Event Detection and Summarization in Soccer Videos Using Bayesian Network and Copula

Mostafa Tavassolipour, Mahmood Karimian, and Shohreh Kasaei, *Senior Member, IEEE*

***Abstract*—Semantic video analysis and automatic concept extraction play an important role in several applications; including content-based search engines, video indexing, and video summarization. As the Bayesian network is a powerful tool for learning complex patterns, a novel Bayesian network-based method is proposed for automatic event detection and summarization in soccer videos. The proposed method includes efficient algorithms for shot boundary detection, shot view classification, mid-level visual feature extraction, and construction of the related Bayesian network. The method contains of three main stages. In the first stage, the shot boundaries are detected. Using the hidden Markov model, the video is segmented into large and meaningful semantic units, called *play-break* sequences. In the next stage, several features are extracted from each of these units. Finally, in the last stage, in order to achieve high level semantic features (events and concepts), the Bayesian network is used. The basic part of the method is constructing the Bayesian network, for which the structure is estimated using the Chow–Liu tree. The joint distributions of random variables of the network are modeled by applying the Farlie-Gumbel-Morgenstern family of Copulas. The performance of the proposed method is evaluated on a dataset with about 9 h of soccer videos. The method is capable of detecting seven different events in soccer videos; namely, goal, card, goal attempt, corner, foul, offside, and nonhighlights. Experimental results show the effectiveness and robustness of the proposed method on detecting these events.**

***Index Terms*—Bayesian network, copula distribution, semantic video analysis, video summarization.**

I. Introduction

THE RATE of audio-visual data produced in the form of image and video has increased rapidly in recent years. With the growth of computational power and electronic storage capacity, the necessity for large digital image/video libraries has also increased. For example, in the Internet there exists a large amount of anonymous image and video data which are the bases of many entertainments, educational, and commercial applications. This makes the search among image/video data a challenging issue for users. Thus, an appropriate digital image/video library must provide easy access to information and facilitates the retrieval of the content.

Manuscript received March 15, 2012; revised July 13, 2012; accepted November 30, 2012. Date of publication January 28, 2013; date of current version February 4, 2014. This paper was recommended by Associate Editor T. Zhang.

The authors are with the Department of Computer Engineering, Sharif University of Technology, Tehran 11155, Iran (e-mail: tavassolipour@ce.sharif.edu; mkarimian@ce.sharif.edu; skasaei@sharif.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2013.2243640

When the total length of videos reaches thousands of hours, users need a system for summarizing and abstracting them in order to have an efficient and effective search. Although text-based search algorithms assist users to find a specific image or a segment of long videos, in most cases, the system outputs many irrelevant videos to ensure the retrieval of the objective video. Intelligent indexing and summarization systems are an essential need in the process of video retrieval. Among all video types, sports videos attract many viewers and usually last for long hours. Sports videos, in general, are composed of some interesting events which capture users' attention. For most people, a summarized version of the sports video is more attractive than the full length version. Although a generic sports video summarization system is sufficient and useful, the summarization system in a domain-specific manner, such as soccer videos, may offer more facilities to users. Most sports broadcasters use some editing effects such as slow-motion replay scenes, and super-imposed text captions to distinguish the key events. Therefore, high level semantics can be detected using these editing effects and the audio-visual features that are automatically extracted.

Processing of sports videos (e.g., detection of important events and creation of summaries) makes it possible to deliver sports videos over narrow-band networks (such as the Internet and wireless), since the valuable semantics generally occupy only a small portion of the whole content [1].

One of the challenging problems in a video event detection method is the event boundary detection. Some methods, such as [2], propose a frame-based algorithm for event detection, while other methods, such as [1] and [3], use the temporal video segments for extracting more meaningful semantic units for event detection. In our method, like [3], we have used the "*play-break*" sequence as a semantic unit in our event detection. Each "*play-break*" consists of two sections called "play" and "break." In soccer videos, the game is in a "play" mode when the game is going on and the "break" mode is the complement set; that is, whenever the game is halted because of occurrence of an event.

A novel method for shot boundary detection and shot view classification is proposed. For shot boundary detection, both compressed and spatial domain features are used. For shot view classification, a new method based on object detection is proposed.

This paper also presents a novel method for segmenting the video into its *play-break* sequences. From each *play-break* sequence some features are extracted which are classified by

using a Bayesian network. The Bayesian network consists of a single hidden variable and several observable random variables. Network structure is determined by the Chow–Liu algorithm. We have proposed a novel method for calculating the joint distributions of continuous random variables in the Bayesian network, using the Copula theory.

The performance of the proposed method has been tested on a dataset of 10 different broadcast soccer videos (about 9 h). The proposed algorithm has successfully detected and classified soccer highlights; including goal, red/yellow card, goal attempt, corner, foul, offside, and nonhighlights.

The rest of this paper is organized as follows. Section II discusses the related work on soccer video event detection. Section III describes the proposed method for temporal video segmentation and our motivations for using *play-break* sequences as the semantic unit for event detection. Section IV explains the Bayesian network and application of Copula for extraction of high level semantic features from low level features. The experimental results are reported in Section V. Finally, the conclusion and future work are outlined in Section VI.

II. Related Work

For more than a decade, researchers have actively tried to find an automatic solution for sports video semantic analysis. High level semantics in sports videos can be detected by occurrences of specific audio-visual features extracted automatically from the video. Object motion analysis and tracking is another approach for detection of high level semantics. However, these methods require expensive computations and usually need customized cameras [4].

High level semantics in sports videos can be detected using superimposed text displayed during specific video events. This text gives more useful information about the key events to audiences; e.g., updating the scoreboard after a goal. However, character recognition in videos is still a challenging process especially for sports videos; because the characters are usually in a small size with low contrast and dpi, and often have a complex background [5], [6].

In general, soccer video semantic analysis methods can be classified into two main categories; pattern recognition methods [2], [3], [7], [8] and object tracking methods [9]–[12].

Often, object tracking methods use static cameras (motionless cameras). Use of these cameras not only improves the accuracy of the whole system, but also decreases the computational complexity of the method. But, usually, because of using customized cameras, implementation of these approaches is very expensive. However, there are some tracking-based methods which use broadcast videos for event detection [9]. The most important advantage of tracking-based methods, in relation to pattern recognition methods, is that they achieve high accuracy in detection of events that follow specific motion patterns; such as the soccer goal (ball passes the goal line). However, they cannot be used for events that have no specific motion pattern; such as the "foul" event in soccer videos. For example, in [10] a method is proposed that uses six fixed cameras for detecting the offside event. In [11], by utilizing four cameras, a method is proposed for detecting the goal and other events around the goal mouth.

Pattern recognition-based methods usually extract some audio-visual features (called low- or mid-level features) and then by using a classifier, the events or high level semantics (high level features) are detected. In [7], an automatic method is proposed that utilizes a subspace-based data mining method for feature extraction. That method is generic such that it does not use any prior knowledge in the detection process and can be considered as a domain-free method. It uses a C4.5 decision tree classifier. In [8], another method is proposed that uses a specific dimension reduction method, called mixture modality projection (MMP), to obtain high level features from low- and mid-level features. Some alternative pattern recognition techniques include the use of a dynamic Bayesian network (DBN) for capturing the temporal pattern of extracted features during soccer events [2]. For sports video highlight detection, a hybrid approach that integrates some audio-visual statistics into logical rule-based models is reported in [3]. It utilizes the *play-break* sequence as a semantic unit of sports videos. The method has been applied to different sports; including soccer, basketball, and Australian football.

Pattern recognition-based methods can be divided into two main categories: machine learning (statistical-based) methods and rule-based approaches (such as [3]). The machine learning methods use a statistical model to capture specific patterns of audio-visual features in sports videos (such as [2] and [12]), For example, a hidden Markov model (HMM) can be trained to detect some events in sports videos [12]. One of the most powerful methods for classifying and modeling features is the Bayesian network. For instance, [2] uses a DBN for capturing the temporal patterns between video frames during soccer events.

In some semantic video analysis methods, the human body is considered as the most important object in the scene. Thus, video events are defined by several human activities in the video [13] and [14]. Zhu *et al.* [13] propose a method for recognizing several human actions; e.g., making cell phone call, putting an object down, and pointing to something. A similar method has proposed in [14] which detects other human activities; e.g., hand wave, pick-up, two-hand wave, and jumping-jacks. These methods can also be used for improving the event detection process in sports videos. For example, in human actions such as jump and kick, the ball can be considered as mid-level semantic features for detecting high level events.

There are some work that propose a generic method for event detection in different types of videos [15]. Osadchy *et al.* [15] use methods of object recognition (such as face recognition) to discriminate more important frames. As mentioned before, methods of video event detection can also be used for video retrieval and browsing [16]. For example, in [17] a data mining-based method is proposed which analyzes video databases using video event detection. It uses a 2-level HMM for event detection. The work reported in [18] proposes an event detection method for wild-life videos. This method uses a neural network as a classifier for this purpose. Foresti *et al.* [19] propose a multilayer perceptron (MLP)

method for surveillance applications. Hung and Hsieh [20] propose an automatic event detection in broadcast baseball videos. It uses the Bayesian network as a classifier for event detection. A method for unusual event detection in videos is proposed in [21]. It uses the invariant subspace analysis (ISA) method for feature extraction. Time-evolving properties of these features are then modeled via an HMM. There are also some methods that act completely different from common event detection methods. For example, in [22] the method uses the webcast text for event detection of broadcast sports videos. Webcast text is a text broadcast channel for sports games which is easily obtained from the web. They first analyze the webcast text to cluster and detect text events and then the boundaries of video events are detected by a structural analyzes of the video.

Although recent approaches use hybrid methods for video event detection (as reported in [23]), these approaches in video indexing and summarization suffer from two main drawbacks of: 1) lack of an appropriate video semantic unit (i.e., where the event starts and ends), and 2) lack of a generic feature set for different events [3]. For example, Ekin *et al.* [1] have proposed a method for goal detection. They consider each goal to occur between two long-view shots, such that the first contains the goal event and the next contains the restart of the match.

III. Proposed Temporal Video Segmentation Method

Block diagram of our proposed semantic analysis system is illustrated in Fig. 1. As seen in this figure, first the video shot boundaries are detected and then the video is segmented to *play-break* sequences; by using the output of replay detection and shot view classification processes. In the next step, several features are extracted from each paly-break sequence and then the related event is detected by using the Bayesian network. Finally, an appropriate weight is assigned to each *play-break* sequence according to the importance of its detected event. The weight is then used to construct the summarized version of the query video by solving the 0–1 knapsack problem.

A. Shot Boundary Detection

Shot boundary detection is the first step of any semantic video analysis and many other applications; such as retrieval, indexing, and video summarization. For this purpose, in the proposed method, we extract two types of features: spatiotemporal and compressed domain features. We have used the hue histogram difference as a feature of the first type, and frame types and resultant motion vectors as features of the second type. After feature extraction, a linear support vector machine (SVM) is used to classify the frames into boundary and nonboundary classes.

The first feature used for shot boundary detection is the hue histogram difference of consequent frames. Calculation of histogram difference is the most popular approach for detecting shot boundaries, which achieves a tradeoff between accuracy and speed [24]. Among different types of histograms, the hue histogram gains a good performance

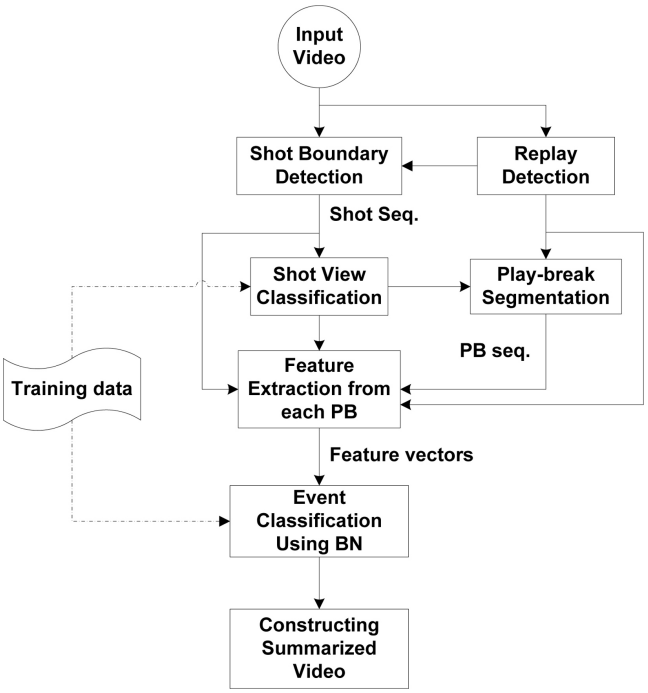


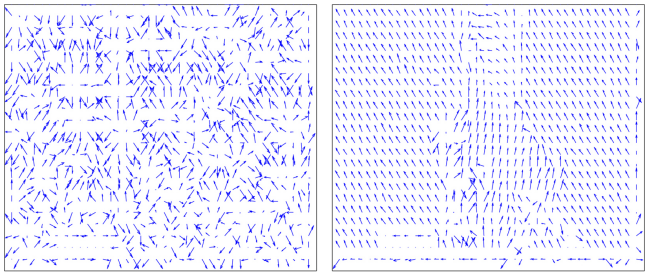
Fig. 1. Block diagram of the proposed method.

and has a high sensitivity to shot changes. Besides, hue histograms are robust to camera and object motions. If two consecutive frames belong to the same shot, then the hue histogram difference will be trivial. Thus, this discrepancy can be used to discriminate shot boundaries. We have used a block-wised histogram difference as reported in [25] to improve the performance of detection process. Several measures are introduced for comparing histograms, such as correlation, X^2-squared, Bhattacharyya, intersection, and the forth. We have used the Bhattacharyya measure defined by

$$d(H_1, H_2) = \sqrt{1 - \frac{1}{N\sqrt{\bar{H}_1 \bar{H}_2}} \sum_i \sqrt{H_1(i) \cdot H_2(i)}} \quad (1)$$

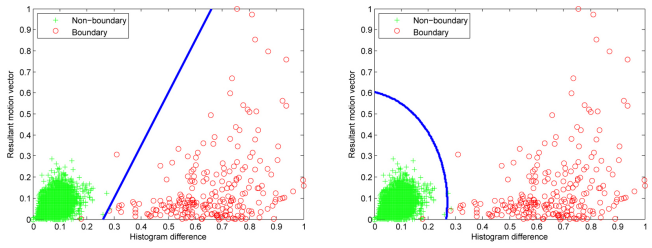
where $\bar{H}_k = 1/N \sum_j H_k(j)$ and N denotes the number of histogram bins.

The use of compressed domain features has several benefits, for example using these features not only improves the performance of the detection process but also does not increase the computational cost of it. In some video standards, such as MPEG-2 and MPEG-4, motion vectors of each frame are calculated and stored in the resulting bit stream of the video. If two consecutive frames are laid on the same shot, then their motion vectors follow a kind of regularity so that the near vectors have the same length and direction (Fig. 2). This regularity is due to the fact that neighboring vectors belong to a single object in the image. But, if two consecutive frames lie in different shots, their motion vectors will scatter without any regulation throughout the frame which causes the length of the resultant vector to become very small (Fig. 2). As such, we have used the length of the resultant motion vector as the second feature for detecting the shot boundaries.



(a) (b)

Fig. 2. Motion vectors. (a) Boundary frame (first frame of a shot), resultant vector length: 874.51 pixel. (b) Nonboundary frame (second frame of the same shot), resultant vector length: 10103.3 pixel.



(a) (b)

Fig. 3. SVM classifier result for shot boundary detection. (a) Linear-SVM. (b) SVM with RBF kernel (the features are scaled in interval [0, 1]).

Frame type is another compressed domain feature that can be used to detect shot changes. In general, each video frame can be coded by one of the three different types: I-frame, B-frame, and P-frame. During video coding, if the content of a frame cannot be predicted by its previous and next frames with a low error rate, the encoder decides to code it as an I-frame. Usually, since the content of the first frame of an abrupt shot boundary (cut) has a high difference with its previous frames, the frame is coded as an I-frame. Our experimental results show that about 84% of shots' first frames are coded as an I-frame. Therefore, the frame type has been chosen as the third feature of the shot boundary detector.

After extracting above features (hue histogram difference, resultant motion vector, and frame type), a 3-D feature vector is obtained for each video frame. Now, an appropriate classifier can be used to detect boundary frames. Here, the linear-SVM is used; because of its high accuracy where instances are linearly separable.

Fig. 3 illustrates the histogram difference and 'resultant motion vector for a soccer video. As seen in this figure, boundary and nonboundary frames are almost linearly separable. Therefore, a linear SVM can obtain a rather good performance; although Fig. 3(b) shows the superiority of a SVM classifier with RBF kernel over a linear-SVM, linear-SVM is more generalizable and faster than RBF kernel.

B. Shot View Classification

Information about the shot view type can lead to useful cues on semantic content of the video. Most methods in the field of sports video analysis try to detect the view type of each shot (or frame) to improve the performance event detection process. Awareness of shot view type not only enriches the semantic analysis method, but also interferes when different processes are to be chosen for different view types. In general, four classes of shot views can be defined in soccer videos: long view, medium view, close-up view, and out-of-field view. Fig. 4 illustrates a sample of each view types.

(a) (b) (c) (d)

Fig. 4. View types in soccer videos. (a) Long-view. (b) Medium-view. (c) Close-up view, and (d) out-of-field view.

Several methods are proposed for shot view classification in sports videos (such as [26] and [1]). Some of them use dominant color of frames for classifying their view type. Ekin *et al.* [23] use the dominant color ratio for detecting view type of each frame (dominant color ratio is defined as the ratio of dominant color area to image area). That method assumes that the dominant color of long views is the grass color; and therefore their dominant color ratio has a large value. In other words, they assume that in other views (medium, close-up, and out-of-field) the dominant color is not green. Although, this method works in some cases, there are several cases for which this assumption is invalid. For example, Fig. 4 shows some medium- and close-up frames for which the dominant color is also green. Method in [26] uses some further processing for improving the accuracy of close-up view detection. It uses a simple skin detection method for close-up views. That method assumes that in the close-up frames, the face of player is placed exactly at the center of the frame. But, obviously there are many cases for which that assumption is invalid [see Fig. 4(b)].

We believe that one of the best features that can be used to discriminate among frame view types is the size of players. It is clear that the size of players in the long views is much smaller than those in the medium and close-up views.

Finding the size of players in soccer videos is an easy process, because the background of players has an almost uniform green texture. Therefore, by subtracting the background we can almost find the players and calculate their bounding boxes. Since this process needs more calculations, we use a hybrid method that merges the method proposed in [1] and our idea. Fig. 5 shows the flowchart of the method in detail. As seen in this figure, if the dominant color of the frame is not green, then its view type is a close-up or an out-of-field, otherwise the largest object size in the field (dominant color region) is calculated and by using two thresholds, T_{low} and T_{high}, the view type of the query frame is determined. Since the shot view type does not change during a shot, the view detection of a single frame of it is sufficient. Fig. 6 shows the output of our method for several view types.

C. Replay Detection

Replays are important sections in sports video semantic analysis. They are used to provide more information and details on occurred events. In some cases, replays are shown

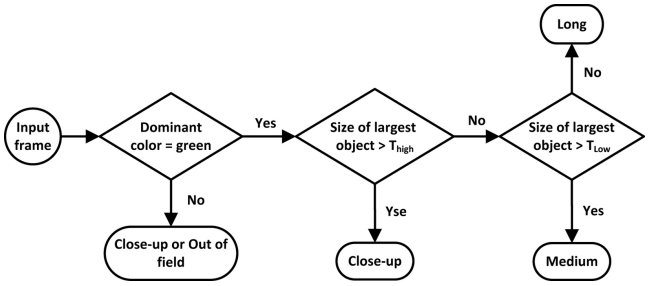


Fig. 5. Flowchart of the proposed shot view classifier.

(a) (b) (c)

Fig. 6. Finding largest object in grass field for shot view classification.

(a) (b)

Fig. 7. Example of two logos shown before and after replays.

with slow motion. But, usually they are played in an ordinary speed. Some methods try to detect them by detecting slow motions [25]. Most sports video broadcasters place the replays between two graphical logos. Therefore, by detecting these logos we can find the boundaries of replays (Fig. 7).

In [3], a threshold-based replay logo detection method is proposed. Every frame is converted to a binary image. Then, the number of white pixels is counted and according to this number the replay logos are detected. It considers that most logos are brilliant and thus their binary images are covered with white pixels. But, there are several cases for which, although the large parts of the binary frames are covered with white pixels, they contain no logos. Also, there are many logos that are not brilliant. Fig. 8 illustrates some of such cases.

If we have an instance of the logo used in the video sequence, then we can calculate its histogram. Thus, in our detection process, the replay logos are detected by comparing this histogram with the histogram of each video frame. This method has a high accuracy and speed and does not append any extra computation to the system, because frames' histograms are already calculated in the shot boundary detection process.

D. Play-Break Segmentation

Temporal video segmentation plays an important role in many video applications; including video summarization and indexing. It refers to partitioning the video into temporal regions which have some common features [27]. For example,

(a) (b) (c) (d)

Fig. 8. Cases for which [3] fails to detect logos. (a) and (c) Original frames. (b) and (d) Corresponding binary images.

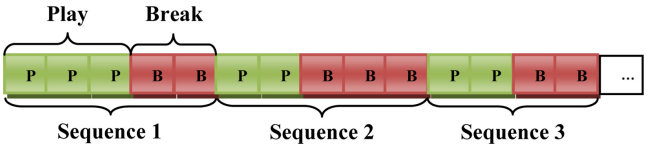


Fig. 9. Play-break scheme in sports videos.

partitioning a video into its shots is a type of temporal video segmentation.

In this section, another semantic unit is introduced for sports videos called *play-break* sequence. Each *play-break* sequence consists of several shots. These semantic units are considered as the smallest (first level of) semantic unit in sports and surveillance videos [16]. In soccer videos, the game is in a "play" mode when the ball is in the field and the game is going on; a "break" mode is the complement set, that is, whenever the game is halted because of occurrence of an event (e.g., goal, foul, corner) [16]. Fig. 9 shows the *play-break* sequence of an exemplar sports video. Using the *play-break* sequence as a semantic unit has several benefits; as *play-break* sequences are larger units than shots, more semantic features can be extracted from them. On the other hand, usually in soccer videos, each *play-break* sequence contains only one event; and thus the event detection process is limited to each *play-break* sequence.

Now, we will specify how to segment a video into *play-break* sequences. There are some methods like [3] and [28] which try to solve this problem. Tjondronegoro and Chen [3] use the view type of shots and apply some heuristic rules on them to classify each shot to "play" or "break" classes. The work reported in [28] uses an HMM-based algorithm for this purpose. In our method, we use an HMM for classifying shots. It has two hidden states of play and break and four observation states which are the view types of shots. The structure of the designed HMM is showed in Fig. 10.

In the training step, state transitions and observation probabilities are trained. For this process, we have used a labeled dataset; which is explained in Section V. In the test step, the view type sequence of shots is feeded to the HMM as the observation sequence and then by using the Viterbi decoding method the *play-break* sequences (state sequences) are obtained.

The method proposed in [21] has two differences with our method. First, it is frame-based (not shot-based), which causes the system to encounter more processes. Second, it uses the "dominant color ratio" and "motion intensity" as the observation states of the HMM. Experiments show that the accuracy of the proposed method is about 97%, while the reported accuracy of method [21] is 83.5%.

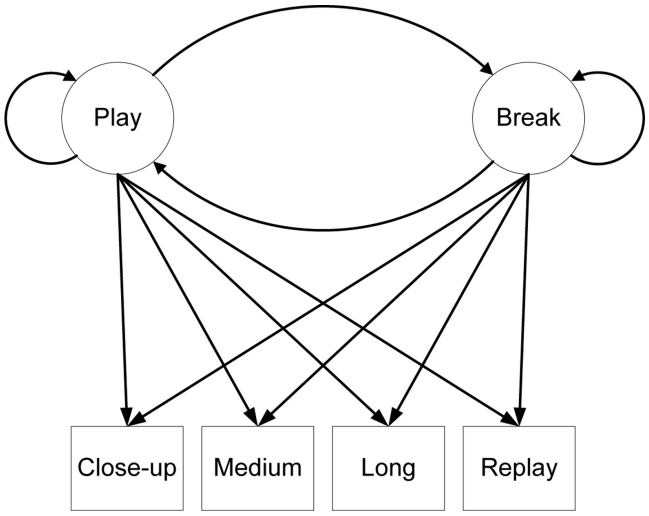


Fig. 10. Structure of the proposed HMM method to segment a sports video into *play* and *break* sequences.

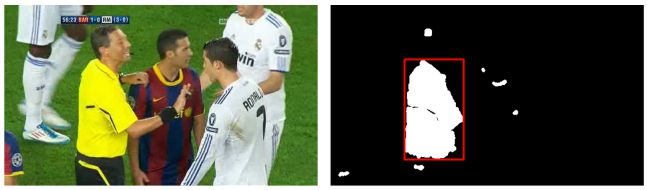
It is worth mentioning that since the *play-break*s are large units (compared to shots), they may contain multiple events. But, our experiments show that the number of such *play-break*s is very low, such that only 3% of all *play-break*s contain multiple events.

E. Other Mid-Level Features

As mentioned above, in order to attain semantics and concepts of videos, usually three levels of features are extracted. Mid-level features are very important since the system highly depends on them. When the semantics of mid-level features are close to a high level, their influence improves. For example, detection of white lines in the grass field can be considered as mid-level features, but we can reach more meaningful features by processing these lines (for instance, we can detect the penalty box region by them). It is clear that the "penalty box region" is a more meaningful feature. In the proposed method, for detecting several events of soccer, some mid-level features are extracted from each *play-break* sequence. These features include: break duration, replay duration, number of close-up shots in break section, number of players in penalty box, penalty box region, referee, goal-keeper, graphical captions, etc. Calculation of some of these features (e.g., referee, penalty box region, graphical caption) requires further image and video processing. In the following, we briefly explain how these features are extracted.

Referees and goal-keepers play an important role in soccer matches, so that many events are tied to them. Therefore, the shots displaying these objects have more importance than the other shots. The only feature that discriminates goal-keepers and referees from other players in the field is the different color of their shirts. By knowing that color we can search the entire close-up and medium-view frames for finding large objects with desired colors (Fig. 11).

The most important region of a soccer field is the penalty box. Most important events (such as goal, goal attempt, and corner) usually take place in this region. Hence, detecting this region can help to better detect these events. The main features

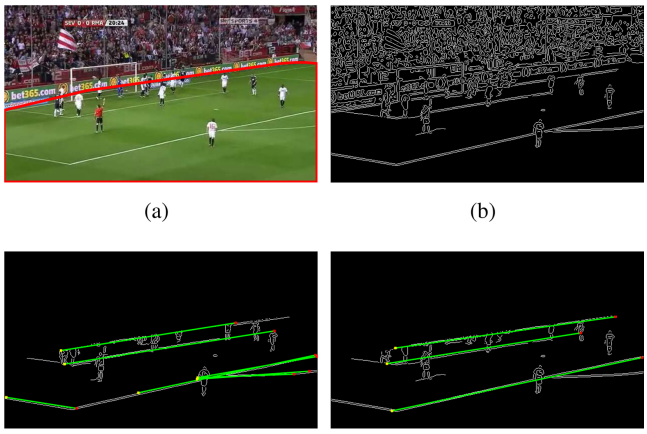


(a) (b)

Fig. 11. Referee detection. (a) Original frame showing the referee. (b) Finding components with color close to referee's shirt color (detected referee is shown by a box).

TABLE I
PROPOSED PENALTY BOX DETECTION ALGORITHM

1) Detect the boundaries of grass field by finding the convex hull of the largest green area of the image.
2) Use Canny edge detector to detect edges.
3) Detect lines which are laid in the grass field using the linear Hough transform.
4) Detect three parallel lines.



(a) (b) (c) (d)

Fig. 12. Steps of proposed penalty box detection algorithm. (a) Detecting boundaries of grass field. (b) Result of Canny edge detector. (c) Result of Hough transform to find lines (the edges placed out of grass field are ignored). (d) Detecting three parallel lines.

of the penalty box are white lines that surround it. Thus, by detecting these lines, we can detect and discriminate it from other regions of the field. As the only region in the field which consists of three parallel lines is the penalty box [1], we use the algorithm mentioned in Table I for detecting the parallel lines of the penalty box. Fig. 12 illustrates the outputs of this algorithm step by step.

One of the other mid-level features that can be extracted from the shots is the number of players in the penalty box. As mentioned earlier, for detection of shots' view type, all of the objects (players) in the grass field are found and by using the size of the largest object, the frame view type is determined (Fig. 5). Therefore, by detecting the penalty box region we can determine the number of players placed in that region.

IV. BAYESIAN NETWORK

The main and final goal of any semantic analysis method is to obtain high level features. These features are indeed

the events or concepts of the video. In soccer videos, several events (goal, foul, ...) are considered as high level features. In the previous sections, the low- and mid-level feature extraction methods were described. This section describes our high level feature extraction method.

In order to classify the events of a video, several classifiers can be used; such as SVM, Bayesian classifier, neural network, decision tree, HMM, and the forth [16]. Among these, the most powerful tool that can be used as a classifier for semantic video analysis is the Bayesian network [2], [20].

In the proposed method, several events are detected using a Bayesian network. It is an acyclic directed graph that represents conditional dependences among a set of random variables [29]. A directed edge represents the conditional relation between two random variables, which are parameterized by conditional probability distribution models. Structure of the network shows the relationship among hidden and observable variables.

The main feature of a Bayesian network is its ability to capture dependences among extracted features. In fact, extracted features are considered as the random variables of the network. Most other classifiers do not have any assumption on the dependences among extracted features. The Bayesian network is a graphical model for Bayesian classifier which is called an optimal classifier because of its minimum classification error [30].

Often, in classification problems there are some dependences among extracted features and therefore the lack of their consideration leads to reduce the classifier accuracy. For example, in linear-SVM there is no assumption on the dependences among the used features.

Another benefit of the Bayesian network is its ability to define a cause–effect relationship among variables. For example, with a Bayesian network we can calculate the importance of each feature on occurred event; because in the Bayesian network the conditional probability distribution function is calculated and therefore features with more conditional density have more influence on the detected event. Since a Bayesian network can model the cause–effect relationship among variables, it can be compared to rule-based algorithms. In the rule-based classifiers, there are several deterministic rules that are used to classify each instance. Thus, the Bayesian network can be considered as a probabilistic rule-based method.

The training and test phases of a Bayesian network are simple. In the training phase, the conditional distributions of dependent variables are calculated and in the test phase the probability of a given query is computed.

We have used a Bayesian network with a single hidden state and some observation variables. The hidden state contains the type of occurred event in the video and observation variables are the extracted mid-level features. For using a Bayesian network, in the training phase, for each class (event type) a distinct network is trained. For example, if we intend to detect two events (e.g., goal and corner) we have to train a Bayesian network for each of them.

Although the structure of the Bayesian network of all events is similar, the network parameters are estimated separately. In other words, for each event, there is a Bayesian network for which the parameters (parameters of distribution models) are estimated so that to maximize the likelihood of the training observation vectors for that event.

TABLE II
RANDOM VARIABLES USED IN THE DESIGNED BAYESIAN NETWORK [D: DISCRETE, C: CONTINUOUS, P: POISSON, MOG: MIXTURE OF GAUSSIAN, B: BERNOULLI]

Variable Name	Type	Dist.	Comments
event	D	—	Event of *play-break* sequences
numBrkShot	D	P	No. of break shots
numRepShot	D	P	No. of replay shots
numBrkShotBefRep	D	P	No. of break shots before replay
numCloseUpShot	D	P	No. of close-up break shots
numRefereeShot	D	P	No. of shots displaying referee
numGKRep	D	P	No. of replay shots displaying goal-keepers
numGKShots	D	P	No. of shots displaying goal-keepers
breakDuration	C	MoG	Break duration
repDuration	C	MoG	Replay duration
nearGoal	D	B	Last play shot displays penalty box
highPlayerInPB	D	B	High number of players in penalty box
caption	D	B	Graphical caption
breakInRep	D	B	Frozen replay

For constructing a Bayesian network, the network structure and the distribution model of each random variable must be determined. Furthermore, the conditional probability of dependent random variables should also be calculated.

A. Structure Learning of Bayesian Network

For determining the structure of a Bayesian network, the type and number of random variables must be determined. Table II lists the random variables along with their types and distribution models. As seen in this table, most random variables are discrete except for two. The distribution model of each random variable is determined according to its histogram. Fig. 13 illustrates the histogram and the corresponding distribution for variables numCloseUpShot and brkDuration. As seen in Table II, 13 features are extracted from each *play-break* sequence (11 features are discrete and 2 features are continuous).

Now, to design the structure of a Bayesian network, the dependency among different features should be determined. There are several methods for determining the structure [2]. The Chow–Liu algorithm can be used for estimating the structure. It is a very simple method and contains discrete variables. In this method, the values of dependences between each pair of variables are calculated using the mutual information. Therefore, a weighted complete graph is obtained in which the weight of each edge is the mutual information of its two vertices. Then, an appropriate estimation of the Bayesian network is calculated by finding the maximum spanning tree

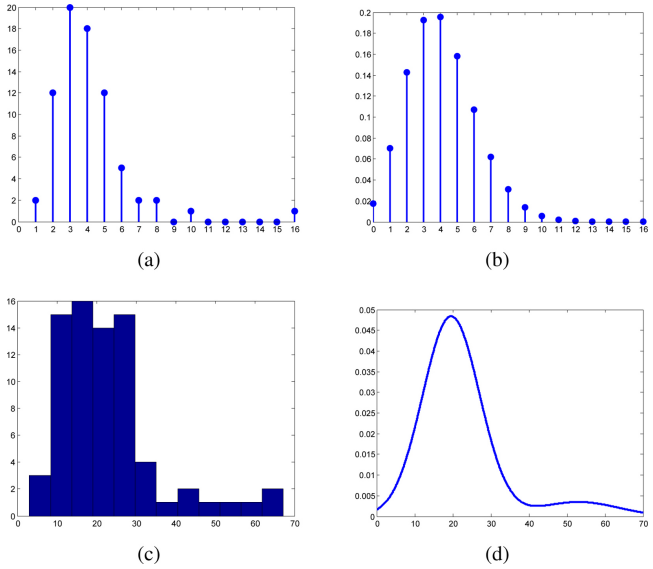


Fig. 13. Histogram and the corresponding distribution function for two random variables of the Bayesian network. (a) Histogram of *numBrkShot*. (b) Fitted Poisson distribution model for *numBrkShot*. (c) Histogram of *brkDuration*. (d) Fitted mixture of Gaussian model for *brkDuration* using *Expectation-Maximization* (EM) algorithm.

TABLE III
CHOW--LIU ALGORITHM FOR ESTIMATING A BAYESIAN NETWORK STRUCTURE

1) For each pair of variables A and B, use the training data to estimate $P(A, B)$, $P(A)$, and $P(B)$.
2) For each pair of variables A and B, calculate the mutual information.

$$I(A, B) = \sum_{a} \sum_{b} P(a, b) \log_2 \frac{P(a, b)}{P(a)P(b)} \tag{2}$$

3) Calculate the maximum spanning tree over the set of variables, using edge weights, $I(A, B)$, (given N variables, this costs only $O(N^2)$ time).
4) Add arrows to edges to form a directed-acyclic graph.

of the complete graph. Table III shows the steps of Chow–Liu algorithm for constructing a Bayesian network.

The work reported in [31] proves that the Chow–Liu tree minimizes the Kullback–Leibler (KL) divergence measure, that is

$$KL(P(X_1, \ldots, X_n) \| T(X_1, \ldots, X_n)) = \sum_{x_1, \ldots, x_n} P(x_1, \ldots, x_n) \log_2 \frac{P(x_1, \ldots, x_n)}{T(x_1, \ldots, x_n)} \tag{3}$$

where P and T are two distribution functions that are compared [32]. Bishop in [32] shows that $KL(P\|T) \geq 0$, with equality if, and only if, $P(X_1, \ldots, X_n) = P(X_1, \ldots, X_n)$. Actually, KL is a measure for comparing distributions so that its small value leads to more similarity between them.

Fig. 14 shows the estimated structure of the Bayesian network by this algorithm for our discrete variables. Since the Chow–Liu algorithm cannot be used for continuous variables, our continuous variables (brkDuration and repDuration) do not

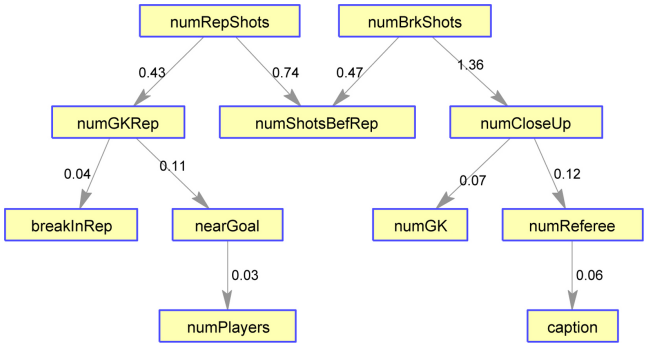


Fig. 14. Estimated Bayesian network for discrete variables using Chow–Liu algorithm, edge weights represent the mutual information.

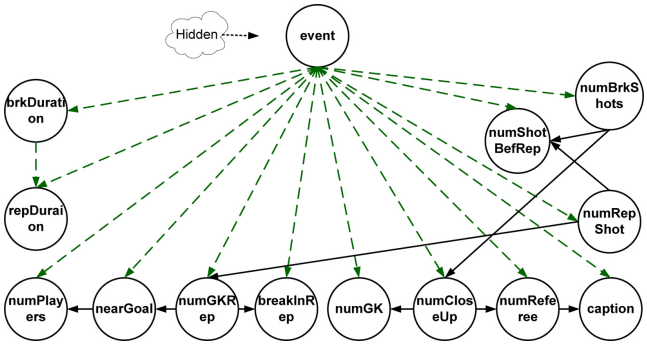


Fig. 15. Structure of designed Bayesian network.

exist in the figure. By adding the two continuous variables and the event variable to the network of Fig. 14, the complete Bayesian network is obtained, as shown in Fig. 15. Dashed edges in the figure are designed manually without any training process, because all features (observable random variables) depend on the event variable.

It is worth mentioning that there are some generalizations of Chow–Liu tree, called t-cherry junction trees. It is proven that these trees provide a better (or at least as good) approximation as the Chow–Liu tree for a discrete multivariate probability distribution. Chow–Liu tree is in fact the second-order t-cherry junction tree [33], [34].

B. Estimating Joint Distributions of Random Variable

For calculating conditional distribution of continuous random variables, we are required to calculate the joint densities. According to the Bayesian rule, we have

$$P(A|B) = \frac{P(A, B)}{P(B)}. \tag{4}$$

For Bernoulli random variables, we can easily calculate the conditional distributions; because by counting the occurrence of different values (true or false) of these variables we can estimate the conditional distributions. For infinite countable variables (such as Poisson variables), we can use intuition and our prior knowledge about the nature of these variables to calculate the conditional distributions. For example, the conditional distribution of two random variables numBrkShot

and numCloseUpshot follow the Binomial distribution

$$P(numCloseUpShot{=}x|numBrkShot{=}n) = \binom{n}{x} p^x (1-p)^{n-x} \quad (5)$$

where p is the average number of close-up shots. Note that $numCloseUpshot \geq numBrkShot$.

The challenging part of the problem is to find the joint distribution of two dependent continuous variables brkDuration and repDuration, which do not follow the Gaussian distribution model. In order to find the joint distribution of these two random variables, we have used Copula which is explained in the next section.

C. Copula Distribution Function

Calculation of joint distribution of variables in a Bayesian network is a challenging problem. The joint distributions of random variables are required for calculating conditional densities. For example, if X and Y are two random variables of a Bayesian network, where X is the parent of Y, then their conditional distribution is

$$f_{Y|X}(y|x) = \frac{f_{XY}(x, y)}{f_X(x)}. \quad (6)$$

Therefore, for calculating the conditional distribution, $f_{Y|X}(y|x)$, we need to have the joint distribution, $f_{XY}(x, y)$.

Now, the main problem is how to obtain the joint distributions. In the case of independent variables, their joint distribution is simply the product of marginal distributions

$$F_{XY}(x, y) = F_X(x)F_Y(y). \quad (7)$$

But, when the variables are dependent, we cannot use (6). In such cases, we need to find a way to obtain the joint distribution from marginal distributions. In other words, if X and Y are two dependent variables with distribution $F_X(x)$ and $F_Y(y)$, and joint distribution $F_{XY}(x, y)$, then we need to find a specific relation among $F_{XY}(x, y)$ and its marginal distributions, $F_X(x)$ and $F_Y(y)$. In general, such a relation may not exist. But, in some cases, we can model this relation as a function. For instance, we may have

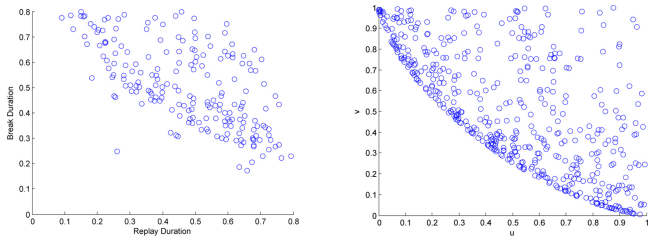
$$F_{XY}(x, y) = C(F_X(x), F_Y(y)). \quad (8)$$

This equation states that the joint distribution of X and Y is a function of marginal distributions. This function is called Copula. In this section, we will introduce the details and concepts of Copula.

The standard definition of a copula is a multivariate cumulative distribution function (CDF) defined on the unit cube $[0, 1]^n$, with uniformly distributed marginal distributions [35]. In other words, Copula is the joint-CDF of several uniform random variables. We can use Copula for finding the joint distribution of several random variables.

It is easy to show that random variable $U = F_X(X)$, where F_X is CDF of X, has uniform distribution in the interval of $[0, 1]$. The transform $X \rightarrow F_X(X)$ is usually referred to as the probability-integral transformation.

There is a theorem known as Sklar's theorem which is the foundation of many applications of Copula in statistics [35].



(a) (b)

Fig. 16. Scatter plots. (a) Scatter plot of random variables brkDuration vs. repDuration. (b) Scatter plot of random generated points using Clayton Copula.

Sklar's theorem elucidates the role that Copulas play in the relationship among multivariate CDFs and their margins. This theorem is as follows:

Sklar's theorem. Let $F_{XY}(x, y)$ be a joint cumulative distribution function with margins F_X and F_Y. Then, there exists a Copula C such that for all (x, y) in $\Re^2$ we have

$$F_{XY} = C(F_X(x), F_Y(y)).$$

If F_X and F_Y are all continuous distributions, then C is unique; otherwise C is uniquely determined on $Ran(F_X) \times Ran(F_Y)$, where $Ran(F_X)$ and $Ran(F_Y)$ are the range of F_X and F_Y, respectively.

For the proof, see [36]. In fact, Sklar's theorem states that the joint-CDF of random variables, $F_XY(x, y)$, is equal to the joint-CDF of their probability-integral transformed variables, $C(F_X(x), F_Y(y))$. In other words, the Copula of probability-integral transformed variables is equal to joint-distribution of original random variables.

According to the type and the amount of correlation between uniform random variables, their joint distribution may have different behaviors. Due to this fact, several Copula families have been introduced; such as Frank, Clayton, FGM, and the forth. Indeed, there are many Copula families with different properties which are used for modeling the joint distribution of different uniform random variables [35].

Fig. 16 shows the scatter plot of random variables brkDuration and repDuration after probability-integral transformation. As seen in this figure, there exists a high similarity between the scatter of these variables and random points that are generated by Clayton Copula. If we plot the scatter of transformed random variables brkDuration and repDuration given a specific event (e.g., goal, foul, etc.) the value of dependency will decrease, so that their corresponding Copula follows the FGM family. FGM family is one of the popular families of Copulas which are used to model weak dependences between random variables. It is calculated by

$$C_\theta(u, v) = uv + \theta uv(1-u)(1-v), \qquad \theta \in [-1, 1] \quad (9)$$

where θ is a parameter which is estimated from the training data using Kendall's τ measure. For estimation of this parameter, we need to calculate the concordance and discordance of the training data. This procedure takes $O(n^2)$, where n is the number of training samples [35].

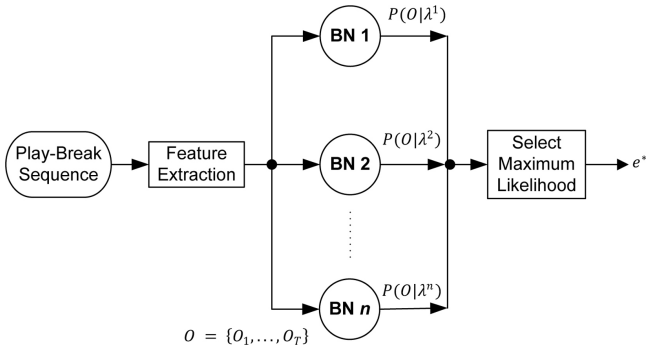


Fig. 17. Block diagram of the proposed event detection using several trained Bayesian Networks (BNs).

By using Sklar's theorem and following the above mentioned equations, we can calculate the joint and conditional PDF of random variables brkDuration and repDuration, by

$$X = brkDuration \tag{10}$$

$$Y = repDutation \tag{11}$$

$$U = F_X(X) \tag{12}$$

$$V = F_Y(Y) \tag{13}$$

$$f_{XY}(x, y) = \frac{\partial^2 H(x, y)}{\partial x \partial y} = \frac{\partial^2 C(u, v)}{\partial u \partial v} \cdot f_X(x) \cdot f_Y(y) \tag{14}$$

where $f_{XY}(x, y)$, $F_{XY}(x, y)$, and $C(u, v)$ represent the joint-PDF, the joint CDF, and the Copula, respectively. Also, $f_X(x)$ and $f_Y(y)$ denote the marginal PDF of X and Y, respectively.

D. Using Bayesian Network Classifier

As mentioned above, in the training step, we have trained a distinct Bayesian network for each event. In the test phase, each instance video (*play-break* sequence) is fed to each of these trained networks to find the network with the highest likelihood for the input instance; i.e., for each unknown event which is to be recognized, the process illustrated in Fig. 17 must be carried out which includes: computing the observation variables $O = O_1, \ldots, O_T$, via a feature analysis of the input video, followed by the calculation of model likelihoods for all possible models, $P(O|\lambda^i)$, followed by the selection of the event whose model likelihood is highest, or

$$e^* = \arg \max_{1 \le i \le N} \{P(O|\lambda^i)\} \tag{15}$$

where λ^i denotes the i^{th} trained Bayesian network.

E. Video Summarization and Knapsack Problem

So far we have explained how the events of soccer videos are detected. But, for building a summarized video we need a policy by which we can discriminate the video shots. The work reported in [37] proves that the video summarization problem can be reduced to the 0–1 knapsack problem. In this problem, given a set of items (each with a weight w_i and a value v_i) a collection of items is selected so that the total weight is less than (or equal to) a given limit, and the total value is as large as possible [38]. In other words, in the 0–1 knapsack problem, the optimization problem

$$A = \arg \max_{\mathcal{A}} \sum_{i \in \mathcal{A}} v_i, \qquad subject\ to: \sum_{i \in \mathcal{A}} w_i \le W \tag{16}$$

is solved, where $\mathcal{A}$ is an arbitrary subset of items, and A contains the set of selected items. Dynamic programming is used to solve this problem [38].

If an appropriate value is assigned to each *play-break* sequence according to the importance of its detected event, then the video summarization problem is converted to the 0–1 knapsack problem; we want to build a summarized video with the maximum length of L from several *play-break* sequences each of which has a weight v and length l. Therefore, by solving the 0–1 knapsack problem, we can build an appropriate summarized video which holds two desired properties.

i. Its length is less than (or equal to) a specific value L.
ii. Contains as valuable *play-break* sequences as possible.

V. Experimental Results

A. Dataset

One of the most important requirements in statistical pattern recognition and machine learning problems is the existence of an appropriate dataset. Without a suitable dataset we cannot achieve an accurate and reliable evaluation on the learning algorithm. Necessity of dataset for supervised learning algorithms is more than unsupervised and semisupervised algorithms. Unfortunately, there is no available labeled dataset for soccer videos. Consequently, we decided to build a dataset that is labeled manually. An appropriate dataset must consist of several soccer videos which are captured in different conditions by various broadcasters. Our constructed dataset consists of about 9 h of 10 soccer videos which are gathered from several countries and broadcasters; e.g., England premier league, Spain first division league, UEFA champion league, FIFA world cup, and the forth. Some of these videos correspond to the soccer matches who are held on daylight and others are held at night-time. The resolution of the videos is 640×368. Videos are coded using MPEG-4 with frame-rate 25 f/s.

In the dataset, shot boundaries and *play-break* sequences are manually labeled. The dataset consists of 3452 shots for which the average length is about 9 s. From each shot, the following features are manually extracted.

1) **View type:** long-view, medium-view, close-up view, and out-of-field view.
2) **Content:** referee, goal-keeper, graphical captions, penalty box, number of players in the penalty box.
3) **Type:** replay, slow-motion.
4) **Event:** goal, goal attempt, foul, out, corner, yellow/red card, offside, free kick.

Some extracted events such as "free-kick" and "out" are not used in the proposed method, but for the completeness of the dataset these events are also extracted to be used in future work.

TABLE IV
CONFUSION MATRIX OF PROPOSED CLASSIFIER, IN DETECTION OF SEVERAL SOCCER EVENTS

Actual class	Predicted class							
	Goal	Card	Goal attempt	Corner	Offside	Foul	Non-highlight	Miss-classified
Goal	34	0	0	0	0	0	0	1
Card	0	27	0	0	0	0	0	1
Goal attempt	3	0	64	0	0	0	3	5
Corner	0	0	1	33	1	0	0	0
Offside	0	0	1	0	9	0	9	6
Foul	1	1	1	0	1	73	40	12
Non-highlight	0	0	6	5	0	12	301	9
Recall(%)	97.14	96.43	85.33	94.29	36.00	61.24	90.39	–
Precision(%)	89.47	96.43	87.67	86.84	90.00	73.15	85.27	–

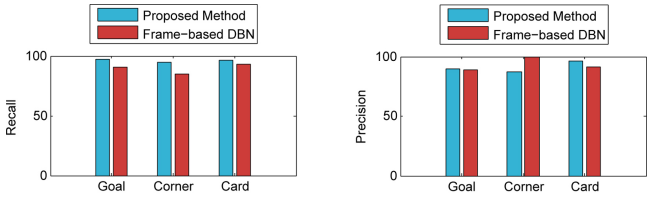


Fig. 18. Comparison of proposed method with frame-based DBN [2].

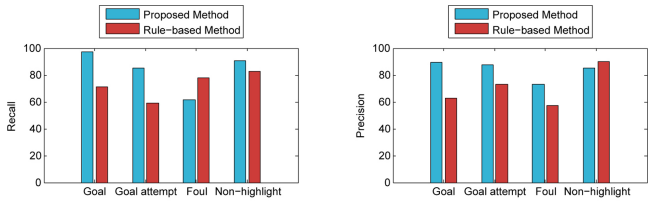


Fig. 19. Comparison of the proposed method with rule-based method [3].

B. Experimental Results

In the proposed method, we detect seven events; namely goal, goal attempt, red/yellow card, corner, foul, offside, and nonhighlights. Table IV shows the confusion matrix of the method. The right-most column indicates the number of instances which are miss-classified. The main diagonal of the matrix indicates the number of instances which are classified correctly. According to this matrix, from 660 instances (*play-break* sequences), 540 are correctly classified, so the total accuracy of the method is 81.8%. By using the method of "one against all" for our classification problem, we can convert it to a seven binary classification problem. The recall and precision measures of these classifiers are listed in the bottom of Table IV.

Fig. 18 compares the result of our proposed method with [2]. As seen in this figure, for more important events the precision and recall rates of our method are superior. In some cases, our method has less accuracy than other methods; for example, in [2] the precision of corner event detection is about 13% more than ours. But, its recall rate is approximately 9% less than ours. High precision of [2] for corner detection may be due to its low recall for this event.

The work reported in [2] uses a DBN, as a tool for soccer video analysis. The method is frame-based (features are extracted from each frame). Hence, it requires more processes than our method. Since they process the video frame by frame, their Bayesian network variables are Boolean. Our method uses more meaningful features (because we use *play-break* sequences as semantic unit for video events) and thus enables us to extract high level features. Although our proposed method uses more complicated features, its time complexity is very lower than [2]. The reason is that the frame-based method needs more computations. As an example, for shot view classification, determining the view type of a frame within a shot is enough; because usually the shot view type does not change during a shot. Often, the features extracted from consequent frames are highly dependent on each other; and thus in the DBN each variable is dependent to its value in the previous time slot (previous frame). This leads the network to be over-fitted on nonimportant information which decreases the accuracy of that system.

TABLE V
PERFORMANCE COMPARISON OF PROPOSED METHOD (BN) WITH SVM AND HMM [R: RECALL, P: PRECISION]

Event	BN		SVM(RBF)		HMM	
	R	P	R	P	R	P
Goal	91.14	89.47	85.71	88.24	94.29	28.45
Goal Attempt	85.33	87.67	73.33	72.37	77.33	63.04
Card	96.43	96.43	35.71	66.67	92.86	48.15
Corner	94.29	86.84	94.29	84.62	97.14	80.95
Foul	61.24	73.15	48.06	56.88	66.67	38.05
Offside	36.00	90.00	28.00	63.64	52.00	59.09
Nonhighlight	90.39	85.27	89.19	78.99	26.73	82.14

Fig. 19 compares the performance of our method with that of the method reported in [3]. That method uses a *play-break* sequence as the semantic unit. They have proposed a heuristic rule-based method for event detection in each *play-break* sequence. Their method extracts three statistics for each feature including minimum, maximum, and average value of it, without considering the dependences (correlations) among features. While, as seen in Fig. 14, many features are dependent on each other. Besides, these three statistics may not be sufficient for representing all properties of a feature. Hence, their method can detect only four event types.

TABLE VI
ACCURACY OF SHOT BOUNDARY DETECTION METHOD

	Linear-SVM (%)	**SVM with RBF kernel** (%)	**Method of [1]** (%)
Recall	99.12	99.50	97.30
Precision	99.26	98.97	91.70

Table V shows the performance comparison of our proposed method with other commonly used classifiers, such as SVM and HMM, on our dataset. Table V lists the precision and recall rates of the Bayesian network, SVM, and HMM. As seen in this table, the performance of the Bayesian network is better than the other two. The reason is that the Bayesian network captures the dependences among extracted features. In addition, it determines the amount of affection of each feature on different events. For example, the feature "referee" has more importance than the other features for detecting the "card" event. In other words, we have

$$P(event = card|referee) > P(event = e|referee), \qquad \forall e \neq card. \tag{17}$$

We can conclude that the Bayesian network can assign different weights to each feature (variable), while SVM and HMM cannot. HMM has more recall rate than the Bayesian network and SVM, but its precision is very low. Therefore, the confidence of HMM is low.

C. Shot Boundary Detection

The accuracy of shot boundary detection is shown in Table VI. The accuracy of the proposed shot boundary detector is presented with two classifiers; SVM with linear kernel and SVM with RBF kernel. The accuracy of SVM with RBF kernel is a bit better than linear-SVM. The proposed method, especially in detection of cut boundaries, has better recall and precision rates than [1]; because we have used more features such as motion vectors and frame types, whiles [1] uses only the histogram of frames. Since motion vectors and frame types are extracted from the compressed domain, no extra computation is needed in comparison with [1]. The results are obtained from a soccer video with 67615 frames, which contains 233 cut boundaries and 15 gradual boundaries.

D. Shot View Classification

Shot view classification is another part of the system. Our proposed method is similar to [1] except that we use the within field objects size to improve the accuracy of view type classification. The results of our proposed view type classifier are shown in Table VII. These results show the superiority of the proposed method compared to [1]. The overall accuracy of our proposed method is 91.04%. As mentioned in Section III-B, there are many cases in which the assumption of [1] is invalid, thus its accuracy is less than our proposed method.

E. Summarization

Table VIII shows the properties of summarized videos for three long time soccer videos. We have used the scores (values) for each event as follows, goal: 150, goal attempt: 50, card: 30, corner: 15, offside: 6, foul: 4, nonhighlight: 1. Scaling these scores does not affect the summarized video. However, for obtaining a good summarized video, the relative values of events scores are very important. The 0–1 knapsack problem guarantees that the summarized version contains the maximum possible score. Since the "goal" event has a high score, all summarized videos contain them. But, some other events are missed in the summarized video due to the length restriction. In this experiment, the output video length is set to 6 min. The last row of Table VIII shows the sum of the event scores. Note that although the input video length is about eight times longer than the output video length, the output video score is higher than the half of the input video score.

TABLE VII
PERFORMANCE COMPARISON OF SHOT VIEW TYPE DETECTION METHOD [L: LONG, M: MEDIUM, C: CLOSE-UP]

	Our method			**Method in [1]**		
	L	**M**	**C**	**L**	**M**	**C**
Recall	96.40	80.82	92.63	97.30	53.42	41.05
Precision	99.07	85.51	86.27	83.08	40.21	75.00

TABLE VIII
STATISTICS OF SUMMARIZATION METHOD FOR THREE VIDEOS [I: INPUT VIDEO, S: SUMMARIZED VIDEO]

	Video 1		**Video 2**		**Video 3**	
	I	**S**	**I**	**S**	**I**	**S**
Video length	45:07	5:57	48:04	6:00	47:39	5:58
Goal	2	2	2	2	2	2
Goal Attempt	9	4	7	3	1	1
Card	0	0	4	0	4	1
Corner	0	0	1	1	2	2
Foul	7	2	11	0	20	1
Offside	4	0	2	0	1	0
Nonhighlight	29	0	22	0	24	1
Total Score	831	508	863	465	610	415

VI. CONCLUSION AND FUTURE WORK

An efficient method for event detection and summarization of soccer videos was proposed in this paper. The Bayesian network was used as a classifier for soccer event detection. Despite some prior methods that are based on frame or shot, our proposed method used the *play-break* sequence as a semantic unit which leads to extraction of more meaningful features from the video and also decreases the required processing cost. Our main contribution was the use of Copula and Chow–Liu tree for calculating the joint distributions in the Bayesian network, which enabled us to use more complicated distribution models for network variables. Chow–Liu tree minimized the KL divergence measure for the joint distribution of discrete variable estimation. Also, novel approaches for shot boundary detection, shot view classification, and *play-break* segmentation were proposed.

It is easily observable that there are some temporal relations between consecutive events in soccer video. For example, the probability of "goal" occurrence after a "corner" is more than its occurrence after an "out" event. Hence, for future work, we intend to use the DBN due to its ability to capture temporal

dependences among random variables. Another applicable improvement is the use of t-cherry junction tree for structure estimation of the Bayesian network.

ACKNOWLEDGMENT

The authors acknowledge M. H. Mirhashemi for his edition.

REFERENCES

[1] A. Ekin, A. M. Tekalp, and R. Mehrotra, "Automatic soccer video analysis and summarization," *IEEE Trans. Image Process.*, vol. 12, no. 7, pp. 796–807, Jul. 2003.

[2] C. L. Huang, H. C. Shih, and C. Y. Chao, "Semantic analysis of soccer video using dynamic Bayesian network," *IEEE Trans. Multimedia*, vol. 8, no. 4, pp. 749–760, Aug. 2006.

[3] D. W. Tjondronegoro, and Y. P. Chen, "Knowledge-discounted event detection in sports video," *IEEE Trans. Syst., Man, Cybern. Part A: Syst. Humans*, vol. 40, no. 5, pp. 1009–1024, Sep. 2010.

[4] C. Yajima, Y. Nakanishi, and K. Tanaka, "Querying video data by spatiotemporal relationships of moving object traces," in *Proc. 6th IFIP Work. Conf. Visual Database Syst.*, 2002, pp. 357–370.

[5] T. Sato, T. Kanade, E. K. Hughes, and M. A. Smith, "Video OCR for digital news archive," in *Proc. IEEE Int. Workshop Content-Based Access Image Video Database*, Jan. 1998, pp. 52–60.

[6] S. C. Antani, D. Crandall, and R. Kasturi, "Robust extraction of text in video," in *Proc. 15th Int. Conf. Pattern Recognit.*, vol. 1. 2000, pp. 831–834.

[7] M. Shyu, Z. G. Xie, M. Chen, and S. Chen, "Video semantic event/concept detection using a subspace-based multimedia data mining framework," *IEEE Trans. Multimedia*, vol. 10, no. 5, pp. 252–259, Feb. 2008.

[8] J. Shen, D. Tao, and X. Li, "Modality mixture projections for semantic video event detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 11, pp. 1587–1596, Nov. 2008.

[9] J. Liu, X. Tong, W. Li, T. Wang, Y. Zhang, and H. Wang, "Automatic player detection, labeling and tracking in broadcast soccer video," *Pattern Recognit. Lett.*, vol. 30, no. 2, pp. 103–113, Jan., 2009.

[10] T. D'Orazio, M. Leo, P. Spagnolo, P. L. Mazzeo, N. Mosca, M. Nitti, and A. Distante, "An investigation into the feasibility of real-time soccer offside detection from a multiple camera system," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 12, pp. 1804–1818, Dec. 2009.

[11] T. D'Orazio, M. Leo, P. Spagnolo, M. Nitti, and N. Mosca, "A visual system for real time detection of goal events during soccer matches," *Comput. Vision Image Understanding*, vol. 113, no. 5, pp. 622–632, May, 2009.

[12] C. Wu, Y.-F. Ma, H.-J. Zhang, and Y.-Z. Zhong, "Events recognition by semantic inference for sports video," in *Proc. IEEE Int. Conf. Multimedia Expo*, Aug. 2002, pp. 805–808.

[13] G. Zhu, M. Yang, K. Yu, W. Xu, and Y. Gong, "Detecting video events based on action recognition in complex scenes using spatio-temporal descriptor," in *Proc. ACM MM*, 2009, pp. 165–174.

[14] Y. Ke, R. Sukthankar, and M. Hebert, "Event detection in crowded videos," in *Proc. ICCV*, 2007, pp. 1–8.

[15] M. Osadchy and D. Keren, "A rejection-based method for event detection in video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 4, pp. 534–541, Apr. 2004.

[16] A. Bovik, *The Essential Guide to Video Processing,* 2nd ed., The Netherlands: Elsevier, 2009, pp. 440–442.

[17] S. Guler, W. H. Liang, and I. A. Pushee, "A video event detection and mining framework," in *Proc. Conf. CVPRW*, Jun. 2003, p. 42.

[18] N. Haering, R. J. Qian, and M. I. Sezan, "A semantic event-detection approach and its application to detecting hunts in wildlife video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 6, pp. 857–868, Sep. 2000.

[19] G. L. Foresti, L. Marcenaro, and C. S. Regazzoni, "Automatic detection and indexing of video-event shots for surveillance applications," *IEEE Trans. Multimedia*, vol. 4, no. 4, pp. 459–471, Dec. 2002.

[20] M. H. Hung and C. H. Hsieh, "Event detection of broadcast baseball videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 12, pp. 1713–1726, Dec. 2008.

[21] I. P. Malinici and L. Carin, "Infinite hidden Markov models for unusual-event detection in video," *IEEE Trans. Image Process.*, vol. 17, no. 5, pp. 811–822, May 2008.

[22] C. Xu, Y. F. Zhang, G. Zhu, Y. Rui, H. Lu, and Q. Huang, "Using webcast text for semantic event detection in broadcast sports video," *IEEE Trans. Multimedia*, vol. 10, no. 7, pp. 1342–1355, Nov. 2008.

[23] V. S. Tseng, J.-H. Su, J.-H. Huang, and C.-J. Chen, "Integrated mining of visual features, speech features and frequent patterns for semantic video annotation," *IEEE Trans. Multimedia*, vol. 10, no. 2, pp. 260–267, Feb. 2008.

[24] G. Akrivas, G. Stamou, and S. Kollias, "Semantic association of multimedia document descriptions through fuzzy relational algebra and fuzzy reasoning," *IEEE Trans. Syst., Man, Cybern. A: Syst., Humans*, vol. 34, no. 2, pp. 190–196, Mar. 2004.

[25] A. Nagasaka and Y. Tanaka, "Automatic video indexing and full-video search for object appearances," in *Visual Database Systems II*, E. Knuth and L. M.Wegner, Eds. Amsterdam, The Netherlands: North-Holland, 1992, pp. 113–127.

[26] D. A. Sadlier and N. E. Oconnor, "Event detection in field sports video using audio-visual features and a support vector machine," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 10, pp. 1225–1233, Oct. 2005.

[27] A. M. Tekalp. *Digital Video Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1995.

[28] L. Xie, S. F. Chang, A. Divakaran, and H. Sun, *Structure Analysis of Soccer Video with Hidden Markov Models,* New York, USA: Columbia University, Apr. 2011.

[29] F. V. Jensen, *An Introduction to Bayesian Networks,* New York, USA: Springer, 1996.

[30] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification,* 2nd ed., New York: Wiley, 2001.

[31] C. K. Chow and C. N. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE Trans. IT-14*, vol. 3, no. 3, pp. 462–467, May 1968.

[32] C. M. Bishop, *Pattern Recognition and Machine Learning*, New York, USA: Springer Science Business Media LLC, pp. 55–57, 2006.

[33] E. Kovács and T. Szántai, "On the approximation of a discrete multivariate probability distribution using the new concept of t-Cherry junction tree," *Lecture Notes in Economics and Mathematical Systems*, Berlin: Springer, 2010, pp. 39–56.

[34] T. Szántai and E. Kovács, "Hypergraphs as a mean of discovering the dependence structure of a discrete multivariate probability distribution," *Ann. Oper. Res.*, vol. 193, no. 1, pp. 71–90, 2012.

[35] R. B. Nelsen, *An Introduction to Copulas,* 2nd ed. New York: Springer, 2006, pp. 7–14.

[36] A. Sklar, "Random variables, distribution functions, and copulas—a personal Look backward and forward," in *Distributions with Fixed Marginals and Related Topics*, Hayward, CA, USA: Institute of Mathematical Statistics, 1996, pp. 1–14.

[37] M. Fayzullin, A. Picariello, M. L. Sapino, and V. S. Subrahmanian, "The CPR model for summarizing video," *Multimedia Tools Application*, vol. 26, no. 2, pp. 153–173, 2005.

[38] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms,* 3rd ed., Cambridge, MA, USA: MIT Press, 2009.

Mostafa Tavassolipour received the B.Sc. degree with the highest honors from the Department of Engineering, Shahed University, Tehran, Iran, in 2009, and the M.Sc. degree from the Department of Computer Engineering, Sharif University of Technology, Tehran, Iran, in 2011.

He is a member of Image Processing Laboratory (IPL) since 2009. His current research interests include image processing, content-based video analysis, machine learning, and statistical pattern recognition.

Mahmood Karimian received the B.Sc. degree in computer engineering from the Department of Engineering, Ferdowsi University of Mashhad, Mashhad, Iran, in 2009, and the M.Sc. degree from the Department of Computer Engineering, Sharif University of Technology, Tehran, Iran, in 2011.

He joined Image Processing Laboratory (IPL) since 2009. His current research interests include machine learning, pattern recognition, and computer vision.

Shohreh Kasaei (M'05–SM'07) received the B.Sc. degree from the Department of Electronics, Faculty of Electrical and Computer Engineering, Isfahan University of Technology, Khomeynishahr, Iran, in 1986, the M.Sc. degree from the Graduate School of Engineering, Department of Electrical and Electronic Engineering, University of the Ryukyus, Okinawa, Japan, in 1994, and the Ph.D. degree from Signal Processing Research Centre, School of Electrical and Electronic Systems Engineering, Queensland University of Technology, Queensland, Australia, in 1998.

She is currently a Full Professor and the Director of Image Processing Laboratory (IPL) with the Sharif University of Technology since 1999. Her current research interests include multiresolution texture analysis, 3-D computer vision, 3-D object tracking, scalable video coding, image retrieval, video indexing, face recognition, hyperspectral change detection, video restoration, and fingerprint authentication.