Applied Intelligence (2019) 49:2899–2917
https://doi.org/10.1007/s10489-019-01410-x

# Replay and key-events detection for sports video summarization using confined elliptical local ternary patterns and extreme learning machine

Ali Javed[1] · Aun Irtaza[2] · Yasmeen Khaliq[3] · Hafiz Malik[4] · Muhammad Tariq Mahmood[5]

Published online: 12 February 2019

**Abstract**

Sports broadcasters generate enormous amount of video content viewed all over the world. To capture the user interests in the rebroadcasted content, the sports videos are summarized that need the manual inspection and analysis. However, the huge repository and long duration of videos make manual analysis and summarization a laborious and time-consuming job. To overcome this problem, efforts have been made for automatic video summarization. In this paper, a novel framework to summarize sports videos is presented. It has been observed that the replays within a sports video represent key-events and these events can be used for video summarization. It has been noted that replays are usually sandwiched between start and stop of gradual-transitions. A thresholding-based approach is used to identify gradual transition effect (i.e. fade-in, fade-out) in sports video. The Gaussian mixture model (GMM) is then applied to key-event candidates to extract silhouettes and generate motion history image (MHI) for each key-event. The MHIs are processed using Confined Elliptical Local Ternary Patterns (CE-LTPs) for feature extraction. Extreme learning machine (ELM) classifier is used to learn the underlying model for events. A trained ELM-based classifier is then used for key-event detection. The output of classifier is then used for key-event labeling, replay detection, and complete game summarization. Performance of the proposed framework is evaluated on a dataset consisting of 20 videos of four different sports. Experimental results indicate the effectiveness of the proposed framework in terms of replays and key-events detection from selected dataset.

**Keywords** Gradual transition · Highlights · Replay detection · Events detection

## 1 Introduction

There has been a lot of exponential growth of multimedia content production, and sharing spark research activities in the domains of video content analysis, management, and summarization. Video summarization methods analyze the input videos and generate a concise video of significant events. A large number of applications i.e. surveillance [1], healthcare [2], media [3], and life logging [4] take significant benefits from video summarization. Another area that take significant advantages from video summarization is sports [5, 6] due to large video collections, and massive viewership interested in significant events of the game. The transmission over the sports broadcasting channels also impose time constraints that creates a need to summarize the long duration sports videos. The summarized sports videos are also used for tournament promotions by gaining more viewership and sponsors that have significant impact over the revenue of the broadcasters. However, to generate the concise videos comprising key-events of the game need manual inspection of the long duration videos that is a laborious job. Therefore, the automatic sports video summarization is a significant requirement for broadcasting companies to facilitate the human agents. Existing state-of-the-art on sports video summarization has mainly focused on key-events detection [7–11], replays detection [6, 12–15], video indexing [16, 17], and shot classification [7, 8].

The key-events detection methods automatically discover significant moments in a sports video and generate the game highlights. These methods can be categorized as either non-replay based [7–9] or replay [6, 12–15] based methods. In non-replay based sports video summarization domain, Mao et al. [8] used textual cues, shot transition information, and Bayesian Belief Networks (BBN) to propose an event classifier for baseball. Similarly, Tavassolipour et al. In [14], proposed a BBN-based architecture to detect seven different key-events in soccer videos. Hue histogram difference

✉ Muhammad Tariq Mahmood
tariq@koreatech.ac.kr

Extended author information available on the last page of the article.

comparison was employed for shot boundary detection. Shot classification was performed by analyzing the size of players against the specified threshold values. Finally, Bayesian Network was employed for key-events detection and summarization in soccer videos. In [5], Javed et al. proposed a framework based on audio-visual cues for key-events detection and summarization of cricket videos. Similarly, in [9], Kolekar et al. proposed an automated method based on audio-visual cues to generate the highlights of soccer video broadcasts. Short time audio energy features were used to detect the excited clips. In the next phase, scenes in the excited clips were classified into events. A Bayesian belief network (BBN) was applied to assign semantic labels to the excited clips i.e. goals, fouls, saves, kicks, etc. in soccer videos. Finally, these labeled clips were selected to generate the summarized video.

Replays repeat the key-events of a game in slow motion. During live broadcasting of sports events, replays are frequently used to highlight the occurrence of key-events, which is the motivation behind using the replays for sports video summarization. Replay based methods consider the replays as independent game summaries [12–15, 18]. In literature, replay detection is usually performed through the frame motion movements [12, 13, 19] or frame logo transitions [20–24] detection. In [12], Pan et al. used Hidden Markov model (HMM) and Viterbi algorithm to analyze frame motion movements for replay detection. Duan et al. used motion features to develop a logo detection technique based on mean shift [15]. The main drawback of the motion-movement based methods is that these methods significantly depends upon the homogeneity of replay speeds and fail when replay speed variates. Zhao et al. used SURF descriptor to detect the logo patterns in sports videos [25]. Similarly, logo detection has been performed in [20, 22, 26] using the statistical features for replay detection. The similar approach of logo transition has also been adopted in [27–30]. However, performance of these methods depends upon the accuracy of logo detection process that is a challenging task due to variations in the shape, color, design, size, and placement of logos amongst different broadcasters and sports. Moreover, logo transitions based replay detection methods have dependency on the replay structure. The performance of these methods is expected to degrade for multiple sports because the structure and representation of replays vary among various sports.

Replay detection based video summarization methods suffer from various limitations i.e. computational complexity of logo frame detection, differentiating between logos of various sizes, design, and placement, replay speed variations, frame transition variations (i.e. abrupt or gradual), etc. In addition, the research has overlooked further analysis of the detected replay frames. However, our findings suggest that the analysis of detected replays have multi-dimensional benefits for the broadcasters i.e. replay categorization in terms of key-events, game-events association (e.g. facilitating broadcasters to organize replays of specific events in entire tournament), key-event categorization (e.g. best catch, best drive, best goal of the event), etc.

Therefore, the purpose of this research is to present a robust mechanism for replay-event detection and categorization. For replay detection, we applied a dual-threshold based method to detect the gradual transition (GT) segments. The video frames between two successive GT segments are identified as replay frames. Replay events categorization is performed in the subsequent step. For this purpose, the Gaussian mixture model (GMM) is applied to extract the silhouettes from replay segment frames followed by generating a motion history image (MHI) by combining the extracted silhouettes. Each MHI is then represented through a novel confined elliptical local ternary patterns (CE-LTPs) descriptor that are used to train the ELM classifier for key-events detection. In the proposed work, a novel approach is presented for key-events detection through processing the replay frames as the existing systems have never used replay segments to classify various key-events in sport videos. The proposed replay detection approach is robust against the limitations of existing replay detection systems i.e. camera variations, replay speed, computational complexity of logo frame detection, logo design, size and placement, etc. The performance of the proposed framework is evaluated on a dataset of four different sports categories i.e. Cricket, Tennis, Baseball, and Basketball. Experimental results signify the effectiveness of the proposed framework in terms of replay and key-events detection. The main contributions of the proposed research work are summarized as:

- A novel key events detection method for sports videos is presented in this paper that exploits the replay frames for key-event detection.
- For video frame representation a novel feature extraction approach referred to as confined elliptical local ternary patterns is presented that is particularly designed for the sports videos.
- The potential of ELM classifier is highlighted for the analysis of visual content of the sports videos is still an unexplored topic in the domain of sports video summarization.

The rest of the paper is organized as follows: Section 2 presents the detailed framework of the proposed system. Section 3 presents the comprehensive discussion on the results and various experiments that are designed to evaluate the performance of the proposed system. Finally, Section 4 presents the conclusion of the proposed work.

## 2 Proposed method

The architecture of the proposed framework is presented in Fig. 1. The proposed framework consists of two main steps: replay segment extraction and key-events detection.

### 2.1 Replay segment extraction

Replay segments in sports videos are usually contained between the frames comprising the gradual transition (GT) effects such as *wipes, dissolves, fade-in/out*, etc. The characteristics of multiple GTs thus can be used to identify the boundaries of a replay segment. However, the challenge in GT detection is that the histogram difference between consecutive frames is very small compared to an abrupt transition where a significant difference occurs between the frames of two shots. Hence, that makes it difficult to detect the GTs for replay detection. Additionally, the accumulative histogram difference of the first transition frame and the consecutive frames grows gradually to an abrupt transition level (i.e. $T_U$). Accumulative histogram difference represents the aggregate of all the consecutive differences (i.e. value of successive histogram difference between frames exceeds the threshold, $T_L$) computed between the consecutive frames as shown in Fig. 2.

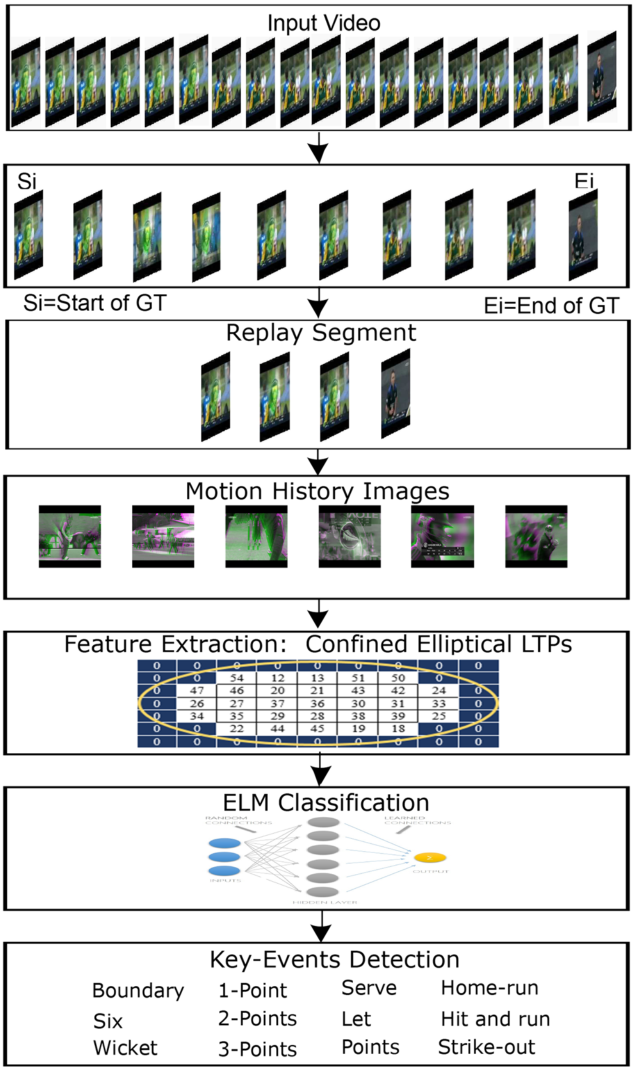


**Fig. 1** Architecture of the proposed framework

In the proposed work, accumulative histogram difference is used to detect the GT frames. For this purpose, in the first step histogram difference of luminance component of consecutive frames is computed as follows.

$$D_i = \sum_{j=1}^{B} |H_i(j) - H_{i+1}(j)| \tag{1}$$

Where $D_i$ represents the obtained difference between the histograms of consecutive frames. $H_i(j)$ and $H_{i+1}(j)$ are the histograms of the $i_{th}$ and $(i+1)_{th}$ frames with total number of $B$ bins. The value of $B$ in our experiments is 256. However, different values of $B$ can also be explored; but as in our case $B = 256$ provides the satisfactory results therefore, we have not emphasized on experimentation with different values of $B$. The accumulative histogram difference represents the aggregate of all the histogram

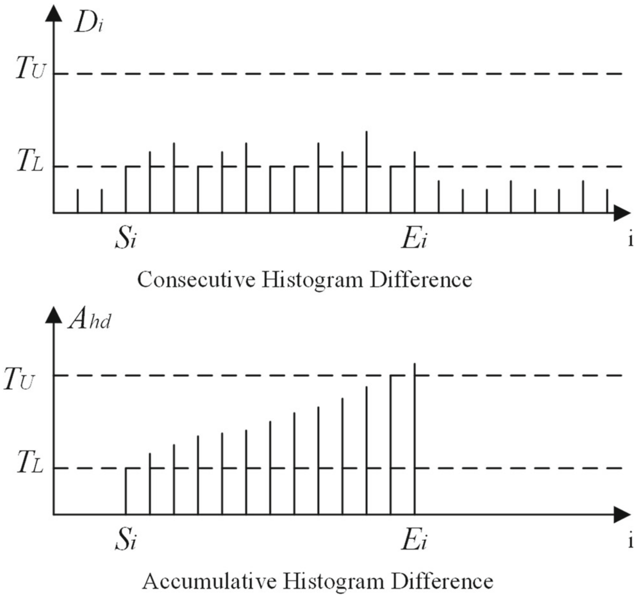


**Fig. 2** Dual-threshold method for GT Detection

differences between consecutive frames obtained in (1) and expressed as follows:

$$A_{hd} = \sum_{i=1}^{n-1} D_i \tag{2}$$

Where $n$ represents the number of frames in a GT segment. The histogram difference between the consecutive frames is insignificant for GTs due to same camera shot, unlike abrupt transitions where camera shots transform suddenly. However, the accumulative difference $(A_{hd})$ ultimately becomes significantly large and appears similar to the histogram difference of consecutive frames in the abrupt transitions as shown in (Fig. 2). Afterwards, the dual-threshold-based method [6] is applied for GT detection. For dual-threshold based method, $D_i$ is compared against a computed threshold $T_L$ to detect the start of GT frames. Similarly, $A_{hd}$ is compared against the computed threshold $T_U$ to detect the end of GT frames. More explicitly, a segment is marked as GT if $D_i$ is less than $T_L$ and the $A_{hd}$ is greater than $T_U$ as shown in Fig. 2. In GT method a lower threshold $T_L$ is used to identify the candidate frames that start a transition. Whereas, the threshold $T_U$ is used to detect the end of a transition.

The distance (in terms of number of frames) between the two successive $GTs$ is computed to detect the candidate replay segment. Let $S_i$ and $E_i$ denote the start and end of $i_{th}$ GT, and $N_R$ represents the distance between frame indices of $S_i$ and $E_{(i+1)}$, then a segment between two consecutive $GTs$ is labeled as candidate replay segment if it satisfies the following condition,

$$2N_{GT} + N_{RL} \leq E_{(i+1)} - S_i \leq 2N_{GT} + N_{RU} \tag{3}$$

Where $N_{RL}$ and $N_{RU}$ represent minimum and maximum duration of a replay segment and set to 50 and 500 respectively. The foremost reason for these values is that the replays usually consist of the video events between 2 and 20 seconds, therefore $N_{RL}$ and $N_{RU}$ are assigned the values of 50 (i.e. 2 seconds of 25 fps) and 500 (i.e. 20 seconds of 25 fps). However, these values can still be modified according to the requirements. The $N_{GT}$ represents the minimum length of GT frames either at the start or end of the replay segment, and $2N_{GT}$ represents the total GT frames in a replay segment. Hence, $(2N_{GT} + N_{RL})$ represents the replays of minimum duration and $(2N_{GT} + N_{RL})$ represents the replays of maximum duration. More specifically, a candidate segment is selected as a GT in case the distance between start and end of GT frame-indices is $(\geq N_{GT})$. For clear illustration the process is explained through the start of a GT frame segment as shown in Fig. 3. As it can be observed that there are about 12 GT frames in a segment shown in Fig. 3. We have computed the average for various GTs and used this average as the $N_{GT}$ value. Therefore, in our implementation the value of $N_{GT}$ is set to 10. $N_{RL}$ and $N_{RU}$ captures the replays of various lengths. To test effectiveness of this approach, we applied it on the selected videos from the dataset at random. Shown in Fig. 4 is the start $(S_i)$ and end $(E_{(i+1)})$ of candidate replay segments for cricket, tennis, baseball, and basketball videos.

## 2.2 Key-events detection

We propose a novel approach for key-events detection through processing the replay frames as the existing systems have never used replay frames to classify various key-events in sport videos. Once the replay segments are detected in the underlying sports videos, we use these segments for key-events detection. For this, we apply the GMM over the replay frames that outputs the silhouettes to represent the individual frames comprising the replay segment. Afterwards, a motion history image (MHI) is computed through the extracted silhouettes to represent the entire replay segment in form of a single image. The MHIs of the various replay segments are then represented in the form of feature vectors through the confined elliptical local

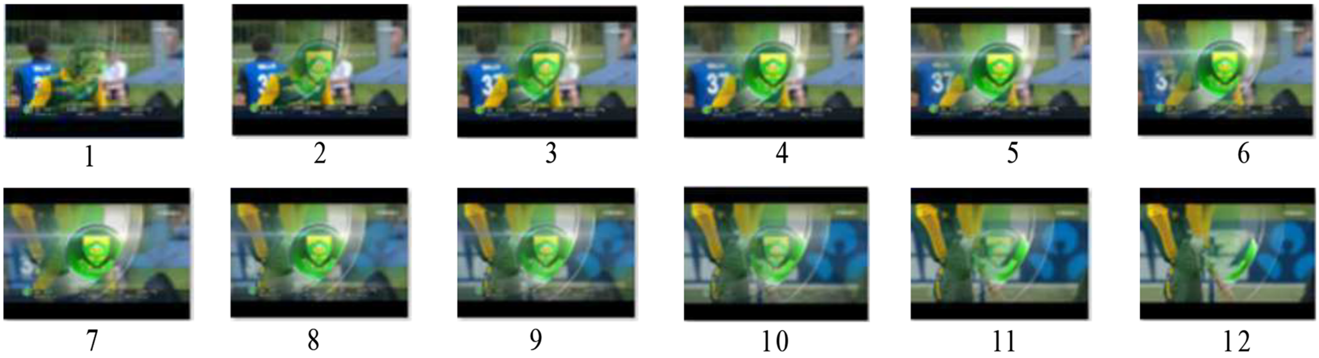
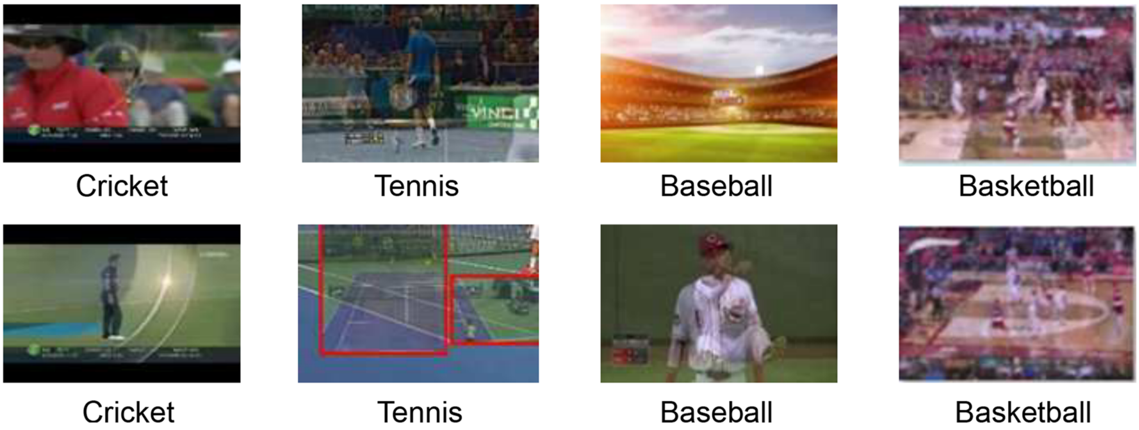


**Fig. 3** GT frames of a cricket replay


Cricket Tennis Baseball Basketball

Cricket Tennis Baseball Basketball

**Fig. 4** Top Row: Start GT Frames, Bottom Row: End GT frames

ternary patterns that is a novel feature extraction approach particularly designed for key-events classification through the extreme learning machines. Moreover, the ELM is also never explored in the sports video summarization research.

### 2.2.1 Motion pattern detection

Sports videos contain drastic variations in the illumination conditions due to field and camera views, lighting sources, etc. and have complex (non-stationary) backgrounds that are needed to be suppressed for key-events detection. Therefore, as a first step of modeling the replay segments for key-events detection, we discriminate between the background and foreground pixels through GMM [31]. We applied GMM to extract the accurate motion information through frames present in the replay segment, as the GMM updates the pixel data by adding new frames and discards the older frames. Hence, GMM is more robust against the variations in illumination conditions and improves the performance of foreground segmentation. After silhouette extraction from a given sequence of replay frames, we generate the MHI to model the key-events in sports videos. MHI [32] represents a motion pattern that creates the temporal layers of successive image differences in the form of a static image. Motion history can easily be predicted from the pixel intensity as the brighter intensities correspond to more recent motion. The intensity gradients in the MHI are used to measure the directional information of silhouettes in the given replay segment. MHI is robust as the motion information lies at the contours of the moving objects and ignores the undesired motion in the internal regions of object contours. MHI is well renowned due to its ability to represent short duration movement and low computation cost. In the proposed work, MHI creates a unified image of a given sequence of replay video frames that records the spatial and temporal information regarding the motion. We generated the MHI image against each video shot as given below

$$MHI_{\psi}(x, y, t) = \begin{cases} \psi, & if\ B(x, y, t) = 1 \\ \max(0, MHI_{\psi}(x, y, t-1) - 1), & otherwise \end{cases} \tag{4}$$

Where $B(x, y, t)$ represents the grayscale image of differences between the frames and $\psi$ represents the maximum duration of motion that is set to 25 in our experiments.

### 2.2.2 Feature extraction

Effective feature extraction is an indispensable requirement for precise event classification. Therefore, in the proposed work we have extended the existing research of sports videos summarization with representing the video scenes through the unique features. Low level features such as color, texture, etc. are commonly employed for events detection [17]. The potential of extracting prominent features from the input image is the main reason behind using the texture analysis in computer vision applications. Local patterns (Local Binary patterns, Local Ternary Patterns, Local Tetra Patterns) are effective texture feature descriptors that are frequently employed in image classification approaches [33]. For CE-LTPs representation in the proposed method, given a central pixel $(x, y)$ and two perpendicular axis (horizontal and vertical), an elliptic region $f_1\ (x, y)$ of the image patch of size $M \times N$ is the set of pixels which satisfy the following relation.

$$f_1\ (x, y) = \left\{(i, j)\ \mid\ |i - x| \leq \left\lfloor \frac{M}{2} \right\rfloor \cos\theta \wedge |j - y| \leq \left\lfloor \frac{N}{2} \right\rfloor \sin\theta \right\} \tag{5}$$

Where, $\lfloor M/2 \rfloor$ is the length of horizontal axis of the ellipse, $\lfloor N/2 \rfloor$ is the length of vertical axis of the ellipse, $\theta$ indicates the angle and $(i, j)$ represents the indices within the image patch. For instance, the ellipse of $7 \times 9$ image patch obtained by (5) is illustrated in Fig. 5, which demonstrates that the pixel values where ellipse cuts the image matrix are discarded and a perfect ellipse is obtained. As a result, the subsequent matrix gives more appropriate presentation of confined elliptic arrangements.

The geometric representation of image effects on the outcome of the classifier in terms of key-events detection. We quantized the threshold values $t$ of neighboring pixels $g_p$ around the central pixel $g_c$ as follows:

$$f\ (x,\ g_c,\ t) = \left\{ \begin{array}{ll} +1 & x \geq g_c + t \\ 0 & |x - g_c| < t \\ -1 & x \leq g_c - t \end{array} \right. \Bigg|_{x=(g_p - g_c)} \tag{6}$$

The confined elliptic matrix shown through Fig. 5 is converted to elliptic ternary patterns matrix for $t = 1$ as shown in Fig. 6. Elliptic ternary patterns are quantized in six different dimensions to extract maximum information from $N$ neighboring pixels through the following expression.

$$\mu_i = \frac{1}{N} \sum_{k=1}^{N} f_{k\theta} \tag{7}$$

Where $f_{k\theta}$ is the value of $k_{th}$ element in confined elliptic ternary matrix of $\theta$ dimension such that $\theta \in \left[0,\ \pi,\ \frac{\pi}{4},\ \frac{3\pi}{4},\ \frac{5\pi}{4},\ \frac{7\pi}{4}\right]$. And $N$ is the number of values of the matrix elements along the directions in $P = 1$. The values of $\mu_i$ at $\theta = 0$ dimension is computed as shown in Fig. 7. Further, elliptic co-occurrence values among the quantized values in 8 dimensions (0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°, 360°) are computed as follows:

$$f_2\ (x, y) = \left\{ \begin{array}{rl} 1 & if\ x = y = 1 \\ -1 & if\ x = y = -1 \\ 0 & else \end{array} \right. \tag{8}$$

Finally, after identifying the confined elliptic local ternary pattern we represented the frame through a histogram as follows:

$$H(k) = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} f_3(CE - LTP(i, j), k) \tag{9}$$

$$f_3(x, y) = \left\{ \begin{array}{ll} 1, & if\ x = y \\ 0, & otherwise \end{array} \right\} \tag{10}$$

### 2.2.3 ELM classification

In the proposed method, a novel feature descriptor Confined Elliptic Local Ternary Patterns (CE-LTPs) is used to train the Extreme Learning Machine (ELM) for key-events detection. ELM [34] was originally proposed for single hidden layer feed-forward networks (SLFNs) where the hidden layer is not required like a neuron; hence hidden layer is not tuned. For binary classification problems the output function of ELM for single output unit is:

$$f_L(x) = \sum_{i=1}^{L} \beta_i h_i(x) = h(x)\beta \tag{11}$$

Where $\beta = \{\beta_1, ..., \beta_L\}^T$ is the vector having output weights between the hidden layer of $L$ nodes and the output node and $h(x) = \{h_1(x), ..., h_L(x)\}$ is the output vector. For binary classification problems the decision function of ELM is:

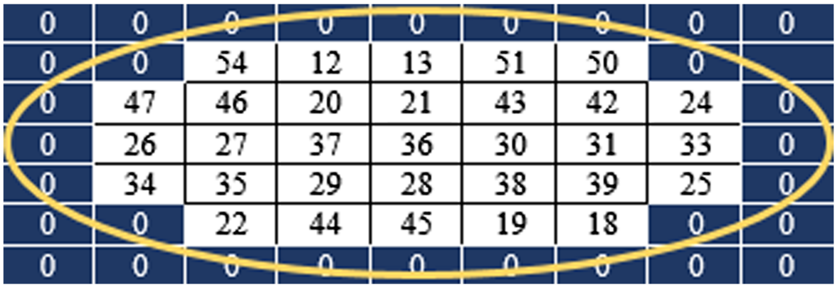
$$f_L(x) = sign\ (h(x)\beta) \tag{12}$$

In order to have the better generalization performance of the network, the ELM targets to reach the smallest training error and smallest norm of the output weights by minimizing the following objective function:

$$Minimize :\ ||H\beta - T||^2\ and\ ||\beta|| \tag{13}$$

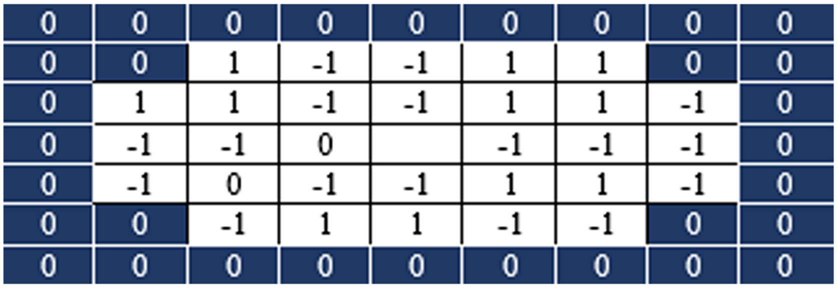
Where $H$ represents the hidden layer output matrix

$$H = \begin{bmatrix} h(x_1) \\ \vdots \\ h(x_N) \end{bmatrix} = \begin{bmatrix} h_1(x_1) & \cdots & h_L(x_1) \\ \vdots & \vdots & \vdots \\ h_1(x_N) & \vdots & h_L(x_N) \end{bmatrix} \tag{14}$$



| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 54 | 12 | 13 | 51 | 50 | 0 | 0 |
| 0 | 47 | 46 | 20 | 21 | 43 | 42 | 24 | 0 |
| 0 | 26 | 27 | 37 | 36 | 30 | 31 | 33 | 0 |
| 0 | 34 | 35 | 29 | 28 | 38 | 39 | 25 | 0 |
| 0 | 0 | 22 | 44 | 45 | 19 | 18 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Fig. 5** Confined Elliptic 7 × 9 Matrix



| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | -1 | -1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | -1 | -1 | 1 | 1 | -1 | 0 |
| 0 | -1 | -1 | 0 | | -1 | -1 | -1 | 0 |
| 0 | -1 | 0 | -1 | -1 | 1 | 1 | -1 | 0 |
| 0 | 0 | -1 | 1 | 1 | -1 | -1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Fig. 6** Confined Elliptic Local Ternary Patterns

To minimize the norm of the output weights, $||\beta||$ maximizes the distance of the separating margins of both classes in ELM feature space: $2/||\beta||$ by defining the minimal least square method as:

$$\beta \;=\; H^{\dagger}T \tag{15}$$

Where $H^{\dagger}$ is the moore-penrose generalized inverse of matrix that can be computed through the orthogonalization method, orthogonal projection method, and singular value decomposition. The multi class scenario consists of minimization of the following objective function:

$$\begin{aligned} &Minimize : L_{P_{ELM}} = \tfrac{1}{2}||\beta||^2 + C\tfrac{1}{2}\sum_{i=1}^{N}||\xi_i||^2 \\ &Subject\,to : h(x_i)\beta = {t_i}^T - {\xi_i}^T, \qquad i = 1, ..., N \end{aligned} \tag{16}$$

Where $t_i = [t_{i,1}, ...t_{i,m}]^T$ is the target output vector and $\xi_i = [\xi_{i,1}, ...\xi_{i,m}]^T$ is the training error vector of the $m$ output nodes. Training of the ELM consists of solving the following dual optimization problem based on KKT theorem:

$$L_{D_{ELM}} = \frac{1}{2}||\beta||^2 + C\frac{1}{2}\sum_{i=1}^{N}||\xi_i||^2 - \sum_{i=1}^{N}\sum_{j=1}^{m}\alpha_{i,j}(h(x_i)\beta_j - t_{i,j} + \xi_{i,j}) \tag{17}$$

where $\beta_j$ is the vector of weights linking hidden layer to the $j^{th}$ output unit and $\beta = [\beta_1, ..., \beta_m]$ . The corresponding optimally conditions based on KKT are as follows:

$$\begin{aligned} &\frac{\partial L_{D_{ELM}}}{\partial \beta_j} = 0 \rightarrow \beta_j = \sum_{i=1}^{N}\alpha_{i,j}h(x_i)^T \rightarrow \beta = H^T\alpha \\ &\frac{\partial L_{D_{ELM}}}{\partial \xi_i} = 0 \rightarrow \alpha_i = C\xi_i, \qquad i = 1, ..., N \\ &\frac{\partial L_{D_{ELM}}}{\partial \alpha_i} = 0 \rightarrow h(x_i)\beta - {t_i}^T + {\xi_i}^T = 0, \qquad i = 1, ..., N \end{aligned} \tag{18}$$

where $\alpha_i = [\alpha_{i,1}, ..., \alpha_{i,m}]^T$ and $\alpha = [\alpha_1, ..., \alpha_N]^T$. From (18), we have:

$$\begin{aligned} &\beta = CH^T\xi \\ &\xi = \tfrac{1}{C}\left(H^T\right)^{\dagger}\beta \end{aligned} \tag{19}$$

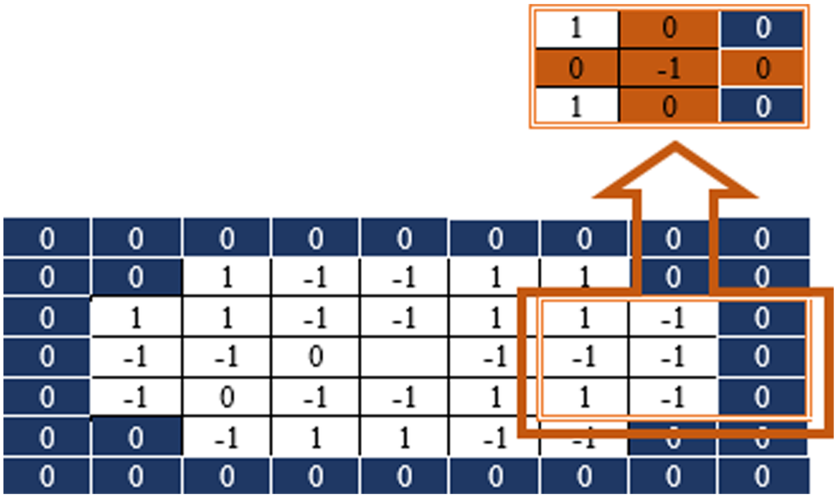
and

$$\begin{aligned} &H\beta - T + \tfrac{1}{C}(H^T)^{\dagger}\beta = 0 \\ &H^T\left(H + \tfrac{1}{C}(H^T)^{\dagger}\right)\beta - = H^T T \\ &\beta = \left(\tfrac{1}{C} + H^T H\right)^{-1} H^T T \end{aligned} \tag{20}$$

The output function of the ELM classifier is:

$$f(x) = h(x)\beta = h(x)\left(\frac{1}{C} + H^T H\right)^{-1} H^T T \tag{21}$$

For a given testing sample the class-label is the index of output node that has the highest output.



| | | |
|---|---|---|
| 1 | 0 | 0 |
| 0 | -1 | 0 |
| 1 | 0 | 0 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | -1 | -1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | -1 | -1 | 1 | 1 | -1 | 0 |
| 0 | -1 | -1 | 0 | | -1 | -1 | -1 | 0 |
| 0 | -1 | 0 | -1 | -1 | 1 | 1 | -1 | 0 |
| 0 | 0 | -1 | 1 | 1 | -1 | -1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Fig. 7** Calculated $\mu_i$ values for $\theta = 0$

## 3 Experiments and results

This section analyzes the results of the experiments designed to evaluate the performance of the proposed system for replay detection and categorization.

### 3.1 Datasets

Performance of the proposed system is evaluated on a dataset consisting of 20 real-world sports videos. The dataset comprises of the videos from five of the major broadcasters namely ESPN, Ten Sports, Sky Sports, Fox Sports, and Eurosport for Cricket, Baseball, Tennis, and Basketball. The cricket videos comprise of 2015 test series between Pakistan and Sri Lanka, and 2014 One Day International series between South Africa and New Zealand. The tennis videos comprise of US Open, ATP Rogers Cup, Tennis Master's Cup, ATP World Tour 2011 finals, and Wimbledon. The basketball videos comprise of 2013 Champion Classic; and the baseball videos are comprised of 2015 Major League Baseball.

There is no standard dataset for sports video summarization. It is a common practice to create a custom video repository to evaluate the performance of video summarization systems. For video selection we have ensured that the video content is diverse. Therefore, the selected videos differentiate in terms of illumination conditions (i.e. day and night videos), video length (1 minute to 30 minutes), and editing effects (i.e. the gradual transitions for replays). Each video in the dataset has a frame resolution of 640 × 480 pixels with a frame rate of 25 fps. A few video frames from our dataset are provided in Fig. 8. We have also provided the dataset for research purposes at [35].

### 3.2 Evaluation metrics

We employed following objective metrics for performance evaluation: precision, recall, F-1 score, accuracy, and error

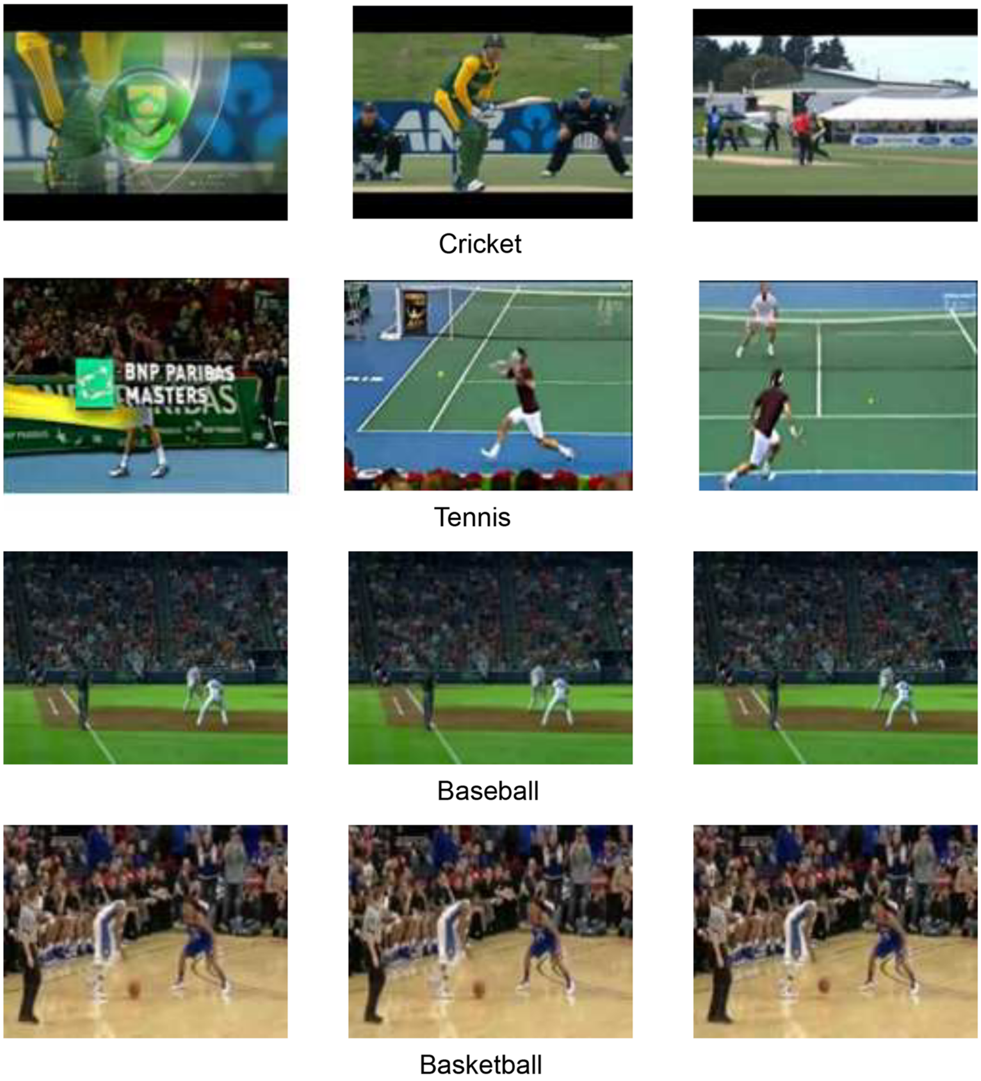


**Fig. 8** Snapshots of dataset for Cricket, Tennis, Baseball and Basketball

rate. Precision represents the ratio of correctly labeled replay frames to the total detected replay frames; and similarly, for key-events detection the ratio of correctly classified key-events against the classified key-events of a particular class. We computed the precision as follows:

$$Precision = \frac{TP}{TP + FP} \tag{22}$$

Where in (22), TP represents true positives, and FP represents false positives in terms of frames labeled as replay or a key-event. Recall represents the ratio of true detection of replay frames against the actual number of replay frames in the video. For key-event detection, recall is the ratio of correctly classified key-events against actual number of key-events. We computed the recall as follows:

$$Recall = \frac{TP}{TP + FN} \tag{23}$$

Where $FN$ in (23) represents the false negatives i.e. the replay frames or the key-events that are misclassified. $F-1$ score is weighted average representation of precision and recall. $F-1$ score was computed to overcome the scenarios where precision and recall amongst the comparative methods overlap i.e. some methods have higher precision and lower recall and vice versa. $F-1$ score was computed as follows:

$$F-1\ \ Score = 2 * \frac{(Precision * Recall)}{Precision + Recall} \tag{24}$$

Accuracy rate represents the ratio of the correctly labeled replay/non-replay frames or the key-events/non key-events to the total number of frames or events. Accuracy rate was computed as follows:

$$Accuracy\ Rate = \frac{TP + TN}{P + N} \tag{25}$$

Where, in (25), P and N represents the total number of positive and negative samples, respectively. Error rate represents the ratio of mislabeled replay frames (both FP and FN) to the total number of frames. Error rate is computed as:

$$Error\ Rate = \frac{FP + FN}{P + N} \tag{26}$$

## 3.3 Performance evaluation for replay detection

For replay detection, the proposed system was tested for each video in the dataset (Table 1). Our method achieves **95.09%, 95.94%, 95%, 95.77%, and 4.23%**, as average precision, recall, f1-score, accuracy, and error rate respectively for all sports videos. The performance of the proposed method was remarkable for cricket, tennis, and basketball, where we were able to achieve **98.27%, 97.89%, and 97.28%** average accuracy values respectively; whereas, for baseball, the results are slightly lower, where the average accuracy is 89.64%. The reason behind the slight performance drop in the case of baseball is attributed to the fact that the selected baseball-videos display some replay segments without the gradual transitions, hence, replay frames remain undetected.

We also evaluated the performance of the proposed system through ROC curves by ploting the true positive rate (TPR) against the false positive rate (FPR). The TPR and FPR were computed as:

$$TPR = \frac{TP}{TP + FN} \tag{27}$$

$$FPR = \frac{FP}{FP + TN} \tag{28}$$

The area under the curve (AUC) of the ROC presented in Fig. 9 signifies that the proposed method is effective in terms of replay frame detection.

### 3.3.1 Camera angle variation

Broadcasters capture and display the replays from different camera-angles; therefore, a fundamental requirement of any replay detection method is appear to be robust against the camera-angle variations. Thus, in this experiment we evaluated the replay detection performance of our method when sports videos present replays from different camera-angles. For this experiment, we selected 4 videos from the dataset that display the similar replay event from three of the camera-angles i.e. front, back, and side. The results of the experiment are presented in Table 2. Our method achieved average accuracy of 94.50% that shows effectiveness of the proposed method in terms of replay detection when multiple views of the replay events are captured.

### 3.3.2 Replay speed

In our next experiment, we measured the performance of the proposed method when replays are broad-casted at different speeds. Robustness against the replay speed is also a significant requirement as replay speeds can easily deceive the replay detection methods to consider the replay-frames as non-replay frames [12, 13]. Thus, a replay detection method must be able to detect the replays independent of the speed. For this experiment, we selected 4 videos that display the similar replay event in different speeds. Our method achieved an average precision, recall, F-1 score, accuracy, and error rate of 98.72%, 98.24%, 0.98, 98.05%, and 1.95% respectively as shown in Table 3. From the table, it can be observed that the proposed method successfully detected the replays which are displayed at different speeds. Thus, the experimental results indicate the effectiveness of the proposed method in terms of replay detection at various speeds.

**Table 1** Replay detection results for cricket, tennis, baseball, basketball

| Video Type | TP | TN | FP | FN | Precision | Recall | Accuracy | Error | F-1 Score |
|---|---|---|---|---|---|---|---|---|---|
| Cricket | | | | | | | | | |
| Crick1 | 292 | 22 | 0 | 2 | 100% | 99.31% | 99.36% | 0.64% | 0.99 |
| Crick2 | 292 | 25 | 2 | 2 | 99.31% | 99.31% | 99.06% | 0.94% | 0.99 |
| Crick3 | 420 | 294 | 0 | 17 | 100% | 96.11% | 97.67% | 2.33% | 0.98 |
| Crick4 | 584 | 189 | 11 | 16 | 98.15% | 97.34% | 96.63% | 3.37% | 0.98 |
| Crick5 | 695 | 242 | 8 | 5 | 98.86% | 99.28% | 98.63% | 1.37% | 0.99 |
| Average | | | | | **99.27%** | **98.27%** | **98.27%** | **1.73%** | **0.99** |
| Tennis | | | | | | | | | |
| Tennis1 | 140 | 583 | 0 | 5 | 100% | 96.55% | 99.32% | 0.68% | 0.98 |
| Tennis2 | 342 | 592 | 41 | 4 | 89.29% | 98.84% | 95.40% | 4.60% | 0.94 |
| Tennis3 | 226 | 249 | 0 | 5 | 100% | 97.83% | 98.95% | 1.05% | 0.99 |
| Tennis4 | 740 | 241 | 9 | 10 | 98.79% | 98.67% | 98.10% | 1.90% | 0.98 |
| Tennis5 | 689 | 180 | 10 | 11 | 98.56% | 98.43% | 97.64% | 2.36% | 0.98 |
| Average | | | | | **97.33%** | **98.06%** | **97.89%** | **2.11%** | **0.97** |
| Baseball | | | | | | | | | |
| Base1 | 322 | 610 | 100 | 21 | 76.30% | 93.87% | 88.50% | 11.50% | 0.84 |
| Base2 | 367 | 391 | 123 | 22 | 74.89% | 94.34% | 83.94% | 16.06% | 0.83 |
| Base3 | 198 | 409 | 51 | 72 | 79.52% | 73.34% | 83.15% | 16.85% | 0.76 |
| Base4 | 446 | 240 | 15 | 14 | 96.74% | 96.95% | 95.94% | 4.06% | 0.96 |
| Base5 | 682 | 280 | 15 | 18 | 97.84% | 97.42% | 96.68% | 3.32% | 0.97 |
| Average | | | | | **85.05%** | **91.18%** | **89.64%** | **10.36%** | **0.87** |
| Basketball | | | | | | | | | |
| Basket1 | 266 | 349 | 10 | 2 | 96.37% | 99.25% | 98.09% | 1.91% | 0.98 |
| Basket2 | 134 | 82 | 0 | 14 | 100% | 90.54% | 93.92% | 6.08% | 0.95 |
| Basket3 | 211 | 139 | 0 | 6 | 100% | 97.23% | 98.31% | 1.69% | 0.99 |
| Basket4 | 562 | 209 | 9 | 8 | 98.42% | 98.60% | 97.84% | 2.16% | 0.98 |
| Basket5 | 679 | 179 | 8 | 7 | 98.83% | 98.87% | 98.28% | 1.72% | 0.98 |
| Average | | | | | **98.72%** | **96.89%** | **97.28%** | **2.72%** | **0.98** |

Average measures are provided in bold

**Fig. 9** ROC curves for Replay Detection of Sports videos (cricket, tennis, baseball, baseketball)

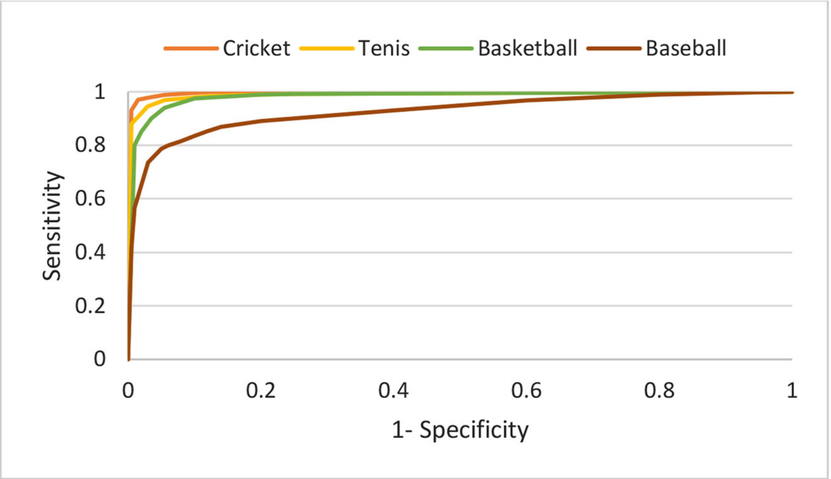


**Table 2** Detection performance of replay events for camera angle variations

| Video type | Camera views | TP | TN | FP | FN | Precision | Recall | Accuracy | Error |
|---|---|---|---|---|---|---|---|---|---|
| Crick1 | Front, Back | 292 | 22 | 0 | 2 | 100% | 99.31% | 99.36% | 0.64% |
| Crick4 | Front, Back | 584 | 189 | 11 | 16 | 98.15% | 97.34% | 96.63% | 3.37% |
| Tennis4 | Front, Side | 740 | 241 | 9 | 10 | 98.79% | 98.67% | 98.10% | 1.90% |
| Base2 | Front, Side | 367 | 391 | 123 | 22 | 74.89% | 94.34% | 83.94% | 16.06% |
| Average | | | | | | **92.95%** | **97.41%** | **94.50%** | **5.50%** |

Average measures are provided in bold

## 3.4 Performance comparison for replay detection

To justify the proposed method as an authoritative approach for replay detection, we conducted the performance comparison against several existing replay detection methods. The brief description of the comparative schemes and comparison results are provided in this section.

### 3.4.1 Comparative schemes

We compared our replay detection scheme against [10, 12, 15, 18–20, 22–24, 26, 28]. The reason to select these methods for comparison is due to their relevance in terms of the concerned problem i.e. replay detection. The brief description of these methods is as under: In [12], Pan et al. used Hidden Markov model (HMM) and Viterbi algorithm to analyze frame motion movements for replay detection. In [15], Duan et al. used motion features to develop a logo detection technique based on mean shift for replay detection. In [18], Zawbaa et al. applied SVM and NN to detect the logo frames that were then used for replay detection. In [19], Wang et al. proposed a generic framework to detect slow motion replays in sports videos based on the difference of motion between the normal and slow-motion video shots. In [23], Su et al. proposed a transition effect detection technique to identify the likely highlight sequences in baseball videos. A transition effect template was generated for the archived video by selecting a collection of video segments. The slow-motion video segments were then searched by comparing the candidate frames with this template. The extracted slow-motion segments were processed further to obtain more accurate highlights by training HMM to classify four distinct events in baseball videos. In [20], Xu et al. presented an approach to detect the logos by computing the difference in frames to create the logo template from the frames in soccer videos. In [24], Wang et al. proposed a replay detection technique using the context information, which learned from the transition of shot types. The dependency on shot classification degrades the accuracy of this technique [24]. In [22, 26, 28], statistical features have been used to identify the logo transitions that are then used to detect the replay frames.

### 3.4.2 Comparative analysis

Detection performance of proposed and comparative replay detection approaches [10, 12, 15, 18–20, 22–24, 26, 28], in terms of precision and recall are shown in Fig. 10. From the results it can be observed that the proposed method has achieved highest precision and recall rates that are 98.72% and 96.89% respectively. Whereas, in comparative methods Nyugen et al. [28] and Su et al. [23] achieved the second and third highest average precision values that are 94.6% and 91% respectively. Similarly, in terms of recall rates the second and third best recall rates were achieved by Nyugen et al. [28] and Zawbaa et al.[18] that are 95.8% and 95.7% respectively. So the results clearly reveals the robustness of the proposed method.

As mentioned earlier in Section 3.1, there does not exist any standard repository of sports video summarization. Therefore, we followed the similar approach for performance evaluation as adopted by the comparative methods i.e. to create a custom video repository using YouTube video sources with diverse content. The detailed information

**Table 3** Detection performance of replay events for replay speed variations

| Video type | TP | TN | FP | FN | Precision | Recall | Accuracy | Error | F1- score |
|---|---|---|---|---|---|---|---|---|---|
| Crick2 | 292 | 25 | 2 | 2 | 99.31% | 99.31% | 99.06% | 0.94% | 0.99 |
| Tennis3 | 226 | 249 | 0 | 5 | 100% | 97.83% | 98.95% | 1.05% | 0.99 |
| Base4 | 446 | 240 | 15 | 14 | 96.74% | 96.95% | 95.94% | 4.06% | 0.96 |
| Basket5 | 679 | 179 | 8 | 7 | 98.83% | 98.87% | 98.28% | 1.72% | 0.98 |
| Average | | | | | **98.72%** | **98.24%** | **98.05%** | **1.95%** | **0.98** |

Average measures are provided in bold

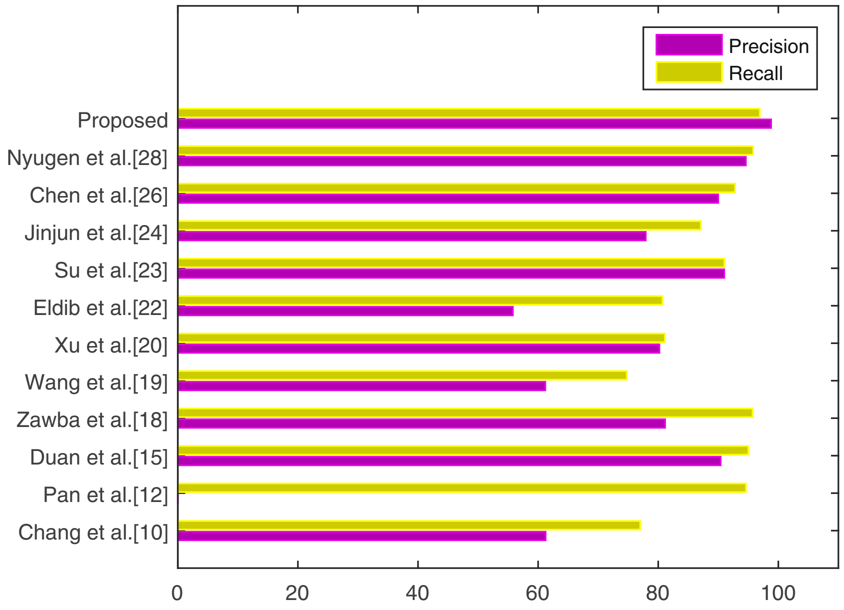


**Fig. 10** Precision and Recall rates of proposed and existing systems

about the format, frame rate, resolution, repository size, and sports categories in the video-sets of our and comparative methods are shown in Table 4. We can observe that the video-set we used for the evaluation of our method is more diverse in terms of size, no. of sports categories, and frame resolution. Our method is evaluated on 20 videos whereas the comparative methods are evaluated over an average of 7.5 videos. Hence, our database is approximately 3 times bigger than the comparative methods. Although the accuracy of a method significantly depends on the dataset size, our method still shows the superior performance. Additionally, our frame resolution is 640 × 480 whereas the resolution of comparative methods is approximately half of our method. In the same way, we evaluated our method on 4 sports categories whereas comparative methods are evaluated either on a single sports category or maximally 2 sports categories. Even in these scenarios where comparative methods are biasedly facilitated still our method outperforms all of the comparative techniques in terms of precision and recall.

### 3.5 Performance evaluation for key-event detection through ELM

The performance of the proposed ELM based key-events detector for particular event category is remarkable; where our method achieved an average precision, recall, F-1 score, accuracy, and error rate of 96.36%, 95.27%, 95.76%

**Table 4** Comparison of the proposed and existing replay detection techniques

| Techniques | Non-standard dataset details | | | | | Precision | Recall |
|---|---|---|---|---|---|---|---|
| | Format | Frame rate | Resolution | No. of videos | Sports category | | |
| Chang et al. [10] | – | – | – | 6 | 1 | 61.25% | 77% |
| Pan et al. [12] | MPEG-2 | 25 fps | 320 x 240 | 14 | 2 | Not Used | 94.60% |
| Duan et al. [15] | – | – | – | 1 | 1 | 90.40% | 95% |
| Zawba et al. [18] | AVI | 30 fps | - | 5 | 1 | 81.15% | 95.70% |
| Wang et al. [19] | – | – | - - | 8 | 2 | 61.20% | 74.77% |
| Xu et al. [20] | X264 | 30 fps | 320 x 240 | 4 | 1 | 80.20% | 81.10% |
| Eldib et al. [22] | – | – | – | 10 | 1 | 55.80% | 80.70% |
| Su et al. [23] | MPEG-2 | 30 fps | 352 x 240 | 10 | 1 | 91% | 91% |
| Jinjun et al. [24] | – | – | – | 3 | 1 | 77.93% | 87.10% |
| Chen et al. [26] | MPEG-2 | 30 fps | 480x352 | 10 | 1 | 90% | 92.80% |
| Nyugen et al. [28] | – | – | – | 3 | 1 | 94.60% | 95.80% |
| Proposed system | AVI | 25 fps | 640 x 480 | 20 | 4 | 98.80% | 96.90% |

**Table 5** Selected key-events for four sports categories

| Sports categories | Key-event classes | | |
|---|---|---|---|
| Cricket | Boundary | Six | Wicket |
| Tennis | Serve | Let | Point |
| Baseball | Home Run | Strikeout | Hit and Run |
| Basketball | 1-Point | 2-Points | 3-Points |

95.75%, and 4.25% respectively. We evaluated our method by training the ELM classifier over 12 key-event classes of four sports categories. The number of key-event classes for each of the sports categories are presented in Table 5.

The results for cricket videos on three key-event classes i.e. boundary, six, and wicket are shown in Fig. 11. The proposed method provides an average precision, recall, F-1 score, accuracy, and error rate of 96.18%, 95.11%, 95.54% 95.5%, and 4.5% respectively for cricket videos. The results for basketball videos on three key-events i.e. 1-point, 2-points, and 3-points are presented in Fig. 12. For basketball videos, the proposed key-events detection method achieves an average precision, recall, F-1 score, accuracy, and error rate of 96.1%, 94.66%, 95.32% 95.46%, and 4.54% respectively. For baseball videos, we selected homerun, hit and run, and strikeout as key-event classes. The results obtained for these three key-event classes on baseball videos are shown in Fig. 13. The average precision, recall, F-1 score, accuracy, and error rate of 95.83%, 94.16%, 94.97% 95%, and 5% are attained respectively.

Similarly, for tennis videos, we computed the objective evaluation metrics and results obtained are provided in Fig. 14. The average precision, recall, F-1 score, accuracy, and error rate of 97.33%, 97.16%, 97.24% 97.06%, and 2.94% are obtained on serve, let, and point key-event classes in tennis videos.

## 3.6 Performance comparison for key-events detection based replay event categorization

To validate the effectiveness of the proposed key-events detection method for replay categorization, we conducted the performance comparison against existing state-of-the-art event detection methods. The brief description of the comparative schemes and comparison results are provided in this section.

### 3.6.1 Comparative schemes

There is no existing research work that classify the key-events based on analyzing the replays from the sports videos. Hence, performance of the proposed key-events detection method is compared with the existing key-events detection methods. We compared our proposed scheme against [5, 8, 11, 14, 36–39]. The brief description of these methods is as follows: In [5], Javed et al. used audio-visual features to propose an automated method for key-events detection and summarization of cricket videos. As a first step, rule-based induction was applied to detect the excited audio frames in the cricket videos. In the second step, the corresponding video frames of the excited audio clips were analyzed through a decision tree to detect the key-events. Video skims were generated against each key-event

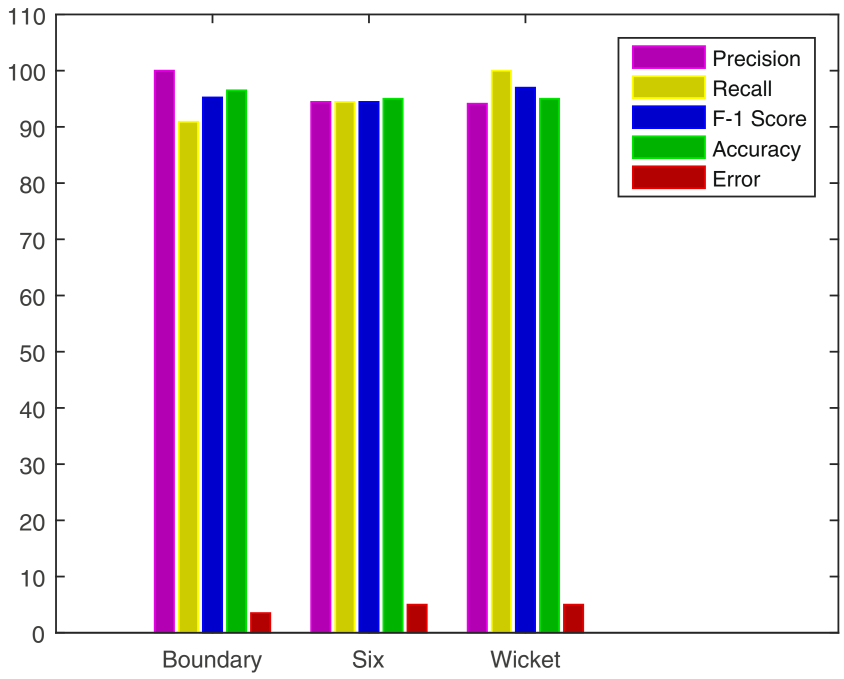
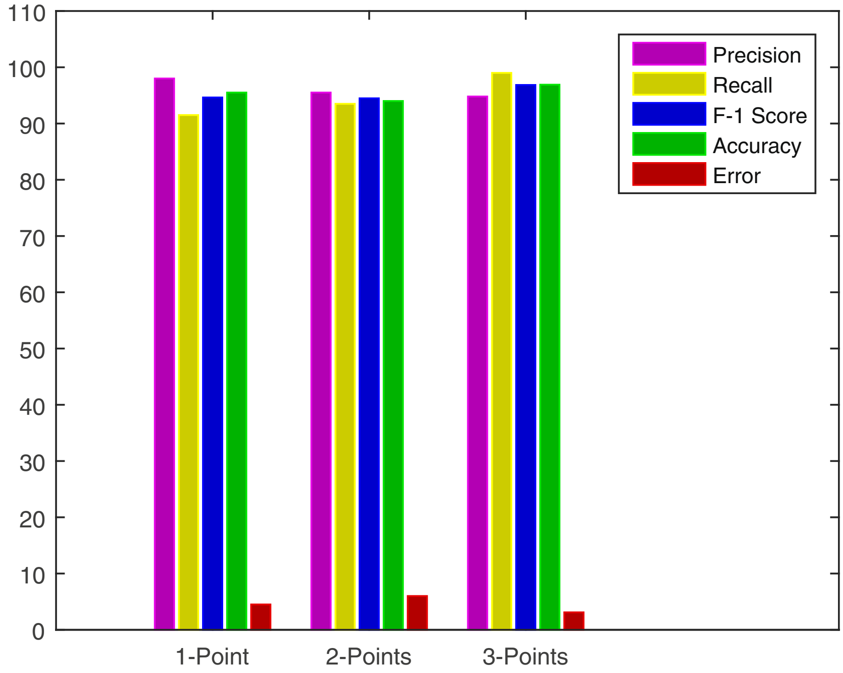


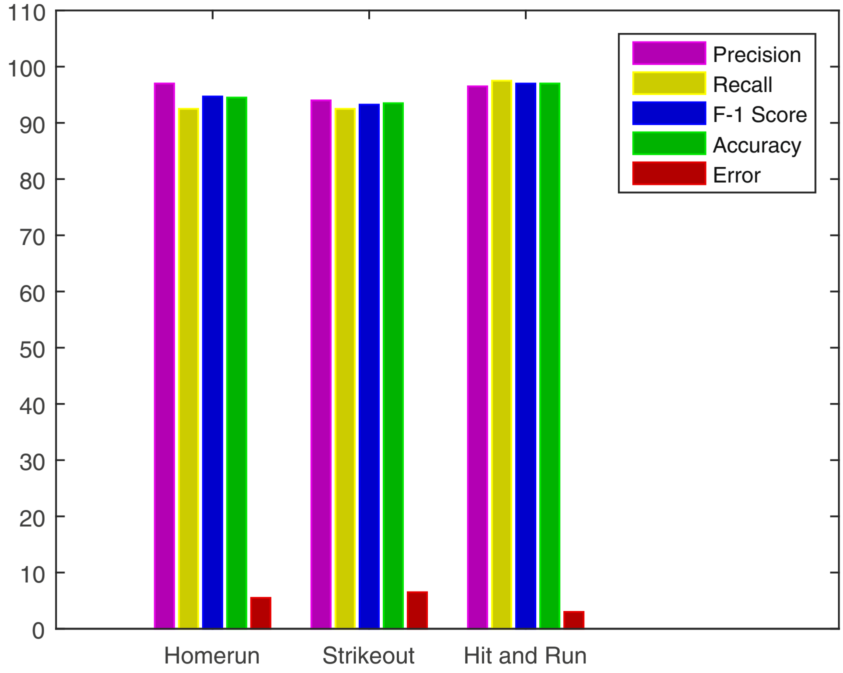
**Fig. 11** Key-Events Detection for Cricket Videos

**Fig. 12** Key-Events Detection for Basketball Videos



and arranged in a chronological order to produce the summarized video. In [11], Jiang et al. proposed a soccer event detection system using a deep learning method based on convolution neural networks (CNNs) and recurrent neural network (RNN). In [14], Tavassolipour et al. used Bayesian network-based scheme to detect significant events for soccer video summarization. In [36], Wilson et al. applied the HMM to propose an event driven system for sports video classification. This method [36] has a dependency on sequence knowledge for accurate events classification and performs better on those videos which have well-defined events. In [37], Midhu et al. proposed a two-step summarization framework for cricket videos. In the first step, low level features were used to classify various shots, replay detection, and frames labeling for various events. In the second step, concept mining was performed using the apriori algorithm and labeled frames to generate the summarized video. In [38], Wang et al. proposed

**Fig. 13** Key-Events Detection for Baseball Videos



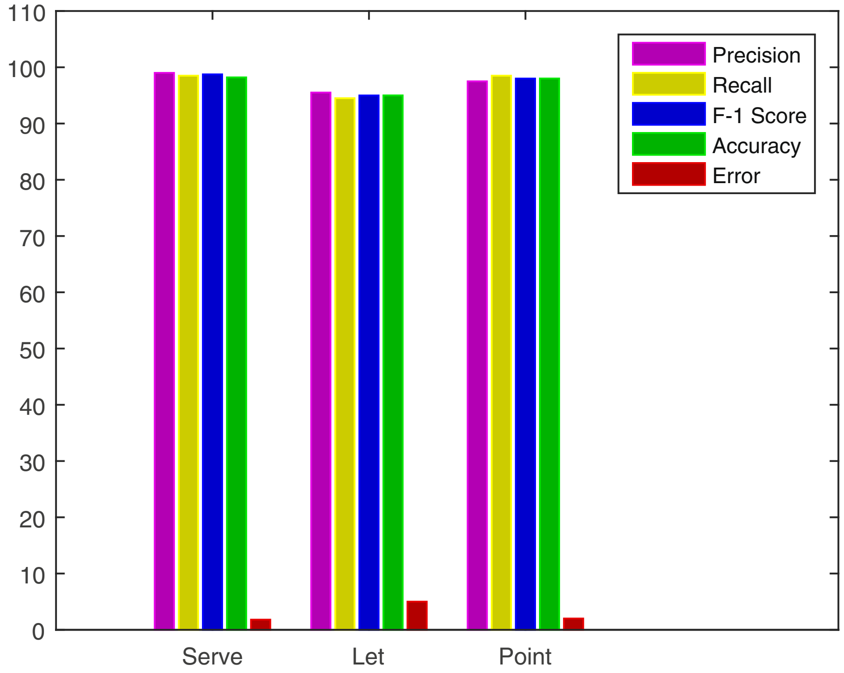


**Fig. 14** Key-Events Detection for Tennis Videos

a framework for key-events annotation, event boundaries detection, soccer field classification, and whistle detection in soccer videos. Similarly, in [39], Godi et al. applied a deep CNN method to summarize ice-hockey videos.

### 3.6.2 Comparative analysis

The performance of the proposed key-events detection method is compared against the comparative approaches [5, 8, 11, 14, 36–39] in terms of precision and recall (see Table 6). From the results it can be clearly observed that the proposed method has achieved highest precision and recall values that are 96.36% and 95.27% respectively. Whereas, [8] and [11] have achieved second and third best precision rates of 95% and 92.37% respectively. Similarly, [8] and [38] have achieved second and third best recall rates of 92% and 91% respectively. The comparative method of [8] is specifically designed for baseball videos, whereas, [11] and [38] are designed for soccer videos and unable to detect the events for multiple sports. In comparison, our method successfully detects the key-events for multiple sports (cricket, tennis, baseball, basketball). In addition, we evaluated the performance of our method on a larger video repository as compared to comparative methods. The detailed information about the format, frame rate, resolution, repository size, and sports categories in the video-sets of our and comparative methods are shown in Table 6. Precision and recall rates

**Table 6** Comparison for key-events detection of the proposed and existing replay detection techniques

| Techniques | Non-standard dataset details | | | | | Precision | Recall |
|---|---|---|---|---|---|---|---|
| | Format | Frame rate | Resolution | Video length | Sports category | | |
| Javed et al. [5] | AVI | 25 fps | 640 x 480 | 6 hrs | 1 | 91.87% | 89.85% |
| Hung et al. [8] | MPEG-1 | 30 fps | 352 x 240 | Not specified | 1 | 95% | 92% |
| Jiang et al. [11] | – | – | 256 x 256 | – | 1 | 92.37% | 87.79% |
| Tavassolipour et al. [14] | MPEG-4 | 25 fps | 640 x 368 | 9 | 1 | 86.90% | 80.10% |
| Wilson et al. [36] | AVI | 25 fps | 320 x 240 | 5.5 hrs | 5 | 73.63% | 76.37% |
| Midhu et al. [37] | AVI | 30 fps | – | 03 hrs | 1 | 88.01% | 87.91% |
| Wang et al. [38] | – | 25fps | – | 17 | 1 | 91.30% | 91% |
| Godi et al. [39] | – | 30 fps | 100 x 100 | 2 | 1 | 69% | 84% |
| Proposed system | AVI | 25 fps | 640 x 480 | 10 hrs | 4 | 96.36% | 95.27% |

are employed for performance comparison of our key-events detection method. From the results in Table 6, it can be clearly observed that our method provides superior precision and recall rates and outperforms the comparative methods. The main reason our proposed approach outperforms the comparative methods is that, the representation of the key-frames through CE-LTP provides effective feature extraction that helps ELM to achieve better generalization; hence, results in form of higher classification accuracy. As ELM provides smallest training error and considers the magnitude of weights as well, therefore, ELM provides better generalization performance. Whereas, the comparative methods only focus on achieving minimum training error without considering the generalization performance.

## 4 Conclusion

In this paper, we proposed an effective method for key-event detection and categorization and their applications to summarize the sports videos. It has been demonstrated that the replays represent key-events in the input video and can be used for video summarization. A novel two-stage framework for sports video summarization is proposed here. The first stage processes input video and extracts the key-events. Specifically, it uses computationally efficient thresholding-based approach for gradual transition detection, which represents temporal boundaries of a replay. The key-events then processed at the second stage for motion history images (MHIs) generation. A novel confined elliptical local ternary pattern features are then used to capture characteristics from MHIs. The key-events are then classified using the ELM classifier. Robustness of the proposed framework have also been evaluated against camera variations, replay speed, logo (design, size and placement), broadcasters, and sports category. Performance of the proposed video summarization framework is also evaluated on a video dataset consisting of 20 videos of four different sports categories i.e. Cricket, Basketball, Baseball, and Tennis. It has been shown that the proposed method achieves an average accuracy of 95.8% that illustrates the significance of the proposed method in terms of replay and key-events detection for video summarization. It has been noted that the proposed replay detection method shows a marginal drop in performance for the videos containing relays without gradual transitions.

Our current research efforts are focused on developing methods for replay in the absence of gradual transitions. The research can be extended by incorporating the recent advancement in autonomous learning [40–42]. Developmental network will be examined for learning features and events classification autonomously from the audio-visual content of sports videos results will be examined and analyzed.

**Acknowledgments** This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2016R1D1A1B03933860).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## References

1. Panda R, Roy-Chowdhury AK (2017) Multi-view surveillance video summarization via joint embedding and sparse optimization. IEEE Trans Multimedia 19(9):2010–2021
2. Muhammad K, Ahmad J, Sajjad M, Baik SW (2016) Visual saliency models for summarization of diagnostic hysteroscopy videos in healthcare systems. Springer Plus 5(1):1495
3. Tran QD, Hwang D, Lee OJ, Jung JE (2017) Exploiting character networks for movie summarization. Multimed Tools Appl 76(8):10357–10369
4. Varini P, Serra G, Cucchiara R (2017) Personalized egocentric video summarization of cultural tour on user preferences input. IEEE Trans Multimed 19(12):2832
5. Javed A, Bajwa KB, Malik H, Irtaza A, Mahmood MT (2016) A hybrid approach for summarization of cricket videos. In: IEEE International conference on consumer electronics-Asia (ICCE-Asia). IEEE, pp 1–4
6. Javed A, Bajwa KB, Malik H, Irtaza A (2016) An efficient framework for automatic highlights generation from sports videos. IEEE Signal Process Lett 23(7):954–958
7. Li B, Pan H, Sezan I (2003) A general framework for sports video summarization with its application to soccer. In: ICASSP, 2003: Proceedings of 30th IEEE international conference on acoustics, speech and signal processing; 2003 April 6; Hong Kong. IEEE, pp 169–172
8. Hung MH, Hsieh CH (2008) Event detection of broadcast baseball videos. IEEE Trans Circ Syst Vid Technol 18(12):1713–1726
9. Kolekar MH, Sengupta S (2015) Bayesian network-based customized highlight generation for broadcast soccer videos. IEEE Trans Broadcast 61(2):195–209
10. Chang P, Han M, Gong Y (2002) Extract highlights from baseball game video with hidden Markov models. In: ICIP, 2002: Proceedings of 9th IEEE international conference on image processing; 2002 Sep 22-25; Pittsburgh, USA. IEEE, pp 609–612
11. Jiang H, Lu Y, Xue J (2016) Automatic soccer video event detection based on a deep neural network combined CNN and RNN. In: Proc int conf in tools with artificial intelligence, San Jose, CA, USA, November 2016. IEEE, pp 490–494
12. Pan H, Van Beek P, Sezan MI (2001) Detection of slow-motion replay segments in sports video for highlights generation. In: ICASSP, 2001: Proceedings of 28th IEEE international conference on acoustics, speech and signal processing; 2001 May 7-11; Utah, USA. IEEE, pp 1649–1652
13. Pan H, Li B, Sezan MI (2002) Detection of slow-motion replay segments in sports video for highlights generation. In: ICASSP, 2001: Proceedings of 28th IEEE international conference on acoustics, speech and signal processing; 2001 May 7-11; Utah, USA. IEEE, pp 1649–1652
14. Tavassolipour M, Karimian M, Kasaei S (2014) Event detection and summarization in soccer videos using Bayesian network and copula. IEEE Trans Circ Syst Video Technol 24(2):291–304

15. Duan LY, Xu M, Tian Q, Xu CS et al (2004) Mean shift based video segment representation and applications to replay detection. In: ICASSP, 2004: Proceedings of 29th IEEE international conference on acoustics, speech and signal processing; 2004 May 17-21; Montreal, Canada. IEEE, pp 709–712
16. Soleymani M, Larson M, Pun T, Hanjalic A (2014) Corpus development for affective video indexing. IEEE Trans Multimed 16(4):1075–1089
17. Kapela R, McGuinness K, O'Connor NE (2017) Real-time field sports scene classification using colour and frequency space decompositions. J Real-Time Image Process 13(4):725–737
18. Zawbaa HM, El-Bendary N, Hassanien AE, Kim TH (2011) Machine learning-based soccer video summarization system. In: Multimedia, computer graphics and broadcasting. Springer, Berlin, pp 19–28
19. Wang L, Liu X, Lin S, Xu G, Shum HY (2004) Generic slow-motion replay detection in sports video. In: ICIP, 2004: Proceedings of 11th IEEE international conference on image processing; 2004 Oct 24-27; Singapore. IEEE, pp 1585–1588
20. Xu W, Yi Y (2011) A robust replay detection algorithm for soccer video. IEEE Signal Process Lett 18(9):509–512
21. Zhao F, Dong Y, Wei Z, Wang H (2012) Matching logos for slow motion replay detection in broadcast sports video. In: ICASSP, 2012: Proceedings of 37th IEEE international conference on acoustics, speech and signal processing; 2012 Mar 25; Kyoto, Japan. IEEE, pp 1409–1412
22. Eldib MY, Zaid BSA, Zawbaa HM, El-Zahar M, El-Saban M (2009) Soccer video summarization using enhanced logo detection. In: ICIP, 2009: proceedings of 16th IEEE international conference on image processing; 2009 Nov 7-10; Cairo, Egypt. IEEE, pp 4345–4348
23. Su PC, Lan CH, Wu CS, Zeng ZX, Chen WY (2013) Transition effect detection for extracting highlights in baseball videos. EURASIP J Image Vid Process 2013(1):1–6
24. Wang J, Chng E, Xu C (2005) Soccer replay detection using scene transition structure analysis. In: ICASSP, 2005: Proceedings of 30th IEEE international conference on acoustics, speech, and signal processing; 2005 Mar 19-23; Philadelphia, PA, USA. IEEE, pp 433–436
25. Zhao Z, Jiang S, Huang Q, Zhu G (2006) Highlight summarization in sports video based on replay detection. In: ICME, 2006: Proceedings of international conference on multimedia and expo; 2006 Jul 9-12; Toronto, Canada. IEEE, pp 1613–1616
26. Chen CM, Chen LH (2015) A novel method for slow motion replay detection in broadcast basketball video. Multimed Tools Appl 74(21):9573–9593
27. Chen CM, Chen LH (2014) Novel framework for sports video analysis: a basketball case study. In: ICIP, 2014: Proceedings of international conference on image processing; 2014 Oct 27-30; Paris, France. IEEE, pp 961–965
28. Nguyen N, Yoshitaka A (2012) Shot type and replay detection for soccer video parsing. In: ISM, 2012: Proceedings of IEEE international symposium on multimedia; 2012 Dec 10-12; California, USA. IEEE, pp 344–347
29. Dang Z, Du J, Huang Q, Jiang S (2007) Replay detection based on semi-automatic logo template sequence extraction in sports video. In: ICIG 2007: Proceedings of 4th international conference on image and graphics; 2007 Aug 22; Sichuan, China. IEEE, pp 839–844
30. Li W, Chen S, Wang H (2009) A rule-based sports video event detection method. In: CISE, 2009: Proceedings of 21st IEEE international conference on computational intelligence and software engineering; 2009 Dec 11-13; Wuhan, China: IEEE, pp 1–4
31. Chen M, Wei X, Yang Q, Li Q, Wang G, Yang MH (2017) Spatiotemporal GMM for background subtraction with superpixel hierarchy. IEEE Trans Pattern Anal Mach Intell 40(6):1518
32. Bilen H, Fernando B, Gavves E, Vedaldi A (2017) Action recognition with dynamic image networks. IEEE Transactions on Pattern Analysis and Machine Intelligence
33. Murala S, Maheshwari R, Balasubramanian R (2012) Local tetra patterns: a new feature descriptor for content-based image retrieval. IEEE Trans Image Process 21(5):2874–2886
34. Huang GB, Zhou H, Ding X, Zhang R (2012) Extreme learning machine for regression and multiclass classification. IEEE Trans Syst Man Cybern Part B (Cybern) 42(2):513–529
35. Javed A, Malik H, Bajwa K, Irtaza A, Mahmood MT Data from: replay detection framework for automatic highlights generation from sports videos. Dryad Digital Repository. https://doi.org/10.5061/dryad.5b880
36. Wilson S, Mohan CK, Murthy KS (2014) Event-based sports videos classification using HMM framework. In: Computer vision in sports. Springer, Cham, pp 229–244
37. Midhu K, Padmanabhan NA (2018) Highlight generation of cricket match using deep learning. In: Computational vision and bio inspired computing. Springer, Cham, pp 925–936
38. Wang Z, Yu J, He Y (2017) Soccer video event annotation by synchronization of attack-defense clips and match reports with coarse-grained time information. IEEE Trans Circ Syst Vid Technol 27(5):1104–1117
39. Godi M, Rota P, Setti F (2017) Indirect match highlights detection with deep convolutional neural networks. In: Proc int conf on image analysis and processing, Catania, Italy, September 2017, pp 87–96
40. Wang D, Xin J (2018) Emergent spatio-temporal multimodal learning using a developmental network. Appl Intell 1–18
41. Song X, Zhang W, Weng J (2015) Types, locations, and scales from cluttered natural video and actions. IEEE Trans Auton Ment Dev 7(4):273
42. Wang D, Wang J, Liu L (2017) Developmental network: an internal emergent object feature learning. Neural Process Lett 1–25

**Ali Javed** received the B.Sc. degree with honors in Software Engineering from UET Taxila, Pakistan in 2007. He got 3rd position in Software Batch-2003F. He received his MS and Ph.D. degrees in Computer Engineering from UET Taxila, Pakistan in 2010 and 2016. He received Chancellor's Gold Medal in MS Computer Engineering degree. Currently, he is serving as an Assistant Professor in the Department of Software Engineering at UET Taxila, Pakistan. Dr. Javed has served as a visiting PhD research scholar at ISSF Lab in University of Michigan, USA in 2015. He was awarded HEC scholarship to pursue his Ph.D. research work at University of Michigan, USA. His areas of interest are Digital Image Processing, Computer vision, Video Content Analysis, Medical Imaging, Machine Learning, Multimedia Signal Processing, Software Quality Assurance and Testing.

Dr. Javed is a recipient of various research grants from HEC Pakistan, National ICT R n D Fund Pakistan and UET Taxila Pakistan. He has also served as an HOD in Software Engineering Department at UET Taxila, Pakistan in 2014. Dr. Javed got selected as an Ambassador of Asian Council of Science Editors from Pakistan in 2016. He is also a member of Pakistan Engineering Council since 2007.

**Aun Irtaza** has completed his PhD from FAST-National University of Computers & Emerging Sciences in 2016. During his PhD he remained working as a research scientist in the Signal and image processing lab in Gwangju Institute of Science and Technology (GIST) South Korea. His research interests include computer vision, pattern analysis, and big data analytics.

**Yasmeen Khaliq** is currently enrolled in Ph.D. from University of Engineering and Technology Taxila. Her major research area is Machine Learning and Image Processing. She has also worked on data mining, sentiment analysis, Hadoop and cloud computing. Her major research interests in Ph.D. is Ensembled Learning and Autonomous Vehicles. She is also working as a Lecturer at Comsats University Islamabad, Wah Campus.

**Hafiz Malik** received the B.E. degree in electronics and communications engineering (with distinction) from the University of Engineering and Technology Lahore, Pakistan, in 1999 and the Ph.D. degree in electrical and computer engineering from the University of Illinois, Chicago, in 2006. After the Ph.D. degree, he joined the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, where he worked as a Postdoctoral Research Fellow. Currently, he is serving as an Associate Professor in ECE Department at University of Michigan Dearborn, MI. His research interests are in the general areas of digital content protection and digital signal processing, and the focus of current research includes information security, steganography, steganalysis, statistical signal processing, audio analysis/synthesis, and digital forensic analysis. He has published more than 15 technical papers and book chapters in refereed conferences and journals in the area of multimedia security, steganography, steganalysis, multimedia processing, audio analysis/synthesis, and statistical signal processing. Dr. Malik has served as organizing committee of the special track on Doctoral Dissertation in the IEEE International Symposium on Multimedia (ISM) 2006. He was a member of technical program committees of several conferences.

**Muhammad Tariq Mahmood** received the MS degree in Computer Science from Blekinge Institute of Technology, Sweden in 2006. He received the Ph.D. degree in Informatics and Mechatronics from Gwangju University of Science and Technology, Republic of Korea in 2011. Currently, he is serving as Assistant Professor at School of computer science and engineering, Korea University of Technology and Education. His research interests include Image Processing, Machine Learning and Pattern Recognition.

## Affiliations

**Ali Javed**[1] · **Aun Irtaza**[2] · **Yasmeen Khaliq**[3] · **Hafiz Malik**[4] · **Muhammad Tariq Mahmood**[5]

Ali Javed
ali.javed@uettaxila.edu.pk

Aun Irtaza
aun.irtaza@uettaxila.edu.pk

Yasmeen Khaliq
yasmeen@ciitwah.edu.pk

Hafiz Malik
hafiz@umich.edu

[1] Software Engineering Department, University of Engineering and Technology, Taxila, 47050, Pakistan

[2] Computer Science Department, University of Engineering and Technology, Taxila, 47050, Pakistan

[3] Computer Science Department, COMSATS University Islamabad, Wah Campus, GT Road, Wah Cantt, 47040, Pakistan

[4] College of Engineering and Computer Science, University of Michgan-Dearborn, 4901 Evergreen Rd, Dearborn, MI 48128, USA

[5] School of Computer Science and Engineering, Korea University of Technology and Education, 1600, Chungjeol-ro, Byeongcheon-myeon, 31253, Cheonan, South Korea