footage. They also motivate the subsequent comparative study of CNN and ViT backbones, which aims to identify architectures that remain robust in such conditions.

5.2.2 CNN vs. ViT backbones for face recognition

The comparative study presented in Section 3.6 evaluates six backbones (one ViT and five representative CNN) under a unified training protocol on VGGFace2 and a diverse suite of verification benchmarks: LFW (unconstrained faces), SCface (surveillance imagery), ROF (real-world occlusions), and UPM-GTI-Face (distance and occlusions). Training and identification results on VGGFace2 show that all networks can reach high accuracy, confirming that they are capable of learning strong facial embeddings under clean conditions.

The differences become visible on the verification benchmarks. On LFW, all models reach near-ceiling performance, with ViT obtaining slightly higher AUC and lower EER but only marginally improving over the best CNN. In SCface, ROF, and especially UPM-GTI-Face, the gaps widen: ViT tends to outperform CNN at medium and long distances and under occlusions, often maintaining substantially higher AUC for the same EER. In some UPM-GTI-Face configurations that aggregate all distances, several CNN exhibit AUC values close to 0.5 (random behavior), while ViT remains above this level, indicating more robust separation between genuine and impostor pairs.

From a practical standpoint, these results suggest that Transformer-based backbones offer improved robustness in the challenging conditions most relevant to PVS, at a moderate cost in parameters and inference time compared to some heavyweight CNN. At the same time, the study also shows that no single architecture dominates across all scenarios: Inception- and EfficientNet-like models retain competitive performance in some settings, and differences in computational cost may favor lightweight CNN when resources are constrained. Overall, the comparative analysis provides a principled basis for selecting a face recognition backbone within the PVS pipeline and highlights the importance of evaluating architectures on stress tests aligned with the target application.

5.3 Personalized Video Summarization (PVS)

The PVS system described in Section 3.7 represents the point where the summarization and face recognition strands converge. The pipeline is explicitly designed around a separation

of concerns: a text-guided summarizer (either SportCLIP or QD-DETR-based) proposes candidate highlight segments, while a face-analysis branch (based on ArcFace or TransFace) provides temporal identity evidence for each target. A final assignment module then decides, for each candidate segment and athlete, whether the segment should be included in that athlete's personalized summary.

The configuration grid explored in Section 4.7 systematically combines these components: two summarizers (SportCLIP vs. QD-DETR), two face recognition models (ArcFace vs. TransFace), two target representations (original gallery images vs. updated in-video image), and two assignment strategies (sequential vs. instant). End-to-end event-level evaluation on the Olympic Highlights dataset shows several consistent trends. First, configurations based on QD-DETR populate the top of the F-score range, reaching per-sport F-scores in the mid-70s to low-80s (e.g., around 84% for long jump using QD-DETR + TransFace + sequential assignment with updated targets). Second, across both summarizers, TransFace systematically outperforms ArcFace, often adding 10–20 percentage points of F-score, which confirms the advantages of transformer backbones observed in the standalone face recognition study.

Target updating and assignment strategy introduce more nuanced trade-offs. Updated targets, where the reference embedding is replaced by a clean in-video crop, tend to improve performance when detections are reliable and the backbone is strong (particularly with TransFace), but can slightly degrade results when the broadcast offers only noisy or occluded views. Sequential assignment, which propagates identity decisions over time, generally increases recall by recovering attempts in which the athlete is briefly hard to detect, while instant assignment yields higher precision by evaluating each segment independently within an expanded temporal window. Both strategies can reach similar overall F-scores, offering different recall–precision balances that practitioners can choose depending on whether missing events or including extra segments is more problematic.

The ablation studies in Section 4.7 clarify how much each component contributes to the final performance. When evaluated purely as highlight detectors (ignoring identity), QD-DETR and SportCLIP reproduce the expected behavior: a QD-DETR model fine-tuned on the Olympic Highlights dataset and evaluated on the same twenty videos effectively acts as an *oracle* summarizer, and therefore achieves higher recall and precision than the zero-shot SportCLIP configuration. This should be interpreted as an upper bound on what a fully supervised, domain-specific transformer can do when dense, fine-grained annotations are available, rather than as a fair head-to-head comparison under equal supervision. Conversely, SportCLIP operates without any training on Olympic Highlights and still delivers strong frame-level performance, trading a modest amount of precision for the ability to generalize across sports. When identity is evaluated under perfect highlight segmentation, TransFace reaches very high F-scores, particularly with sequential assignment and updated targets. Interestingly, when these components are assembled into the full PVS pipeline, the best SportCLIP-based configurations still achieve event-level F-scores not far from those of the QD-DETR-based oracle PVS, despite never being trained on Olympic Highlights. This highlights both the potential of QD-DETR-like architectures in the presence of rich supervision and the ability of zero-shot text-guided summarization to remain competitive at the system level, making

text-guided PVS a compelling option when annotated data is scarce.

Overall, the PVS experiments validate the modular design proposed in Chapter 3: new summarizers and recognition backbones can be plugged into the pipeline with minimal changes, and their impact can be quantified both individually and end-to-end. While the results still expose clear limitations—particularly in sports and broadcast conditions where faces are tiny or seldom visible, and in the reliance on relatively small summarization datasets—they also demonstrate that the core ingredients for identity-aware sports highlights are already in place. In this sense, the PVS system provides a solid and extensible foundation on which future work can build, scaling to richer data, more diverse sports, and increasingly sophisticated forms of personalization.

Chapter 6

Conclusions

This thesis has addressed the central problem of automatically generating sports highlights that are both narratively meaningful and personalized to individual athletes. Starting from the gap between the abundance of raw sports footage and the generic, one-size-fits-all summaries typically offered to viewers, the work has brought together two traditionally separate strands: video summarization, which determines *what* is interesting and *where* it happens in the video, and face recognition, which determines *who* is involved. Within this overarching objective, the thesis has developed and evaluated three tightly connected lines of research: (i) automatic highlight detection in long, untrimmed sports videos; (ii) robust face recognition under variations in distance, occlusions, and image quality; and (iii) their integration into a modular Personalized Video Summarization (PVS) system that produces identity-conditioned highlight reels. Together, these lines of work provide concrete models, datasets, and analysis that move identity-aware sports summarization from an abstract goal toward a working, experimentally validated system. This chapter summarizes the main findings along these strands and reflects on how they collectively advance the goal of identity-aware sports highlights.

The first strand has explored automatic sports highlight detection, progressing from classical motion-based analysis to text-guided and Transformer-based models. In the challenging domain of martial arts tricking, a motion-centric framework combined optical-flow-based descriptors, temporal smoothing, and duration-aware selection to detect sparse, high-impact events in long, user-generated videos. The MATDAT dataset and its associated evaluation protocol provided a controlled test bed in which this classical pipeline achieved very high frame-level accuracy, showing that carefully designed motion features remain highly competitive when supervision is scarce and domain structure is well understood. In addition, two sports summarization datasets, SportCLIP and Olympic Highlights, were created from scratch to support this line of work, offering realistic, carefully annotated benchmarks for text-guided and event-level sports highlight detection that help the field move beyond small, highly specialized datasets. Building on these foundations, a CLIP-based, text-guided framework was developed to retrieve highlightable segments directly from broadcast footage using natural-language queries. By combining vision–language embeddings with prompt engineering and robust filtering, this framework delivered consistently strong performance across multiple sports without sport-specific training, trading a modest loss with respect to a tailored motion

model for substantially greater generality. Finally, a Transformer-based detector (QD-DETR), fine-tuned on Olympic Highlights, was used as a strong supervised reference summarizer when dense, event-level supervision was available, providing contextual upper bounds for our text-guided approach. Taken together, these contributions clarify the trade-offs between classical and learned representations, between supervision and generality, and between sport-specific tuning and open-vocabulary operation, and they leave behind a set of datasets and baselines that can serve as reference points for future work.

The second strand has focused on face recognition under stressors that are particularly relevant in sports and surveillance settings, namely varying capture distances and occlusions. The UPM-GTI-Face dataset was introduced as a dedicated benchmark that jointly factors distance and occlusion under controlled acquisition, filling a gap in existing resources that typically address these challenges in isolation or only with qualitative annotations. The accompanying baseline pipeline, spanning detection, alignment, and embedding extraction, revealed how rapidly the performance of standard CNN-based recognizers degrades as faces become smaller, more distant, or partially occluded, even when those same models perform near-perfectly on unconstrained web benchmarks. To better understand the architectural choices behind these effects, a comprehensive comparison between convolutional and Transformer-based backbones was carried out under a unified training and evaluation protocol. The results showed that the Transformer-based model achieves higher verification and identification accuracy under distance and occlusion stressors, often with a favorable trade-off in model size and inference time. This analysis turns the face recognition strand into more than a benchmark exercise: it provides concrete guidance on which backbones are best suited to operate inside a personalized summarization pipeline, where reliable identity evidence must be extracted from degraded, sports-like imagery. Viewed together, UPM-GTI-Face and the accompanying study establish a rigorous testbed and set of design guidelines for long-range, occlusion-robust face recognition in sports-style scenarios.

The third strand has integrated these developments into the PVS system, an end-to-end pipeline that transforms full-length sports videos into identity-aware highlight reels. The system was designed around a clear separation of concerns: an upstream summarization module proposes candidate highlight segments, while a downstream face-analysis stream detects faces, builds robust descriptors, and measures their similarity to target templates. A modular implementation with standardized file-based interfaces has made it possible to swap summarizers, detectors, and embeddings with minimal changes, to support multiple targets and backbones, and to expose interpretable intermediate outputs such as temporal resemblance curves and assignment visualizations. Experiments on Olympic Highlights have shown that this pipeline adapts well to different sports and broadcasting styles, and that the choice of components has a measurable yet controllable impact on the final summaries. In particular, fine-tuned QD-DETR models act as strong supervised summarizers, while zero-shot SportCLIP remains surprisingly competitive at the system level; similarly, Transformer-based face recognizers provide clear gains over classical CNNs, especially when combined with sequential assignment strategies and updated target representations. Overall, the PVS experiments validate the proposed modular design and demonstrate that the core building blocks needed for identity-aware sports highlights are already in place. In practical terms, the PVS pipeline thus constitutes a complete, end-to-end prototype for identity-aware sports

summarization that is already suitable for offline use and for extension to new sports, models, and personalization criteria.

Beyond the specific methods and datasets, an important conceptual contribution of this thesis is the explicit separation between scoring and selection. Across all summarization variants, the construction of a relevance signal over time—whether derived from motion statistics, deep video features, or vision–language embeddings—is decoupled from the constrained selection step that turns those scores into compact, watchable highlight sets. The same duration- and redundancy-aware selection machinery is reused across methods, ensuring that advances in representation learning translate into tangible improvements in the final summaries rather than being confounded with changes in the post-processing stage. A similar philosophy governs the face-analysis and personalization stages, where detection, embedding, and matching are specified independently but connected through standardized interfaces. This modularity not only simplifies experimentation and ablation, but also provides a practical blueprint for extending the system to new sports, new identity cues, and new forms of personalization. As such, it constitutes one of the main conceptual takeaways of the thesis and is likely to remain relevant as summarization and recognition models continue to evolve.

6.1 Future work and open challenges

The results of this thesis suggest several promising directions for future research, both at the level of individual components and for the overall PVS system.

Stronger motion modeling in Transformer-based summarizers. The success of QD-DETR on Olympic Highlights indicates that Transformer-based detection architectures can learn rich representations for sports highlights when sufficient supervision is available. However, the current formulation relies primarily on RGB appearance. Incorporating explicit motion cues—such as optical flow, 3D convolutions, or video transformers with spatio-temporal attention—into QD-DETR-like models could improve sensitivity to fast, subtle actions (e.g., take-off phases, impact moments) that are not easily captured by static frames. Exploring architectures that fuse motion and appearance, while still benefiting from query-based decoding and text conditioning, is a natural next step.

Improved generalization and data scaling for supervised summarizers. QD-DETR is trained and evaluated on the same twenty Olympic Highlights videos, which limits conclusions about its generalization ability. A key avenue for future work is the collection or aggregation of larger, more diverse sports highlight datasets with event-level annotations and, ideally, identity labels. Training QD-DETR (or similar models) on such datasets would allow testing whether the excellent in-domain performance observed here translates to new events, competitions, and broadcasting styles. Semi-supervised or weakly supervised strategies, leveraging web-scale footage and noisy highlight cues (e.g., crowd reactions, commentator keywords), could help to reduce annotation costs.

Leveraging face detection metadata and tracking in PVS. The current PVS implementation treats face detections as independent observations that are later smoothed in time. Yet, the face-analysis stream already produces rich metadata, including bounding box coordinates, facial landmarks, and detection confidences. Incorporating multi-object tracking to maintain per-athlete tracks across frames would provide more stable identity evidence, reduce the impact of missed detections, and simplify clip assignment. Tracking could also support more sophisticated target updating strategies, for example by selecting the best-quality crop over a whole attempt rather than relying on a single frame.

Richer identity cues beyond faces. Faces are not always visible or reliable in sports broadcasts, especially at long distances or under occlusions. Extending the PVS system to incorporate complementary identity cues—such as jersey numbers, team colors, body pose, or player re-identification features trained on full-body crops—would increase robustness in real-world settings. Combining these cues with faces within a unified identity branch, possibly through multi-modal transformers, could substantially improve assignment accuracy in sports with wide shots and fast camera switches.

Robust prompt and query design for text-guided summarization. Although the CLIP-based framework is relatively robust to reasonable prompt variations, the experiments in Chapter 4 show that performance can degrade when textual descriptions become too mismatched with the visual content. Future work could explore learned prompt tuning or lightweight adapters that adjust CLIP's text embeddings to the sports domain, reducing sensitivity to user phrasing while preserving zero-shot capabilities. Interactive interfaces in which users iteratively refine prompts based on previewed summaries could also help bridge the gap between natural language and the model's internal representation of highlight semantics.

From per-athlete to multi-criteria personalization. The current PVS formulation focuses on one specific axis of personalization: which athlete the user cares about. In practice, viewers may have richer preferences, such as specific event types (e.g., only successful attempts above a certain height), tactical patterns (e.g., counterattacks in team sports), or aesthetic properties (e.g., smooth landings). Extending the PVS pipeline to combine identity with additional text-defined criteria—possibly using multi-query summarizers or hierarchical selection stages—would move closer to fully personalized sports storytelling.

System-level efficiency and real-time constraints. Finally, while the computational analyses in Chapter 4 show that both SportCLIP and the PVS pipeline are feasible for offline processing of long videos, many practical applications involve near real-time summarization during or immediately after a broadcast. Future work could explore model compression, frame subsampling strategies, and cascaded architectures that quickly discard uninformative segments while reserving more expensive processing for promising intervals. Combining such techniques with the modular design of PVS would make identity-aware summarization more suitable for deployment in live or interactive environments.

Taken together, these directions extend the three main strands of the thesis—video summarization, robust face recognition, and personalized highlight generation—toward more

scalable, robust, and flexible systems. The experimental results in Chapter 4 provide a strong foundation, demonstrating that modular pipelines built on top of classical methods, vision–language models, and modern transformers can already deliver high-quality, identity-aware sports summaries. Building on this foundation, future work can further close the gap between research prototypes and the rich, personalized highlight experiences that future audiences will expect.

References

[1] P. F. Alcantarilla, J. Nuevo, and A. Bartoli, "Fast explicit diffusion for accelerated features in nonlinear scale spaces", in *Proceedings of the British Machine Vision Conference*, BMVA Press, 2013, pp. 13.1–13.11.

[2] E. Apostolidis, E. Adamantidou, A. I. Metsai, V. Mezaris, and I. Patras, "Video summarization using deep neural networks: A survey", *Proceedings of the IEEE*, vol. 109, no. 11, pp. 1838–1863, 2021.

[3] T. Badamdorj, M. Rochan, Y. Wang, and L. Cheng, "Joint visual and audio learning for video highlight detection", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 8127–8137.

[4] T. Badamdorj, M. Rochan, Y. Wang, and L. Cheng, "Contrastive learning for unsupervised video highlight detection", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 042–14 052.

[5] P. Benz, S. Ham, C. Zhang, A. Karjauv, and I. S. Kweon, "Adversarial robustness comparison of vision transformer and mlp-mixer to cnns", *arXiv preprint arXiv:2110.02797*, 2021.

[6] J.-Y. Bouguet, "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm", *Intel Corporation*, vol. 5, no. 1-10, p. 4, 2001.

[7] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, et al., "Vggface2: A dataset for recognising faces across pose and age", in *IEEE International Conference on Automatic Face and Gesture Recognition*, 2018, pp. 67–74.

[8] J. Chai, H. Zeng, A. Li, and E. W. Ngai, "Deep learning in computer vision: A critical review of emerging techniques and application scenarios", *Machine Learning with Applications*, vol. 6, p. 100 134, 2021.

[9] R. Chellappa, J. Ni, and V. M. Patel, "Remote identification of faces: Problems, prospects, and progress", *Pattern Recognition Lett.*, vol. 33, no. 14, pp. 1849–1859, 2012.

[10] F. Chen, D. Delannay, and C. De Vleeschouwer, "An autonomous framework to produce and distribute personalized team-sport video summaries: A basketball case study", *IEEE Transactions on Multimedia*, vol. 13, no. 6, pp. 1381–1394, 2011.

[11] S. Chen, Y. Liu, X. Gao, and Z. Han, "Mobilefacenets: Efficient cnns for accurate real-time face verification on mobile devices", in *Proceedings of the Chinese Conference on Biometric Recognition*, Springer, 2018, pp. 428–438.

[12] P. W. Connolly, G. C. Silvestre, and C. J. Bleakley, "Automated identification of trampoline skills using computer vision extracted pose estimation", *arXiv preprint arXiv:1709.03399*, 2017.

[13] C. Cuevas, D. Quilón, and N. García, "Techniques and applications for soccer video analysis: A survey", *Multimedia Tools and Applications*, vol. 79, no. 39, pp. 29 685–29 721, 2020.

[14] J. Dan et al., "Transface: Calibrating transformer training for face recognition from a data-centric perspective", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 20 642–20 653.

[15] A. Deliege et al., "Soccernet-v2: A dataset and benchmarks for holistic understanding of broadcast soccer videos", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4508–4519.

[16] J. Deng, W. Dong, R. Socher, L.-J. Li, et al., "Imagenet: A large-scale hierarchical image database", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[17] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4690–4699.

[18] M. P. Díaz-Pereira, I. Gomez-Conde, M. Escalona, and D. N. Olivieri, "Automatic recognition and scoring of olympic rhythmic gymnastic movements", *Human movement science*, vol. 34, pp. 63–80, 2014.

[19] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, et al., "An image is worth 16x16 words: Transformers for image recognition at scale", *arXiv preprint arXiv:2010.11929*, 2020.

[20] H. Du, H. Shi, D. Zeng, X.-P. Zhang, and T. Mei, "The elements of end-to-end deep face recognition: A survey of recent advances", *ACM Computing Surveys*, vol. 54, no. 10s, pp. 1–42, 2022.

[21] A. Ekin, A. M. Tekalp, and R. Mehrotra, "Automatic soccer video analysis and summarization", *IEEE Transactions on Image Processing*, vol. 12, no. 7, pp. 796–807, 2003.

[22] M. E. Erakın, U. Demir, and H. K. Ekenel, "On recognizing occluded faces in the wild", in *Proceedings of the International Conference of the Biometrics Special Interest Group*, IEEE, 2021, pp. 1–5.

[23] A. Fanizzi et al., "Comparison between vision transformers and convolutional neural networks to predict non-small lung cancer recurrence", *Scientific Reports*, vol. 13, no. 1, p. 20 605, 2023.

[24] M. Fei, W. Jiang, and W. Mao, "Learning user interest with improved triplet deep ranking and web-image priors for topic-related video summarization", *Expert Systems with Applications*, vol. 166, p. 114 036, 2021.

[25] A. George, C. Ecabert, H. O. Shahreza, K. Kotwal, and S. Marcel, "Edgeface: Efficient face recognition model for edge devices", *arXiv preprint arXiv:2307.01838*, 2023.

[26] A. George and S. Marcel, "On the effectiveness of vision transformers for zero-shot face anti-spoofing", in *IEEE International Joint Conference on Biometrics*, 2021, pp. 1–8.

[27] S. Ghatak, S. Rup, B. Majhi, and M. Swamy, "Hsajaya: An improved optimization scheme for consumer surveillance video synopsis generation", *IEEE Transactions on Consumer Electronics*, vol. 66, no. 2, pp. 144–152, 2020.

[28] S. Giancola, M. Amine, T. Dghaily, and B. Ghanem, "Soccernet: A scalable dataset for action spotting in soccer videos", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition workshops*, 2018, pp. 1711–1721.

[29] E. Gonzalez-Sosa, I. Frontelo-Benito, R. Kachach, P. Perez, J. J. Ruiz, and A. Villegas, "Audience meter: A use case of deploying machine learning algorithms over 5G networks with mec", in *IEEE International Conference Consum. Electron.*, 2020, pp. 1–2.

[30] M. Grgic, K. Delac, and S. Grgic, "Scface–surveillance cameras face database", *Multimedia Tools and Applications*, vol. 51, no. 3, pp. 863–879, 2011.

[31] J. Guo, J. Deng, A. Lattas, and S. Zafeiriou, "Sample and computation redistribution for efficient face detection", *arXiv preprint arXiv:2105.04714*, 2021.

[32] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, "Ms-celeb-1m: A dataset and benchmark for large-scale face recognition", in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 87–102.

[33] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool, "Creating summaries from user videos", in *Proceedings of the European Conference on Computer Vision*, Springer, 2014, pp. 505–520.

[34] B. Han, J. Hamm, and J. Sim, "Personalized video summarization with human in the loop", in *IEEE Workshop on Applications of Computer Vision*, 2011, pp. 51–57.

[35] K. Han, Y. Wang, H. Chen, X. Chen, et al., "A survey on vision transformer", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 87–110, 2022.

[36] M. Han, P. Wen, and J. Zhao, "AESHML: An automatic editing method for soccer match highlights using multimodal learning", *IEEE Access*, vol. 12, pp. 129 967–129 974, 2024.

[37] Z. Han, A. B. Azman, M. R. B. Mustaffa, and F. B. Khalid, "Cross-modal retrieval: A review of methodologies, datasets, and future perspectives", *IEEE Access*, vol. 12, pp. 115 716–115 741, 2024.

[38] C. Harris and M. Stephens, "A combined corner and edge detector", in *Proceedings of the 4th Alvey Vision Conference*, 1988, pp. 147–151.

[39] B. He, J. Wang, J. Qiu, T. Bui, A. Shrivastava, and Z. Wang, "Align and attend: Multimodal summarization with dual contrastive losses", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 14 867–14 878.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[41] L. He, X. Ren, Q. Gao, X. Zhao, B. Yao, and Y. Chao, "The connected-component labeling problem: A review of state-of-the-art algorithms", *Pattern Recognition*, vol. 70, pp. 25–43, 2017.

[42] A. G. Howard et al., "Mobilenets: Efficient convolutional neural networks for mobile vision applications", *arXiv preprint arXiv:1704.04861*, 2017.

[43] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.

[44] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.

[45] P. Hu and D. Ramanan, "Finding tiny faces", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 951–959.

[46] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments", in *Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition*, 2008.

[47] H. Huang, Z. Wu, G. Pang, and J. Xie, "An aesthetic-driven approach to unsupervised video summarization", *IEEE Access*, vol. 12, pp. 128 768–128 777, 2024.

[48] Z.-Y. Huang et al., "A study on computer vision for facial emotion recognition", *Scientific Reports*, vol. 13, no. 1, p. 8425, 2023.

[49] T. Hussain, K. Muhammad, W. Ding, J. Lloret, S. W. Baik, and V. H. C. de Albuquerque, "A comprehensive survey of multi-view video summarization", *Pattern Recognition*, vol. 109, p. 107 567, 2021.

[50] O. Issa and T. Shanableh, "Cnn and hevc video coding features for static video summarization", *IEEE Access*, vol. 10, pp. 72 080–72 091, 2022.

[51] Jianbo Shi and Tomasi, "Good features to track", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1994, pp. 593–600.

[52] P. Kadam et al., "Recent challenges and opportunities in video summarization with machine learning algorithms", *IEEE Access*, vol. 10, pp. 122 762–122 785, 2022.

[53] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, et al., "Transformers in vision: A survey", *ACM Computing Surveys*, vol. 54, no. 10s, pp. 1–41, 2022.

[54] D. Kim, D. Joo, and J. Kim, "Tivgan: Text to image to video generation with step-by-step evolutionary generator", *IEEE Access*, vol. 8, pp. 153 113–153 122, 2020.

[55] Y. Kong, Z. Wei, and S. Huang, "Automatic analysis of complex athlete techniques in broadcast taekwondo video", *Multimedia Tools and Applications*, vol. 77, no. 11, pp. 13 643–13 660, 2018.

[56] J. Lei, T. L. Berg, and M. Bansal, "Detecting moments and highlights in videos via natural language queries", *Advances in Neural Information Processing Systems*, vol. 34, pp. 11 846–11 858, 2021.

[57] H. Li, Q. Ke, M. Gong, and T. Drummond, "Progressive video summarization via multimodal self-supervised learning", in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, 2023, pp. 5584–5593.

[58] J. Li, L. Zhou, and J. Chen, "Mobilefaceformer: A lightweight face recognition model against face variations", *Multimedia Tools and Applications*, pp. 1–17, 2023.

[59] S. Li et al., "Probing visual-audio representation for video highlight detection via hard-pairs guided contrastive learning", *arXiv preprint arXiv:2206.10157*, 2022.

[60] T. Li, Z. Sun, and X. Xiao, "Unsupervised modality-transferable video highlight detection with representation activation sequence learning", *IEEE Transactions on Image Processing*, vol. 33, pp. 1911–1922, 2024.

[61] R. W. Lienhart, "Dynamic video summarization of home video", in *Storage and Retrieval for Media Databases 2000*, vol. 3972, 1999, pp. 378–389.

[62] W. Liu, Y. Wen, Z. Yu, M. Li, et al., "Sphereface: Deep hypersphere embedding for face recognition", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017, pp. 212–220.

[63] D. G. Lowe, "Distinctive image features from scale-invariant keypoints", *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[64] K. Mangalam, H. Fan, Y. Li, et al., "Reversible vision transformers", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 830–10 840.

[65] Y. Martindez-Diaz, L. S. Luevano, H. Mendez-Vazquez, M. Nicolas-Diaz, L. Chang, and M. Gonzalez-Mendoza, "Shufflefacenet: A lightweight face architecture for efficient and highly-accurate face recognition", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[66] Rodrigo, M. UPM-GTI-Face Dataset. https://www.gti.ssr.upm.es/data/upm-gti-face-dataset. [Under CC-BY license] (accessed: 08.8.2024).

[67] J. Maurício, I. Domingues, and J. Bernardino, "Comparing vision transformers and convolutional neural networks for image classification: A literature review", *Applied Sciences*, vol. 13, no. 9, p. 5521, 2023.

[68] P. Meena, H. Kumar, and S. K. Yadav, "A review on video summarization techniques", *Engineering Applications of Artificial Intelligence*, vol. 118, p. 105 667, 2023.

[69] J. Meng, H. Wang, J. Yuan, and Y.-P. Tan, "From keyframes to key objects: Video summarization by representative object proposal selection", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1039–1048.

[70] M. Mirjalili, E. A. Gutiérrez, E. F. Fernández, V. G. Castro, and W. Tanveer, "Human-centric video summarization via identity-aware tracking", *Jornadas de Automática*, vol. 46, 2025.

[71] S. Mishra, P. Majumdar, R. Singh, and M. Vatsa, "Indian masked faces in the wild dataset", in *IEEE International Conference Image Processing*, 2021, pp. 884–888.

[72] W. Moon, S. Hyun, S. Park, D. Park, and J.-P. Heo, "Query-dependent video representation for moment retrieval and highlight detection", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 23 023–23 033.

[73] G. Mujtaba, A. Malik, and E.-S. Ryu, "LTC-SUM: Lightweight client-driven personalized video summarization framework using 2D CNN", *IEEE access*, vol. 10, pp. 103 041–103 055, 2022.

[74] J. Neves, J. Moreno, and H. Proença, "Quis-campi: An annotated multi-biometrics data feed from surveillance scenarios", *IET Biometrics*, vol. 7, no. 4, pp. 371–379, 2018.

[75] N. Ottakath et al., "Vidmask dataset for face mask detection with social distance measurement", *Displays*, vol. 73, p. 102 235, 2022.

[76] H. Pan, P. Van Beek, and M. I. Sezan, "Detection of slow-motion replay segments in sports video for highlights generation", in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, 2001, pp. 1649–1652.

[77] Y. Pan et al., "Exploring global diversity and local context for video summarization", *IEEE Access*, vol. 10, pp. 43 611–43 622, 2022.

[78] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition", *Brit. Machine Vision Conference*, pp. 1–12, 2015.

[79] M. Peronikolis and C. Panagiotakis, "Personalized video summarization: A comprehensive survey of methods and datasets", *Applied Sciences*, vol. 14, no. 11, p. 4400, 2024.

[80] S. Priyadharshini and A. Mahapatra, "Mohasa: A dynamic video synopsis approach for consumer-based spherical surveillance video", *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 290–298, 2023.

[81] A. Radford et al., "Learning transferable visual models from natural language supervision", in *International Conference on Machine Learning*, PMLR, 2021, pp. 8748–8763.

[82] M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy, "Do vision transformers see like convolutional neural networks?", *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 116–12 128, 2021.

[83] B. Reily, H. Zhang, and W. Hoff, "Real-time gymnast detection and performance analysis with a portable 3D camera", *Computer Vision and Image Understanding*, vol. 159, pp. 154–163, 2017.

[84] M. Rodrigo, C. Cuevas, D. Berjón, and N. García, "Automatic highlight detection in videos of martial arts tricking", *Multimedia Tools and Applications*, vol. 83, no. 6, pp. 17 109–17 133, 2024.

[85] M. Rodrigo, C. Cuevas, and N. García, "Comprehensive comparison between vision transformers and convolutional neural networks for face recognition tasks", *Scientific reports*, vol. 14, no. 1, p. 21 392, 2024.

[86] M. Rodrigo, C. Cuevas, and N. García, "Text-guided sports highlights: A CLIP-based framework for automatic video summarization", *IEEE Access*, 2025, Accepted for publication.

[87] M. Rodrigo, E. González-Sosa, C. Cuevas, and N. García, "UPM-GTI-Face: A dataset for the evaluation of the impact of distance and masks in face detection and recognition systems", in *IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2022, pp. 1–8.

[88] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015, pp. 815–823.

[89] A. Senior, "Tracking people with probabilistic appearance models", in *ECCV workshop on Performance Evaluation of Tracking and Surveillance Systems*, 2002, pp. 48–55.

[90] H.-C. Shih, "A survey of content-aware video analysis for sports", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 5, pp. 1212–1231, 2017.

[91] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", *arXiv preprint arXiv:1409.1556*, 2014.

[92] H. Solberg, M. H. Sarkhoosh, S. Gautam, S. S. Sabet, P. Halvorsen, and C. Midoglu, "PlayerTV: Advanced player tracking and identification for automatic soccer highlight clips", in *IEEE International Symposium on Multimedia*, IEEE, 2024, pp. 93–97.

[93] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes, "Tvsum: Summarizing web videos using titles", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5179–5187.

[94] J. Sul, J. Han, and J. Lee, "Mr. hisum: A large-scale dataset for video highlight detection and summarization", *Advances in Neural Information Processing Systems*, vol. 36, pp. 40 542–40 555, 2023.

[95] M. Sun, A. Farhadi, and S. Seitz, "Ranking domain-specific highlights by analyzing edited videos", in *Proceedings of the European Conference on Computer Vision*, 2014, pp. 787–802.

[96] S.-W. Sun, Y.-C. F. Wang, F. Huang, and H.-Y. M. Liao, "Moving foreground object detection via robust sift trajectories", *Journal of Visual Communication and Image Representation*, vol. 24, no. 3, pp. 232–243, 2013.

[97] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification", *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[98] Z. Sun and G. Tzimiropoulos, "Part-based face recognition with vision transformers", *arXiv preprint arXiv:2212.00057*, 2022.

[99] C. Szegedy et al., "Going deeper with convolutions", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[100] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1701–1708.

[101] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks", in *International Conference on Machine Learning*, PMLR, 2019, pp. 6105–6114.

[102] M. Taskiran, N. Kahraman, and C. E. Erdem, "Face recognition: Past, present and future (a review)", *Digital Signal Processing*, vol. 106, p. 102 809, 2020.

[103] A. Tejero-de-Pablos, Y. Nakashima, T. Sato, N. Yokoya, M. Linna, and E. Rahtu, "Summarization of user-generated sports video by using deep action recognition features", *IEEE Transactions on Multimedia*, vol. 20, no. 8, pp. 2000–2011, 2018.

[104] V. Tiwari and C. Bhatnagar, "A survey of recent work on video summarization: Approaches and techniques", *Multimedia Tools and Applications*, vol. 80, no. 18, pp. 27 187–27 221, 2021.

[105] P. Tome, J. Fierrez, R. Vera-Rodriguez, and D. Ramos, "Identification using face regions: Application and assessment in forensic scenarios", *Forensic Science International*, vol. 233, no. 1-3, pp. 75–83, 2013.

[106] S. Tuli, I. Dasgupta, E. Grant, and T. L. Griffiths, "Are convolutional neural networks or transformers more like human vision?", *arXiv preprint arXiv:2105.07197*, 2021.

[107] I. Ul Haq, A. Ullah, K. Muhammad, M. Y. Lee, and S. W. Baik, "Personalized movie summarization using deep cnn-assisted facial expression recognition", *Complexity*, vol. 2019, pp. 1–10, 2019.

[108] L. Vadicamo et al., "Evaluating performance and trends in interactive video retrieval: Insights from the 12th vbs competition", *IEEE Access*, vol. 12, pp. 79 342–79 366, 2024.

[109] V. Vasudevan and M. Sellappa Gounder, "Advances in sports video summarization–a review based on cricket videos", in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, 2021, pp. 347–359.

[110] B. Q. Vo and V. H. Vo, "Integrate the temporal scheme for unsupervised video summarization via attention mechanism", *IEEE Access*, vol. 13, pp. 38 147–38 162, 2025.

[111] M. Voronina, "Automated camera motion control for rhythmic gymnastics using deep learning", M.S. thesis, Tallinn University of Technology, School of Information Technologies, 2019.

[112] M. Vrigkas, E.-A. Kourfalidou, M. E. Plissiti, and C. Nikou, "Facemask: A new image dataset for the automated identification of people wearing masks in the wild", *Sensors*, vol. 22, no. 3, p. 896, 2022.

[113] H. Wang, Y. Wang, Z. Zhou, X. Ji, et al., "Cosface: Large margin cosine loss for deep face recognition", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5265–5274.

[114] M. Wang and W. Deng, "Deep face recognition: A survey", *Neurocomputing*, vol. 429, pp. 215–244, 2021.

[115] R. Webster, J. Rabin, L. Simon, and F. Jurie, "Detecting overfitting of deep generative networks via latent recovery", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 273–11 282.

[116] F. Wei, B. Wang, T. Ge, Y. Jiang, W. Li, and L. Duan, "Learning pixel-level distinctions for video highlight detection", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3073–3082.

[117] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2011, pp. 529–534.

[118] G. Wu, J. Lin, and C. T. Silva, "Intentvizor: Towards generic query guided interactive video summarization", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 503–10 512.

[119] M. Xu, H. Wang, B. Ni, R. Zhu, Z. Sun, and C. Wang, "Cross-category video highlight detection via set-based learning", in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 7970–7979.

[120] C. Yan, X. Li, and G. Li, "A new action recognition framework for video highlights summarization in sporting events", in *IEEE International Conference on Computer Science & Education*, 2021, pp. 653–666.

[121] M. Yan, M. Zhao, Z. Xu, Q. Zhang, G. Wang, and Z. Su, "Vargfacenet: An efficient variable group convolutional neural network for lightweight face recognition", in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019.

[122] S. Yang, P. Luo, C.-C. Loy, and X. Tang, "Wider face: A face detection benchmark", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5525–5533.

[123] Y. Yao, B. R. Abidi, N. D. Kalka, N. A. Schmid, and M. A. Abidi, "Improving long range and high magnification face recognition: Database acquisition, evaluation, and

enhancement", *Computer Vision and Image Understanding*, vol. 111, no. 2, pp. 111–125, 2008.

[124] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch", *arXiv preprint arXiv:1411.7923*, 2014.

[125] H. Yoon and J.-H. Han, "Content-based video retrieval with prototypes of deep features", *IEEE Access*, vol. 10, pp. 30 730–30 742, 2022.

[126] S. Zahan, G. M. Hassan, and A. Mian, "Learning sparse temporal video mapping for action quality assessment in floor gymnastics", *arXiv preprint arXiv:2301.06103*, 2023.

[127] J. Zhang, J. Huang, S. Jin, and S. Lu, "Vision-language models for vision tasks: A survey", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 8, pp. 5625–5644, 2024.

[128] J. Zhang, K. He, D. Xu, and H. Shi, "CLIP-based natural language-guided low-redundancy fusion of infrared and visible images", *IEEE Transactions on Consumer Electronics*, vol. 71, no. 1, pp. 931–944, 2025.

[129] T. Zhang, D. Wen, and X. Ding, "Person-based video summarization and retrieval by tracking and clustering temporal face sequences", in *Imaging and Printing in a Web 2.0 World IV*, SPIE, vol. 8664, 2013, 86640O.

[130] Z. Zhang, "Cross-category highlight detection via feature decomposition and modality alignment", in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, 2023, pp. 3525–3533.

[131] Y. Zhong and W. Deng, "Face transformer for recognition", *arXiv preprint arXiv:2103.14803*, 2021.

[132] H.-Y. Zhou, C. Lu, S. Yang, and Y. Yu, "Convnets vs. transformers: Whose visual representations are more transferable?", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2230–2238.

[133] P. Zhou et al., "Character-oriented video summarization with visual and textual cues", *IEEE Transactions on Multimedia*, vol. 22, no. 10, pp. 2684–2697, 2020.

[134] Y. Zhu, H. Cai, S. Zhang, C. Wang, and Y. Xiong, "Tinaface: Strong but simple baseline for face detection", *arXiv preprint arXiv:2011.13183*, 2020.

[135] Z. Zhu et al., "Webface260m: A benchmark unveiling the power of million-scale deep face recognition", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 492–10 502.