Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Shot classification and replay detection for sports video summarization*

Ali JAVED†‡, Amen ALI KHAN

Department of Software Engineering, University of Engineering and Technology, Taxila 47050, Pakistan

†E-mail: ali.javed@uettaxila.edu.pk

Received Aug. 16, 2020; Revision accepted Mar. 25, 2021; Crosschecked Feb. 23, 2022

Abstract: Automated analysis of sports video summarization is challenging due to variations in cameras, replay speed, illumination conditions, editing effects, game structure, genre, etc. To address these challenges, we propose an effective video summarization framework based on shot classification and replay detection for field sports videos. Accurate shot classification is mandatory to better structure the input video for further processing, i.e., key events or replay detection. Therefore, we present a lightweight convolutional neural network based method for shot classification. Then we analyze each shot for replay detection and specifically detect the successive batch of logo transition frames that identify the replay segments from the sports videos. For this purpose, we propose local octa-pattern features to represent video frames and train the extreme learning machine for classification as replay or non-replay frames. The proposed framework is robust to variations in cameras, replay speed, shot speed, illumination conditions, game structure, sports genre, broadcasters, logo designs and placement, frame transitions, and editing effects. The performance of our framework is evaluated on a dataset containing diverse YouTube sports videos of soccer, baseball, and cricket. Experimental results demonstrate that the proposed framework can reliably be used for shot classification and replay detection to summarize field sports videos.

Key words: Extreme learning machine; Lightweight convolutional neural network; Local octa-patterns; Shot classification; Replay detection; Video summarization

https://doi.org/10.1631/FITEE.2000414 **CLC number:** TP391

1 Introduction

Sports broadcasters generate multimedia content exponentially these days. Effective management and handling of such massive multimedia content is very challenging. Manual video content analysis is a taxing activity. Therefore, automated video content analysis techniques have been proposed to effectively handle the enormous amount of multimedia content available in cyberspace. Additionally, viewers are unable to watch full-length sports broadcasts due to time constraints. This fact creates an urgent need to develop effective summarization solutions (Javed et al., 2016, 2019), which are capable of providing a brief summary of significant sports video events. Replays are usually displayed by broadcasters to show significant sports video events in slow motion. Therefore, replay detection (Javed et al., 2016) can be used to produce a video summary of significant events of the game.

Sports video content analysis requires effective segmentation and classification of shots into different views (i.e., long, medium, close-up, and out-field). Accurately classifying sports video shots into different views has potential benefits such as key event detection and video summarization. Existing works (Fani et al., 2017; Kapela et al., 2017; Minhas et al., 2019; Javed et al., 2020) employ various conventional

‡ Corresponding author

* Project supported by the Directorate of Advanced Studies, Research & Technological Development, University of Engineering and Technology Taxila (No. UET/ASRTD/RG-1002-3)

ORCID: Ali JAVED, https://orcid.org/0000-0002-1290-1477

machine learning and deep learning methods for classification of sports video shots. Shot classification methods have also employed various low-, mid-, or high-level features alone or in combination. Kapela et al. (2017) employed color and texture features using a radial-basis decomposition function along with the Gabor wavelet transform to train support vector machines (SVMs) for field sports scene classification. This method is well suited for real-time sports video scene classification, but the performance of event extraction can be further improved using additional audio-visual features. In our prior work (Javed et al., 2020), a decision tree architecture was employed to propose various rules that were then used to classify field sports video shots. Although this method provides better accuracy for shot classification in cricket videos, this technique is less effective to some extent for close-up shot classification due to lower face detection performance. This happens only in close-up shots that involve the face exposure of the batsman wearing a helmet. Many shot classification methods (Ekin et al., 2003; Raventós et al., 2015; Choroś and Gogol, 2016) use low-level features (e.g., grass-field color-pixel ratio and pitch-field pixel ratio) to classify shots into long, medium, and close-up ones. In Ekin et al. (2003), the grass-field color-pixel ratio was employed to classify video shots into long, medium, and close-up views. Specifically, the grass-field color-pixel ratio of selected regions was used to train the Bayesian network for classification of long and medium shots. The accuracy of this method can be improved by integrating more robust features. However, this approach is only tailored to soccer videos and is unable to generate summaries of different sports. In Choroś and Gogol (2016), a rule-based thresholding approach using the grass-field color-pixel ratio was proposed to detect long shots in soccer videos. This approach is limited to detecting only a single shot category. Similarly, the grass-field color-pixel ratio was employed in Raventós et al. (2015) for soccer video shot classification. Shot classification methods that use these low-level features operated with hard-coded thresholds often fail to achieve good performance in real-time conditions, e.g., variations in illumination and color similarity of foreground and grass-field. Moreover, the grass-field color-pixel ratio feature is unreliable for discrimination between long and medium shots because medium shots with a high grass-field color-pixel ratio are often misclassified as long shots. Low-level features have also been used with mid- or high-level features for shot classification of sports videos. In Tavassolipour et al. (2014), the grass-field color-pixel ratio was used in combination with player size for shot classification of soccer videos into long, medium, and close-up views. The performance of event detection can be enhanced to capture the temporal dependencies among various game events. Likewise, in Bagheri-Khaligh et al. (2012), low- and mid-level features were fused to classify soccer video shots. This technique achieves reasonable accuracy on high-resolution videos, but is unable to perform well on low-resolution videos.

Existing techniques (Wang DH et al., 2004; Jiang and Zhang, 2011; Kapela et al., 2017) use hybrid feature descriptors for effective classification of multiple shots in sports videos. In Wang DH et al. (2004), the grass-field color-pixel ratio was fused with motion features to train the C4.5 decision tree for shot classification. In Jiang and Zhang (2011), edge and optical flow based motion features were employed with an SVM to classify different types of tennis video shots. These techniques (Wang DH et al., 2004; Jiang and Zhang, 2011) are not robust to variations in the motion, resulting in misclassification of different shot categories. Existing shot classification approaches have employed various deep learning (DL) models for shot classification. Minhas et al. (2019) employed an eight-layer AlexNet convolutional neural network (CNN) to classify soccer and cricket videos into long, medium, close-up, and out-field views. Similarly, in Fani et al. (2017), a local and global feature fusion based deep learning model, using the camera's zoom and out-field information, was used to classify soccer video shots. A CNN-oriented model was employed in Tien et al. (2007) to classify basketball video shots into long and close-up views. These works (Tien et al., 2007; Fani et al., 2017) have limited applicability in terms of sports genre; classification for only soccer video shots was considered in Fani et al. (2017) and for only basketball in Tien et al. (2007).

Existing shot classification methods are limited in certain ways; e.g., they are genre-specific, computationally complex, and dependent on deviations in camera, illumination conditions, game structure, shot speed, occluded objects, broadcasters, etc. To better resolve these aforementioned challenges, we

propose an efficient lightweight CNN based method for classifying field sports video shots.

After shot segmentation and classification in sports videos, we can perform various tasks such as replay detection and/or key event detection for video summarization. The fact that broadcasters frequently use replays to repeat significant or key events of a game in slow motion during live game broadcasts motivated us to detect replays for summarization of sports videos. Existing methods (Pan et al., 2001; Choroś and Gogol, 2016) use only replays to produce sports video summaries. Approaches based on frame motion (Pan et al., 2001, 2002; Duan et al., 2004; Wang L et al., 2004) and frame logo transitions (Wang JJ et al., 2005; Eldib et al., 2009; Xu and Yi, 2011; Zhao F et al., 2012; Su et al., 2013) have been designed for replay detection in the past. Pan et al. (2001) employed the hidden Markov model and Viterbi algorithm to identify movements in video frames for replay detection. Duan et al. (2004) proposed a mean shift oriented logo detection technique using motion features. Frame motion based methods (Pan et al., 2001; Duan et al., 2004) are dependent on the replay speed and are unable to accurately detect replays when the replay speed varies. Choroś and Gogol (2016) employed a threshold-based method using the difference in average contrast values between consecutive frames to detect the logo frames. These methods (Pan et al., 2001; Choroś and Gogol, 2016) experience performance degradation under certain conditions, e.g., deviations in shape, color, and logo (design, size, and position). Zhao Z et al. (2006) employed speeded up robust features (SURF) for logo pattern detection in sports videos. Statistical features were used to identify the logo frames for replay detection (Eldib et al., 2009; Xu and Yi, 2011; Chen and Chen, 2015). Similarly, logo transition oriented methods were used (Dang et al., 2007; Li et al., 2009; Nguyen and Yoshitaka, 2012; Chen and Chen, 2014). Logo frame detection simplifies the process of replay detection; however, the replay detection performance depends on accurate logo detection, which is very challenging because of the huge diversity in logos (e.g., color, shape, design, size, and placement) used by different broadcasters and sports. Furthermore, logo transition oriented methods depend on the replay structure and are unable to accurately detect replays in multiple sports due to the variations in structure and representation of the replays among different sports.

In our prior work (Javed et al., 2016), we employed a thresholding-based approach to detect the gradual transitions that were then used to identify the candidate replay segments. Later, score-caption presence/absence detection was employed for replay detection. This method achieves remarkable results in daylight videos, but is unable to perform well on artificial light videos. Additionally, this method is unable to detect replays where score-captions are not removed from the video during replays.

Existing replay detection oriented summarization approaches have various limitations, e.g., computational complexity of logo detection, dependency on logos (size, design, and position), variations in replay speed and frame transition, and dependency on editing effects. To address the aforementioned challenges, we propose an effective shot classification method and a replay detection method to summarize field sports videos. The main contributions of the proposed research work are as follows:

1. We present an effective and lightweight shot classification method that can reliably be used to classify shots into long, medium, close-up, and outfield views.

2. We propose a feature descriptor, local octapattern (LoP), for effective representation of video frames.

3. We present an effective replay detection based video summarization method that can accurately classify the replay and non-replay (live) frames in sports videos.

4. The proposed framework is robust to variations in camera, illumination conditions, editing effects, frame transitions, sports genre, shot length, shot type, broadcasters, logos (size, design, and position), etc.

2 Proposed framework

This section provides a discussion of the proposed shot classification and replay detection methods for sports video summarization. The architecture of the proposed framework is provided in Fig. 1. The details of each method are provided in the subsequent sections.

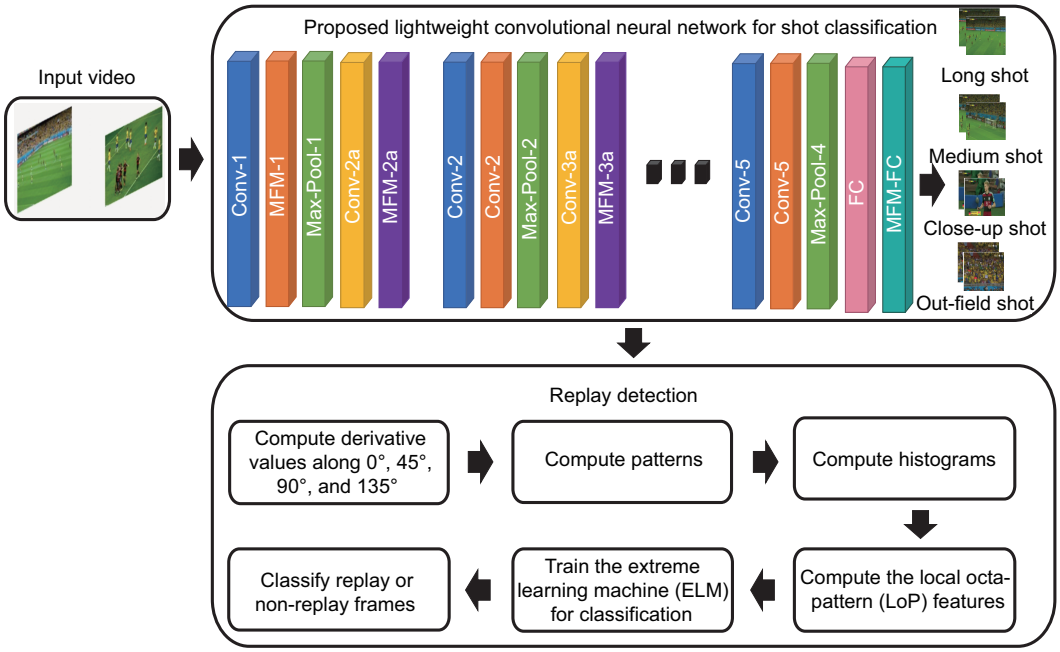

Fig. 1 Architecture of the proposed framework

2.1 Shot classification

The proposed shot classification method employs the lightweight CNN deep learning framework to categorize the shots into long, medium, close-up, and out-field ones. We employ the lightweight CNN deep learning model to develop a shot classification method for two reasons. First, the lightweight CNN model is computationally fast and well suited for real-time video processing. Second, lightweight CNN employs the maximum feature map (MFM) activation function, which is more robust to noise and can accurately separate the noise and informative content. Because sports videos contain various types of noise, the MFM-based lightweight CNN framework is used to better address this limitation. The details of the lightweight CNN framework used for shot classification are explained in the following.

The lightweight CNN method used for shot classification employs a nine-layer lightweight CNN framework comprising five convolutional layers, four network-in-network (NIN) layers, six MFM layers, four maximum pooling layers, and one fully connected layer. In the proposed architecture, we define the input layer as preprocessing where input frames are down-sampled to 128×128 pixels to reduce the computational cost. We use five convolutional layers, where we employ 96 5×5 kernels in the first convolutional layer, 192 3×3 kernels in the second convolutional layer, 384 3×3 kernels in the third convolutional layer, and 256 3×3 kernels each in the remaining two convolutional layers. Additionally, we adopt a stride of one for each convolutional layer. We use MFM activation and a maximum pooling layer after each convolutional layer. For each maximum pooling layer, we use a 2×2 kernel with a stride of two as well. The facts that NIN can perform feature selection between the convolutional layers and that the number of parameters can be reduced using small convolution kernels motivate us to integrate NIN along with a small-size convolution kernel and MFM. Specifically, we use a 1×1 convolution kernel before every convolutional layer except the first one. Finally, we use the fully connected layer followed by MFM activation to classify the shots into long, medium, close-up, and out-field ones.

Because sports videos contain noisy frames, we need a more robust method that can accurately classify the shots in the presence of noise in the video frames. To ensure that the CNN does not learn a biased result, errors produced by these noisy patterns must be handled using a robust activation function in CNN. The rectified linear unit (ReLU) activation function is employed in CNNs to segregate the noisy content from the informative content via a threshold that decides the outcome of a neuron, i.e.,

active or non-active. It has been observed that this threshold often results in loss of some informative content, particularly for initial convolutional layers (Wu et al., 2018). To address this issue, we employ the MFM activation function, which uses a competitive relationship to suppress only a few neurons, thus making the CNN light and robust. The benefits of MFM activation are threefold: (1) effective segregation of informative and noisy signals, (2) feature selection, and (3) developing an efficient model. The details of our lightweight CNN model are shown in Table 1.

Table 1 Proposed lightweight CNN model

Layer	Filter size	Stride/Pad	Output size
Conv-1	5×5	1/2	128×128×96
MFM	–	–	128×128×48
Max-Pool-1	2×2	2	64×64×48
Conv-2a	1×1	1	64×64×96
MFM-2a	–	–	64×64×48
Conv-2	3×3	1/1	64×64×192
MFM-2	–	–	64×64×96
Max-Pool-2	2×2	2	32×32×48
Conv-3a	1×1	1	32×32×192
MFM-3a	–	–	32×32×96
Conv-3	3×3	1/1	32×32×384
MFM-3	–	–	32×32×192
Max-Pool-3	2×2	2	16×16×192
Conv-4a	1×1	1	16×16×384
MFM-4a	–	–	16×16×192
Conv-4	3×3	1/1	16×16×256
MFM-4	–	–	16×16×128
Conv-5a	1×1	1	16×16×256
MFM-5a	–	–	16×16×128
Conv-5	3×3	1/1	16×16×256
MFM-5	–	–	16×16×128
Max-Pool-4	2×2	2	8×8×128
FC-1	–	–	512
MFM-FC-1	–	–	256

2.2 Replay detection

Broadcasters use slow-motion replay segments in live broadcasts after any key event in a game. Replay frames are sandwiched between logo frames. We exploit this fact to detect the logo transition frames in the first step. Next, the frames between two consecutive batches of logo frames are selected as the replay frames. Note that our video dataset comprises replays where broadcasters have used different chromatic, spatial, and chromatic–spatial effects during replays. Moreover, different tournaments in each sports category have employed distinct logos in replays that are diverse in terms of color, shape, size, spatial arrangements, placement in the frame, etc. This demands the development of a robust feature descriptor that can extract the relevant information from such diverse logo frames. For this purpose, we propose the local octa-patterns (LoP) descriptor by extending the local tetra-pattern (LTrP) descriptor (Murala et al., 2012). Our LoP features can capture the traits of various logo frames containing chromatic and spatial variations, because they encode the texture information of more neighboring pixels as they consider more directions while computing the derivatives as compared to existing local patterns (e.g., LTrP). We represent the frames using an LoP descriptor and train the extreme learning machine (ELM) for classification. The details of LoP feature computation and ELM classification are presented in the following subsections.

2.2.1 Feature extraction

Because sports videos contain varying illumination conditions and various textures, especially in the replay frames where broadcasters use different chromatic–spatial effects and diverse logos (size, placement, shape, colors, etc.), we propose the LoP features to effectively represent the given input frame for better classification.

For LoP feature representation of the given image I, the first-order derivatives along 0°, 45°, 90°, and 135° are represented as $I_1^{\theta}(p_n)$, where $\theta = 0°$, 45°, 90°, 135°. Let p_c, p_v, and p_h represent the center, vertical, and horizontal pixels, respectively, in I. We can represent the first-order derivatives at p_c as Eqs. (1)–(5) where Eq. (5) is presented on the top of the next page.

$$I_1^{0°}(p_c) = I(p_h) - I(p_c), \tag{1}$$

$$I_1^{90°}(p_c) = I(p_v) - I(p_c), \tag{2}$$

$$I_1^{45°}(p_c) = I(p_{d_1}) - I(p_c), \tag{3}$$

$$I_1^{135°}(p_c) = I(p_{d_2}) - I(p_c), \tag{4}$$

where d_1 and d_2 refer to the directions of 45° and 135°, respectively.

After computing the direction of the referenced pixel, the n^{th}-order LoP can be represented as

$$I_1^d(p_c)=\begin{cases}1, & \left(I_1^{0^\circ}(p_c)\geq 0\right)\wedge\left(I_1^{90^\circ}(p_c)\ \geq 0\right)\wedge\left((I_1^{45^\circ}(p_c)\vee I_1^{135^\circ}(p_c))\geq 0\right),\\ 2, & \left(I_1^{0^\circ}(p_c)<0\right)\wedge\left(I_1^{90^\circ}(p_c)\ \geq 0\right)\wedge\left((I_1^{45^\circ}(p_c)\vee I_1^{135^\circ}(p_c))\geq 0\right),\\ 3, & \left(I_1^{0^\circ}(p_c)<0\right)\wedge\left(I_1^{90^\circ}(p_c)\ <0\right)\wedge\left((I_1^{45^\circ}(p_c)\vee I_1^{135^\circ}(p_c))\geq 0\right),\\ 4, & \left(I_1^{0^\circ}(p_c)\geq 0\right)\wedge\left(I_1^{90^\circ}(p_c)\ <0\right)\wedge\left((I_1^{45^\circ}(p_c)\vee I_1^{135^\circ}(p_c))\geq 0\right),\\ 5, & \left(I_1^{0^\circ}(p_c)\geq 0\right)\wedge\left(I_1^{90^\circ}(p_c)\ \geq 0\right)\wedge\left((I_1^{45^\circ}(p_c)\vee I_1^{135^\circ}(p_c))<0\right),\\ 6, & \left(I_1^{0^\circ}(p_c)<0\right)\wedge\left(I_1^{90^\circ}(p_c)\ \geq 0\right)\wedge\left((I_1^{45^\circ}(p_c)\vee I_1^{135^\circ}(p_c))<0\right),\\ 7, & \left(I_1^{0^\circ}(p_c)<0\right)\wedge\left(I_1^{90^\circ}(p_c)\ <0\right)\wedge\left((I_1^{45^\circ}(p_c)\vee I_1^{135^\circ}(p_c))<0\right),\\ 8, & \left(I_1^{0^\circ}(p_c)\geq 0\right)\wedge\left(I_1^{90^\circ}(p_c)\ <0\right)\wedge\left((I_1^{45^\circ}(p_c)\vee I_1^{135^\circ}(p_c))<0\right).\end{cases}\tag{5}$$

follows:

$$\mathrm{LoP}_n(p_c)=\Big(f_1\left(I_1^d(p_c),I_1^d(p_1)\right),f_1\left(I_1^d(p_c),I_1^d(p_2)\right),\ldots,f_1\left(I_1^d(p_c),I_1^d(p_8)\right)\Big),\tag{6}$$

where $(m=1,2,\ldots,8)$

$$f_1(I_1^d(p_c),I_1^d(p_m))=\begin{cases}0, \text{ if } I_1^d(p_c)=I_1^d(p_m),\\ I_1^d(p_m), \text{ otherwise.}\end{cases}\tag{7}$$

From Eq. (6), we obtain the LoP code that is further used to obtain seven 8-bit binary patterns based on the direction of p_c:

$$\mathrm{LoP}^n_{p\{\overline{D}|\forall D,\ni\neg I_1^d(p_c)\}}=f_2\left(\mathrm{LoP}^n_{p\{\overline{D}|\forall D,\ni\neg I_1^d(p_c)\}}\right),\tag{8}$$

where

$$f_2\left(\mathrm{LoP}^n_p(p_c)_{\{\vec{D}\in\overline{D}\}}\right)=\begin{cases}1, \text{ if } \mathrm{LoP}^n_p(p_c)=\vec{D},\\ 0, \text{ otherwise,}\end{cases}\tag{9}$$

where $\overline{D}$ represents the set of all quadrants except the quadrant of the reference pixel and $\vec{D}$ denotes one of the quadrants of $\overline{D}$. Finally, we generate the LoP code as

$$\mathrm{LoP}^n_{p\{\overline{D}|\forall D,\ni\neg I_1^d(p_c)\}}=\sum_{p=1}^{P}2^{p-1}\cdot f_2\left(\mathrm{LoP}^n_{p\{\overline{D}|\forall D,\ni\neg I_1^d(p_c)\}}\right).\tag{10}$$

Similarly, we obtain the 8-bit binary pattern for each of the remaining seven directions with respect to the central pixel. The magnitude pattern is merged with all of the binary patterns computed for different directions:

$$\mathrm{MP}=\sum_{n=1}^{N}2^{n-1}\cdot f_3\left(M_{I(p_m)}-M_{I(p_c)}\right),\tag{11}$$

where

$$M_{I(p_m)}=\Big(\sum\Big((I^{0^\circ}_{n-1}(p_m))^2+(I^{45^\circ}_{n-1}(p_m))^2+(I^{90^\circ}_{n-1}(p_m))^2+(I^{135^\circ}_{n-1}(p_m))^2\Big)\Big)^{1/2}.\tag{12}$$

Fig. 2 shows the local octa-patterns for a central pixel (highlighted in green) around the neighboring pixels (highlighted in red). $D(c)$ represents the central direction, whereas $D(1)$ to $D(8)$ represent the directions of the eight neighboring pixels. Similarly, $M(c)$ represents the central pixel magnitude, whereas $M(1)$ to $M(8)$ represent the magnitudes of neighboring pixels. If the directions of the central and neighboring pixel are the same, then the LoP bit is set to 0; otherwise, the LoP bit is set to the direction of the neighboring pixel. In the example provided in Fig. 2, we obtain the LoP of 41383183. Later, we divide this pattern into seven binary patterns where the first pattern is obtained by replacing 2 with 1 and setting the remaining values to 0 in the LoP. The second pattern is obtained by replacing 3 with 1 and setting the rest of the values with 0. Similar operations are done for the remaining patterns. We determine the magnitude pattern by comparing the magnitude values of neighboring pixels with that of the central pixel. If the magnitude of a neighboring pixel is greater than that of the central pixel, then the corresponding bit of the magnitude pattern is set to 1; otherwise, it is set to 0. In our example shown in Fig. 2, we obtain the pattern 11100001 because the values of $M(1)$, $M(2)$, $M(3)$, and $M(8)$ are greater than $M(c)$. These eight patterns are then used to depict the texture of an image. Although we expect the higher-order octa-patterns to retrieve more information, our experimental analysis reveals the superiority of the second-order octa-patterns over others.

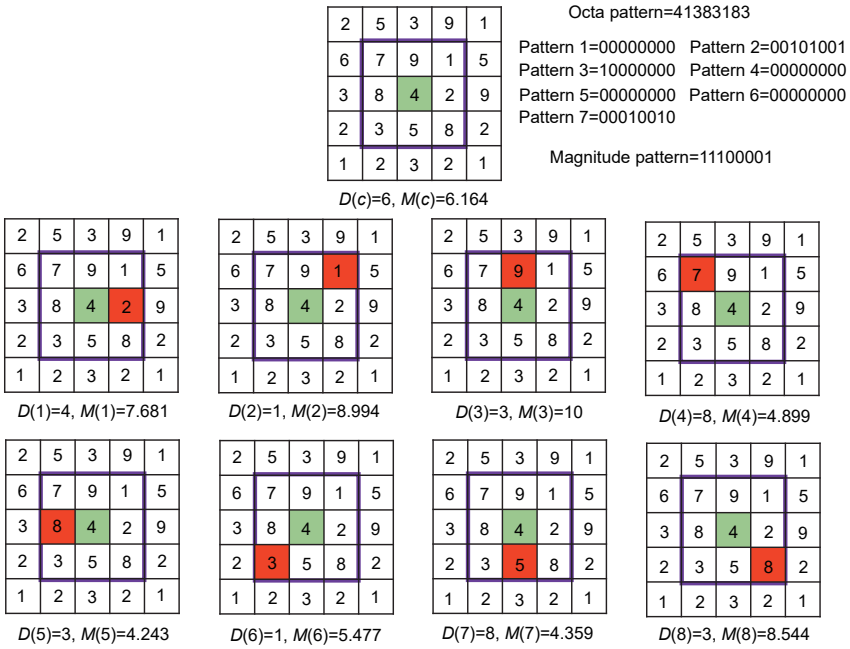

Fig. 2 Local octa-pattern computation (References to color refer to the online version of this figure)

2.2.2 Classification

In the proposed work, we employ the LoP feature descriptor to represent the replay and non-replay frames and train the ELM for classification. We adopt the ELM for classification because it has the smallest training error and norm of output weights. The ELM was initially developed for the single hidden layer feed-forward neural network, where the hidden layer is not required as it is in a neuron. For binary classification, the output function of the ELM for a single output unit is represented as

$$f_z(x) = \sum_{z=1}^{Z} \beta_z h_z(x) = \boldsymbol{h}(x)\boldsymbol{\beta}, \qquad (13)$$

where $\boldsymbol{\beta} = [\beta_1, \beta_2, ..., \beta_Z]^{\mathrm{T}}$ represents the vector of the output weights between the hidden layer of Z nodes and the output node; $\boldsymbol{h}(x) = [h_1(x), h_2(x), ..., h_Z(x)]$ represents the output vector of the hidden layer with respect to the input x. For binary classification, we compute the decision function of ELM as

$$f_z(x) = \sin(\boldsymbol{h}(x)\boldsymbol{\beta}). \qquad (14)$$

Finally, these replay frames are used to present a summary of key events that occurred in the game.

3 Experimental results and discussion

In this section, we provide the experimental results to evaluate the performance of our framework. The details of the dataset are also provided. We employ the precision, recall, F1-score, accuracy, and error rate metrics for performance evaluation, as also adopted by the comparative methods.

3.1 Dataset

To evaluate the performance of the proposed framework, we used YouTube sports videos of three kinds of sports genre (soccer, baseball, and cricket); YouTube videos were also adopted by the comparative approaches (Choroś and Gogol, 2016; Javed et al., 2016, 2019, 2020; Fani et al., 2017; Kapela et al., 2017; Minhas et al., 2019) for performance evaluation. Our dataset (https://datadryad.org/stash/share/rG2gQHc23ElD9xpxyMXL7gdY7ys_USqPM9XFm3P6uC8) includes 50 videos of 100-hour duration. We selected a diverse collection of field sports videos in terms of field illumination conditions, shot and replay types, replay speed, length, genre, editing effects, events, broadcasters, etc. Moreover, replay frames in our sports videos contain distinct logos that have various colors, sizes, shapes, placements, and chromatic–spatial effects.

We used the videos of six renowned broadcasters from Entertainment and Sports Programming Network (ESPN), Star Sports, Ten Sports, Sky Sports, Fox Sports, and Euro Sports. We also selected the videos from different tournaments for each sports genre. For soccer videos, we selected the games of 2014 and 2018 World Cup tournaments. For baseball videos, we selected Major League Baseball games of 2015 and the Peach Belt Conference (PBC) baseball tournament of 2019. For cricket videos, we selected games from six different tournaments. We selected videos from each of these three cricket formats: One Day International (ODI), Test Match, and Twenty-20. From the ODI format, we selected the 2006 bilateral series between South Africa and Australia, and the 2007 bilateral ODI tournament between South Africa and New Zealand. From the Test Match format, we selected the 2018 bilateral tournament between Pakistan and Australia. Finally, from the Twenty-20 format, we selected the videos from the 2010, 2014, and 2016 World Cup tournaments. We used 70% of the frames in our dataset for training and the rest for testing. Fig. 3 shows a few live and replay images in each sports category (i.e., cricket, soccer, and baseball) of our dataset.

3.2 Performance evaluation of the shot classification method

We designed an experiment to evaluate the performance of our lightweight CNN deep learning based shot classification method. We trained our model using cricket, baseball, and soccer sports video frames to classify the shots into long, medium, close-up, and out-field views. The results of our shot classification method for each kind of sports videos are presented in Fig. 4. For soccer videos, we obtained 94.3% precision, 96.7% recall, 95.5% F1-score, 97% accuracy, and 3% error rate. For baseball videos, we obtained 95.3% precision, 96.5% recall, 95.9% F1-score, 97.2% accuracy, and 2.8% error rate. For cricket videos, we obtained 95.9% precision, 96.7% recall, 96.3% F1-score, 97.4% accuracy, and 2.6% error rate. The proposed shot classification method has the best detection performance on cricket videos and marginally lower detection performance on baseball and soccer videos. Our method has remarkable detection performance on different field sports videos. On average, the proposed shot classification method achieves 95.2% precision, 96.6% recall, 95.9% F1-score, 97.2% accuracy, and 2.8% error rate. From these results, we argue that our shot classification method can be effectively used to classify sports video shots.

We also designed an experiment to illustrate the effectiveness of the MFM activation function in our lightweight CNN model for better shot classification of noisy videos. For this purpose, we selected three videos in each sports category that contained a moderate level of noise in the frames. In the first stage of this experiment, we fed these videos

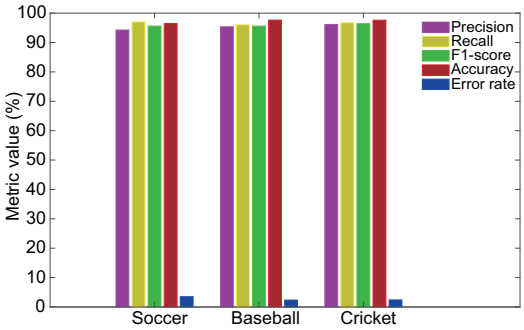

Fig. 4 Performance of shot classification (References to color refer to the online version of this figure)

Fig. 3 Live and replay frames in sports videos of our dataset

to the lightweight CNN model using MFM activation, and performed classification of shots into long, medium, close-up, and out-field views. On average, we achieved 93.3% precision, 94.1% recall, 93.7% F1-score, 95.3% accuracy, and 4.7% error rate. In the second stage, we replaced the MFM activation function with ReLU while keeping the same CNN architecture. On average, we obtained 89.44% precision, 90.12% recall, 89.78% F1-score, 90.91% accuracy, and 9.09% error rate. These results revealed that the MFM activation function is more robust to noisy conditions than other activation functions (e.g., ReLU). From the results of this experiment, we can conclude that the MFM activation function in the CNN model is more effective for real-world sports videos containing different degradations such as noise.

3.3 Performance evaluation of the replay detection method

We designed this experiment to evaluate the performance of the proposed replay detection method. For this purpose, we represented the replay and non-replay frames of input sports videos through LoP, and trained the ELM to classify the frames into replay and non-replay ones. Again, we selected soccer, baseball, and cricket videos for experimentation. The results of our replay detection method on each kind of sports videos are presented in Fig. 5. For soccer videos, we obtained 96.9% precision, 97% recall, 96.95% F1-score, 96.7% accuracy, and 3.3% error rate. For baseball videos, we obtained 95.3% precision, 98.1% recall, 96.7% F1-score, 96.2% accuracy, and 3.8% error rate. For cricket videos, we obtained 96.7% precision, 95.7% recall, 96.2% F1-score, 97.6% accuracy, and 2.4% error rate. On average, our replay detection method achieves 96.3% precision, 96.9% recall, 96.6% F1-score, 96.8% accuracy, and 3.2% error rate. These results verify that our LoP features effectively capture the patterns of diverse logo frames that include different chromatic and spatial variations. Thus, we conclude from this experiment that the proposed LoP features effectively represent the input video frames and make them more suitable for ELM to better classify replay and non-replay frames. Therefore, our replay detection technique can be reliably used to detect replays in sports videos of different genres.

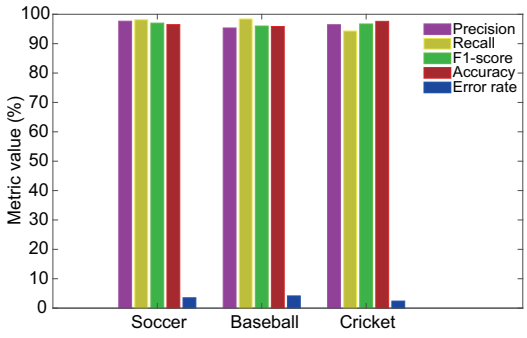

Fig. 5 Performance of replay detection (References to color refer to the online version of this figure)

3.4 Performance comparison with existing methods

In this experiment, we compared the performance of our framework with that of existing state-of-the-art summarization methods for sports videos. First, we compared the performance of our shot classification method with other comparative shot classification methods (Fani et al., 2017; Kapela et al., 2017; Minhas et al., 2019; Javed et al., 2020), and the results are reported in Table 2. For shot classification, the method proposed by Kapela et al. (2017) has the worst performance based on the lowest precision (82.5%) and recall (84.2%), whereas our prior work (Javed et al., 2020) performs the second best and achieved 94.6% precision and 96.2% recall. The proposed shot classification method performs marginally better than our prior work (Javed et al., 2020) and significantly better than other comparative methods (Fani et al., 2017; Kapela et al., 2017; Minhas et al., 2019). Our prior work on shot classification (Javed et al., 2020) has low accuracy for close-up shots in cricket videos due to misdetection of the batsmen's faces, which are obscured by helmets. The proposed lightweight shot classification method successfully addressed this limitation, where deep features better capture the information available in the frames against each view including the close-up view. From this comparative analysis, we can clearly see that our shot classification method provides superior detection performance and can be reliably used to classify sports video shots.

Second, Table 3 provides a comparison of our replay detection method with existing replay detection methods (Choroś and Gogol, 2016; Javed et al., 2016, 2019). For replay detection, the method proposed in

Table 2 Comparative analysis of shot classification methods

Shot classification method	Sports genre	Precision (%)	Recall (%)
Kapela et al. (2017)	Field sports	82.5	84.2
Javed et al. (2020)	Field sports	94.6	96.2
Minhas et al. (2019)	Field sports	90.6	91.3
Fani et al. (2017)	Soccer	90.6	91.3
Proposed method	Field sports	95.2	96.6

Table 3 Comparative analysis of replay detection methods

Replay detection method	Sports genre	Precision (%)	Recall (%)
Javed et al. (2019)	Field sports	95.09	95.94
Javed et al. (2016)	Field sports	92.97	94.70
Choroś and Gogol (2016)	Field sports	86.19	64.35
Proposed method	Field sports	96.30	96.90

Choroś and Gogol (2016) achieves the lowest precision (86.19%) and recall (64.35%). Our previous method (Javed et al., 2019) performs the second best and achieved 95.09% precision and 95.94% recall. The proposed replay detection method performs the best by obtaining the highest precision (96.3%) and recall (96.9%) among all the comparative replay detection methods. Note that the proposed replay detection method is independent of the absence/presence of score captions, unlike our previous replay detection framework (Javed et al., 2016). The dependency of our prior work (Javed et al., 2016) on score captions is unable to achieve good results on videos where score captions are not omitted by broadcasters during replay segments. However, the proposed replay detection method is robust to the presence/absence of score captions and other editing effects, and can successfully detect replay segments from sports videos. From this comparative analysis, we can clearly observe that our replay detection method provides superior detection performance under diverse conditions and editing effects, and can be reliably used to detect replays in sports videos.

4 Conclusions

In this paper we have presented an effective summarization framework based on shot classification and replay detection for field sports videos. Our lightweight CNN based shot classification method successfully addresses the limitation of our prior work, which was unable to accurately detect close-up shots in cricket videos where batsmen wear helmets. Additionally, we have detected the batch of successive logo frame sets to detect replays. We proposed the local octa-pattern features to represent the frames and employed the ELM for classification. Our replay detection method successfully addresses the limitation of our prior replay detection method, which depends on score captions for replay detection. The proposed framework is robust to variations in camera, replay speed, shot speed, illumination conditions, game structure, sports genre, broadcasters, logo design and placement, frame transitions, editing effects, score captions, etc. The performance of our framework has been evaluated using a YouTube sports video dataset that includes videos with varying illumination conditions, shot and replay types, replay speed, length, genre, editing effects, events, broadcasters, logo design, size and placement, etc. The effectiveness of our framework has been illustrated in terms of average precision and recall of 95.2% and 96.6% for shot classification, and 96.3% and 96.9% for replay detection, respectively. The possibility exists for incorrect replay detection due to unsuccessful detection of successive logo frame sets. In the future, we plan to further enhance the performance of our replay detection method.

Contributors

Ali JAVED developed the proposed method and designed the research. Amen ALI KHAN collected and processed the dataset. Ali JAVED and Amen ALI KHAN wrote the code, performed the experimentation, and drafted the paper. Ali JAVED revised and finalized the paper.

Compliance with ethics guidelines

Ali JAVED and Amen ALI KHAN declare that they have no conflict of interest.

References

Bagheri-Khaligh A, Raziperchikolaei R, Moghaddam ME, 2012. A new method for shot classification in soccer sports video based on SVM classifier. Proc IEEE Southwest Symp on Image Analysis and Interpretation, p.109-112. https://doi.org/10.1109/SSIAI.2012.6202465

Chen CM, Chen LH, 2014. Novel framework for sports video analysis: a basketball case study. Proc Int Conf on Image Processing, p.961-965. https://doi.org/10.1109/ICIP.2014.7025193

Chen CM, Chen LH, 2015. A novel method for slow motion replay detection in broadcast basketball video. *Multimed Tools Appl*, 74(21):9573-9593. https://doi.org/10.1007/s11042-014-2137-5

Choroś K, Gogol A, 2016. Improved method of detecting replay logo in sports videos based on contrast feature and histogram difference. Proc 8th Int Conf on Computational Collective Intelligence, p.542-552. https://doi.org/10.1007/978-3-319-45243-2_50

Dang ZH, Du J, Huang QM, et al., 2007. Replay detection based on semi-automatic logo template sequence extraction in sports video. Proc 4th Int Conf on Image and Graphics, p.839-844. https://doi.org/10.1109/ICIG.2007.73

Duan LY, Xu M, Tian Q, et al., 2004. Mean shift-based video segment representation and applications to replay detection. Proc 29th IEEE Int Conf on Acoustics, Speech, and Signal Processing, p.709-712. https://doi.org/10.1109/ICASSP.2004.1327209

Ekin A, Tekalp AM, Mehrotra R, 2003. Automatic soccer video analysis and summarization. *IEEE Trans Image Process*, 12(7):796-807. https://doi.org/10.1109/TIP.2003.812758

Eldib MY, Zaid BSA, Zawbaa HM, et al., 2009. Soccer video summarization using enhanced logo detection. Proc 16th IEEE Int Conf on Image Processing, p.4345-4348. https://doi.org/10.1109/ICIP.2009.5413649

Fani M, Yazdi M, Clausi DA, et al., 2017. Soccer video structure analysis by parallel feature fusion network and hidden-to-observable transferring Markov model. *IEEE Access*, 5:27322-27336. https://doi.org/10.1109/ACCESS.2017.2769140

Javed A, Bajwa KB, Malik H, et al., 2016. An efficient framework for automatic highlights generation from sports videos. *IEEE Signal Process Lett*, 23(7):954-958. https://doi.org/10.1109/LSP.2016.2573042

Javed A, Irtaza A, Khaliq Y, et al., 2019. Replay and keyevents detection for sports video summarization using confined elliptical local ternary patterns and extreme learning machine. *Appl Intell*, 49(8):2899-2917. https://doi.org/10.1007/s10489-019-01410-x

Javed A, Malik KM, Irtaza A, et al., 2020. A decision tree framework for shot classification of field sports videos. *J Supercomput*, 76(9):7242-7267. https://doi.org/10.1007/s11227-020-03155-8

Jiang H, Zhang M, 2011. Tennis video shot classification based on support vector machine. Proc IEEE Int Conf on Computer Science and Automation Engineering, p.757-761. https://doi.org/10.1109/CSAE.2011.5952612

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. https://doi.org/10.1007/s11554-014-0437-7

Li W, Chen SJ, Wang HB, 2009. A rule-based sports video event detection method. Proc Int Conf on Computational Intelligence and Software Engineering, p.1-4. https://doi.org/10.1109/CISE.2009.5366226

Minhas RA, Javed A, Irtaza A, et al., 2019. Shot classification of field sports videos using AlexNet convolutional neural network. *Appl Sci*, 9(3):483. https://doi.org/10.3390/app9030483

Murala S, Maheshwari RP, Balasubramanian R, 2012. Local tetra patterns: a new feature descriptor for content-based image retrieval. *IEEE Trans Image Process*, 21(5):2874-2886. https://doi.org/10.1109/TIP.2012.2188809

Nguyen N, Yoshitaka A, 2012. Shot type and replay detection for soccer video parsing. Proc IEEE Int Symp on Multimedia, p.344-347. https://doi.org/10.1109/ISM.2012.69

Pan H, Van Beek P, Sezan MI, 2001. Detection of slow-motion replay segments in sports video for highlights generation. Proc IEEE Int Conf on Acoustics, Speech, and Signal Processing, p.1649-1652. https://doi.org/10.1109/ICASSP.2001.941253

Pan H, Li BX, Sezan MI, 2002. Automatic detection of replay segments in broadcast sports programs by detection of logos in scene transitions. Proc IEEE Int Conf on Acoustics, Speech, and Signal Processing, p.IV-3385-IV-3388. https://doi.org/10.1109/ICASSP.2002.5745380

Raventós A, Quijada R, Torres L, et al., 2015. Automatic summarization of soccer highlights using audio-visual descriptors. *SpringerPlus*, 4(1):301. https://doi.org/10.1186/s40064-015-1065-9

Su PC, Lan CH, Wu CS, et al., 2013. Transition effect detection for extracting highlights in baseball videos. *EURASIP J Image Video Process*, 2013(1):27. https://doi.org/10.1186/1687-5281-2013-27

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. https://doi.org/10.1109/TCSVT.2013.2243640

Tien MC, Chen HT, Chen YW, et al., 2007. Shot classification of basketball videos and its application in shooting position extraction. Proc IEEE Int Conf on Acoustics, Speech and Signal Processing, p.1085-1088. https://doi.org/10.1109/ICASSP.2007.366100

Wang DH, Tian Q, Gao S, et al., 2004. News sports video shot classification with sports play field and motion features. Proc IEEE Conf on Image Processing, p.2247-2250. https://doi.org/10.1109/ICIP.2004.1421545

Wang JJ, Chang E, Xu CS, 2005. Soccer replay detection using scene transition structure analysis. Proc IEEE Int Conf on Acoustics, Speech, and Signal Processing, p.433-436. https://doi.org/10.1109/ICASSP.2005.1415434

Wang L, Liu X, Lin S, et al., 2004. Generic slow-motion replay detection in sports video. Proc Int Conf on Image Processing, p.1585-1588. https://doi.org/10.1109/ICIP.2004.1421370

Wu X, He R, Sun ZN, et al., 2018. A light CNN for deep face representation with noisy labels. *IEEE Trans Inform Forens Secur*, 13(11):2884-2896. https://doi.org/10.1109/TIFS.2018.2833032

Xu W, Yi Y, 2011. A robust replay detection algorithm for soccer video. *IEEE Signal Process Lett*, 18(9):509-512. https://doi.org/10.1109/LSP.2011.2161287

Zhao F, Dong Y, Wei Z, et al., 2012. Matching logos for slow motion replay detection in broadcast sports video. Proc IEEE Int Conf on Acoustics, Speech and Signal Processing, p.1409-1412. https://doi.org/10.1109/ICASSP.2012.6288154

Zhao Z, Jiang SQ, Huang QM, et al., 2006. Highlight summarization in sports video based on replay detection. Proc Int Conf on Multimedia and Expo, p.1613-1616. https://doi.org/10.1109/ICME.2006.262855