Information Sciences 578 (2021) 702–724

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Multimodal soccer highlight identification using a sparse subset of frames integrating long-term sliding windows

Carolina L. Bez, João B.O. Souza Filho *, Luiz G.L.B.M. de Vasconcelos, Thiago Frensch, Eduardo A.B. da Silva, Sergio L. Netto

Electrical Engineering Program (PEE/COPPE), Federal University of Rio de Janeiro, PO Box 68504, RJ 21941-972, Brazil

ARTICLE INFO

Article history:
Received 30 September 2020
Received in revised form 23 April 2021
Accepted 19 July 2021
Available online 21 July 2021

Keywords:
Highlight detection
Video summarization
Soccer video indexing
Frame selection
Random Forest

ABSTRACT

The massive growth of audiences eager for sport content has substantially increased workers' demand in this profitable segment. Highlight identification is vital for summarizing football matches. Decision support tools can significantly reduce the number of company employees required to tackle such a task, widely benefiting workforce resource allocation. This paper discusses the development of an automatic football highlight detector. The proposed system exploits discriminative low-level audio and video features extracted from a compact set of irregularly time–spaced frames that integrate a long-term sliding window. A new mixed wrapper-probabilistic algorithm leverages a cost-effective selection of the most significant frames submitted to a robust multi-frame consensus classification scheme. By considering a comprehensive database integrating 30 full matches, the proposed approach achieves a highlight identification rate of 100% (including all annotated goals), conjugated with a match-time compression rate of about 94%, when employing a Random Forest classifier.

1. Introduction

Recent developments in the capture, storage, and video retrieval have substantially increased sports-related media availability, especially after the popularization of the over-the-top (OTT) media services. Motivated by the massive growth in audience's interest, many content providers have been expanding the number and types of sporting events broadcasted globally through a range of platforms. As a result, the automation of video-related tasks, like production, tagging, annotation, and highlight detection has become an urgent need, strongly affecting broadcaster's workforce allocation [1], due to the rising number of releases with tight schedules.

In today's football broadcasts, highlight annotation requires the assignment of one employee just for watching the whole match, or as many, in case of multiple simultaneous transmissions. Automatic highlight detection systems (AHDSs) may significantly reduce companies' efforts in accomplishing this task. Such a tool makes it viable that a single operator carries off several matches in parallel, as now in charge of the much simpler task of discarding misjudged events or fine-tuning the starting and ending times of identified highlights.

It is worth mentioning that highlights in football broadcasts constitute long-term episodes, commonly described by slow time-varying audio and video features, especially when considering typical frame rates (30 frames/s). Usually, the relevant

* Corresponding author at: Federal University of Rio de Janeiro, Av. Athos da Silveira Ramos, 149, Building H, 2nd floor, Room 219(20), University City, Rio de Janeiro, RJ, Brazil.

https://doi.org/10.1016/j.ins.2021.07.066

information for accomplishing such a task is spread in a compact subset of frames integrating long-lasting frame windows, which should be adequately identified and accessed, representing a hard task in most situations.

This work discusses the development of an automatic football highlight detector, solely exploiting low-level audio and video features. Such a system was produced to attend a specific broadcaster company demand. The highlights of interest are represented by any scoring attempt, no matter how feeble, including all times the players shoot the ball in the goal direction, even if it is deviated or blocked during its trajectory. Identifying highlights represent a priority task for game broadcasters when summarizing game dynamics. Therefore, this kind of system may constitute a handy supporting tool for employees of broadcast companies.

As compared to the state-of-the-art, the proposed work distinguishes by considering the dynamic behaviour of frame features both in feature generation and decision-making. The feature extraction exploits a long-term sliding window, integrating past and future frames around the frame under analysis. To keep the learning process feasible, we propose a hybrid wrapper-probabilistic approach that selects the most highlight distinguishing frames, according to a complexity hyperparameter. This process assumes a non-parametric frame relevance modelling, which may exploit any classification technique with some intrinsic feature selection, in this work, an Adaboost [2] classifier. By wisely integrating the long-term behaviour of frames features, the proposed approach reduces the false-positive highlight rate from 51.5% to 11.3%, as compared to a single frame classification alternative, both assuming a decision threshold settled to a true-positive rate of 97.0%. Additional system improvements, such as adopting a decision voting filter and a Random Forest classifier [2], can further reduce this false-positive rate to 4.7%.

Another noteworthy model characteristic is the significant reduction in the number of video frames effectively involved in highlight identification by a factor of 12, considering that it operates over a set only including 61 frames. These frames are identified in long-term windows (24s) that, at a frame rate of 30 frames/s, contain 721 ($1 + 2 \times 12 \times 30$) frames. As a result, since each frame produces 16 features, the feature vector considered for highlight classification has its dimensionality reduced from 11536 (721×16) to just 976 (61×16) components. Such a number of features is readily manageable by most classification algorithms. This means that by generating a concise set of highly discriminative features from such long-lasting frame windows, the system may achieve satisfactory performance with a wide range of classifiers, thus becoming somewhat "agnostic" to the classifier choice. Besides, the best classification technique can still be selected by cross-validation. In addition, by including a consensus decision approach, the proposed system also reduces the false-alarms, only identifying a highlight when the current frame as its corresponding neighbours are classified accordingly.

Experimental evaluation includes a fully-annotated comprehensive database, comprised of 30 full matches (total of 58 h of video), covering many tournaments (consequently, different production patterns), game time (day, evening, or night), stadia, and teams.

The paper is organized as follows: Section 2 conducts a brief review on football highlight detection. Section 3 introduces the proposed system design approach, discussing the multi-frame classification scheme, the generation of discriminating audio and video features (Subsection 3.1), and the processes of time aggregation and frame selection (see Subsections 3.2 and 3.3, respectively). Section 4 describes the database (Subsection 4.1) and the performance assessment strategy (SubSection 4.2). Finally, Section 5 reports the experimental results, and Section 6 summarizes the conclusions, emphasizing main paper contributions.

2. Related work

In an AHDS, highlights of interest may include many sorts of match events, restricting to scenes of goals and goal attempts, or consider a brother range of possibilities, such as the application of penalty cards [3,4], penalty faults, free kick, corner moments [5], all types of boots [4], and periods of intense competition and emotion [3]. Relatively to system outcome, this may restrict to the occurrence of a highlight or not, or include its classification in categories, such as goal, penalty fault, yellow or red cards, among others [4,6,7].

Highlight prediction models often exploit video features. These may be classified as (i) low-level [7–9], such as the dominant-colour, the camera motion, and some moving object information; (ii) mid-level [10,7], for instance, the presence of straight white lines and the shot-type; and (iii) high-level, such as the presence of the goal frame, penalty-boxes, scorebox, spectators, the referee and players' position [11,4,9,12], or even social media content [11]. The inclusion of audio information, considering simple excitement measures, like signal energy and pitch, may be beneficial, especially in broadcasts to which narrators become loud and very excited in imminent goal moments. Multimodal schemes represent a well-succeeded trend in football highlight identification [11,13,1,14].

Motivated by the release of several open-access datasets, such as UCF101, HDMB51, Activity-Net, and THUMOS-14, the action recognition in videos has attracted a lot of attention lately [15–19], especially considering Deep Learning solutions [20–22]. In this direction, Transfer Learning (TL) was exploited by [23,24,14] for football highlight classification, aiming at mitigating the computational efforts, dataset size requirements, time, and task complexity related to developing a deep model from scratch. However, the scarcity of training data strictly covering the problem of interest (i.e., football highlights) makes this approach being of questionable efficacy [14]. Moreover, typical action dataset scenarios radically differ from those of interest in football matches, as including different sports [25], broadcasting dynamics [25], and highlight definitions [26]. Consequently, in such restricted data scenarios [27], hand-feature engineered and compact machine learning models, developed by exploiting domain knowledge [28], represent competitive or better alternatives to more complex deep learning techniques.

Up to the authors' best knowledge, only a few works exploited models or strategies that include temporal information when addressing this task. Some examples include hidden-Markov models (HMM) [10], temporal confusion networks [6], video-density modelling [29], Recursive Neural Networks [23], and Long-Short-Time Memories (LSTM) [14]. Nonetheless, these alternatives may not strictly represent highlight detectors but event classifiers instead. In contrast, others may assume only a reduced number of matches, short excerpts, or attain low or regular performance.

Therefore, the primary motivation for this work is building an end-to-end high-performance plus computationally attractive football highlight detector, exploiting low-level audiovisual features to strengthen system robustness to different operational conditions, experimentally validating the proposed approach with a comprehensive dataset. For this purpose, this work presents a new feature generation methodology that efficiently incorporates the past and the future behaviour of highly discriminative multimodal features generated over long-term audio and video frame sequences, resulting in moderate-size feature vectors. Such design achievement leads the highlight classification system being almost "agnostic" to the classifier choice, as confirmed by an extensive set of experiments including AdaBoost, k Nearest Neighbours, Support Vector Machines, Extreme Learning Machines, and Random Forests classifiers, the latter related to the best true-positive × false-positive trade-off. Moreover, the system underwent a robust performance assessment process, including 30 full football matches covering a wide range of game scenarios and highlight contents, in contrast to most previous works that solely consider short game excerpts. The use of full matches also represents a new evaluation paradigm in this context, better emulating the real operational scenarios of such a system, and contributes to more realistic performance results.

3. Proposed AHDS

Fig. 1 depicts a high-level view of the proposed automatic highlight detection system (AHDS). By aiming to improve decision robustness, the mth frame classification, denoted here as d_m, considers the outcomes of several multiple-frame classification modules (MFCMs), operating in audio and video frame packs $\mathbf{P}_i$ $(m - k_l \leqslant i \leqslant m + k_u)$ integrating the set

$$\mathscr{E} = \{d_{m-k_l}, d_{m-k_l+1}, \cdots, d_m, \cdots, d_{m+k_u-1}, d_{m+k_u}\}, \tag{1}$$

where the individual MFCM decisions are represented by $d_i \in \{0, 1\}$, and the constants k_l and k_u are design hyperparameters. Majority voting over $\mathscr{E}$ defines the system outcome d.

The pack $\mathbf{P}_i$ feeding the ith MFCM block is constituted by synchronized audio $\mathbf{s}_i$ and video $\mathbf{V}_i$ frames integrating a window $\mathscr{W}$ defined around the ith frame $(i - l_l \leqslant i \leqslant i + l_u)$, with size equal to $W = l_l + l_u + 1$, where the constants l_l and l_u represent design hyperparameters. As illustrated in Fig. 2, MFCM includes the following stages:

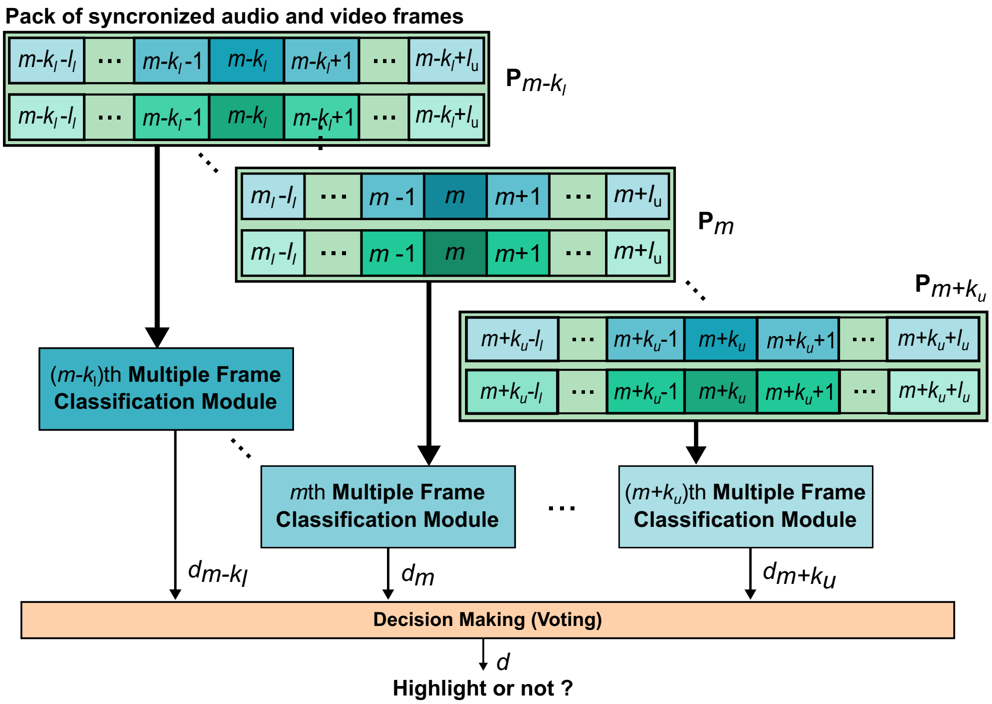
Pack of syncronized audio and video frames
m-k_l-l_l … m-k_l-1 m-k_l m-k_l+1 … m-k_l+l_u
P_{m-k_l}
m_l-l_l … m-1 m m+1 … m+l_u
P_m
P_{m+k_u}
m+k_u-l_l … m+k_u-1 m+k_u m+k_u+1 … m+k_u+l_u
(m-k_l)th Multiple Frame Classification Module
mth Multiple Frame Classification Module
(m+k_u)th Multiple Frame Classification Module
d_{m-k_l}
d_m
d_{m+k_u}
Decision Making (Voting)
d
Highlight or not ?

Fig. 1. Top level description of the proposed football highlight detection system..

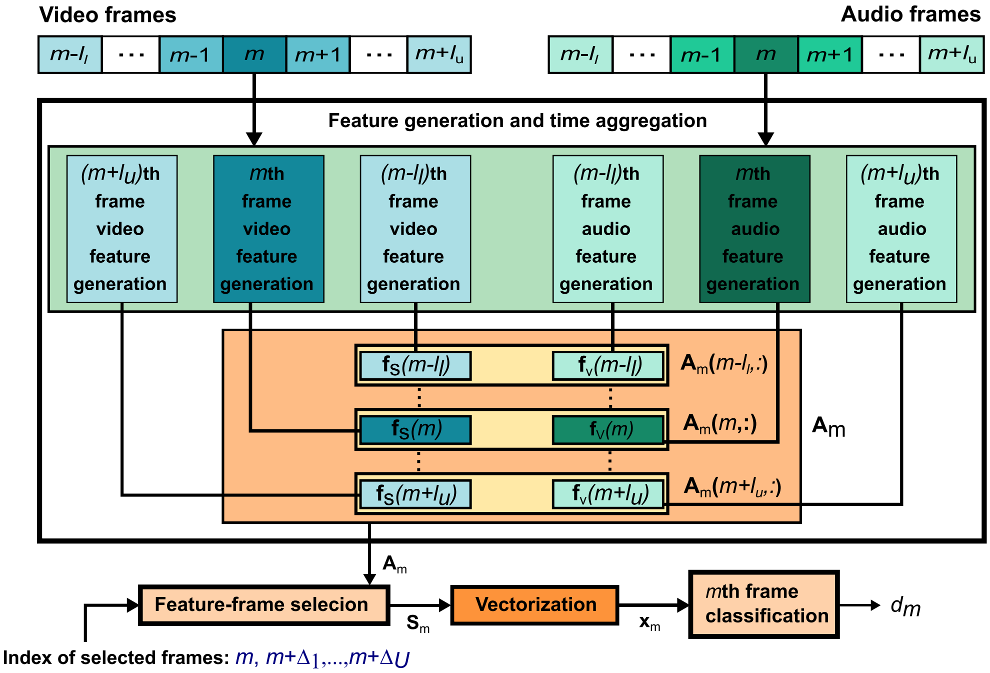
Video frames
m-l_l
m-1
m
m+1
m+l_u
Audio frames
m-l_l
m-1
m
m+1
m+l_u
Feature generation and time aggregation
(m+l_U)th frame video feature generation
mth frame video feature generation
(m-l_l)th frame video feature generation
(m-l_l)th frame audio feature generation
mth frame audio feature generation
(m+l_U)th frame audio feature generation
f_S(m-l_l)
f_v(m-l_l)
A_m(m-l_l,:)
f_S(m)
f_v(m)
A_m(m,:)
A_m
f_S(m+l_U)
f_v(m+l_U)
A_m(m+l_u,:)
A_m
Feature-frame selecion
S_m
Vectorization
x_m
mth frame classification
d_m
Index of selected frames: m, m+Δ_1,...,m+Δ_U

Fig. 2. Block diagram of the multiple frame classification module (MFCM)..

1. **Feature generation and time-aggregation**: this first stage aims to extract highly discriminating highlight features from $\mathscr{W}$, as will be discussed in SubSection 3.1. Generally speaking, this stage is responsible for generating a total of N_A^s audio and N_A^v video features, respectively represented by the scalars $a_{i_s}^s(m)$ and $a_{i_v}^v(m)$ ($1 \leqslant i_s \leqslant N_A^s$, $1 \leqslant i_v \leqslant N_A^v$), for each frame integrating the pack $\mathbf{P}_m$. The audio $\mathbf{f}_s(m)$ and video $\mathbf{f}_v(m)$ feature-frame vectors, defined as $\mathbf{f}_s(m) = [a_1^s(m)a_2^s(m)\cdots a_{N_A^s}^s(m)]$ and $\mathbf{f}_v(m) = [a_1^v(m)a_2^v(m)\cdots a_{N_A^v}^v(m)]$, summarize these features. This process is repeated to all frames included in $\mathscr{W}$, resulting in a set of audio $\mathbf{a}_{i_s}^s(m)$ and video $\mathbf{a}_{i_v}^v(m)$ feature vectors related to $\mathscr{W}$, such that $\mathbf{a}_{i_s}^s(m) \in \mathbb{R}^{\mathbb{W}}$ ($W = l_l + l_u + 1$) integrates the current value of $a_{i_s}^s(m)$ plus l_l past - $a_{i_s}^s(m - l_l), \cdots, a_{i_s}^s(m-1)$ - and l_u future - $a_{i_s}^s(m+1), \cdots, a_{i_s}^s(m + l_u)$ - values, according to $\mathbf{a}_{i_s}^s(m) = [a_{i_s}^s(m - l_l)\cdots a_{i_s}^s(m)\cdots a_{i_s}^s(m + l_u)]$. The vector $\mathbf{a}_{i_v}^v(m)$ is defined similarly to $\mathbf{a}_{i_s}^s(m)$. Finally, these vectors are concatenated in the general feature vector $\mathbf{a}_i(m) = [\,\mathbf{a}_i^s(m) \quad \mathbf{a}_i^s(m)\,] \in \mathbb{R}^{N_A}$, satisfying $N_A = N_A^s + N_A^v$, that is summarized by the matrix of general aggregated features $\mathbf{A}_m \in \mathbb{R}^{W \times N_A}$, given by $\mathbf{A}_m = [\,\mathbf{a}_1(m) \quad \mathbf{a}_2(m) \quad \cdots \quad \mathbf{a}_{N_A}(m)\,]$.
2. **Frame selection**: the third stage is responsible for selecting the U ($U \ll W$) most relevant frames, whose indexes are given by $\mathscr{S} = \{m, m + \Delta_1, m + \Delta_2, \cdots, m + \Delta_U\}$, $-l_l \leqslant \Delta_i \leqslant l_u$, where U represents a system hyperparameter. Section 3.3 discusses in more details the process for obtaining $\mathscr{S}$. The matrix of selected frame features $\mathbf{S}_m \in \mathbb{R}^{\mathbb{U} \times \mathbb{N}_\mathbb{A}}$ is formed by the lines of $\mathbf{A}_m$ specified by $\mathscr{S}$, being subsequently vectorized to produce the classifier input vector $\mathbf{x}_m \in \mathbb{R}^{\mathbb{U}\mathbb{N}_\mathbb{A}}$.
3. **Classification**: since the proposed frame selection algorithm can identify a concise but representative subset $\mathscr{S}$, MFCM may consider most standard classification algorithms. The experiments conducted in this work included the AdaBoost, k Nearest Neighbors (kNN), Support Vector Machines (SVM), Extreme Learning Machines (ELM), and Random Forests (RF), which are briefly described below.
   (a) **AdaBoost**: AdaBoost [2] represents an additive model that combines several "weak" classifiers, i.e., with low-discriminative power, usually constituted by "shallow" classification trees (CTs), aiming to result in a highly-discriminative classification system. CTs are hierarchical models that exploit sequential binary data space partitioning, realized by a set of variables identified as the most relevant for class prediction. During tree induction, new partitions are generated, targeting to solve local class confusions. Each tree is induced over a Bootstrap sample of the training set [2], and optimally combined in a step-wise fashion.
   (b) ***k*NN**: this algorithm has the premise that neighbour feature vectors are prone to belong to the same class, assigning to a given testing set instance the most frequent class-label observed among its k-nearest neighbours, usually identified using the Euclidean distance [30].
   (c) **SVM**: this model exploits the kernel trick for implicitly mapping input data into a feature space, producing a binary classifier based on a hyperplane that maximizes the margin of separation between two classes of interest [30].

(d) **ELM**: this method consists of a single hidden-layer neural network, whose output neurons are linear. In contrast to the standard multilayer perceptron (MLP), ELM hidden-layer neurons have random weights and may include non-differentiable activation functions [31]. The much simpler and fast ELM learning process, as compared to MLP, solely consists of estimating output-layer parameters by ordinary linear regression.

(e) **RF**: the Random Forest is a powerful ensemble technique that fuses multiple classification trees outcomes, induced upon random data subspaces and bootstrap dataset samples [2], considering majority voting for decision making.

These development stages are detailed in the following.

3.1. Feature generation

Feature generation comprises of extracting audio and video frame attributes, like audio signal energy, pitch, dominant colour, camera movement, and measures about their dynamics inside $\mathscr{W}$, as described in the sequence.

For synchronising audio and video match contents, the discrete-time audio signal $s(n)$ was split into M non-overlapping frames $s_m(n')$ of N samples each, with $m = 1, 2, \cdots, M$ and $n' = 0, 1, \cdots, (N-1)$. Then, these frames were multiplied by a Hamming window function [32], producing the set of audio-frame vectors $\mathbf{s}_m = [s_m(0); \quad \cdots \quad s_m(N-1)]$ associated with the set of video-frames $\mathbf{V}_m$. The window length was set to 33.3 ms to allow synchronization with the NTSC video frame-rate of 29.97 frames/s, as will be detailed in SubSection 3.1.2.

3.1.1. Audio features

The vector $\mathbf{f}_s(m)$ resumes the nine audio-features produced to the mth frame, including the simplified pitch frame estimate $g_0(m)$, the short-time frame narrator voice energy $e_{\mathrm{st}}(m)$, the Comb-filtered frame narrator voice energy $e_{\mathrm{cst}}(m)$, and two dynamic behaviour measures for each audio attribute, indistinguishably refereed now as f for conciseness: the difference on feature means $\mu^f_{av}(m)$ and the feature's mean ascending indicator $\mu^f_I(m)$. The first can be stated as

$$\mu^f_{av}(m) = \mu^f_b(m) - \mu^f_a(m), \tag{2}$$

where $\mu^f_b(m)$ and $\mu^f_a(m)$ are the average values of a given signal attribute f, before and after, respectively, the mth frame, that is,

$$\mu^f_b(m) = \sum_{k=m-N_b}^{m-1} f(k), \quad \mu^f_a(m) = \sum_{n=m+1}^{m+N_a} f(k), \tag{3}$$

with the values of N_b and N_a corresponding to durations of 10s and 3s, respectively, as suggested in [33]. The second is defined as

$$\mu^f_I(m) = \begin{cases} 1, & \text{if } \mu^f_a(m) - \mu^f_b(m) \geqslant 0 \\ 0, & \text{if } \mu^f_a(m) - \mu^f_b(m) < 0 \end{cases}. \tag{4}$$

The processes and rationales behind generating the previously mentioned features are discussed in the sequence.

Pitch. The main idea behind the pitch-estimation process is to exploit the almost-periodic behaviour during voiced intervals of speech signals. Among the several pitch estimation algorithms, the autocorrelation method is quite reliable at a reasonably low-computational cost. In such a technique, one computes the autocorrelation function $R_m(\tau)$ of a given audio signal frame $s_m(n)$ as

$$R_m(\tau) = \sum_{n=1}^{N} s_m(n)s_m(n-\tau), \tag{5}$$

and the corresponding pitch period $T_0 = 1/f_0$ can be determined by the first significant peak of $R_m(\tau)$ for $\tau > 0$, as illustrated in Fig. 3.

To mitigate non-speech components, one may compute a simplified pitch estimate $g_0(m)$ by discarding unusual pitch values, such as the ones outside the interval $50 < f_0 < 500$ Hz [34]. Additionally, by restricting $g_0(m)$ computation to audio-frames whose energy is above certain level, one may avoiding including silent or unvoiced moments. Fig. 4 illustrates the original $f_0(m)$ and the simplified $g_0(m)$ pitch behaviour. One may observe how the narrator's pitch increases and becomes smooth during the highlight event between 9.5s and 12s.

Frame Energy. Football highlights are often associated with a higher energy level in the narrator's voice. Therefore, one may use the short-time energy function

$$e_{\mathrm{st}}(m) = \sum_{n'=0}^{N-1} s^2_m(n'). \tag{6}$$

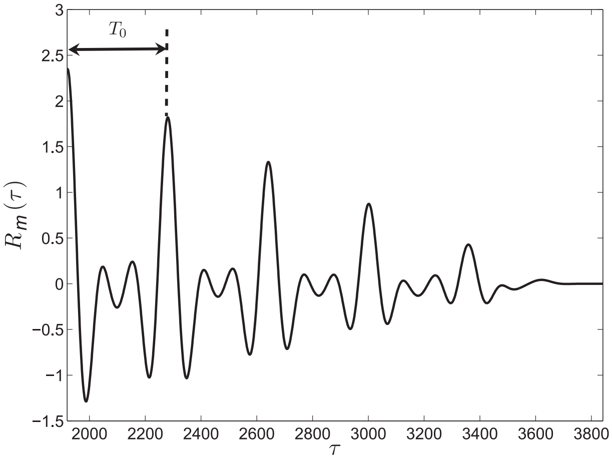

Fig. 3. Speech signal pitch-period T_0 estimation using the autocorrelation method.

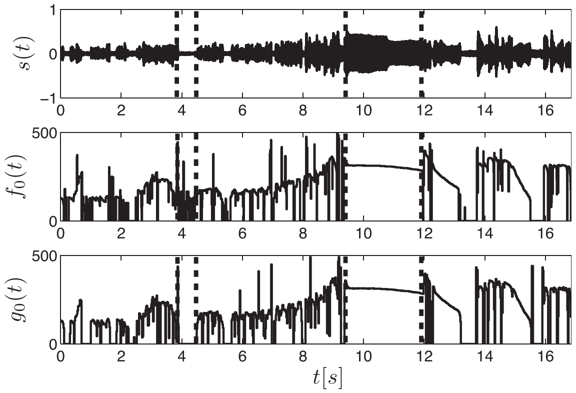

Fig. 4. An audio signal and the evolution of its pitch features in time, illustrating a pitch increase during a highlight event. Center plot: the pitch $f_0(t)$ of the signal $s(t)$ in the upper plot; bottom plot: the simplified pitch $g_0(t)$.

However, since the audio signal carries sound components other than the narrator's voice (such as background cheers), the narrator's signal can be emphasized by first applying a low-pass filter to mitigate noisy high-frequency components. Subsequently, the resulting low-pass audio-frame filtered vector $\mathbf{s}_{\mathrm{lp},m}$ can be submitted to a Comb-filter $C_m(f)$, described by

$$C_m(f) = \sum_{k=1}^{d} W(f - kf_0), \tag{7}$$

where f_0 is the narrator's pitch, d is the number of harmonics being summed up, and $W(f)$ represents a frequency-domain window function. Fig. 5 illustrates the Comb filter effect over an arbitrary $\mathbf{s}_{\mathrm{lp},m}$ vector, which is emphasizing all audio components around f_0 and its harmonics. Thus, the resulting comb-filtered signal energy is given by

$$e_{\mathrm{ce}}(m) = \sum_{n'=0}^{N-1} s^2_{\mathrm{ce},m}(n'), \tag{8}$$

with

$$s_{\mathrm{ce},m}(n') = \mathscr{F}^{-1}\{C_m(f).\mathscr{F}\{\mathbf{s}_{\mathrm{lp},m}\}\}, \tag{9}$$

where $\mathscr{F}$ and $\mathscr{F}^{-1}$ denote the direct and inverse Fourier transforms, respectively. Based on [33], the low-pass filter cut-off frequency was settled to 4400-Hz, $W(f)$ used a rectangular 50 Hz windows, and Eq. (7) adopted $d = 10$.

Fig. 6 depicts the energy features $e_{\mathrm{st}}(m)$ and $e_{\mathrm{ce}}(m)$ from an arbitrary match excerpt. From this figure, one may observe a strong association between a given highlight event (between 20 and 25s) and both features, noticing that $e_{\mathrm{ce}}(m)$ is seemingly related to a better discriminating capability.

Algorithm 1 summarizes the generation of the mth frame audio-features for convenience.

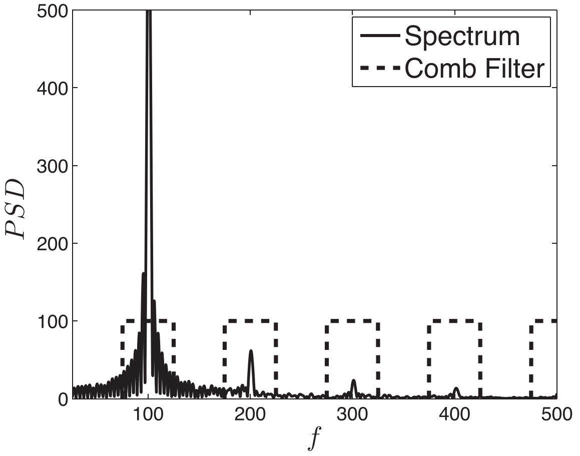

Fig. 5. Example of comb filter (dashed lines) applied to an audio signal to emphasize speech components against other audio sources.

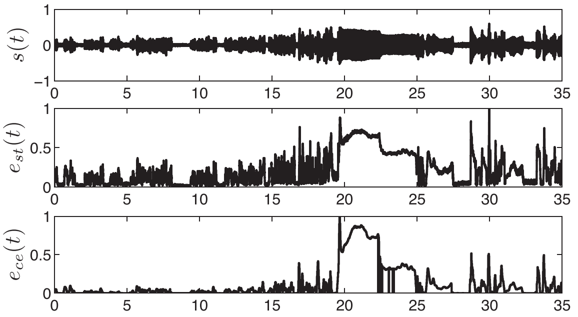

Fig. 6. An audio signal (upper plot) and the corresponding energy features (from second-top to bottom: $e_{\text{st}}(m)$ and $e_{\text{ce}}(m)$), showing a strong correlation with the event of interest that is within the time interval $20 \leqslant t \leqslant 25$s.

Algorithm 1. Generating the mth audio-frame features

```
Inputs: the current audio-frame $\mathbf{s}_m$; the previous and subsequent audio-frames to $\mathbf{s}_m$: $\mathbf{s}_i$, $(m - l_l) \leqslant i \neq m \leqslant (m + l_u)$
Outputs: (1) the current frame pitch $g_0(m)$; (2) the current audio-frame energy $e_{st}(m)$; (3) the current Comb-filtered audio-frame energy $e_{ce}(m)$; (4, 5, 6) the difference on pitch's means $\mu_{av}^{g_0}(m)$, the difference on audio-frame energy means $\mu_{av}^{e_{st}}(m)$, and the difference on audio-frame Comb-filtered energy means $\mu_{av}^{e_{ce}}(m)$; (7, 8, 9) the pitch's mean ascending indicator $\mu_I^{g_0}(m)$, the audio-frame energy ascending indicator $\mu_I^{e_{st}}(m)$, and the Comb-filtered audio-frame energy ascending indicator $\mu_I^{e_{ce}}(m)$.
Compute $R_m(\tau)$ (Eq. 5)
Determine $T_0(m)$ using $R_m(\tau)$ and compute $f_0(m) = \frac{1}{T_0(m)}$ (Section 3.1.1)
Compute $g_0(m)$ (Section 3.1.1)
Estimate the audio-frame energy $e_{\text{st}}(m)$ (Eq. 6)
Compute the Comb-filter $C_m(f)$ using $f_0(m)$ (Eq. 7).
Produce the low-pass filtered audio-feature vector $\mathbf{s}_{\text{lp},m}$ (Section 3.1.1).
Filter the vector $\mathbf{s}_{\text{lp},m}$ using the Comb-filter $C_m(f)$ (Eq. 9)
Estimate the Comb-filtered signal energy function $e_{\text{ce}}(m)$ (Eq. 8)
Produce the before-frame $\mu_b^{g_0}(m)$ and after-frame mean pitch $\mu_a^{g_0}(m)$ (Eq. 3)
Compute the difference on pitch's means $\mu_{av}^{g_0}(m)$ (Eq. 2)
Compute the pitch's mean ascending indicator $\mu_I^{g_0}(m)$ (Eq. 4)
Produce the before-frame $\mu_b^{e_{\text{st}}}(m)$ and after-frame mean audio-frame energy $\mu_a^{e_{\text{st}}}(m)$ (Eq. 3)
Compute the difference on audio-frame energy means $\mu_{av}^{e_{\text{st}}}(m)$ (Eq. 2)
Compute the audio-frame energy mean ascending indicator $\mu_I^{e_{\text{st}}}(m)$ (Eq. 4)
Produce the before-frame $\mu_b^{e_{\text{c}}}(m)$ and after-frame mean Combo-filtered audio-frame energy $\mu_a^{e_{\text{c}}}(m)$ (Eq. 3)
Compute the difference on Comb-filtered audio-frame energy means $\mu_{av}^{e_{\text{c}}}(m)$ (Eq. 2)
Compute the Comb-filtered audio-frame energy ascending indicator $\mu_I^{e_{\text{c}}}(m)$ (Eq. 4)
```

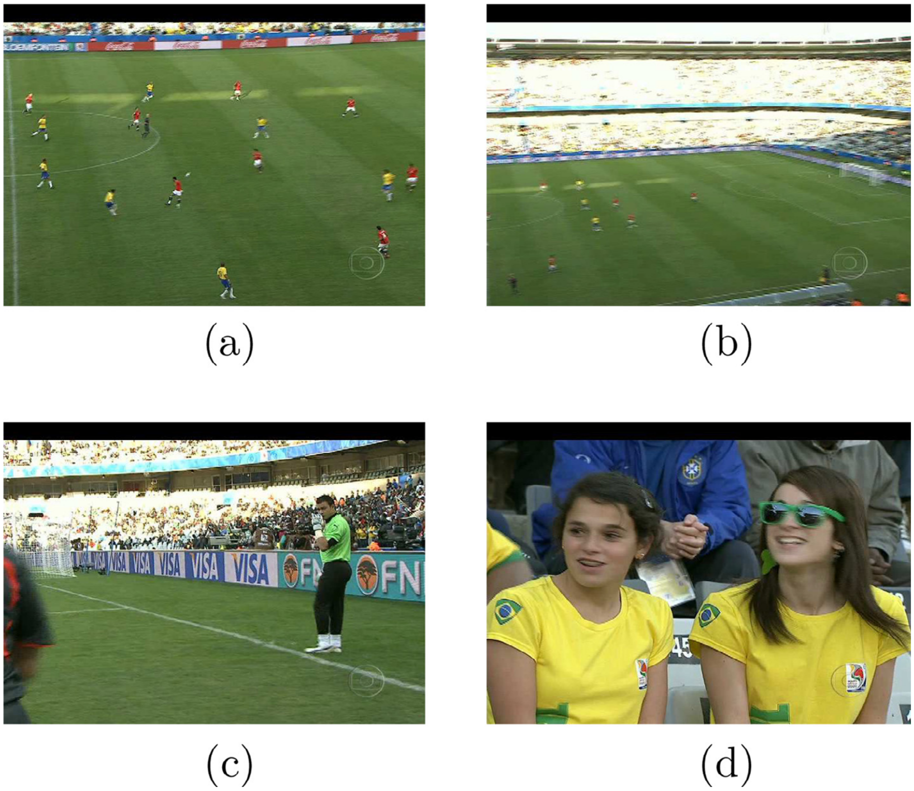

Fig. 7. Examples of screenshots: (a) Football field showing a dominant green colour; (b) Football field and sunny bleaches; (c) Football field and goalkeeper close-up; (d) Supporters showing a yellow-jersey dominant colour.

3.1.2. Video features

Summarized by the vector $\mathbf{f}_v(m)$, a total of seven video-frame features are produced for the mth frame: the dominant colour mean hue $\bar{h}(m)$, the percentage of image pixels with a colour similar to the dominant one $r_S(m)$, as well as the camera movement estimates, defined by the magnitude $\Delta(m)$, the direction $\theta(m)$, and the confidence $\rho(m)$ values. This set also includes inter-frame magnitude $\sigma^2_\Delta(m)$ and direction $\sigma^2_\theta(m)$ variances.

Dominant Colour. Traditional football broadcast shots contemplate not only a general view of the field but also close-ups or even audience shots, as exemplified in Fig. 7. Except in extreme snowing conditions, one can assume that the field colour is green. However, different lighting conditions, either natural or artificial, or even grass types, can significantly change the field's colour. What we denoted above by colour is, more precisely, the colour hue, which is the attribute that differentiates, for instance, a violet colour from one that is yellow. For these reasons, when computing the dominant colour of a frame, it is better to map the video signal from the RGB (red-green-blue) to the HSI (hue-saturation-intensity) domain, wherein similar/distinct colours are associated with similar/different representations [35].

In the HSI domain, H stands for the colour hue, S denotes its saturation, i.e., how much the colour is diluted in white, and I represents the associated intensity. This means that the hue of a pixel is defined in the HSI representation by the single component H, as opposed to RGB domain, which exploits all three R, G, and B components. It is important to emphasize that HSI representation characterizes black colour by a very low-intensity level. Similarly, low saturation corresponds to grey-level descriptions, ranging from black to white.

To convert a colour from the RGB to the HSI domain, one should use the following definitions:

$$r = \tfrac{R}{R+G+B},\ g = \tfrac{G}{R+G+B},\ b = \tfrac{B}{R+G+B}, \tag{10}$$

$$A = \sqrt{\left(r - \frac{1}{3}\right)^2 + \left(b - \frac{1}{3}\right)^2 + \left(g - \frac{1}{3}\right)^2}, \tag{11}$$

$$B = \frac{2}{3}\left(r - \frac{1}{3}\right) - \frac{1}{3}\left(b - \frac{1}{3}\right) - \frac{1}{3}\left(g - \frac{1}{3}\right), \tag{12}$$

$$\theta = \arccos\left(\frac{B}{A\sqrt{\frac{2}{3}}} - \frac{180}{\pi}\right), \tag{13}$$

to compute H, S and I components according to [36,37]:

$$H = \begin{cases} \theta, & \text{if } g \geqslant b \\ 360^\circ - \theta, & \text{otherwise} \end{cases}, \tag{14}$$

$$S = 1 - 3\min(r, g, b), \tag{15}$$

$$I = \frac{R + G + B}{3}. \tag{16}$$

Following the approach described in [38], the dominant colour of a given video frame can be inferred from the peak value of the hue histogram, as exemplified in Fig. 8. In this figure, $I_{\max}$ represents the maximum incidence value, achieved by a value of H equal to $H_{\max}$.

As mentioned before, when performing this analysis, however, one must disregard the frame pixels with low saturation or very low-intensity levels, which may lead to unreliable hue information. Therefore, the dominant colour hue mean $\bar{h}(m)$ should be estimated by assuming an interval around $H_{\max}$, with extremes given by H_{left} and H_{right}, correspondent to the incidence values I_{left} and I_{right}, respectively, such as $I_{\text{left}} = I_{\text{right}} = KI_{\max}$, where $K \in (0, 1)$ represents a design hyperparameter. In this case, the value of $\bar{h}(m)$ can be computed as

$$\bar{h}(m) = \frac{\sum_{i \in [H_{\text{left}} H_{\text{right}}]} H_i I_i}{\sum_{i \in [H_{\text{left}} H_{\text{right}}]} I_i}, \tag{17}$$

thus corresponding to a weighted average of hue values observed inside the interval $[H_{\text{left}} H_{\text{right}}]$. In this work, some experiments guided us to adopt $K = 0.2$.

Artificial graphic insertions (for advertising or public-announcements purposes, for instance) may present many pixels with the same colour, introducing sharp peaks in the hue histogram that are far away from the dominant colour hue. Nonetheless, by submitting the hue histogram incidences to a moving-average filter previously to performing peak detection, such artefacts can be dramatically reduced, with a negligible effect on $\bar{h}(m)$.

Colour-dominance analysis results, considering the video frames exhibited in Fig. 7, are shown in Fig. 9. Note that black regions include pixels with similar hue values to the dominant colour. In all cases, the dominant colour was successfully determined, despite several degrading aspects, such as strong sunlight, reduced field shot, or publicity signs.

Finally, to contribute to match field identification, one may consider quantifying the proportion of image pixels whose hue is similar (up to some similarity level S) to the dominant frame colour $r_S(m)$. If the mth video-frame $\mathbf{V}_m$ has dimensions $p \times q$, and their pixel hue values are defined by the hue pixel image vector $\mathbf{h}(m) = [h_1(m) \quad h_2(m) \quad \cdots \quad h_{p\times q}(m)]$, the feature $r_S(m)$ can be computed as

$$r_S(m) = \frac{1}{p \times q} \sum_{i=1}^{p\times q} \mathbb{1}(|\bar{h}(m) - h_i(m)| < S), \tag{18}$$

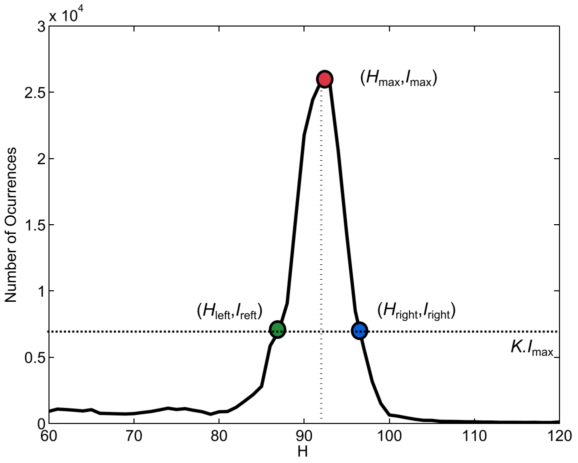

Fig. 8. Hue histogram of Fig. 7a, indicating a peak in its most dominant colour..

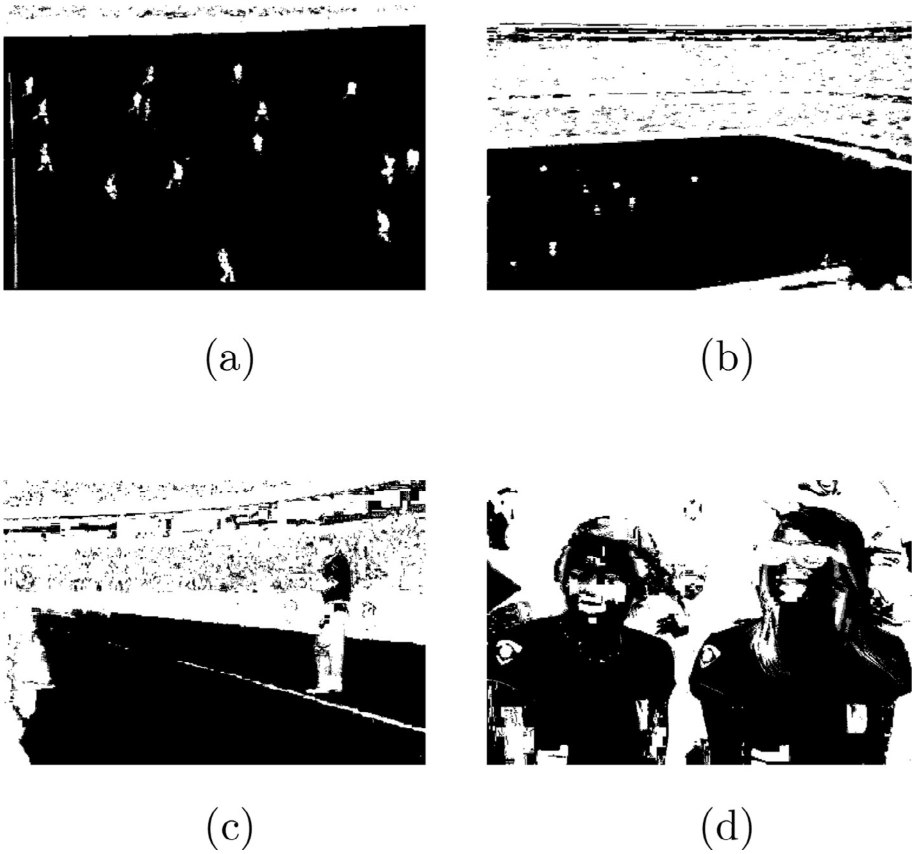

Fig. 9. Binary masks indicating regions of dominant colour for the four screenshots shown in Fig. 7.

where $\mathbb{I}(\cdot)$ is an indicator function, returning 1 for a true argument, otherwise zero. Based on some experiments, we adopted $S = 30^\circ$.

Camera Movement & *Panoramic Shots*. Current football transmissions use several cameras at distinct view planes/distances to cover the whole game action. All worldwide football broadcasts tend to follow a similar dynamics: close-ups are dedicated to moments to which the ball is not moving significantly, while the moments of interest, such as fast-breaks or goal shots, are all shown in a panoramic view. Therefore, characterizing the camera type or its movement may assist the classifier in identifying a highlight [39].

An estimate of camera movement for each frame can be derived by computing the phase correlation $C_m(x,y)$ between the current video frame $\mathbf{V}_m$ and the previous one $\mathbf{V}_{m-1}$, as follows [36]:

$$C_m(x,y) = \mathscr{F}^{-1}\left[\frac{\mathscr{F}[\mathbf{V}_m]\mathscr{F}^*[\mathbf{V}_{m-1}]}{|\mathscr{F}[\mathbf{V}_m]\mathscr{F}^*[\mathbf{V}_{m-1}]|}\right], \tag{19}$$

where * denotes the complex conjugate operator. Ideally, suppose that the video-frame $\mathbf{V}_m$ corresponds to a translated version of the frame $\mathbf{V}_{m-1}$, i.e., $\mathbf{V}_m(x,y) = \mathbf{V}_{m-1}(x - v_x, y - v_y)$. In this case, their Fourier transforms are related as

$$\mathscr{F}[\mathbf{V}_m] = \mathscr{F}[\mathbf{V}_{m-1}]e^{-2\pi j(f_x v_x + f_y v_y)}, \tag{20}$$

where f_x and f_y represent frequencies in the x and y-axes, respectively. Consequently,

$$C_m(x,y) = \delta(x - v_x, y - v_y), \tag{21}$$

and the coordinates v_x and v_y are easily distinguishable by the prominent peak in the $C_m(x,y)$ plot.

Noteworthily, the particular movement of objects in a football match, such as players, ball, referee, and others, are superimposed to the camera movement, leading to a noisy correlation function. As a result, the peak location defined by $C_m(v_x, v_y)$ only indicates the leading (probably the camera's) movement. Notably, this peak amplitude $\rho = |C_m(v_x, v_y)|$ may provide a confidence measure about this estimate: a large peak indicates that many image pixels follow a similar direction, a behavior often observed in panoramic shots. Additionally, it is convenient to exploit the polar coordinate system to represent the camera movement using the subsequent formula

$$\Delta(m) = \sqrt{v_x^2 + v_y^2}, \quad \theta(m) = \arctan\left(\frac{v_y}{v_x}\right), \tag{22}$$

where the parameters Δ and θ indicate the intensity and the angular direction of the detected movement, respectively.

Fig. 10 illustrates the time evolution of the camera movement parameters Δ, θ, and ρ during an arbitrary video excerpt, which starts with a close-up shot (scene 1), followed by a panoramic view (scene 2), and an audience shot (scene 3), ending with another panoramic view (scene 4). From this figure, one can infer that the camera movement features Δ and θ become stable during the panoramic takes, case where the image pixels are mostly moving in the same direction, thus leading to larger ρ values.

By the previous observations, one may conclude that the variance of Δ_m and θ_m in panoramic views tends to be lower than in other scene modalities. To exemplify such behaviour, Fig. 11 exhibits the variance of such features during a 15-frame window, which corresponds to approximately 0.5s in the NTSC standard, considering the same video excerpt from Fig. 10. It is noteworthy that these statistics can aid the classifier in distinguishing panoramic views. The variance of some feature f, in this case Δ_m or θ_m, inside a window integrating the current plus R past frames can be defined as

$$\sigma_f^2(m) = \frac{1}{R-1} \sum_{i=m-R+1}^{m} (f_i - \bar{f}_m)^2 \tag{23}$$

$$\bar{f}_m = \frac{1}{R} \sum_{i=m-R+1}^{m} f_i, \tag{24}$$

where f_i refers to the ith frame feature value. In this work, we adopted $M = 15$, in accordance with some trials.

Algorithm 2 resumes the process of generating all video-frame features considered in this work.

Algorithm 2. Generating the mth frame video-features

```
Inputs: the current video-frame $\mathbf{V}_m$; the previous and subsequent video-frames to $\mathbf{V}_m$: $\mathbf{V}_i$, $(m - l_l) \leqslant i \neq m \leqslant (m + l_u)$
Outputs: (1) the dominant colour hue mean $\bar{h}(m)$; (2) the percentage of image pixels with a colour similar to the
    dominant one $r_S(m)$; (3,4,5) the magnitude $\Delta(m)$, the direction $\theta(m)$, and the confidence $\rho(m)$ of camera movement
    estimates; (6,7) the inter-frame magnitude $\sigma_\Delta^2(m)$ and direction $\sigma_\theta^2(m)$ variances.
 1: Normalize the RGB components of all pixels integrating the $m$th frame image (Eq. 10).
 2: Compute $\theta$ (Eq. 13) for all normalized RGB components using Eqs. 11,12.
 3: Compute the hue values of all normalized RGB components (Eq. 14).
 4: Determine the dominant hue $H_{\max}$ and the interval limits $H_{\text{left}}$ and $H_{\text{right}}$ from the $m$th frame hue histogram
    (Section 3.1.2).
 5: Compute the dominant colour hue mean $\bar{h}(m)$ (Eq. 17).
 6: Compute the percentage of image pixels with a colour similar to the dominant one $r_S(m)$ (Eq. 18).
 7: Compute the phase correlation function $C_m(x, y)$ between the frames $\mathbf{V}_m$ and $\mathbf{V}_{m-1}$ (Eq. 19)
 8: Identify the pair of coordinates $(v_x, v_y)$ to which $C_m(x, y)$ is maximum (Section 3.1.2).
 9: Compute $\Delta(m)$, $\theta(m)$ (Eq. 22), and $\rho = |C_m(v_x, v_y)|$.
10: Compute the inter-frame camera movement magnitude $\sigma_\Delta^2(m)$ and direction $\sigma_\theta^2(m)$ variances (Eq. 23).
```

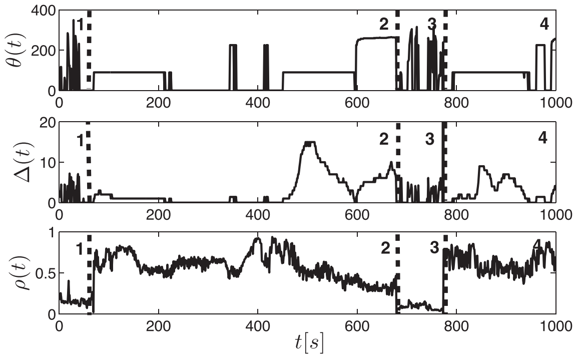

Fig. 10. Time evolution of movement features θ, Δ, and ρ during a video excerpt including four takes: close-up (scene 1); panoramic view (scene 2), audience shot (scene 3), and another panoramic view (scene 4).

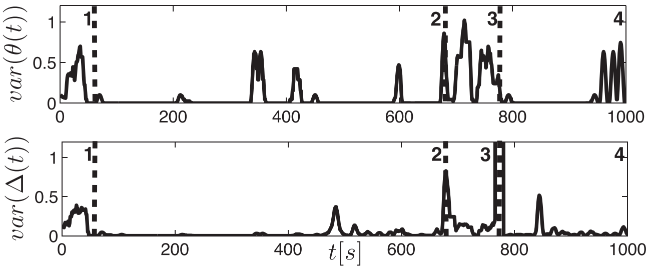

Fig. 11. Time evolution of the variances of θ and Δ for the same video excerpt as in Fig. 10.

3.2. Feature time-aggregation

Defining lower (l_l) and upper (l_u) window boundaries constitutes a central issue to the feature time-aggregation stage (see Section 3). More oversized windows enable the system to account for a richer feature panorama during decision making. However, it turns the subsequent feature selection process harder, especially when considering keeping the classifier's input dimensionality manageable for most classification techniques. In our case, the raw feature vector would have 11536 ($16 + 16 \times 30 \times 12 \times 2$) elements, considering the adoption of a symmetrical window centred in the current frame with 24 s of duration, a frame rate of 30 frames/s, and a total of sixteen features produced to each frame.

We propose a simplified low-resolution cross-validation approach to define such window limits, for simplicity assumed as symmetrical. In the related experiments, to allow a fair comparison among the multiple window length alternatives, the classifier input dimensionality was kept frozen by proportionally subsampling the frames in time while increasing the corresponding trial windows' length. The process adopted here is quite simple, starting with a trial considering a hypothetical reference window with $(2M + 1)$ frames lasting for $L_1 = t_0$ s. In the second trial, the window's length is increased to $L_2 = 2L_1 = 2t_0$ but only includes the odd frames. Subsequent trials solely consider windows lasting for integer multiples of L_1, i.e., $L_i = kL_1, k \in \mathbb{Z}, k \geqslant 3$, but only takes frames subsampled from the original sequence by a factor k. Naturally, this procedure is equivalent to sampling a single line from each group of k adjacent lines in the matrix of aggregated features $\mathbf{A}_m$. Finally, the best window size can be defined by considering the Occam-razor principle [30].

3.3. Selection of Time-Aggregated Features

The development of an AHDS involves the crucial task of identifying the subset of frames most representative of a highlight in long-term windows. The strategy proposed here considers an irregular frame subsampling in time based on frames relevance, inferred by a non-parametric modelling scheme, as illustrated in Fig. 12. This wrapper-like approach [30] can use any classifier with intrinsic feature selection, in our case, an Adaboost classifier. Roughly, this process consists of first defining a training set composed of frames regularly subsampled in time. The rationale behind that is keeping the number of features feeding the classifier involved in this task under control. In the sequence, an AdaBoost classifier is trained over this set, and the resulting model is accessed for inferring each frame's relevance. Then, the remaining frames have their relevance computed by interpolating the values observed for their neighbours integrating the training set. Finally, an irregular frame subsampling procedure exploiting such values defines the set of frame indexes $\mathscr{S}$, not necessarily uniformly time–spaced, that best describes the problem at hand. This procedure is detailed in the following steps:

1. **Training audio and video frames set definition:** this stage consists of selecting a subset of synchronized and consecutive (in time) audio $\mathbf{s}_i$ and video-frames ($\mathbf{V}_i, 1 \leqslant i \leqslant q$) to integrate the training set exploited in this frame selection process.
2. **Time aggregated audio and video features generation:** based on the q pairs of audio and video frames defined in the previous step, this step produces a total of q matrices of aggregated features $\mathbf{A}_i$ ($1 \leqslant i \leqslant q$), as described in subsection 3.1.
3. **Regular frame subsampling**: this process aims at reducing the computational efforts when inferring frames' relevance. For this, a subsampling factor p is defined by the user to produce the matrix $\mathbf{SA}_i$, a subsampled version of the feature matrix $\mathbf{A}_i$ ($1 \leqslant i \leqslant q$), whose jth-line corresponds to $\mathbf{SA}_m(j, :) = \mathbf{A}_m(1 + (j - 1) \times p, :), 1 \leqslant j \leqslant W/p$, assuming W as an integer multiple of p.
4. **Subsampled frame relevance wrapper inference**: the relevance of any feature-frame pair may be inferred by the number of times it integrates some node decision in the AdaBoost model. Therefore, this process involves training an auxiliary AdaBoost model for highlight identification that considers the matrices $\mathbf{SA}_i$ ($1 \leqslant i \leqslant q$) generated in the previous step. After, the subsampled frame-feature relevance matrix $\mathbf{SR} \in \mathbb{R}^{W/p \times N_A}$ can be determined by accessing the resulting model, wherein each entry SR_{ij} would represent the relevance of the ith feature associated with the jth frame. Finally, the relevance of each subsampled frame is summarized by the subsampled frame relevance vector $\mathbf{sr}$, defined by summing up all $\mathbf{SR}$ columns.

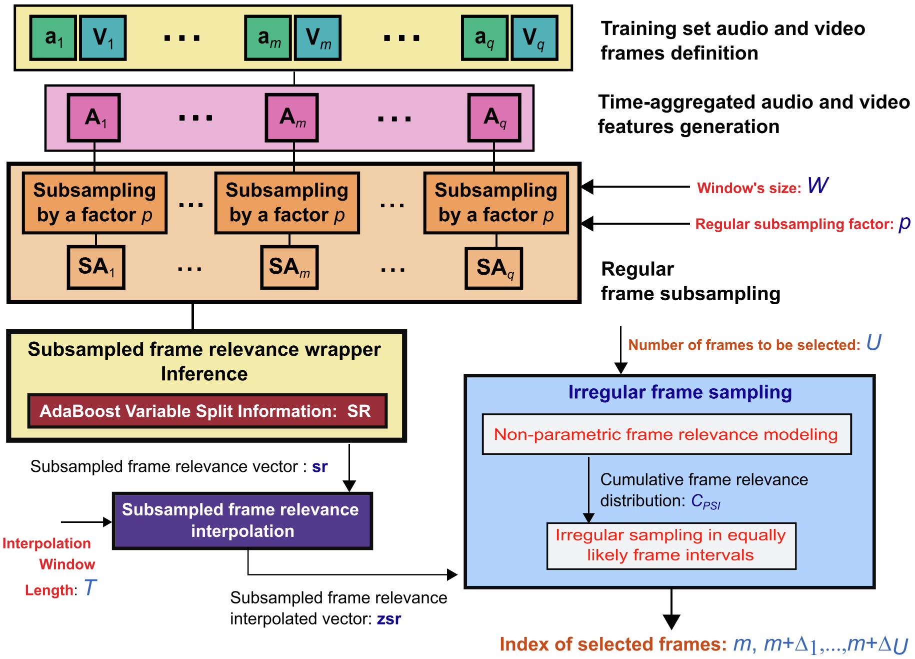
a1 V1 ... am Vm ... aq Vq
Training set audio and video frames definition
A1 ... Am ... Aq
Time-aggregated audio and video features generation
Subsampling by a factor p
SA1 ... SAm ... SAq
Window's size: W
Regular subsampling factor: p
Regular frame subsampling
Subsampled frame relevance wrapper Inference
AdaBoost Variable Split Information: SR
Number of frames to be selected: U
Irregular frame sampling
Non-parametric frame relevance modeling
Cumulative frame relevance distribution: C_PSI
Irregular sampling in equally likely frame intervals
Subsampled frame relevance vector : sr
Subsampled frame relevance interpolation
Interpolation Window Length: T
Subsampled frame relevance interpolated vector: zsr
Index of selected frames: m, m+Δ1,...,m+ΔU

Fig. 12. Block diagram of the proposed frame selection method..

5. **Subsampled frame relevance vector interpolation**: motivated by the slow time-varying nature of frame features, a sufficiently accurate estimate of the original frame sequence relevance may be achieved by interpolating $\mathbf{sr}_m$. For this, the vector $\mathbf{sr}$ must be first zero-padded to produce the vector $\mathbf{zsr} = [sr(1), 0, \cdots, 0, sr(2), 0, \cdots, 0, sr(W/p), 0, \cdots, 0], \mathbf{zsr} \in \mathbb{R}^W$. Subsequently, it should be convolved with a rectangular T-length unitary window $h(k)$, defined as

$$h(k) = \begin{cases} 1, & 0 < k < T, \\ 0, & \text{otherwise}, \end{cases} \tag{25}$$

resulting in the estimated frame relevance vector $\mathbf{r} = [r(1); \cdots; r(W)] \in \mathbb{R}^W$. In this work, experiments have led to $T = 16$.

6. **Irregular frame sampling guided by the inferred relevance**: the rationale here is sampling frames with a probability defined by their relevance. Thus, by considering a sliding window defined between the time-intervals $(m - l_l)$ and $(m + l_u)$ including a total of W frames, each one indexed by some integer i, such that $1 \leqslant i \leqslant W$, the sampling process will assume the probability of selecting the ith frame proportional to the corresponding frame relevance estimate $r(i)$, as follows:

$$P_{SF}(i) = \frac{r(i)}{\sum_{j=1}^{W} r(j)}, \quad 1 \leqslant i \leqslant W. \tag{26}$$

If one defines a set of U frame time-indexes given by $\mathscr{F} = \{\gamma_1, \gamma_2, \cdots, \gamma_l, \cdots, \gamma_U\}$, such that $1 = \gamma_1 < \gamma_2 < \cdots < \gamma_U = W$, the probability of selecting any frame from the interval $I_l = [\gamma_{l-1}, \gamma_l]$, where $2 \leqslant l \leqslant U$, will be

$$P_{SI}(l) = \sum_{j=\gamma_{l-1}}^{\gamma_l} P_{SF}(j), \tag{27}$$

For defining each γ_l, one should make three assumptions:

(i) Only one frame is sampled from each interval;

(ii) The intervals are defined such that $P_{SI}(l) = \frac{1}{U}$, i.e., they are equally likely to be sampled. This last assumption is equivalent to saying that all intervals would similarly contribute for solving the problem at hand. Therefore, short-time intervals will be assigned to packs of frames associated with highly relevant periods of time, while longer intervals will be dedicated to those whose relevance is more spread in time;

(iii) The current frame (m) is always included.

For simplicity, one may assume the upper-interval limit (γ_l) index for defining which frame will be sampled from the corresponding interval $I(l)$, which seems reasonable, considering that the number of time-intervals U is large enough and the features are slowly-varying. From the above, for defining each γ_l, one may consider the inverse cumulative distribution function related to P_{SI}, here denoted as $C^{-1}_{P_{SI}}(\cdot)$, as follows:

$$\gamma_l = \left\lfloor C^{-1}_{P_{SI}}\left(\frac{l}{U}\right)\right\rfloor, \quad 1 \leqslant l \leqslant U, \tag{28}$$

where the operator $\lfloor x \rfloor$ denotes the largest integer smaller than or equal to x. Therefore, by assuming that the time intervals of the selected frames are given by $\mathscr{S} = \{m, m + \Delta_1, \cdots, m + \Delta_U\}$, the value of Δ_j would correspond to

$$\Delta_j = \gamma_j - l_l - 1, \qquad 1 \leqslant j \leqslant U. \tag{29}$$

Algorithm 3 summarizes the frame selection algorithm proposed.

Algorithm 3: Proposed frame selection algorithm

```
Inputs: Training audio $\mathbf{s}_i$ and video frames $\mathbf{V}_i$ $(1 \leqslant i \leqslant q)$
Outputs: Indexes of the selected frames $\mathscr{S} = \{m, m + \Delta_1, \cdots, m + \Delta_U\}$
1: Identify a proper value for the window length $W$ by a coarse-search (Section 3.2).
2: Generate the matrices of aggregated features $\mathbf{A}_i$ $(1 \leqslant i \leqslant q)$ related to the pairs of audio $\mathbf{s}_j$ and video $\mathbf{V}_j$ frames $(1 \leqslant j \leqslant q)$ (Section 3.1).
3: Define a proper subsampling factor $p$ and compute the subsampled aggregated feature matrix $\mathbf{SA}_i$ $(1 \leqslant i \leqslant q)$
4: Train an AdaBoost classifier with the training set $\mathscr{T} = \{\mathbf{SA}_1, \mathbf{SA}_2, \cdots, \mathbf{SA}_q\}$.
5: Access the AdaBoost model to produce the subsampled frame-feature relevance matrix SR and the subsampled frame relevance vector sr.
6: Define the size $T$ of the interpolating window $h(k)$ (Eq. 25).
7: Produce the interpolated frame relevance vector r by convolving the window $h(k)$ with the zero-padded relevance vector zsr.
8: Define the number of frames to be selected $U$.
9: Compute the probability of selecting the $i$th frame $P_{SF}(i)$ (Eq. 26).
10: Compute the probability of selecting any frame inside the $i$th interval $P_{SI}(i)$ (Eq. 27).
11: Compute the offsets $\Delta_i$ from $\mathscr{S}$ (Eq. 29) using (Eq. 28).
```

4. Experimental methodology

4.1. Database description

Frequently, the database represents a critical factor in the development of most classification systems due to requiring the coverage of all the distinct dynamics over the problem under solution. In our case, the inclusion of different narrators, football championships, production teams, stadiums, periods of the day, and even weather conditions represents a useful strategy for enforcing system robustness to multiple operational scenarios. In addition, the size of the dataset must enable that some matches can be exclusively dedicated to inferring model's parameters (model training), while others are solely dedicated for tuning model hyperparameters.

As pointed by the work in [14], most relevant soccer datasets are under copyright infringement. As a consequence, accessing comprehensive datasets tends to be challenging, especially when considering entire matches instead of small excerpts for system development and evaluation, as it is our case. Besides, considering that each football match lasts for approximately 90 min, labelling such a comprehensive database is a relatively cumbersome and tedious task, even for fans. This fact is particularly true when one enforces the required consistency in such a subjective duty. Therefore, our experiments were restricted to a moderate-size dataset, defined and gently leased by a Brazilian TV broadcaster. The number of championships, matches, highlight moments, and video hours are well positioned regarding the state-of-the-art. The fact that entire matches were employed during system development and evaluation introduces some additional match variability relative to players' and fans' behaviour. In addition, our database does not include any matches played in snowy conditions. Nonetheless, light and moderate snow might result in a diluted field colour that would affect the corresponding hue negligibly. This is equivalent to saying that the mean hue of the dominant colour and the percentage of image pixels having a hue similar to the dominant colour hue are expected to be close to those obtained in non-snowy conditions. This behaviour is also expected to be observed with camera movement estimates, leading to system robustness regarding such scenarios. For the present work, a single person annotated a total of 30 matches, targeting to maintain, as much as possible, a uniform criterion to identify a highlight and to define its starting and ending times.

In summary, some database characteristics are the following, as detailed in Table 1:

- 5 distinct tournaments, thus guaranteeing different production rules: World Cup 2010 (WC), Confederations Cup 2009 (CC), Brazilian National Championship 2010 (BR), UEFA Champions League 2010 (CL), Libertadores Cup 2010 (LC);

Table 1
General characteristics of all 30 matches annotated: championship (CH); stadium (ST); narrator (NA); daytime (DT); number of goals (G#); number of highlights (H#).

#	Teams	CH	ST	NA	DT	G#	H#
01	Argentina x Germany	WC	S01	N1	D	3	11
02	Argentina x Mexico	WC	S02	N2	N	4	10
03	Argentina x Nigeria	WC	S03	N3	D/N	1	11
04	Argentina x S. Korea	WC	S02	N2	D	5	9
05	Brazil x Chile	WC	S03	N3	N	3	8
06	Brazil x Holland	WC	S04	N3	D/N	3	11
07	Chile x Switzerland	WC	S04	N4	D/N	2	8
08	Denmark x Japan	WC	S05	N1	N	4	12
09	France x Mexico	WC	S06	N3	N	2	6
10	Germany x England	WC	S07	N1	D/N	5	15
11	Germany x Spain	WC	S08	N3	N	1	5
12	Germany x Uruguay	WC	S04	N2	N	5	14
13	Italy x Slovakia	WC	S03	N2	D/N	6	13
14	Holland x Japan	WC	S08	N2	D	1	4
15	Holland x Slovakia	WC	S08	N1	D/N	2	4
16	Portugal x N. Korea	WC	S01	N1	D	7	14
17	Spain x Holland	WC	S03	N3	N	0	8
18	Spain x Portugal	WC	S01	N2	N	1	10
19	Spain x Switzerland	WC	S08	N1	D/N	1	12
20	Uruguay x Holland	WC	S01	N2	N	5	10
21	Uruguay x S. Korea	WC	S04	N1	D/N	3	9
22	Brazil x Italy	CC	S09	N3	N	3	12
23	Spain x USA	CC	S07	N2	N	2	8
24	Brazil 1 x Brazil 2	BR	S10	N1	D/N	3	10
25	Brazil 3 x Brazil 4	BR	S11	N1	D/N	1	9
26	Brazil 5 x Brazil 2	BR	S12	N4	D/N	4	9
27	Spain 1 x Italy 1	CL	S13	N3	N	2	5
28	Germany 1 x Italy 1	CL	S14	N3	D/N	2	8
29	Brazil 6 x Brazil 7	LC	S10	N2	N	3	9
30	Brazil 7 x Brazil 6	LC	S15	N2	N	2	8

- 20 national teams and 10 local teams (7 from Brazil, 1 from Spain, 1 from Italy, and 1 from Germany), yielding to different football uniform patterns/colours;
- 15 stadiums, with distinct grass characteristics and broadcasting conditions;
- 3 different lighting conditions: day (D), evening (D/N), and night (N) matches;
- 4 male Brazilian narrators, imposing different styles and pitch characteristics;
- A total of 282 highlight events, with an average duration of 6.0s, including 86 goals.

All video files are in Standard Definition (SD) format (486 lines x 720 columns), with a frame rate of 29.97 frames/s, encoded in MPEG-2 at 6 Mbps, whereas audio files follow PCM format (48 kHz sampling rate). The 30 matches are represented by 6,290,767 frames, which is equivalent to approximately 58 h of video.

The selection of highlights includes some degree of subjectivity, even when performed by a professional operator. When annotating the database in this work, all (successful or not) goal attempts were considered as highlight events. Another problematic aspect refers to the definition of the beginning and the ending times of a given highlight. In such cases, we assumed as highlight beginnings the moments where the goal attempt becomes clear (a few seconds before the player shoots, for instance). In turn, the moments right after the goal is scored or its attempt fails were taken as highlight ends. Following this framework, when annotating the database, the highlight detection was performed in two rounds. The first was responsible for identifying the highlight existence. In the second, the beginning and ending times of each identified highlight were determined. This procedure allows a higher uniformity within each round, resulting in a more reliable frame classification. In practice, once the system had automatically detected a given highlight, a professional operator can easily make slight adjustments in the corresponding timestamp. Nonetheless, our system performed quite reliably regarding this issue in all our experiments.

4.2. Performance assessment

The database was split into three sets: training (TR), validation (VA), and test (TS), targeting model development, hyperparameter tuning, and final system evaluation, respectively. Training and validation sets did not include any match associated with the narrator N4 (matches 07 and 26 in Table 1) to enforce the system robustness to narrator choice. These matches were exclusively assigned to the test set, comprising of approximately 457,000 frames.

A sevenfold cross-validation [30] procedure defined the training and validation sets, considering folds composed by the whole matches. Therefore, the twelve-eight training and validation matches were partitioned into seven folds. Each fold was formed of four distinct matches, contemplating various narrators, tournaments, teams, stadiums, and lighting conditions. As a result, this process resulted in seven pairs of training and validation sets.

Table 2
Training and Validation Set Composition (see text).

Sevenfold Cross Validation									
			Cross-Validation Iterations (#)						
Fold (#)	Fold Matches	Fold TR Size (in 10^3 frames)	1	2	3	4	5	6	7
	01, 18, 25, 27	12.2	VA	TR	TR	TR	TR	TR	TR
2	08, 14, 21, 23	13.6	TR	VA	TR	TR	TR	TR	TR
3	10, 13, 17, 29	20.3	TR	TR	VA	TR	TR	TR	TR
4	05, 09, 12, 15	14.3	TR	TR	TR	VA	TR	TR	TR
5	03, 11, 16, 24	15.9	TR	TR	TR	TR	VA	TR	TR
6	04, 19, 20, 22	21.2	TR	TR	TR	TR	TR	VA	TR
7	02, 06, 28, 30	17.3	TR	TR	TR	TR	TR	TR	VA
	Total Training Size (in 10^3 frames)		102.6	101.3	94.6	100.5	99.0	93.7	97.6
	Total Validation Size (in 10^3 frames)		793.9	829.6	871.5	682.0	898.4	842.0	916.4
			Final Test (Matches 07 and 29)						
			Total Test Size (in 10^3 frames)						456.9

The training sets were submitted to a majority-class undersampling process [40], aiming to mitigate possible class-imbalance effects, due to the much larger number of non–highlights than highlights (inferior to 1%). By this process, approximately the same number of class instances were destined to train the classifiers. Table 2 summarizes the matches and the number of frames integrating each fold, besides the size of each training (TR) and validation (VA) sets for each trial.

5. Experimental results

The figure of merit for hyperparameter tuning was the false-positive rate (FPR) [30], averaged over all validation sets. The FPR values reported here, denoted as $\text{FPR}_{97\%}$, assumed a decision threshold correspondent to a true-positive rate (TPR) equal to 97%, related to a practical operational system setting. Models were compared utilizing the Receiver Operating Characteristics (ROC) curves [30].

The definition of system hyperparameters related to time-aggregation, frame-selection, and decision making exploited a process named here as greedy cross-validation (CV). In this process, the hyperparameters were optimized in a nested sequence, defined according to our convenience, avoiding the computation burden related to any grid-search alternative.

5.1. Feature aggregation

For simplicity, experiments assumed symmetrical windows ($l = l_l = l_u$) with half-length l equal to $1, 2, 4, 6, 8, 10$, and 12 s. For all, windows were composed of $U = 61$ frames (30 past + the current + 30 future), since the frame rate is 30 frames/s, and the subsampling factor k was increased accordingly to the window length, as described in Section 3.2.

Fig. 13 depicts the $\text{FPR}_{97\%}$ values obtained in each case, including the results of one experiment assuming a window composed of just one frame (0s). The inclusion of feature time-dynamics improved system performance, significantly reducing the $\text{FPR}_{97\%}$ value. The models associated with half-window lengths of 8 to 12 s exhibited similar performance, which corroborates with practice, as football match highlights often last for about 6s.

5.2. Feature selection

Feature selection considered the subsampling factor p equal to 4. Frame relevance inference exploited the Gentle AdaBoost [2], assuming a total of forty trees, each one with three-levels, defined in accordance with some trials.

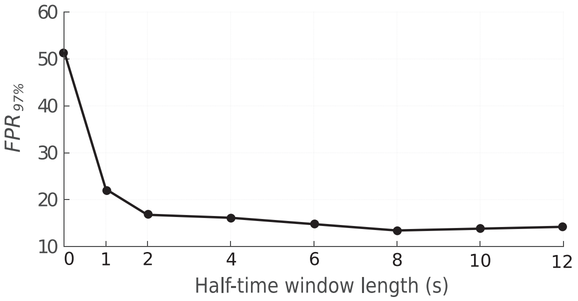

Fig. 13. Values of $\text{FPR}_{97\%}$ for different window lengths L_i.

Motivated by the previous results, the following experiments included symmetric windows whose half-length was made equal to $l = 8$ and $l = 12$ seconds, and compared the proposed frame selection approach with a standard uniform-frame subsampling (in time). Table 3 summarizes the results, assuming windows with $U = 61$ frames. Notably, the proposed method outperforms the uniform-subsampling in both scenarios. It also seems to better exploit the information that is exclusive to the longer (12s) window, considering the higher drop in $FPR_{97\%}$ observed in this case.

By fixing on windows with half-length equal to $l = 12$, the subsequent experiments compared the uniform-sampling solutions, including $U = 61$ and $U = 121$ frames, with the proposed frame selection method, assuming $U = 61$ frames. Fig. 14 exhibits the ROC curves for TPR values ranging from 97% to 99%. Results confirmed the better performance of the proposed frame selection method.

Finally, adopting $l = 12$, simulations varied the number U of frames in the range of 31 to 121. Table 4 summarizes the results, indicating an optimal number of $U = 61$ frames.

5.3. Classifier definition

For implementing the set of classifiers analysed in this work, we used the following toolboxes: MATLAB Statistics and Machine Learning Toolbox [41] (*k*NN, SVM, RF), GML Adaboost Matlab Toolbox [42] (AdaBoost), and the basic ELM code [43] (ELM).

Previously to classifier training and evaluation, data was preprocessed such as all features were normalized to zero mean and unit variance (z-score), except the energy features $e_{st}(m)$ and $e_{cs}(m)$, which were log-normalized before having their z-score computed.

Classifier hyperparameters were tuned by CV, considering $TPR_{97\%}$ as the figure of merit. Table 5 summarizes the range of hyperparameters evaluated for each method and those identified as the best by CV. The hyperparameters not covered in this table were left as default, except for SVM, whose corresponding toolbox has settled the kernel width automatically.

Table 3
Comparison of the proposed frame selection method with ordinary uniform subsampling for different window lengths.

	$FPR_{97\%}$ (%)	
Half-window length	8*s*	12*s*
Uniform subsampling	13.4	14.2
Proposed method	12.5	**11.6**

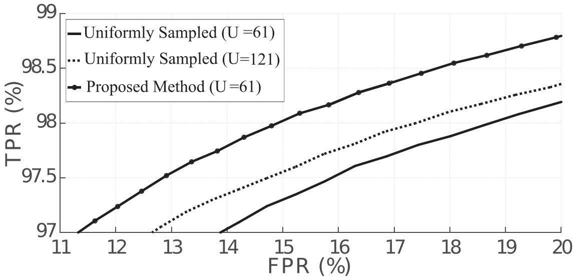

Fig. 14. Comparison of the proposed frame selection method with uniform-frame subsampling, assuming a different number of frames and windows with half-length of 12s.

Table 4
Values of $FPR_{97\%}$ by setting a different number U of frames in the proposed approach.

U	$FPR_{97\%}$(%)
31	13.2
41	13.5
51	13.2
61	**11.6**
71	13.0
81	12.9
91	11.7
121	12.8

Table 5
Hyperparameters (range and choice) exploited in the cross-validation experiments, considering a range of classification algorithms (see text).

Classifiers	Parameter	Range Tested	Chosen Value
Adaboost	Split (S) - Number of Trees (N_T)	$1 \leqslant S \leq 7$; $10 \leqslant N_T \leqslant 150$	$S = 3$; $N_T = 100$
ELM	Number of Hidden Neurons (N_H)	$80 \leqslant N_H \leqslant 200$	$N_H = 150$
kNN	Number of Neighbors (k)	$1 \leqslant k \leqslant 21$	$k = 15$
Random Forest	Number of Trees (N_T)	$50 \leqslant N_T \leqslant 200$	$N_T = 120$
SVM	Kernel Function	Gaussian, Linear and Polynomial of 2nd, 3rd, and 4th orders	Gaussian

Methods were compared by ROC curves, as depicted in Fig. 15. RF achieved the best performance, followed by SVM, and AdaBoost, respectively.

5.4. Voting Filter

Experiments for defining the size of the decision window assumed $k = k_i = k_u$ (symmetrical window) and values for k in the range of 1 to 121 frames. Fig. 16 exhibits the corresponding $\text{FPR}_{97\%}$ values. The use of larger windows has led to a reduction in the $\text{FPR}_{97\%}$ values, as expected, with $U = 61$ frames performing best.

5.5. Final System Evaluation

Aiming to evaluate the effectiveness of each design stage in the proposed approach, Table 6 summarizes the TPR97% values (averaged over validation sets), considering five processing schemes (P1 to P5), each one including a given design procedure in a greedy fashion. Note that when adopting uniformly time-sampled frames (61, in total), integrating a time-interval of ±12s around the current frame (P2), $\text{FPR}_{97\%}$ drops approximately third-seven percentage points. When considering the proposed irregular frame-sampling mechanism replacing the uniform sampling, this gain increases by almost three percentage points (P3). The use of a RF classifier adds more six percentage points (P4). Finally, the inclusion of a voting window (P5) leads to an additional increase of 0.7%. The median and interquartile range (IQR) of $\text{FPR}_{97\%}$ values observed to models following the P5 scheme were 4.2 and 1.7, respectively.

The subsequent analysis focused solely in the model associated with the lowest validation set $\text{FPR}_{97\%}$ (trial 2), selected from those produced by the sevenfold cross-validation process, considering the processing scheme P5.

Table 7 summarizes the performance attained by this model to each match. Matches integrating the validation (8, 14, 21, and 23) and test sets (07 and 26) are in bold. Training matches were not undersampled for this analysis. This table includes

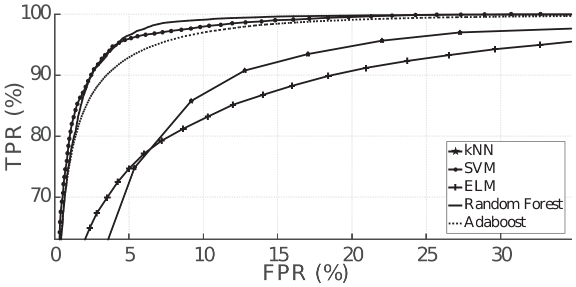

Fig. 15. ROC curves related to different classification techniques (see text).

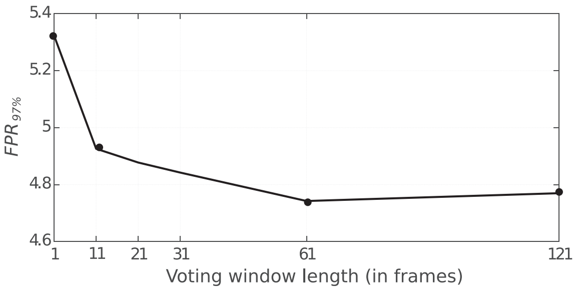

Fig. 16. Values of $\text{TPR}_{97\%}$ for different decision window lengths.

Table 6
Effectiveness of the proposed approach - values of TPR97% - assuming different design choices (see text).

Design choices	Model				
	P0	**P1**	**P2**	**P3**	**P5**
Time window length					
Absent	X				
±12*s*		X	X	X	X
Frame sampling process					
Uniformly spaced		X			
Irregular wrapper-probabilistic approach			X	X	X
Classifier					
Adaboost	X	X	X		
Random Forest				X	X
Voting Filter					
None	X	X	X	X	
2*s*-long voting filter					X
$\text{FPR}_{97\%}(\%)$	51.5	14.2	11.3	5.4	4.7

Table 7
System performance in terms of False Positive Rate (FPR), True Positive Rate (TPR), and Summarization Ratio (SR) for each entire match.

Match #	FPR(%)	TPR (%)	SR (%)	Match #	FPR (%)	TPR (%)	SR (%)
01	5.7	100.0	7.2	18	5.5	100.0	6.5
02	3.6	100.0	5.1	19	2.9	100.0	4.7
03	4.9	100.0	5.9	20	3.4	100.0	4.9
04	5.5	100.0	6.4	**21**	**4.4**	**95.6**	**5.4**
05	5.2	100.0	6.4	22	3.4	100.0	5.2
06	5.4	100.0	6.8	**23**	**4.7**	**99.0**	**5.8**
07	**1.9**	**97.4**	**2.8**	24	4.3	100.0	5.5
08	**6.5**	**99.9**	**7.7**	25	6.7	100.0	7.3
09	1.8	100.0	2.6	**26**	**4.6**	**100.0**	**5.7**
10	5.0	100.0	6.9	27	14.9	100.0	15.5
11	1.7	100.0	2.1	29	11.0	100.0	11.9
12	4.0	100.0	5.8	28	1.5	100.0	2.8
13	2.6	100.0	4.4	30	12.1	100.0	13.0
14	**3.1**	**99.4**	**3.5**				
15	2.2	100.0	3.6	Mean	4.8	99.7	6.0
16	4.2	100.0	6.0	Median	4.4	100.0	5.7
17	1.8	100.0	2.6	IQR	2.6	0.0	2.3

the here denoted Summarization Ratio index (SR), defined as the proportion of highlight frames over all match frames. The average SR was only 6%, meaning that an excerpt with 5.7 min of duration can summarize a regular 90-min match. As this model attained an average TPR of 99.7%, no highlight is lost in practice. We should also mention that the high TPR values associated with the low false-alarm rates observed in this table were only made possible by our system considering highly discriminative features extracted from a concise set of frames integrating long-term windows, a strategy efficiently conjugated with an ensemble decision making process. Nonetheless, by integrating such a tool in daily routine, operator's productivity would significantly increase, as now in charge of the much easier task of just refining highlights automatically preselected by the system.

A more practical quality assessment measure would be the here denoted Operational True Positive Rate (OTPR), which corresponds to the probability of a professional operator in not missing any significant highlight when just watching the summarized match version. For this measure, we arbitrarily assumed that a highlight would be identified in practical settings when at least 10% of the number of frames integrating its corresponding window are positively detected. A remarkable result is that the proposed system attains an OTPR equal to 100% for all matches. Nonetheless, operators can quickly settle the limits of the highlights more precisely to meet more specific task demands.

Fig. 17 exhibits some screenshots taken from the system during a match sequence ending in a goal. In the bottom, a highlight bar provides an indicator proportional to the number of MFCMs pointing out a highlight. As the final decision relatively whether the scene is a highlight or not considers majority voting over all MFCMs, as soon as this bar exceeds the half-scale level, a highlight is automatically identified, and the frame number corresponding to the start of this highlight is displayed. Similarly, when the system detects the end of a highlight, the corresponding frame number is also exhibited. All highlights identified and their related information are stored in a database for subsequent operator processing and evaluation.

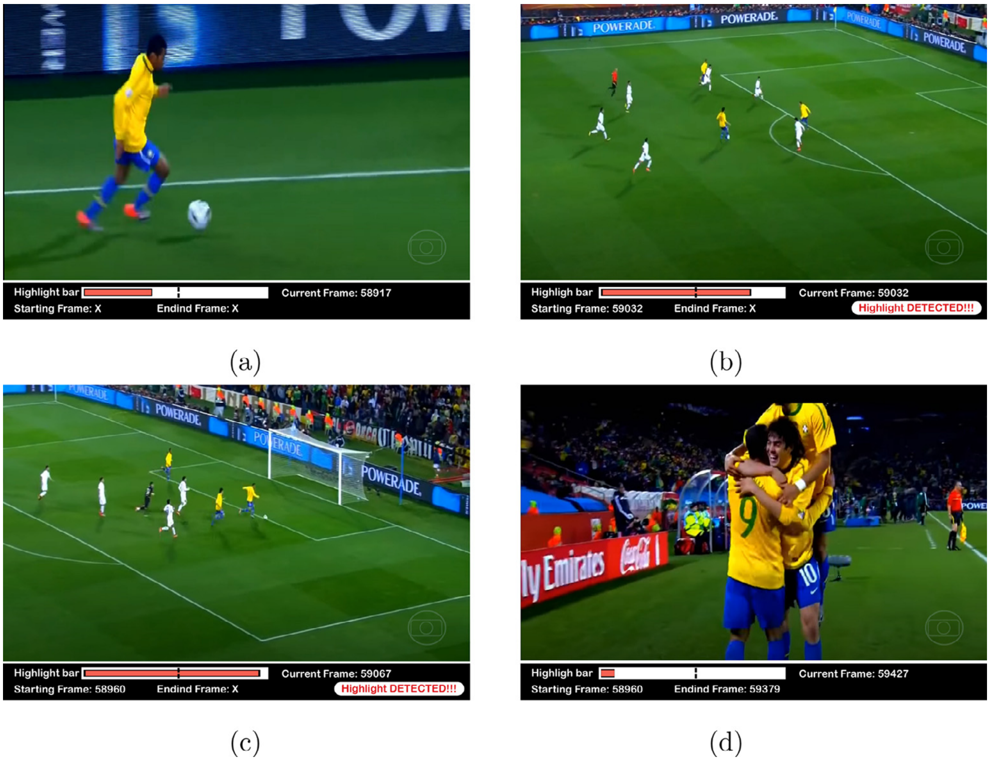

Fig. 17. Example of a highlight sequence being identified by the proposed system: (a) Start of the passing sequence until goal; (b) Last pass before goal shot; (c) Goal shot; (d) Players' celebration after scoring a goal.

Table 8
Performance and dataset size comparison of the proposed and literature methods.

	GDR (%)	TPR or Recall (%)	Number of highlights	Other results	Number of Championships (and games) / Video hours	Number of features	Classifier
Our		Work	100.0	99.7	282	FPR = 4.8%; SR = 6.0%	5 (30)/ 58 h
16	RF (also kNN, SVM, ELM,	Random Forest)					
[44]	-	94.9	108	Precision = 86.0%	1 (5)/ 5 h[2]	NR[3]	Proprietary algorithm
[14]	-	45.7	NR	Precision = 47.0%	2 (20)/ 30 h	1024	LSTM
[24]	-	94.3[4]	800	-	NR	4096	CNN
[6][5]	71.4	68.3[6]	1824	Precision = 79.1%	2 (48)/ 73 h	39	Temporal confusion NN
[46]	84.6	-	142	-	1 (10)/ NR	NR	SVM
[23]	91.9	91.4[7]	256	Precision = 94.3%	3 (30)/ 45 h	512	CNN
[13]	-	94.5	91	Precision = 92.5%	2 (23)/ 31 h	17	Decision tree
[45]	-	83.5	643	Precision = 67.4%	10 (13)/ $\approx 19h$	19	E-HMM
[26]	-	61.5	185	Precision = 44.0%	1 (69)/ $\approx 103h$	NR	kNN

[2] Number of games/hours used for testing. It does not mention the use of training.
[3] Not Reported.
[4] Average of goal attempts and shoot related values.
[5] Considering only the World Cup 2010 set.
[6] Average of shooting and scoring related values.
[7] Metrics related to goal attempts only.

5.6. Comparison with Other Works

Comparing the proposed solution with the literature is difficult, especially due to differences on datasets, objectives, and highlight concepts. Nonetheless, Table 8 provides some quantitative measures to position our proposal regarding the state-of-the-art, focusing on problem coverage, feature/classification issues, and performance achievements. Concerning the last aspect, this table reports the most commonly adopted metrics: the goal-detecting rate (GDR) and true positive rate (TPR), as well as includes additional indexes for some cases, according to the original reference availability. In what concerns the problem coverage, our work is distinguished by the following aspects:

1. the number of championships included, since most works restrict to WorldCups [44,13], WorldCups plus some regional championship [6], or solely regional championships [14,26]. A notable exception is [45];
2. the number of matches, which is among the highest, being only surpassed by [6,26];
3. the number of highlight excerpts, although the references [6,24,45] include more highlights than our work;
4. the number of evaluation hours, which is among the top four [6,23,26].

Nevertheless, some points regarding the methods in the references mentioned above should be emphasized: the reference [6] exploits speech recognition by acoustic and language models, which tends to be less robust and more computationally complex; the reference [45] includes a replay detection mechanism, which may restrict the domains of system application, and the reference [24] considers a dataset composed by short match excerpts (10s) that are strongly positively biased (75% of the excerpts are highlights).

Relatively to the number of matches included in this study, it is comparable to one of most state-of-the-art works reported in Table 8 [13,14,23], and it is large enough to account for a reasonable variability regarding tournaments, clubs, stadiums, and lighting conditions, as discussed in Section 4.1. Besides, in opposition to previous works that typically use short game excerpts, this work innovates by exploiting full-length matches when evaluating system performance. In that sense, the adoption of full matches enabled a better emulation of the operational scenarios typically expected for such a system, leading to more realistic performance estimates. Overall, this database encompasses approximately 6.3 million frames, 50.7 thousand of which are annotated as highlights, such that each game has an average of 210 thousand frames, and each highlight has an average of 1.7 thousand frames. Therefore, concerning the results summarized in Table 7, the FPR values considered an average evaluation of (50.7–1.7) = 49.0 thousand frames, while the TPR values assumed an average evaluation of 1.7 thousand frames, yielding to FPR and TPR estimates with very low-variance.

Regarding system characteristics, our work involves the fewest number of features (multimodal and low-level), especially when compared to deep-learning alternatives [14,23,24]. This characteristic may contribute to a better model generalization in other operational scenarios, such as different matches, championships, narrator, and broadcast conditions. Additionally, our work considered a range of classifier alternatives, differing from most competing proposals that only evaluated one classification model. Moreover, many highlight classification windows [24,26,45] are short-term (duration inferior to 10s); one exception is the work in [14] (30s classification windows). Relatively to conducting or not feature selection, the only other work besides ours that exploits this resource is in [13], but it exploits a filter-based approach. We should stress out that wrapper methods, such as the one used in this work, often outperforms filter-based techniques. It is also noteworthy that, bearing in mind the limitations in this analysis, our system surpasses all competitors in TPR value, attaining also a reduced FPR .

6. Conclusions

This paper describes a complete and robust framework for highlight detection in football broadcasts using audio/video descriptors (multimodal approach). The proposed system accurately identifies a highlight by producing highly discriminating features, extracted from a sparse subset of irregularly time-sampled frames integrating long-term windows, which are submitted to a multi-frame classification approach. A feature selection scheme based on a non-parametric probabilistic frame relevance modelling, adequate to slowly-varying features, is proposed. Experimental results, exploiting a comprehensive 30-match database, each one with an average duration of about 90 min, indicate a compression ratio of 94%, as well as a correct highlight identification of 100%, including all annotated goals and significant goal attempts.

The present work has not considered games played under any form of snowy weather. Therefore, the developed system is not necessarily reliable in these conditions. Future works might consider addressing such liability by expanding the annotated database to include matches played under snow. In this scenario, the development of an updated version of this highlight system can greatly benefit from the design methodology proposed here.

CRediT authorship contribution statement

Carolina L. Bez: Methodology, Software, Validation, Formal analysis, Investigation, Writing - original draft. **João B.O. Souza Filho:** Conceptualization, Methodology, Writing - review & editing, Supervision, Project administration. **Luiz G.L.B. M. de Vasconcelos:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation. **Thiago Frensch:** Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation.

Eduardo A.B. da Silva: Conceptualization, Methodology, Writing - review & editing, Supervision, Project administration, Funding acquisition. **Sergio L. Netto:** Conceptualization, Methodology, Writing - review & editing, Supervision, Project administration, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001. Professors E.A.B. da Silva and Sergio L. Netto are recipients of a career award from the CNPq and FAPERJ.

References

[1] H. Shih, A survey of content-aware video analysis for sports, IEEE Trans. Circuits Syst. Video Technol. 28 (2018) 1212–1231, https://doi.org/10.1109/TCSVT.2017.2655624.

[2] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed., Springer, 2009.

[3] N. Nguyen, A. Yoshitaka, Soccer video summarization based on cinematography and motion analysis, in: Proceedings of IEEE International Workshop on Multimedia Signal Processing, 2014, pp. 1–6. https://doi.org/10.1109/MMSP.2014.6958804.

[4] M.H. Kolekar, S. Sengupta, Bayesian network-based customized highlight generation for broadcast soccer videos, IEEE Trans. Broadcast. 61 (2015) 195–209, https://doi.org/10.1109/TBC.2015.2424011.

[5] J. Assfalg, M. Bertini, A. Del Bimbo, W. Nunziati, P. Pala, Detection and recognition of football highlights using HMM, in: Proceedings of International Conference on Electronics, Circuits and Systems, volume 3, 2002, pp. 1059–1062. https://doi.org/10.1109/ICECS.2002.1046433.

[6] N.M. Pham, Q.H. Vu, Temporal confusion network for speech-based soccer event retrieval, in: Proceedings of International Conference on Advanced Technologies for Communications, 2013, pp. 549–553, https://doi.org/10.1109/ATC.2013.6698176.

[7] M. Tavassolipour, M. Karimian, S. Kasaei, Event detection and summarization in soccer videos using Bayesian network and copula, IEEE Trans. Circuits Syst. Video Technol. 24 (2014) 291–304, https://doi.org/10.1109/TCSVT.2013.2243640.

[8] A. Raventós, R. Quijada, L. Torres, F. Tarrés, E. Carasusán, D. Giribet, The importance of audio descriptors in automatic soccer highlights generation, in: Proceedings of IEEE International Multi-Conference on Systems, Signals Devices, 2014, pp. 1–6. https://doi.org/10.1109/SSD.2014.6808845.

[9] A. Ekin, A.M. Tekalp, R. Mehrotra, Automatic soccer video analysis and summarization, IEEE Trans. Image Process. 12 (2003) 796–807, https://doi.org/10.1109/TIP.2003.812758.

[10] H.-S. Chen, W.-J. Tsai, A framework for video event classification by modeling temporal context of multimodal features using HMM, J. Vis. Commun. Image Represent. 25 (2014) 285–295, https://doi.org/10.1016/j.jvcir.2013.12.001.

[11] S. Jai-Andaloussi, A. Mohamed, N. Madrane, A. Sekkaki, Soccer video summarization using video content analysis and social media streams, in: Proceedings of IEEE/ACM International Symposium on Big Data Computing, 2014, pp. 1–7, https://doi.org/10.1109/BDC.2014.20.

[12] D. Tran, J. Yuan, D. Forsyth, Video event detection: From subvolume localization to spatiotemporal path search, IEEE Trans. Pattern Anal. Mach. Intell. 36 (2014) 404–416, https://doi.org/10.1109/TPAMI.2013.137.

[13] Y. Yang, S. Chen, M. Shyu, Temporal multiple correspondence analysis for big data mining in soccer videos, in: Proceedings of IEEE International Conference on Multimedia Big Data, 2015, pp. 64–71, https://doi.org/10.1109/BigMM.2015.88.

[14] M. Sanabria, Sherly, F. Precioso, T. Menguy, A Deep architecture for multimodal summarization of soccer games, in: Proceedings of the 2Nd International Workshop on Multimedia Content Analysis in Sports, MMSports '19, ACM, New York, NY, USA, 2019, pp. 16–24. https://doi.org/10.1145/3347318.3355524.

[15] W. Xu, Z. Miao, J. Yu, Q. Ji, Action recognition and localization with spatial and temporal contexts, Neurocomputing 333 (2019) 351–363, https://doi.org/10.1016/j.neucom.2019.01.008.

[16] T. Wang, C. Liu, L. Wang, Action recognition by latent duration model, Neurocomputing 273 (2018) 111–119, https://doi.org/10.1016/j.neucom.2017.07.057.

[17] Y. Yi, Z. Zheng, M. Lin, Realistic action recognition with salient foreground trajectories, Expert Syst. Appl. 75 (2017) 44–55, https://doi.org/10.1016/j.eswa.2017.01.008.

[18] Q.-Q. Chen, Y.-J. Zhang, Cluster trees of improved trajectories for action recognition, Neurocomputing 173 (2016) 364–372, https://doi.org/10.1016/j.neucom.2015.03.124.

[19] G. Varol, A.A. Salah, Efficient large-scale action recognition in videos using extreme learning machines, Expert Syst. Appl. 42 (2015) 8274–8282, https://doi.org/10.1016/j.eswa.2015.06.013.

[20] Z. Zheng, G. An, D. Wu, Q. Ruan, Spatial-temporal pyramid based convolutional neural network for action recognition, Neurocomputing 358 (2019) 446–455, https://doi.org/10.1016/j.neucom.2019.05.058.

[21] Y. Yuan, Y. Zhao, Q. Wang, Action recognition using spatial-optical data organization and sequential learning framework, Neurocomputing 315 (2018) 221–233, https://doi.org/10.1016/j.neucom.2018.06.071.

[22] S. Bai, Growing random forest on deep convolutional neural networks for scene categorization, Expert Syst. Appl. 71 (2017) 279–287, https://doi.org/10.1016/j.eswa.2016.10.038.

[23] H. Jiang, Y. Lu, J. Xue, Automatic soccer video event detection based on a deep neural network combined CNN and RNN, in: Proceedings of IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI), 2016, pp. 490–494, https://doi.org/10.1109/ICTAI.2016.0081.

[24] M. Z. Khan, S. Saleem, M. A. Hassan, M. Usman Ghanni Khan, Learning Deep C3D features for soccer video event detection, in: Proceedings of 14th International Conference on Emerging Technologies (ICET), 2018, pp. 1–6. https://doi.org/10.1109/ICET.2018.8603644.

[25] M. Merler, K.-N.C. Mac, D. Joshi, Q.-B. Nguyen, S. Hammer, J. Kent, J. Xiong, M.N. Do, J.R. Smith, F.R. Schmidt, Automatic curation of sports highlights using multimodal excitement features, IEEE Trans. Multimedia 21 (2018) 1147–1160, https://doi.org/10.1109/TMM.2018.2876046.

[26] T. Decroos, V. Dzyuba, J. V. Haaren, J. Davis, Predicting soccer highlights from spatio-temporal match event streams, in: Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, USA, 2017, pp. 1302–1308.

[27] E. Alves, J.B.S. Filho, A. Kritski, An ensemble approach for supporting the respiratory isolation of presumed tuberculosis inpatients, Neurocomputing 331 (2019) 289–300, https://doi.org/10.1016/j.neucom.2018.11.074.

[28] L. Onofri, P. Soda, M. Pechenizkiy, G. Iannello, A survey on using domain and contextual knowledge for human activity recognition in video streams, Expert Syst. Appl. 63 (2016) 97–111, https://doi.org/10.1016/j.eswa.2016.06.011, http://www.sciencedirect.com/science/article/pii/S0957417416302913..

[29] Z. Wei, X. Yang, A novel soccer video summarization model based on video time density function, International Journal of Digital Content Technology and its Applications 6 (2012) 248–256.

[30] S. Theodoridis, Machine Learning: A Bayesian and Optimization Perspective, Academic Press, Orlando, USA, 2015.
[31] S. Ding, H. Zhao, Y. Zhang, X. Xu, R. Nie, Extreme learning machine: Algorithm, theory and applications, Artif. Intell. Rev. 44 (2015) 103–115, https://doi.org/10.1007/s10462-013-9405-z.
[32] P.S.R. Diniz, E.A.B. da Silva, S.L. Netto, Digital Signal Processing - System Analysis and Design, 2nd ed., Cambridge University Press, Cambridge, UK, 2010.
[33] F. Coldefy, P. Bouthemy, Unsupervised soccer video abstraction based on pitch, dominant color and camera motion analysis, in: Proceedings of Annual ACM International Conference on Multimedia, New York, NY, USA, 2004, pp. 268–271. https://doi.org/10.1145/1027527.1027588.
[34] D. Rocchesso, Introduction to Sound Processing, Universita di Verona (2003).
[35] A.K. Jain, Fundamentals of Digital Image Processing, Prentice-Hall, Upper Saddle River, USA, 1989.
[36] R.C. Gonzalez, R.E. Woods, Digital Image Processing, 4th ed., Prentice-Hall, Upper Saddle River, USA, 2018.
[37] H.R. Myler, A.R. Weeks, The Pocket Handbook of Image Processing Algorithms in C, Prentice Hall, Upper Saddle River, USA, 2009.
[38] A. Ekin, Sports Video Processing for Description, Summarization and Search, Ph.D. thesis, The University of Rochester, Eastman School of Music, 2004.
[39] J. Owens, Television Sports Production, 5th ed.,., Focal Press, 2015.
[40] J.L. Leevy, T.M. Khoshgoftaar, R.A. Bauder, N. Seliya, A survey on addressing high-class imbalance in big data, Journal of Big Data 5 (2018) 42, https://doi.org/10.1186/s40537-018-0151-6. 10.1186/s40537-018-0151-6.
[41] MATHWORKS, Statistics and machine learning toolbox, https://www.mathworks.com/products/statistics.html, Accessed on 19/06/2017.
[42] A. Vezhnevets, GML adaboost matlab toolbox, http://research.graphicon.ru/machine-learning/gml-adaboost-matlab-toolbox.html, Accessed on 19/06/2017.
[43] Q. Zhu, G. Huang, Basic ELM algorithms, http://www.ntu.edu.sg/home/egbhuang/elm_codes.html, Accessed on 19/06/2017.
[44] H. Liu, Highlight extraction in soccer videos by using multimodal analysis, in: Proceedings of International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery, 2017, pp. 2169–2173. https://doi.org/10.1109/FSKD.2017.8393107.
[45] Z. Wang, J. Yu, Y. He, T. Guan, Affection arousal based highlight extraction for soccer video, Multimedia Tools and Applications 73 (2014) 519–546. URL 10.1007/s11042-013-1619-1. https://doi.org/10.1007/s11042-013-1619-1.
[46] R. Sharma, V. Gandhi, V. Chari, C. Jawahar, Automatic analysis of broadcast football videos using contextual priors, Signal, Image and Video Processing 11 (2016), https://doi.org/10.1007/s11760-016-0916-3.