Machine Vision and Applications (2018) 29:891–913
https://doi.org/10.1007/s00138-018-0944-9

ORIGINAL PAPER

Temporal segmentation and recognition of team activities in sports

Cem Direkoğlu[1] · Noel E. O'Connor[2]

Received: 17 January 2017 / Revised: 10 March 2018 / Accepted: 24 April 2018 / Published online: 24 May 2018

Abstract

A method for temporal segmentation and recognition of team activities in sports, based on a new activity feature extraction, is presented. Given the positions of team players from a plan view of the playground at any given time, we generate a smooth distribution on the whole playground, termed the position distribution of the team. Computing the position distribution for each frame provides a sequence of distributions, which we process to extract motion features for activity recognition. We can classify six different team activities in European handball and eight different team activities in field hockey datasets. The field hockey dataset is a new, large and challenging dataset that is presented for the first time for continuous segmentation of team activities. Our approach is different from other trajectory-based methods. These methods extract activity features using the explicitly defined trajectories, where the players have specific positions. In our work, given the specific positions of the team players at a frame, we construct a position distribution for the team on the whole playground and process the sequence of position distribution images to extract activity features. Extensive evaluation and results show that our approach is effective.

Keywords Computer vision · Sport video analysis · Team activity recognition · Temporal segmentation · Motion analysis

1 Introduction

Analyzing complex and dynamic sport scenes for the purpose of team activity recognition is an important task in computer vision. Team activity recognition has a wide range of possible applications such as analysis of team tactic and statistics (i.e., especially useful for coaches and trainers), video annotation and browsing, automatic highlight identification, automatic camera control (useful for broadcasters). Despite the fact that there is much research on vision-based activity analysis for individuals [1], group activity analysis remains a challenging problem. In group activity, there are usually many people located at different positions and moving in different individual directions making it difficult to find effective features for higher-level analysis.

There are mainly two possible sources of sport videos: TV broadcasts and multiple video feeds from fixed cameras around the playing field. We first review group activity analysis techniques using broadcast videos and then review methods which investigate sport videos captured by fixed multi-camera systems.

1.1 Using the TV broadcast

Kong et al. [2] use optical flow-based features and the latent-dynamic conditional random field model to recognize three different actions (i.e., left side attacking, stalemate and left side defending) in soccer videos. Later, Kong et al. [3] proposed an alternative approach to recognize the same activities in soccer videos. They use scale-invariant feature transform (SIFT) key point matches on two successive frames and a linear SVM to classify activities. Wei et al. [4] aim to discriminate group activities in broadcast videos targeting identification of football, basketball, tennis or badminton. They extract space–time interest points and use the probability summation framework for classification. Li et al. [5] proposed a discriminative temporal interaction manifold (DTIM) framework to characterize group motion patterns in American football games. For each class of group activity, they learn a multi-modal density function on the DTIM using the players' role and their motion trajectories. Then a maximum a posteriori (MAP) classifier is used to

✉ Cem Direkoğlu
cemdir@metu.edu.tr

Noel E. O'Connor
noel.oconnor@dcu.ie

[1] Department of Electrical and Electronics Engineering, Middle East Technical University - Northern Cyprus Campus, via Mersin 10, Turkey

[2] INSIGHT Centre, Dublin City University, Dublin, Ireland

recognize activities. They can recognize five different activity types. Swears and Hoogs [6] also present a framework to recognize different offense types in the context of American football. First, a broadcast video is stabilized and registered to another domain. This process normalizes the plays into a common coordinate system and orientation. Players' trajectories are then extracted for activity analysis. The temporal interactions of the players are modeled using a non-stationary kernel hidden Markov model. Ibrahim et al. [7] proposed a group activity recognition framework and experiment on a volleyball dataset. The team activity is predicted based on the dynamics of the individual people performing the activity. They build a deep learning model to capture these dynamics based on long short-term memory (LSTM) models. Shu et al. [8] recognize group activities in volleyball game using a LSTM network that forms a feed-forward deep architecture. Instead of using the common softmax layer for prediction, they introduce an energy layer and estimate the energy of the predictions. There are also some recent approaches, [9,10], using convolutional neural networks for action recognition in ice hockey and football games, respectively.

Despite the existence of such approaches, using a TV broadcast is not effective for group activity analysis, since the camera usually captures the region of interest (such as ball locations) and many players may not be in that region. Using broadcast cameras also suffers from inaccurate player localization because of occlusions, camera motion, etc.

1.2 Using fixed multiple cameras

Most team activity analysis methods [11–17] use a fixed multi-camera system around the playing field to overcome the limitations of using broadcast data. The multi-camera system usually has a camera configuration to cover all locations on the playground and is therefore able to capture all players simultaneously. Player detection and tracking algorithms are employed in the videos to obtain the trajectories, and then these trajectories are transformed into the top view of the playing field for more accurate analysis. In the activity analysis stage, features (e.g., position and speed) are extracted using the explicitly defined trajectories and a model employed (e.g., Bayesian net, hidden Markov models or SVM) to recognize the group activities such as different types of offense and defense. These models are summarized below.

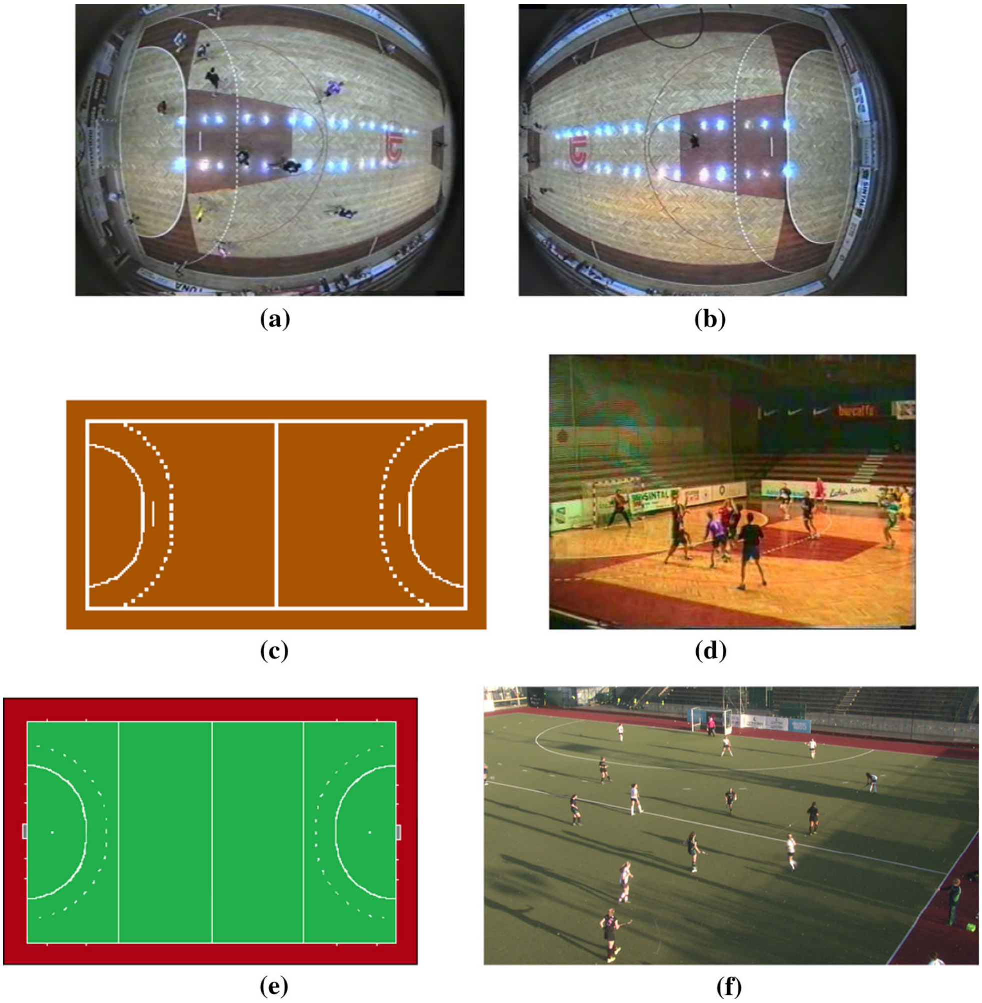
Intille and Bobick [11] use Bayesian belief networks for probabilistically representing and recognizing multi-agent action from noisy trajectories in American football. Blunsden et al. [12] extract features from the trajectory data and classify different offense and defense types in European handball using an SVM. Perse et al. [13] segment the play into three different phases (offense, defense and time-out) in a basketball game using a mixture of Gaussians. Then a more detailed analysis is performed to define a semantic description of the observed activity. Perse et al. [14] also present another approach which uses petri nets (PNs) for the recognition and evaluation of team activities in basketball. Hervieu et al. [15] use a hierarchical parallel semi-Markov model to represent and classify team activities in handball. Recently, Dao et al. [16] have proposed a sequence of symbols which are derived from the distribution of players' positions in a period of time to represent and recognize offensive types (e.g., side attack and center attack) in soccer games. Li and Chellappa [17] also address the problem of recognizing offensive play strategies in American football using a probabilistic model. Varadarajan et al. [18] introduced a topic model approach to represent and classify American football plays. They develop a framework that uses player trajectories as inputs to maximum entropy discriminative latent Dirichlet allocation (MedLDA) for supervised activity learning and classification. Montoliu et al. [19] present a methodology for team activity recognition in handball games based on Author Topic Model (ATM). They use two synchronized and stationary bird's-eye view cameras and extract optical flow-based activity features from the video frames. The evolution of motions and the recognition of team activities are based on the ATM model.

In this paper, we present a framework for temporal segmentation and recognition of team activities which is based on players' trajectories on the top view of the playing field. Our motivation and contribution are explained below.

2 Our motivation and contribution

In team activities, there is a group of people (the team) performing activities on the constrained playground. All of the existing trajectory-based methods analyze the specific positions (set of points) obtained by either vision-based tracking or GPS-based wearable sensors. There are two main drawbacks in these approaches. First, the position information is noisy. Second and the most important drawback is that they use only specific positions and ignore the rest of the playground. By its very nature, team activity takes place over the whole playground as the entire team reconfigures itself to either attack or defend. Thus, we believe that a more holistic approach is required rather than simply considering a collection of specific player locations.

In this paper, we propose an approach that analyzes the entire playground for activity feature extraction. Given the team players' positions from a plan view of the playing field at any given time, we solve a particular Poisson equation to generate a smooth distribution that we term the position distribution of the team. The position distribution is computed at each frame to form a sequence of distributions. Then, we process the sequence of position distributions to



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 1 **a** Sample frame from the fixed Camera 1. **b** Sample frame from the fixed Camera 2. **c** The top view of the handball court. **d** Sample image from the European handball game. **e** The top view of the field hockey playground. **f** Sample image from the field hockey game

extract motion-information images for each frame, where the motion-information images are obtained using frame differencing and optical flow. Finally, we compute weighted moments (up to second order) of these images to represent motion features at each frame. The proposed motion features are experimented with support vector machine (SVM) classification and evaluated on two different datasets: European handball and field hockey datasets. The European handball dataset [20] is publicly available, and the position information (trajectories) of the players is collected using a similar multi-camera capture setup to those reported previously, where sample frames from these cameras are shown in Fig. 1a, b. The top view of the European handball court and a sample image from the handball game are shown in Fig. 1c, d, respectively. We also created a larger dataset for a different game with more activities, the field hockey dataset, to conduct extensive experiments on team activity segmentation and recognition. In the field hockey dataset, the position information is collected using GPS-based wearable sensors. The top view of the field hockey playground and a sample image from the game are shown in Fig. 1e, f. Results show that we can temporally segment and recognize six different team activities in handball, and eight different team activities in field hockey. We also perform better than a method [12] that analyzes the explicitly defined trajectories for recognition, and better than a method based on pretrained convolutional neural network (AlexNet) [21].

Our method is novel and different from other trajectory-based methods presented in Sect. 1.2. These methods extract activity features using the explicitly defined trajectories, where the players have specific positions at any given time, and ignore the rest of the playground. In our work, on the other hand, given the specific positions of the team players at a frame, we construct a position distribution for the team on the whole playground and process the sequence of position distribution images to extract motion features for activity recognition. As no tracking and positioning algorithm (vision based or GPS based) can be 100% accurate, the position dis-

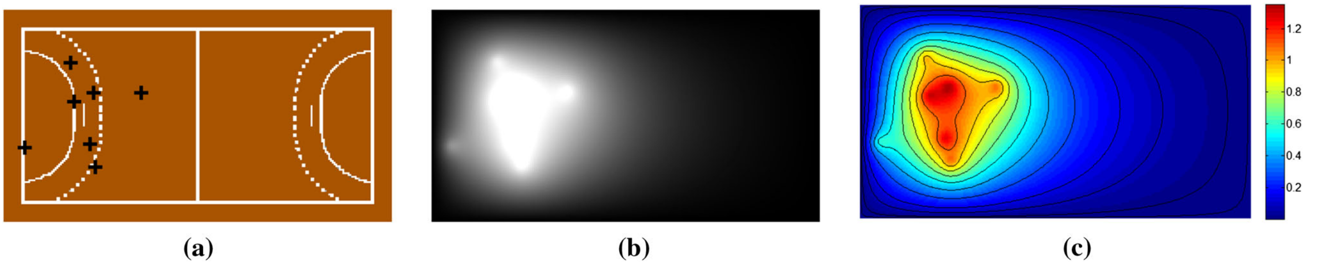


Fig. 2 The Poisson equation is applied to generate the position distribution. **a** The top view of the handball court with player locations. **b** The position distribution of the team. **c** The color-mapped position distribution with level sets

tribution accounts for the uncertainty of players' positions and it is defined on the whole playground which can be considered as an intensity image. Representing the positions of the team players as an intensity image instead of a set of points at any given time allows us to use frame differencing and optical flow, which are important techniques for image motion description. We extract motion features at each frame using the sequence of position distribution images instead of using the explicitly defined trajectories to represent activities.

Earlier versions of this work were presented in [22,23]. In [22] we verified that a particular Poisson equation can be used to determine the region of highest population, corresponding to the area with the highest density of the majority of players, and to estimate the region of intent, corresponding to the region toward which the team is moving as they press for territorial advancement. Then in [23] we significantly extended this early work [22] to perform full classification of team activity. In [23], we were not concerned about the region of intent or the region of highest population, and it was an independent piece of work. However, the continuous classification of team activities in [23] was only investigated on European handball dataset, which is a publicly available small dataset. In this paper, we create and investigate a larger and challenging dataset (i.e., field hockey). We conduct extensive evaluations for our method while comparing with the most related method [12] that is designed to recognize similar activities. In particular, we assess the accuracy in detail, conduct time evaluations and study the effect of window size.

3 Team position distribution generation

We illustrate the problem in the context of European handball, where the top view of the handball field of play with the team player positions is shown in Fig. 2a. (A European handball team has 7 players.) Given the positions of the team players at any time, we aim to generate a position distribution of the team defined on the whole playground. There are many possible probability distribution models (e.g., Gaussians, Laplace or Cauchy distribution), which can be centered on each player position and then summed to generate a position distribution of the team. Since the activity is performed on the bounded playground and players have to be on the playground to be involved in the team-based activity, the position distribution must be zero outside the playground. This can be achieved by using the truncated versions (e.g., truncated Gaussians) of the probability distributions. However, most of the probability distribution models which can be used to create a smooth distribution and account for uncertainty for the positions are parameter dependent and the parameters need to be adjusted to optimize the performance of the team activity recognition. In our work, we choose to solve a particular Poisson equation to generate a position distribution since it has a unique and steady-state solution with respect to the given team player positions. The proposed Poisson equation is parameter-free and can model zero probability outside the playground without any truncation. The solution of the proposed Poisson equation only depends on the players positions.

3.1 Background to the Poisson equation

In mathematics, the Poisson equation is an elliptic-type partial differential equation [24] which arises usually in electrostatics, heat conduction and gravitation. The general form of the Poisson equation, in two dimensions, is given by,

$$\nabla^2 I(\mathbf{x}) = -Q(\mathbf{x}), \tag{1}$$

where Q is a real-valued function of a space vector $\mathbf{x} = (x, y)$ and it is known as the source term, I is the solution which is also a real-valued function, and ∇^2 is the spatial Laplacian operator. Given a source term $Q(\mathbf{x})$, we find a solution for $I(\mathbf{x})$ that satisfies the Poisson equation and the boundary conditions over a bounded region of interest. There are three general types of boundary conditions: Dirichlet, Neumann and mixed. Here, we explain the Dirichlet condition, which is used in our algorithm. In the Dirichlet condition, the boundary values (solutions) are specified on the boundary. These values can be a function of space or can be constant. The Dirichlet condition is represented as $I(\mathbf{x}) = \Phi(\mathbf{x})$, where

$\Phi(\mathbf{x})$ is the function that defines the solution at the boundary layer.

3.2 The proposed Poisson equation and solution

The proposed Poisson equation and the resulting distribution (solution) are obtained based on the following considerations. The top-view image of the field of play is assumed to be a binary image where the player positions are one and the rest of the positions are zero at any time during the game. Although players are expected to be in the play area during the game, players sometimes can move a little outside for a variety of different reasons, such as to serve the ball, when the ball is out or in order to talk to the coach. Thus, we expand the binary image of the field of play to include the possibility that the players may move a little outside the lines. The binary image is defined to be the source term in the Poisson equation. The boundary condition is Dirichlet which has a specific solution, $I(\mathbf{x}) = 0$, at the boundaries of the expanded field of play. This means that there is no possibility for a player to be outside the region of interest. The proposed Poisson equation problem is,

$$\begin{aligned} &\nabla^2 I(x, y) = -\left(\sum_{i=1}^{N} \delta(x - x_i, y - y_i)\right) \\ &I(x, y) = 0, \quad \text{boundary condition}, \end{aligned} \tag{2}$$

where N is the number of players in the team and (x_i, y_i) is the position of player i. The source function is assumed to be a linear combination of Dirac delta functions $\delta(.)$ in two dimensions. It is important to note that the proposed Poisson equation has a unique and steady-state solution at each frame. The solution is parameter-free, and it only depends on the position of the players. Therefore, when players change their position from the previous frame to the current frame, the solution also changes in the current frame.

The numerical solution methods of the Poisson equation can be categorized as direct and iterative methods. In [25], Simchony et al. pointed out that direct methods are more efficient than multi-grid-based iterative methods for solving the Poisson equation on a rectangular domain, since direct methods can be implemented using the fast Fourier transform (FFT). In our work, since the field of play is rectangular, we employ FFT-based direct methods to solve the proposed Poisson equation. The proposed equation has a Dirichlet boundary condition that needs discrete sine transforms (using FFT) to achieve an exact solution, where the detailed description of the solution method is given in [25]. The solution to the proposed equation forms peaks at the player positions. To smooth these peaks, we apply Gauss–Seidel iterations (5 iterations), as a post-processing stage, to relax the surface while maintaining the boundary condition ($I(\mathbf{x}) = 0$) outside the region of interest.

The resultant distribution provides the likelihood of a position to be occupied by players at any given time, and it is called the position distribution of the team. Figure 2b shows the position distribution for the given example, and Fig. 2c shows the same distribution with color mapping and with level sets. For handball, the resolution of the position distribution image is 220 × 120 in our experiments. For field hockey, it is 372 × 240.

4 Motion-information images and feature extraction

Computing the position distribution for each frame provides a sequence of position distributions. We process the sequence of distribution images to generate motion-information images which can describe motion at each frame. The motion-information images are created using frame differencing and optical flow.

4.1 Frame differencing

The simplest way in which we can detect motion is by image differencing. Figure 3a shows the direction of movement of the team players from the current frame to the next frame (50 frames later), where the starting point of the arrow represents the position of the player at the current frame and the end point represents the position of the player at the next frame. We compute the position distribution for the team at the current and at the next frames. Since the team players move from the current positions to the next positions, they create higher position distribution values in the direction of movement. To detect motion with the direction, we apply change detection by simply subtracting the current distribution from the next distribution and keep the positive values while setting the negative values to zero, i.e., $(I(x, y, n+m) - I(x, y, n)) > 0$, where $I(x, y, n)$ represents the position distribution of the team at frame number n and m is the number of frames between the current and the next frames. Frame differencing is applied with 50 frames (i.e., $m = 50$) of temporal extent in our experiments. Figure 3b shows the frame differencing whereby we keep the positive values and set the negative values to zero for the given example.

4.2 Optical flow

Although frame differencing can provide some information about the movement, we cannot exactly see how the distribution points move. In order to describe the position changes at each frame, we compute optical flow vectors that can

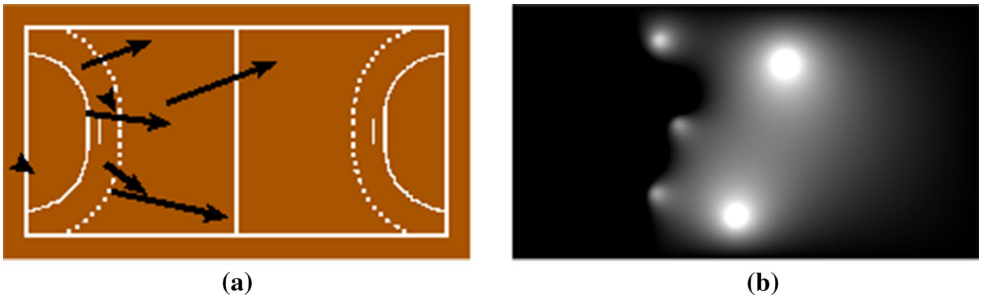


Fig. 3 Generating a motion-information image using frame differencing. **a** Team players movements. **b** The motion-information image

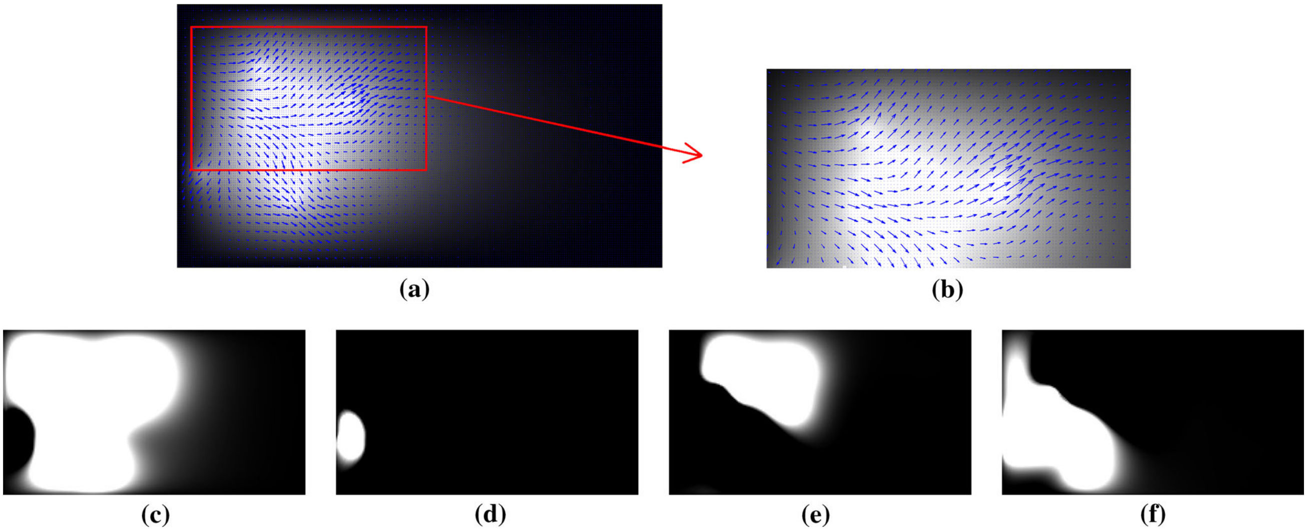


Fig. 4 Computing the directional speed images to represent the motion-information images. **a** The position distribution and the estimated optical flow. **b** The zoomed in image from the red box in (**a**). **c** Directional speed image in the direction of positive x-axis, **d** negative x-axis, **e** positive y-axis and **f** negative y-axis (color figure online)

provide the displacement of the points with directions. We employ the classical Horn and Schunck (HS) method [26] for optical flow estimation. This is a differential approach which combines a data term that assumes constancy of some image property (e.g., brightness constancy and gradient magnitude constancy) with a spatial term that models how the flow is expected to vary across the image (e.g., smoothness constraint). An objective function combining these two terms is then optimized. In our experiments, we observed that using the gradient magnitude constancy assumption (i.e., $|\nabla I(x, y, n)| = |\nabla I(x+u, y+v, n+m)|$) instead of using the brightness constancy (i.e., $I(x, y, n) = I(x+u, y+v, n+m)$) can better estimate the optical flow, where u is the horizontal optical flow and v is the vertical optical flow. Therefore, in our work, we use the gradient magnitude constancy assumption together with the smoothness constraint to compute the optical flow on the playing field. The gradient of the position distribution is computed using the Sobel operator, and the optical flow is computed from the current frame to the next frame with 8 frames of temporal extent. There are also two parameters that affect the solution of the HS method: A parameter that reflects the influence of the smoothness term is set to 0.1, and the number of iterations to achieve the solution is set to 200. Figure 4a shows the position distribution image and the estimated optical flow. For better illustration, Fig. 4b shows the zoomed in image from the red box in Fig. 4a. Note that this is a novel algorithm to compute the motion field on the top view of the playground. Kim et al. [27] compute the motion field on the top view of the playground by interpolating the player's motion vectors, where the player's motion vectors are generated using the specific positions of the players. On the other hand, in our algorithm, we use the specific positions to generate the position distributions, and then estimate the motion field using optical flow.

The motion-information images, using the optical flow, are generated with the following considerations. The horizontal and vertical components (i.e., u and v) of the flow are two different scalar fields. Each of these components is half-wave-rectified to generate four nonnegative channels:

u^+, u^-, v^+, v^-, so that $u = u^+ - u^-$ and $v = v^+ - v^-$. These channels, u^+, u^-, v^+ and v^-, represent directional speed images in the direction of positive x-axis, negative x-axis, positive y-axis and negative y-axis, respectively. Note that the directional speed images have also been used in [28] for individual action recognition, but their usage for group activity recognition as proposed here is novel. The directional speed images are illustrated in Fig. 4c–f for the given example.

4.3 Feature extraction

We use five motion-information images to describe motion at each frame, where one of them is obtained with frame differencing and the other four are obtained with optical flow. Frame differencing is applied with 50 frames of temporal extent, while the optical flow is computed with 8 frames of temporal extent, so that frame differencing captures motion in a longer period of time, while the optical flow captures motion in a shorter period of time. Our experiments show that describing the motion in this way performs better than other options.

Next, we compute weighted moments for each motion-information image to represent motion features at that frame. The discrete form of the equation is,

$$m_{pq} = \sum_x \sum_y w(x, y) x^p y^q \Delta x \Delta y. \qquad (3)$$

Here, m_{pq} is the moment of order p and q, $w(x, y)$ is the weight function, which we substitute for each motion-information image, and $\Delta x = \Delta y = 1$ are spacing sizes of a pixel. We compute moments up to order $p+q = 2$, resulting in 6 moments per image and 30 moments in total to describe the motion at each frame.

5 Classification using the motion descriptors

We investigate the use of the proposed features with support vector machine (SVM) classification. In SVM, a Gaussian radial basis function kernel is used. SVM is a powerful technique in classification; it maps training data to higher-dimensional space and constructs a separating hyperplane such that the distance between the hyperplane and a data point is maximized. Test data are then classified by the discriminant function.

In the handball dataset, the test frame is classified using the 141-by-141 neighborhood frames (141 from past and 141 from future neighborhoods), which is determined experimentally. This means that the window size is 283 (including the test frame). Each of the frames in the window is labeled with the SVM classifier by using the one-against-all method. Then the most frequent class is selected to represent the activity of the test frame. The scaling factor of the Gaussian kernel function is 2.4. The upper bound on the Lagrange parameter (i.e., the soft margin cost function parameter) is 10. In addition, we use the sequential minimal optimization method to find the separating hyperplane since we have a large dataset and this method is computationally efficient.

In the field hockey dataset, the test frame is classified using the 112-by-112 neighborhood frames, which means that the window size is 225 (including the test frame). Each of the frames in the window is labeled with the SVM classifier by using the one-against-all method, and then the most frequent class represents the activity of the test frame. The scale factor of the Gaussian kernel function is 2, and the upper bound on the Lagrange parameters is 10. The sequential minimal optimization method is used to find the separating hyperplane.

Table 1 Team activities in handball with their numbering

1. Slowly going into offense
2. Offense against set up defense
3. Offense fast break
4. Fast returning into defense to prevent fast break
5. Slowly returning into defense
6. Basic defense

6 Evaluation and results

The proposed model is evaluated on European handball and field hockey games. European handball is usually an indoor game; on the other hand, the field hockey is an outdoor game with a larger field of play.

6.1 Evaluation on European handball dataset

In handball, there are seven players and it is played on a 40-by-20-meter court. The dataset for the handball game is from the publicly available CVBASE dataset [20]. The dataset consists of 10 min of a handball game. The playground coordinates of the seven players of the same handball team are available throughout the sequence. The sequence consists of 14978 frames (approximately 10 min). These trajectories are extracted from two bird's-eye view cameras, one above each part of the court plane, with semiautomatic tracking, where the details on trajectory extraction are given in [29]. The dataset providers obtained error estimates on players' positions in the playground between 0.3 and 0.5 meters. There are mainly six different team activities in this dataset, where the starting and end times of the activities are also annotated. The definition of the six team activities with their numbering is given in Table 1. The length of each activity sequence ranges from 125 frames to 1475 frames. It should be noted that some of these activities can be split into more complex activity classes; however, more information is required such as the ball trajectory or the trajectories of the opposing team to represent more complex activities, which is not provided in this dataset.

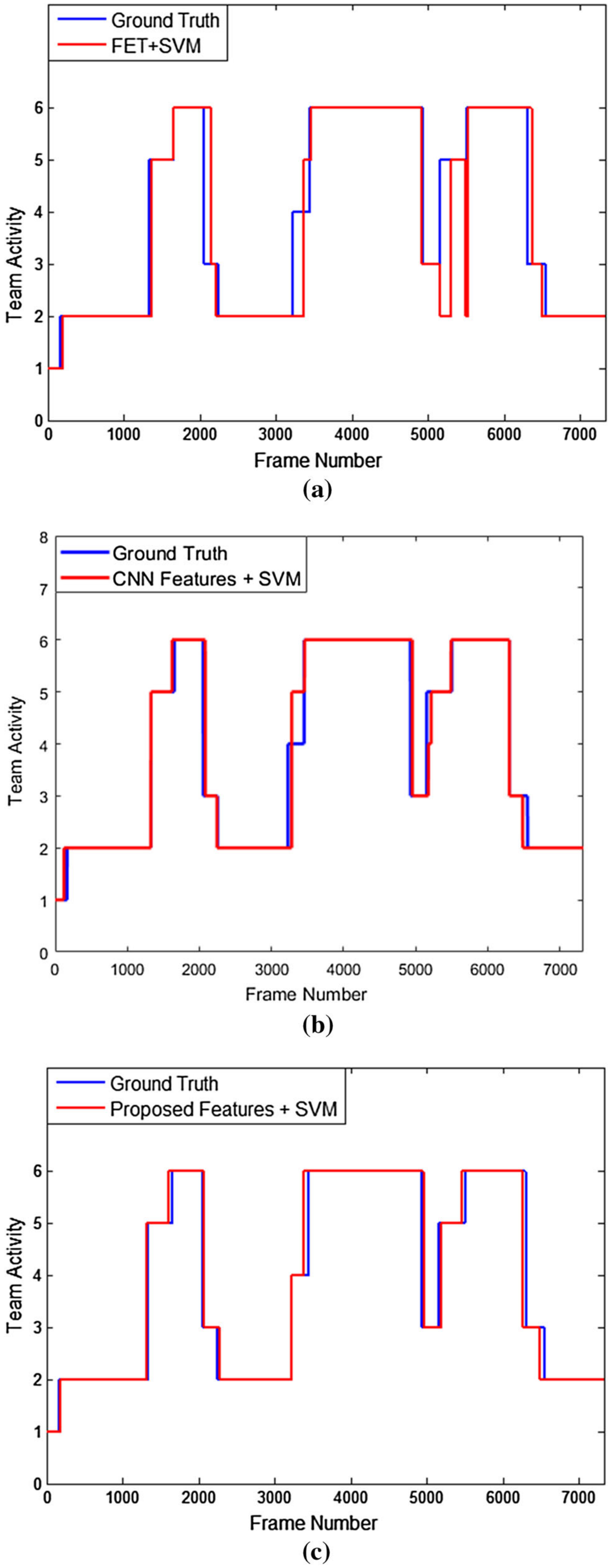
We evaluate our approach while comparing with two other models, namely a model proposed by Blunsden et al. [12] that analyzes the explicitly defined trajectories for team activity recognition, and a model based on a pretrained convolutional neural network (CNN) [21] that extracts CNN features from the motion-information images, defined in Sect. 4, for activity representation. The model proposed by Blunsden et al. [12] was designed to recognize the same activities in the same dataset, which we believe to be one of the best comparisons we can make given the current status of work in this area. They extract 5 features (i.e., positions, speed, directions) from each player trajectory, and then all the players' features are concatenated to form 35-dimensional feature vector to represent the activity at each frame. A SVM classifier is then trained upon these data. They use the one-against-all method for classification. The test frame is classified using the 99-by-99 neighborhood frames that make the window size 199 (including the test frame). Each frame in the window is labeled with the SVM classifier, and then the most frequent label represents the class of the test frame. A Gaussian kernel function is used, and the scaling factor is 2.4. The upper bound on the Lagrange parameters is 10. The sequential minimal optimization method is used to find the separating hyperplane.

We also compare the proposed features with a pretrained convolutional neural network (i.e., AlexNet) [21] as a feature extractor. In particular, we keep the proposed system architecture the same, but use CNN features instead of the proposed features. We generate the five motion-information images as defined in Sect. 4. Each image is resized to AlexNet image requirements (i.e., 227 x 227). Then we extract CNN features for each image using the last layer of the CNN (i.e., fc8 layer of the AlexNet which is the last layer before classification). We extract 1000-dimensional feature vector from each image, and then features extracted from the five different images are concatenated to form 5000-dimensional feature vector. This feature vector represents the activity at each frame. Finally, we use a linear SVM to train and classify activities. We use the one-against-all method for classification. The test frame is classified using the 100-by-100 neighborhood frames that make the window size 201 (including the test frame). Each frame in the window is labeled with the SVM classifier, and then the most frequent label represents the class of the test frame. The upper bound on the Lagrange parameters is 0.01. The sequential minimal optimization method is used to find the separating hyperplane.

6.1.1 Temporal segmentation and recognition

In our evaluation, the second half of the video is used for training (i.e., 7600 frames, 5 min and 4 s) and the first half is used for testing (i.e., 7328 frames, 4 min and 53 s). Both the first and second halves include the six different team activities. In the first half, there are 1, 3, 3, 1, 2 and 3 instances and in the second half there are 3, 3, 2, 2, 2 and 4 instances for activity number 1, 2, 3, 4, 5 and 6, respectively. Since proper training is required for robust classification, we choose the second half for training purposes. The second half includes more activity samples than the first half, e.g., the activity number 1 is performed once in the first half and three times in the second half. In the training, there are at least two segments and at most four segments to represent an activity. On the other hand, in the testing, there are at least one segment and at most three segments to represent an activity. In addition, since we are testing the continuous sequence, there are also time-out segments which occur when the ball is out or when play is stopped. In handball, when it is time-out, teams keep moving and start to perform the next activity, e.g., if they are serving the ball, they move around to create space; on the other hand, if the opponent team is serving the ball, they move around to prevent the pass. Therefore, each of the time-outs in the test sequence is defined to be the following activity in our experiments.

In continuous classification, we classify all individual frames. We evaluate our features with the SVM classification, and all the details related to the classification are provided in Sect. 5. The same evaluations are also conducted for the method proposed by Blunsden et al. [12], and for the method based on CNN features [21] for comparison. In evaluations, the method proposed by Blunsden et al. [12] is denoted by FET + SVM, which means that features are obtained using the explicitly defined trajectories and the classification is achieved using the support vector machines. On the other hand, the method based on CNN features [21] is denoted by CNN + SVM, which means that CNN features are trained and classified using support vector machines. Figure 5a shows the temporal segmentation and recognition results obtained by the FET + SVM, Fig. 5b shows the temporal segmentation and recognition results obtained by the CNN + SVM, while Fig. 5c shows the results obtained by the proposed features with SVM (proposed features + SVM), respectively. The blue graph represents the ground truth and the red graph represents the prediction. It is observed that the proposed features with SVM achieves better temporal segmentation and recognition than the FET + SVM and CNN + SVM. The FET + SVM and CNN + SVM cannot identify activity number 4, which is fast returning into defense, and confuses this with activity number 5, which is slowly returning into defense. The FET + SVM also confuses between activity numbers 2 and 5, which is offense against set up defense and slowly returning



(a)

(b)

(c)

Fig. 5 Temporal segmentation and recognition of activities. **a** FET + SVM (proposed by Blunsden et al. [12]), **b** CNN features with SVM and **c** proposed features with SVM

Table 2 Correct classification rates (CCR%) of the proposed features + SVM, CNN features + SVM and FET + SVM (total frames: 7328)

Methods	FET + SVM [12]	CNN + SVM	Proposed features + SVM
CCR%	89.74%	92.97%	94.61%

into defense, respectively. There are also some errors when the activity switches in FET + SVM and CNN + SVM. The proposed features with SVM can recognize the six different activities, and the errors occur when the activity switches.

As stated by Ward et al. [30], there are two alternatives for scoring in the evaluation of activity recognition: frames and events. Our evaluation is based on scoring the frames which is an acceptable validation method and which we believe puts us in line with best practice especially when we both segment and classify continuous videos. We classify 7328 test frames in the evaluation, and Table 2 shows the correct classification rate (CCR%) for each method. The CCR% is computed as $CCR\% = (C_c/T_c) \times 100$, where C_c is the number of correct classification and T_c is the number of total classification. The FET + SVM achieves 89.74%, the CNN + SVM achieves 92.97%, and the proposed features with SVM achieves 94.61% recognition rate. Results show that the proposed features with SVM performs around 4.9% better than the FET + SVM, and around 1.6% better than the CNN + SVM. Results show that the proposed features perform significantly better than the other features with the same classifier, i.e., SVM.

Table 3 illustrates the precision and recall results, for each activity class, obtained using each method. Here, the precision for a class is defined as $P\% = (P_c/P_t) \times 100$, where P_c is the number of frames correctly predicted as belonging to that class and P_t is the total number of frames predicted as belonging to that class. The recall for a class is defined as $R\% = (R_c/R_t) \times 100$, where R_c is the number of frames correctly predicted and R_t is the total number of frames that actually belong to that class. In this table, both the precision and recall must be high for a method to show that it can handle activity switches and provide sufficient discrimination. There is only one activity, i.e., activity 6, in Table 3, where the FET + SVM [12] has slightly better precision and better recall than the proposed features + SVM. In general, the proposed features + SVM has better performance than the FET + SVM [12]. The main problem of the FET + SVM method is that it cannot discriminate activity number 4 and it is sensitive to activity switches. The CNN + SVM has the same problem with FET + SVM. The CNN + SVM method cannot discriminate activity number 4, and it is also sensitive to activity switches. On the other hand, the proposed features with SVM can discriminate all activities and can handle activity switches better than the FET + SVM and CNN + SVM.

Table 3 Precision (P%) and recall (R%) of the proposed features + SVM, CNN features + SVM and FET + SVM for each activity in handball dataset

Act. No.	# of frames	FET + SVM [12]		CNN + SVM		Prop. features + SVM	
		(P%)	(R%)	(P%)	(R%)	(P%)	(R%)
1.	164	82.41	100.0	100.0	73.78	88.17	100.0
2.	2914	86.48	98.79	94.42	99.97	97.98	98.28
3.	675	96.96	61.48	95.44	80.59	84.76	84.88
4.	225	0.0	0.0	0.0	0.0	97.38	66.22
5.	675	83.92	70.37	75.94	84.15	99.46	82.07
6.	2675	94.19	98.80	96.46	99.70	92.79	98.28

Table 4 Confusion matrix for FET + SVM [12]

	1	2	3	4	5	6
1	1	0	0	0	0	0
2	0.01	0.99	0	0	0	0
3	0	0.15	0.62	0	0	0.23
4	0	0.64	0	0	0.36	0
5	0	0.27	0.02	0	0.70	0.01
6	0	0.01	0	0	0	0.99

The number at row and column is the proportion of row class which is classified as column class at frame level

Table 5 Confusion matrix for CNN features with SVM

	1	2	3	4	5	6
1	0.74	0.26	0	0	0	0
2	0	1	0	0	0	0
3	0	0.11	0.81	0	0	0.08
4	0	0.23	0	0	0.77	0
5	0	0	0.04	0.06	0.84	0.07
6	0	0	0	0	0	1

Table 6 Confusion matrix for proposed features with SVM

	1	2	3	4	5	6
1	1	0	0	0	0	0
2	0.01	0.98	0.01	0	0	0
3	0	0.09	0.85	0	0	0.06
4	0	0	0	0.66	0	0.34
5	0	0	0.05	0	0.82	0.13
6	0	0	0.02	0	0	0.98

Tables 4, 5 and 6 show the confusion matrix for six different activities by FET + SVM [12], CNN + SVM and the proposed features + SVM, respectively. The number at row and column is the proportion of row class which is classified as column class at frame level. For example, in Table 6, 85% of the activity 3 class frames is correctly classified as activity 3 class, 9% is misclassified as activity 2 class, and 6% is misclassified as activity 6 class.

6.1.2 The effect of window size

We present the effect of differing window size in the classification performance (CCR%). Figure 6a shows the CCR% for the proposed features with SVM, for CNN + SVM model [21] and for the FET + SVM model [12]. The window size ranges from 51 to 351 in our evaluation. It is observed that the proposed features with SVM performs better than the other models at each window size. The optimal window size for the proposed features + SVM is 283. For the FET + SVM model, it is 199. For the CNN + SVM model, it is 201.

6.1.3 The effect of motion-information images

We present the influence of motion-information images and report what the temporal segmentation and the classification results would be if only frame differencing or only optical flow was used. Figure 6b shows the result obtained by using only frame differencing (one motion-information image). Figure 6c shows the result using only optical flow (four motion-information images), and Fig. 5c illustrates the result using the combination (five motion-information images). Frame differencing alone achieves 90.96%, optical flow achieves 92.82%, and the combination achieves 94.61%. Results indicate that using the combination improves the CCR% and the discrimination.

6.1.4 The effect of SVM kernel function

We present the influence of different SVM kernel functions in handball dataset. Three different kernel functions are experimented, which are linear kernel function, polynomial kernel function and Gaussian radial basis kernel function. Table 7 shows the performance of the proposed method with respect to these kernel functions. It is observed that the proposed method performs best with the Gaussian radial basis kernel function. Parameters of the kernel functions are selected using the fivefold cross-validation and grid search. We apply the cross-validation and grid search to the training part of the video. In linear SVM, we have a single parameter that is for the soft margin cost function (C). The optimal cost function parameter (C) is 0.25. In polynomial SVM, we have two parameters: the cost function parameter (C) and the order of polynomial function (P). The optimal parameter values are $C = 0.01$ and $P = 3$. Finally, in Gaussian radial basis

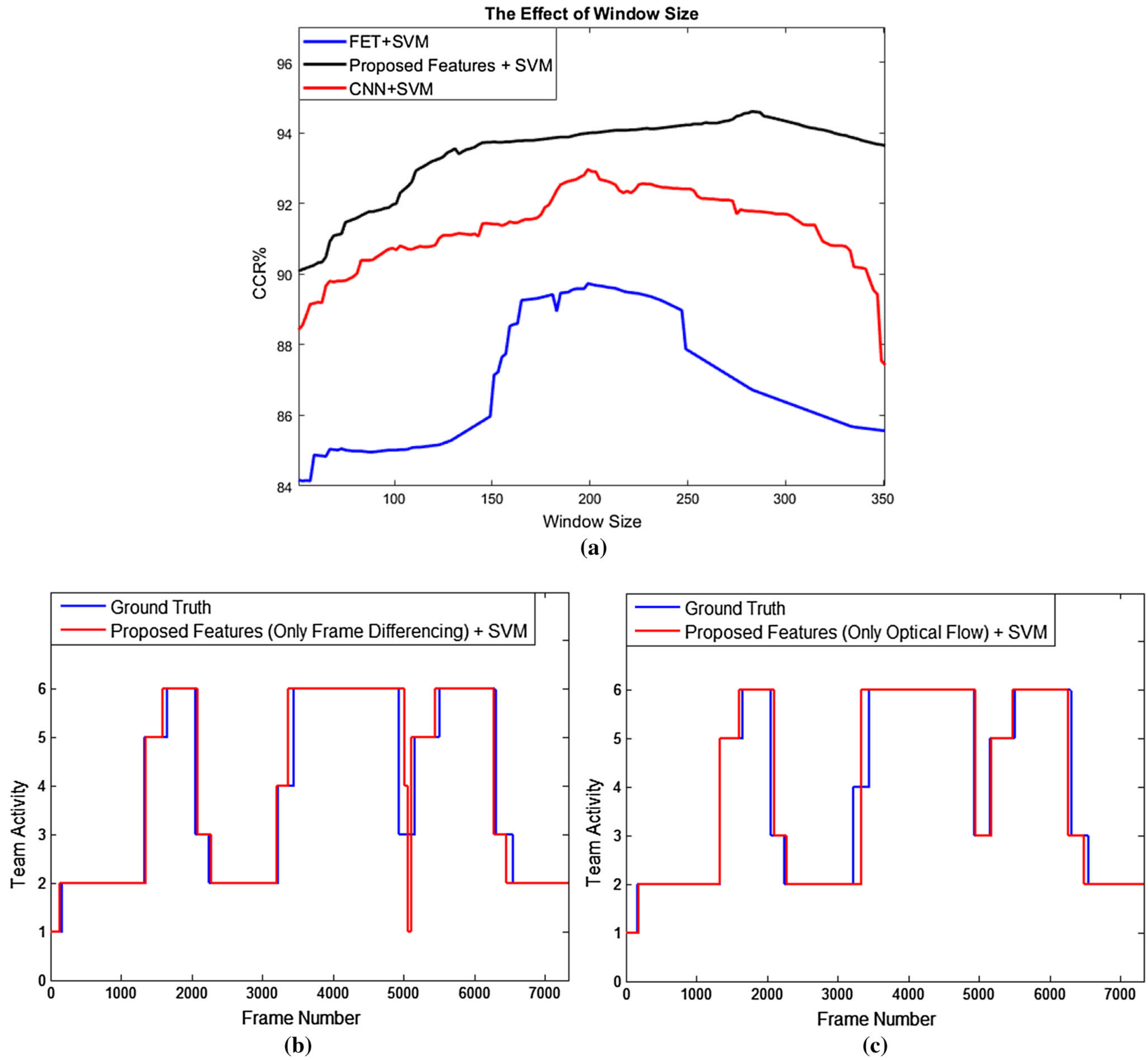


Fig. 6 Effect of window size and the effect of motion-information images. **a** Classification performances (CCR%) with differing window size. **b** Temporal segmentation and recognition using only frame differencing and **c** only optical flow

Table 7 Evaluation of SVM classifier with respect to different kernels in handball dataset

SVM kernel function	CCR%
Proposed features + SVM (Gaussian radial basis function kernel)	94.61
Proposed features + SVM (polynomial kernel)	94.33
Proposed features + SVM (linear kernel)	93.83

kernel function there are two parameters: the cost function parameter (C) and the Gaussian scale factor (S). The optimal parameter values are $C = 10$ and $S = 2.4$.

6.1.5 Computational efficiency

The computational time for each stage of the methods is given in Table 8. We also report the computation times if only frame differencing or only optical flow is used in our method. Results are obtained using MATLAB 7 on a Windows 7 operating system with Intel Core i3-870, 2.93 GHz and 8 MB RAM. It is observed that the FET + SVM method is more efficient than the proposed method with SVM and CNN + SVM especially in feature extraction. Although the proposed features (combination) with SVM is computationally less efficient than FET + SVM in feature extraction, it

Table 8 Computation time for each stage of the methods in handball dataset

Stages	Feature extraction in whole video	Training activities in the second half	Classifying activities in the first half
FET + SVM [12]	1.51 s	19.1 s	0.33 s
CNN + SVM	14,793.42 s	28.31 s	0.29 s
Proposed features + SVM (combination)	13,516.51 s	24.12 s	0.25 s
Proposed features + SVM (frame differencing)	983.28 s	55.23 s	0.27 s
Proposed features + SVM (optical flow)	12,524.34 s	30.5 s	0.24 s

has significantly better classification accuracy in comparison with FET + SVM and CNN + SVM.

6.2 Evaluation on field hockey dataset

Field hockey is an outdoor team sport. There are eleven players in a team, and the size of playground is 91.4 × 55 meters. The position data were collected using a SPIproX Global Positioning System (GPSports Systems Limited, Australia) [31]. The SPIproX equipment was supplied by Statsports company [32] (the UK and Ireland distributor of GPSsports). These GPS devices are one of the most advanced GPS-based tracking technologies on the market. The GPS devices record the position coordinates of players at a frequency of 15 Hz (15 data points per second). The validity and reliability of GPS usage in field hockey were studied by Macleod et al. [33]. They concluded that GPS is a reliable and valid measurement tool for assessing the movement patterns of field hockey players.

We collected a position dataset by recording a match between Irish national teams, i.e., U18s ladies versus U16s ladies. Note that U18s means that all of the team players are under 18 years old. We collected the positions of nine players of U18s team for a period of approximately 35 min of a game. There are eleven players in a team, but unfortunately two of the sensors stopped working during the game, and the position data of two players (i.e., the goalkeeper and a forward player) are missing in our dataset. This is a very natural condition in real-world situations, and our experiments are conducted with nine player positions to segment and recognize team activities.

We also recorded the same field hockey match using two different broadcast cameras, so that we can annotate team activities. The cameras capture 25 frames per second (fps); on the other hand, GPS sensors capture 15 fps. To synchronize the GPS data with the broadcast video data, we simply interpolate the GPS data, using linear interpolation, so that the position data are also at 25 fps. Finally, in total, the sequence consists of 53,201 frames (approximately 35 min). For evaluation, the dataset is divided into two parts, approximately from the middle. The first half has 26,599 frames, and the second half has 26,602 frames. Thus, we can train activities in the first half and test the activities in the second half, or we can train the activities in the second half and test the activities in the first half. There are eight different team activities in this dataset, where the starting and end times of these activities are annotated. The definition of the eight team activities with their numbering is given in Table 9. The length of each activity sequence ranges from 49 frames to 1975 frames.

Table 9 Team activities in field hockey with their numbering

1. Slowly going into offense
2. Offense against set up defense
3. Offense fast break
4. Fast returning into defense to prevent fast break
5. Slowly returning into defense
6. Basic defense
7. Stalemate middle right
8. Stalemate middle left

Note that there are two more activities in this dataset in comparison with the handball dataset. In field hockey, the playground is larger and there are more possible movements for a team. The stalemate middle right activity (i.e., activity number 7) is a transient state between offense and defense. In this activity, the team moves to the middle right to intercept the ball or to pass the ball patiently before the next activity. The stalemate middle left activity (i.e., activity number 8) is also a transient state between offense and defense. The team moves to the middle left to intercept the ball or to pass the ball before the next activity.

6.2.1 Temporal segmentation and recognition

In the first experiment, we train the activities in the first half of the video (i.e., 26,599 frames, approximately 17 min and 44 s) and test the activities in the second half (i.e., 26,602 frames, approximately 17 min and 44 s). Then, in the second experiment, we train the activities in the second half of the video and test the activities in the first half. Both the first and second halves include the eight different team activities.

In the first half, there are 18, 8, 6, 3, 11, 11, 13 and 15 instances and in the second half there are 19, 4, 3, 8, 8, 13,

Table 10 Quantity of the training sets for each activity

Activity number	First half		Second half	
	Total	Training	Total	Training
1.	6257	3500	4840	3500
2.	1776	1750	5282	3500
3.	445	445	957	957
4.	1761	1500	651	651
5.	1809	1800	3094	2500
6.	6264	3000	5131	3000
7.	5174	3300	3487	2500
8.	3116	2800	3083	2500

17 and 14 instances for activity number 1, 2, 3, 4, 5, 6, 7 and 8, respectively. In continuous classification, we classify all individual frames and all the details related to the classification are provided in Sect. 5. The same evaluations are also conducted for the FET + SVM [12] and CNN + SVM [21] for comparison purpose. In FET + SVM, the second half is classified using the 89-by-89 neighborhood of the test frame that makes the window size 179 (including the test frame). The first half is classified using the 81-by-81 neighborhood of the test frame that makes the window size 163 (including the test frame). In SVM, the scaling factor of the Gaussian kernel function is 3.2. The upper bound on the Lagrange parameters is 10. The sequential minimal optimization method is used to find the separating hyperplane. In CNN + SVM, the second half is classified using the 96-by-96 neighborhood of the test frame that makes the window size 193 (including the test frame). The first half is classified using the 100-by-100 neighborhood of the test frame that makes the window size 201 (including the test frame). In SVM, the upper bound on the Lagrange parameters is 0.01. The sequential minimal optimization method is used to find the separating hyperplane.

6.2.2 Training set selection

Training set selection is crucial for supervised classifiers. Large training sets usually increase the computation time and complexity of the training stage. In addition, training sets which include outlier features degrade the performance of the classifier during the testing phase. Outliers in pattern recognition could be defined as patterns whose characteristics are different from the majority of the patterns within the same class. In our domain, outliers could result from human annotation errors (i.e., especially at the activity transition points) or GPS device noise. Therefore, to optimize the computation time of the training stage and enhance the classifier performance, there is a need to clean and reduce the size of training set for each activity.

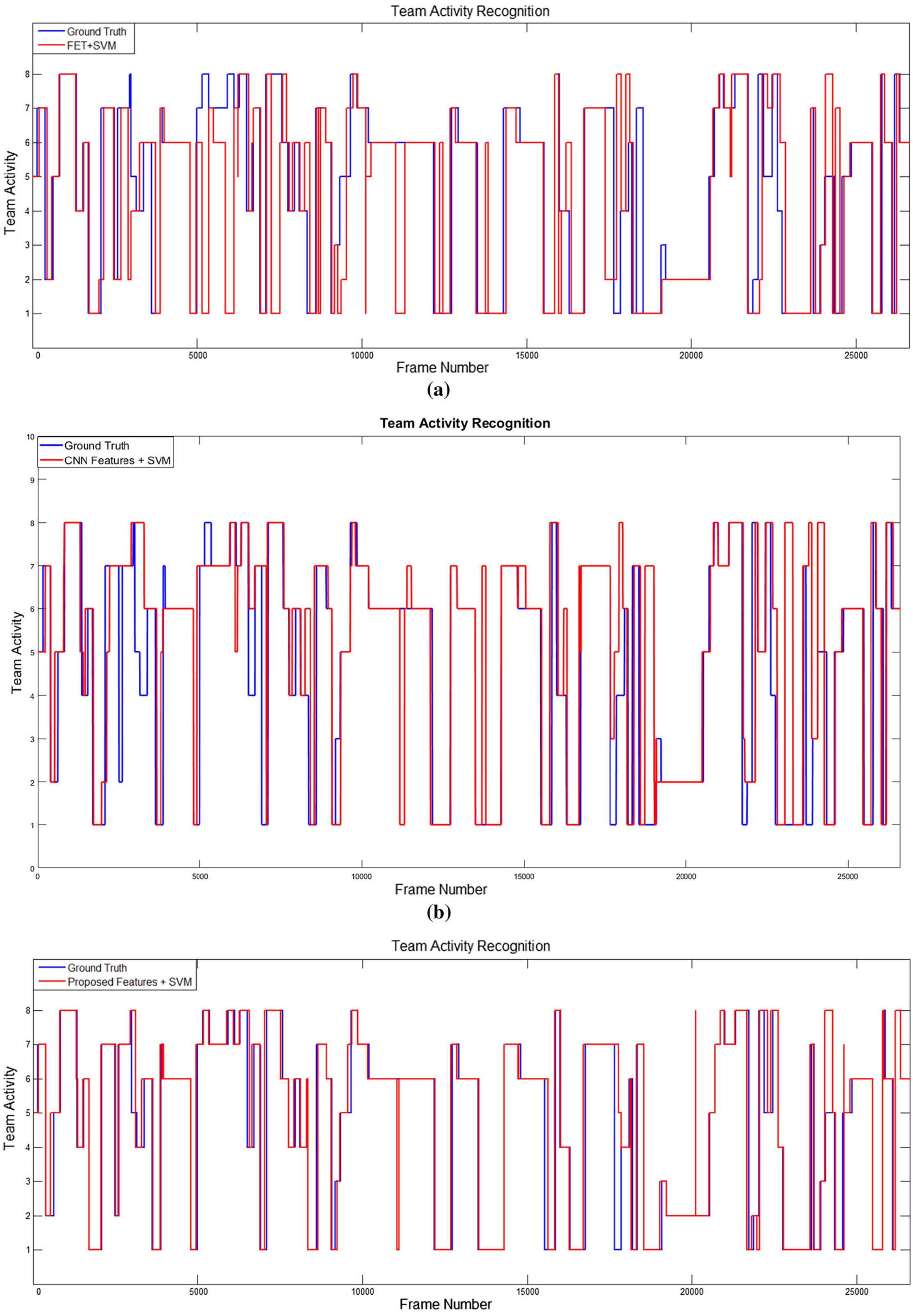
In our work, we use an outlier detection method to reduce the size of training sets. We use local reconstruction weight (LRW) [34,35] to detect and remove the outlier features. The LRW algorithm is based on a dimensionality reduction technique called locally linear embedding [34,35]. It starts with determining the k-nearest neighbors of all the points in the training set. Then it reconstructs each point as a linear combination of its neighbors. The reconstruction is done with linear regression, and points with large reconstruction weights will be outliers in the training set. There are two parameters to be set in this method: The number of nearest neighbors of each data point is 12, and the regularization parameter is 0.001. To reduce the size of training sets, we use the outlier detection MATLAB toolbox provided by Onderwater [36]. Table 10 shows the total number of samples (frames) in the first and second halves, and how many of these samples were included in the training set for each activity. For example, in the second half, there are 5131 samples from the activity class 6, and 3000 samples are used for training. The outlier detection method is used to reduce the size of samples, and the one-against-all method is used to train SVM classifiers. So when we train activity class 6, we use 3000 samples to represent class 6 and use the combination of samples from the other seven classes to represent the non-activity class 6. We select about 500 samples from each of the other seven training sets to form the not-activity class 6 samples. Note that if the total number of samples is less than 1000 for an activity, for example activity class 4 has 651 samples in the second half, all of the 651 samples are included in the non-activity class 6, since the size of the training set is already very small and we do not want to miss important information. The same training sets are used in all of the methods: our method, FET + SVM and CNN + SVM methods.

6.2.3 Second-half testing

We train the first part of the annotated video and test the second part. Figure 7a–c illustrates the temporal segmentation and recognition by the FET + SVM, by the CNN + SVM and by the proposed features with SVM, respectively. The blue graph represents the ground truth, and the red graph represents the prediction. It is observed that the proposed features with SVM performs better temporal segmentation and recognition than the FET + SVM.

We classify 26,602 test frames in the evaluation, and Table 11 shows the correct classification rate (CCR%) for each method. The FET + SVM achieves 71.24%, the CNN + SVM achieves 75.61%, and the proposed features with SVM achieves 88.06% recognition rate. Results show that the proposed features with SVM performs around 17% better than the FET + SVM.

Table 12 shows the precision and recall results, for each activity class, obtained using each method. It is observed



(a)

(b)

(c)

Fig. 7 Temporal segmentation and recognition of activities in the second half. **a** FET + SVM (proposed by Blunsden et al. [12]), **b** CNN features + SVM and **c** the proposed features with SVM

Table 11 CCR% of the proposed features + SVM, CNN features + SVM and FET + SVM in the second half (total frames: 26602)

Methods	FET + SVM [12]	CNN + SVM	Proposed features + SVM
CCR%	71.24%	75.61%	88.06%

Table 13 Confusion matrix for FET + SVM [12] in the second half

	1	2	3	4	5	6	7	8
1	0.79	0.02	0	0	0	0.11	0.05	0.03
2	0.08	0.87	0	0	0.02	0	0.03	0
3	0.21	0.28	0.51	0	0	0	0	0
4	0.03	0	0	0.53	0	0.3	0	0.14
5	0.08	0.11	0.01	0.08	0.33	0.02	0.18	0.19
6	0.05	0	0	0.01	0.01	0.89	0.01	0.03
7	0.08	0.1	0	0	0.04	0.14	0.61	0.03
8	0.27	0.04	0	0.01	0	0.02	0.03	0.63

Table 14 Confusion matrix for CNN + SVM in the second half

	1	2	3	4	5	6	7	8
1	0.65	0.03	0.03	0	0.02	0.04	0.14	0.09
2	0	0.86	0	0	0.07	0	0.06	0
3	0.33	0.31	0.35	0	0	0	0	0
4	0	0	0	0.26	0.07	0.46	0	0.2
5	0.06	0	0	0	0.70	0.03	0	0.21
6	0.05	0	0	0	0	0.86	0.07	0.01
7	0.01	0.01	0	0	0.05	0.01	0.90	0.02
8	0	0.03	0	0	0.03	0	0.10	0.83

Table 15 Confusion matrix for the proposed features with SVM in the second half

	1	2	3	4	5	6	7	8
1	0.85	0.01	0.01	0.01	0.02	0.03	0.06	0.01
2	0.03	0.91	0	0	0.05	0	0.01	0
3	0.13	0	0.77	0	0.03	0	0	0.07
4	0.02	0	0	0.80	0	0.11	0.03	0.04
5	0.02	0	0	0.02	0.59	0.03	0.06	0.28
6	0.03	0	0	0	0	0.97	0	0
7	0.01	0.01	0	0	0	0.03	0.93	0.02
8	0.03	0	0	0	0	0.04	0.03	0.90

that the proposed features with SVM has better precision and recall than the other methods for almost each activity class. This means that our method can handle activity switches and provide sufficient discrimination better than the other methods.

Tables 13, 14 and 15 illustrate the confusion matrix for eight different activities by FET + SVM, CNN features + SVM and the proposed features + SVM, respectively. The number at row and column is the proportion of row class which is classified as column class at frame level. It is observed that the FET + SVM method cannot classify the activity classes 3, 4 and 5 well and confuses them with other activities. On the other hand, our method is significantly better than the other methods in discrimination and classification. In our method, errors generally occur when the activity switches.

6.2.4 First-half testing

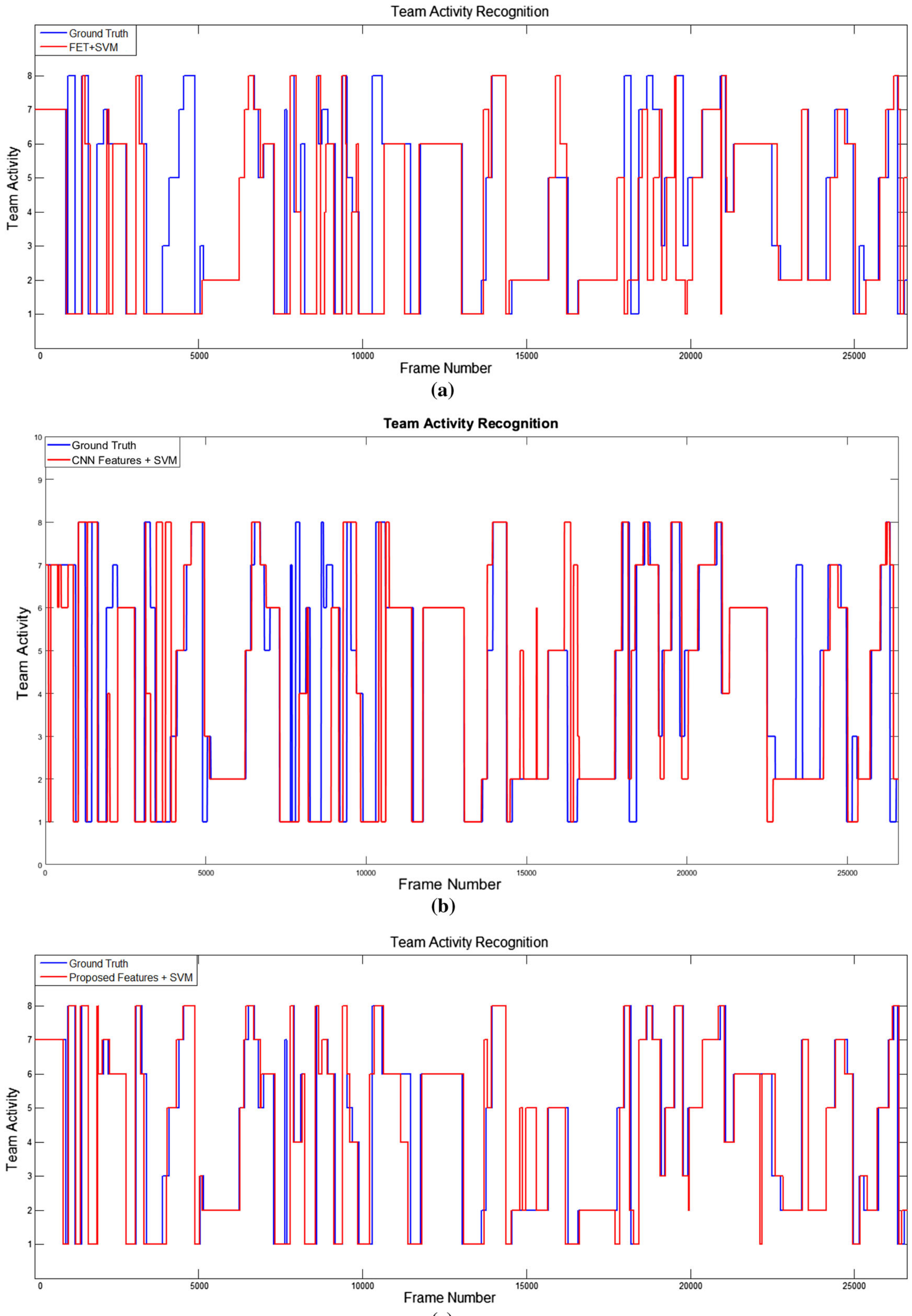
We train the second part of the video and test the first part. Figure 8a–c shows the temporal segmentation and recognition by the FET + SVM, by the CNN + SVM and by the proposed features + SVM, respectively. It can be observed that the proposed method has better temporal segmentation and recognition than the other methods in the first half of the dataset as well.

We classify 26,599 test frames in the first half of the video, and the CCR% for each method is given in Table 16. The FET + SVM achieves 72.69%, the CNN + SVM achieves 74.21%, and the proposed features + SVM achieves 84.05% recognition rate. The proposed method achieves around 12% better performance than the FET + SVM, and around 11% better performance than the CNN + SVM.

The precision and recall results, for each activity class, are shown in Table 17. In general, it is observed that the proposed features with SVM has better results than the other methods.

Table 12 Precision (P%) and recall (R%) of the proposed features with SVM, CNN features + SVM and FET + SVM for each activity in the second half

Act. No.	# of frames	FET + SVM [12]		CNN + SVM		Prop. features + SVM	
		(P%)	(R%)	(P%)	(R%)	(P%)	(R%)
1.	6257	70.90	78.92	86.05	64.60	91.49	84.86
2.	1776	58.79	87.16	76.07	86.43	94.11	90.93
3.	445	90.51	51.46	43.49	35.28	83.82	76.85
4.	1761	82.49	52.98	98.04	25.55	94.39	79.27
5.	1809	67.52	33.44	62.35	69.65	82.18	59.15
6.	6264	72.77	89.29	82.57	86.37	89.02	96.95
7.	5174	79.01	60.90	72.35	90.07	88.16	92.96
8.	3116	64.07	62.71	64.13	83.41	77.70	90.15



(a)

(b)

(c)

Fig. 8 Temporal segmentation and recognition of activities in the first half. **a** FET + SVM (proposed by Blunsden et al. [12]), **b** CNN features + SVM and **c** the proposed features with SVM

Table 16 CCR% of the proposed features + SVM, CNN features + SVM and FET+SVM in the first half (total frames: 26599)

Methods	FET + SVM [12]	CNN + SVM	Proposed features + SVM
CCR%	72.69%	74.21%	84.05%

Tables 18, 19 and 20 show the confusion matrix obtained by FET + SVM, CNN features + SVM and the proposed features + SVM, respectively. It is observed that the FET + SVM method cannot discriminate the activity class 3 and usually confuses with the activity class 1. The FET + SVM also cannot classify the activity classes 5 and 8 well and confuses them with other activities. The proposed features + SVM can discriminate all of the activities and performs better classification than the FET + SVM and CNN + SVM.

6.2.5 Overall performances

We also present overall performances of the FET + SVM and the proposed features + SVM methods. Overall, we classify 53,201 test frames, i.e., 26,602 (second half) + 26,599 (first half) = 53,201. The overall CCR% for each method is shown in Table 21. Our method achieves 86.05%, while FET + SVM achieves 71.97%. The proposed method performs approximately 14% better than the FET + SVM.

The overall precision and recall results for each activity class are also shown in Table 22. The overall confusion matrix for FET + SVM and our method is given in Tables 23 and 24, respectively. The FET + SVM cannot discriminate activity class 3 (i.e., offense fast break) and confuses with the activity class 1 (i.e., slowly going into offense). The FET + SVM method also does not perform well for activity classes 4, 5 and 8. On the other hand, the proposed features with SVM can discriminate all of the activities and achieves better results than the FET + SVM. Results show that the proposed method is effective in temporal segmentation and recognition of activities. Figure 9a–d shows sample frames with the automatically recognized activities by the proposed features with SVM.

Table 18 Confusion matrix for FET + SVM [12] in the first half

	1	2	3	4	5	6	7	8
1	0.87	0.06	0	0	0.01	0.03	0.02	0.01
2	0.03	0.94	0	0	0.01	0	0.02	0
3	0.49	0.28	0.04	0	0	0.16	0.03	0
4	0.04	0	0	0.71	0.03	0.1	0	0.12
5	0.16	0.12	0	0.01	0.50	0.08	0.08	0.05
6	0.17	0	0	0	0	0.82	0	0.01
7	0.11	0.02	0	0.01	0.10	0.05	0.71	0
8	0.33	0.15	0	0	0	0.06	0.02	0.44

Table 19 Confusion matrix for CNN features with SVM in the first half

	1	2	3	4	5	6	7	8
1	0.66	0.05	0.02	0	0.03	0.01	0.06	0.17
2	0.01	0.94	0.01	0	0.04	0	0	0
3	0.51	0.27	0.11	0	0.03	0	0	0.08
4	0.11	0	0	0.85	0	0	0	0.04
5	0	0.09	0	0	0.64	0.04	0.12	0.11
6	0.10	0	0	0.02	0	0.86	0	0.02
7	0.14	0.06	0	0	0.03	0.11	0.62	0.06
8	0.15	0.01	0	0.05	0	0	0.03	0.75

Table 20 Confusion matrix for the proposed features with SVM in the first half

	1	2	3	4	5	6	7	8
1	0.89	0.04	0	0	0	0.03	0	0.04
2	0.04	0.83	0.03	0	0.09	0	0.01	0
3	0.16	0.05	0.65	0	0.05	0.09	0	0
4	0.08	0	0	0.92	0	0	0	0
5	0.05	0.01	0.01	0.03	0.75	0.06	0.06	0.03
6	0.07	0	0	0.05	0	0.85	0.01	0.02
7	0.04	0.01	0.01	0	0.01	0.03	0.83	0.07
8	0.01	0.02	0.01	0.02	0	0.03	0.01	0.90

Table 21 CCR% of the proposed features + SVM and FET + SVM in overall (total frames: 53201)

Methods	FET + SVM [12]	Proposed features + SVM
CCR%	71.97%	86.05%

Table 17 Precision (P%) and recall (R%) of the proposed features with SVM, CNN features + SVM and FET + SVM for each activity in the first half

Act. No	# of frames	FET + SVM [12]		CNN + SVM		Prop. features + SVM	
		(P%)	(R%)	(P%)	(R%)	(P%)	(R%)
1.	4840	55.17	87.23	60.76	66.41	79.93	88.97
2.	5282	77.77	94.04	83.37	94.36	93.86	83.38
3.	957	100.0	3.87	39.55	11.08	70.06	65.52
4.	651	87.08	71.43	67.03	84.64	58.65	92.17
5.	3093	75.33	50.34	81.39	64.05	79.63	75.56
6.	5131	80.89	81.41	88.61	85.54	86.99	84.95
7.	3562	82.92	71.67	74.69	61.62	89.06	82.76
8.	3083	80.93	44.05	59.26	75.25	82.96	90.04

Table 22 Precision and recall of the proposed features with SVM and FET + SVM for each activity in overall

Activity number	# of frames	FET + SVM [12]		Proposed features + SVM	
		Precision (P%)	Recall (R%)	Precision (P%)	Recall (R%)
1.	11097	62.66	82.54	85.93	86.65
2.	7058	72.23	92.31	93.93	85.28
3.	1402	91.72	18.97	74.37	69.12
4.	2412	83.96	57.96	79.78	82.75
5.	4902	72.97	44.10	80.41	69.50
6.	11395	76.03	85.74	88.16	91.55
7.	8736	80.71	65.29	88.50	88.80
8.	6199	70.05	53.43	80.23	90.10

Table 23 Confusion matrix for FET + SVM[12] in overall

	1	2	3	4	5	6	7	8
1	0.82	0.04	0	0	0	0.08	0.04	0.02
2	0.04	0.92	0	0	0.02	0	0.02	0
3	0.40	0.28	0.19	0	0	0.11	0.02	0
4	0.03	0	0	0.58	0.01	0.25	0	0.13
5	0.13	0.12	0	0.04	0.44	0.06	0.11	0.1
6	0.10	0	0	0	0.01	0.86	0.01	0.02
7	0.1	0.06	0	0	0.07	0.10	0.65	0.02
8	0.3	0.1	0	0.01	0	0.04	0.02	0.53

Table 24 Confusion matrix for proposed features with SVM in overall

	1	2	3	4	5	6	7	8
1	0.87	0.02	0.01	0	0.01	0.03	0.03	0.03
2	0.03	0.85	0.02	0	0.09	0	0.01	0
3	0.15	0.03	0.69	0	0.05	0.06	0	0.02
4	0.04	0	0	0.83	0	0.08	0.02	0.03
5	0.04	0.01	0.01	0.02	0.69	0.05	0.06	0.12
6	0.05	0	0	0.02	0	0.91	0.01	0.01
7	0.02	0.01	0	0	0.01	0.03	0.89	0.04
8	0.02	0.01	0	0.01	0	0.04	0.02	0.90

6.2.6 The effect of window size

We study the effect of changing window size in the classification performance (CCR%). Figure 10a, b shows the performance of each method with changing window size for the second half and the first half. The window size ranges from 3 to 551 in our evaluation.

From figures, it is observed that the proposed method performs better than the FET + SVM method and CNN + SVM method at each window size. Table 25 shows the optimal window sizes for each method, in the field hockey dataset, for the second half and the first half.

6.2.7 The effect of motion-information images

We also study the influence of motion-information images in the field hockey dataset. For the second half of the dataset, Figure 11a, b shows the result obtained by using only frame differencing (one motion-information image) and using only optical flow (four motion-information images), respectively. Figure 7c illustrates the result using the combination (five motion-information images).

For the first half of the dataset, Fig. 12a, b shows the result by using only frame differencing and using only optical flow, respectively. Figure 8c shows the result using the combination.

Table 26 also shows the CCR% obtained by only frame differencing, only optical flow and their combination for each stage in the field hockey dataset. In overall, only frame differencing achieves 53.77%, only optical flow achieves 83.55%, and the combination achieves 86.05%. Results indicate that using the combination improves the performance.

6.2.8 The effect of SVM kernel function

We also present the effect of different SVM kernel functions in field hockey dataset. We experiment three different kernel functions: linear kernel function, polynomial kernel function and Gaussian radial basis kernel function. Table 27 shows the performance of the proposed method with respect to these kernel functions. It is observed that the best performance of the proposed method is achieved with the Gaussian radial basis kernel function. Parameters of the kernel functions are selected using the fivefold cross-validation and grid search. For the second-half testing case, we apply the cross-validation and grid search to the first half (training part) of the video. For the first-half testing case, we apply the cross-validation and grid search to the second half (training part) of the video. In linear SVM, we have a single parameter that is for the soft margin cost function (C). The optimal cost function parameter (C) is 0.25 and 0.2 for the second-half testing and first-half testing, respectively. In polynomial SVM, we

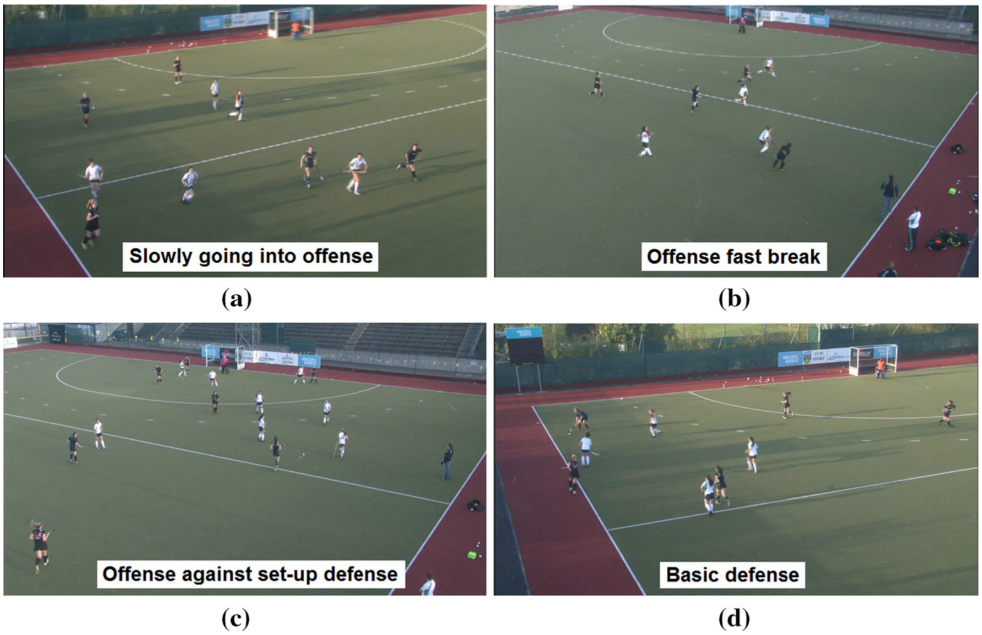


Fig. 9 Activity of the black-uniform-wearing team is automatically recognized by the proposed features with SVM

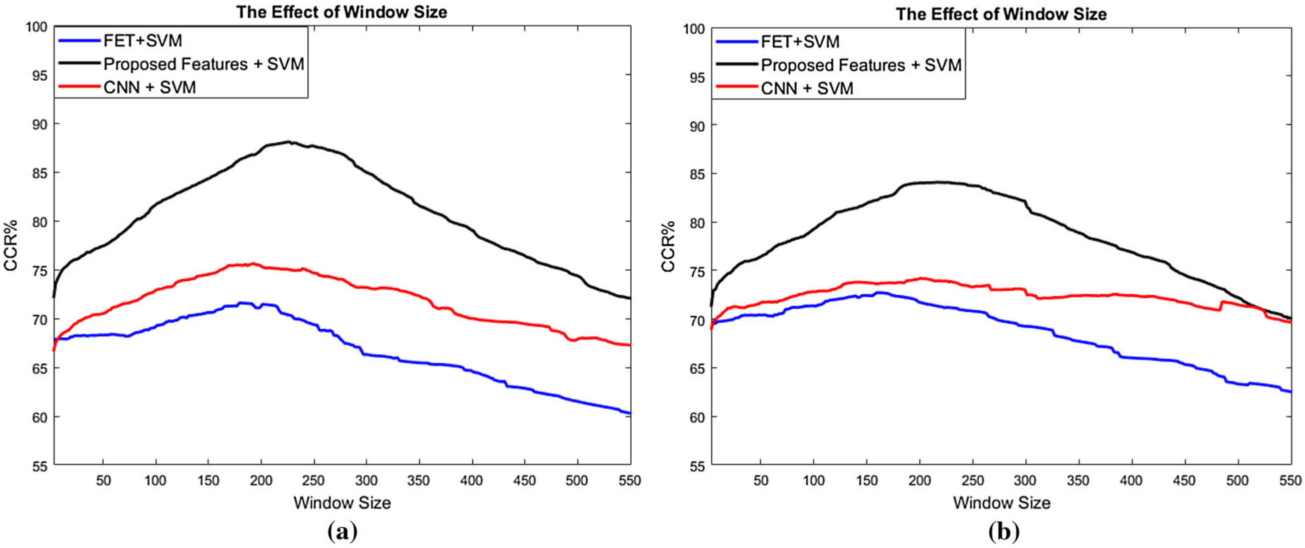
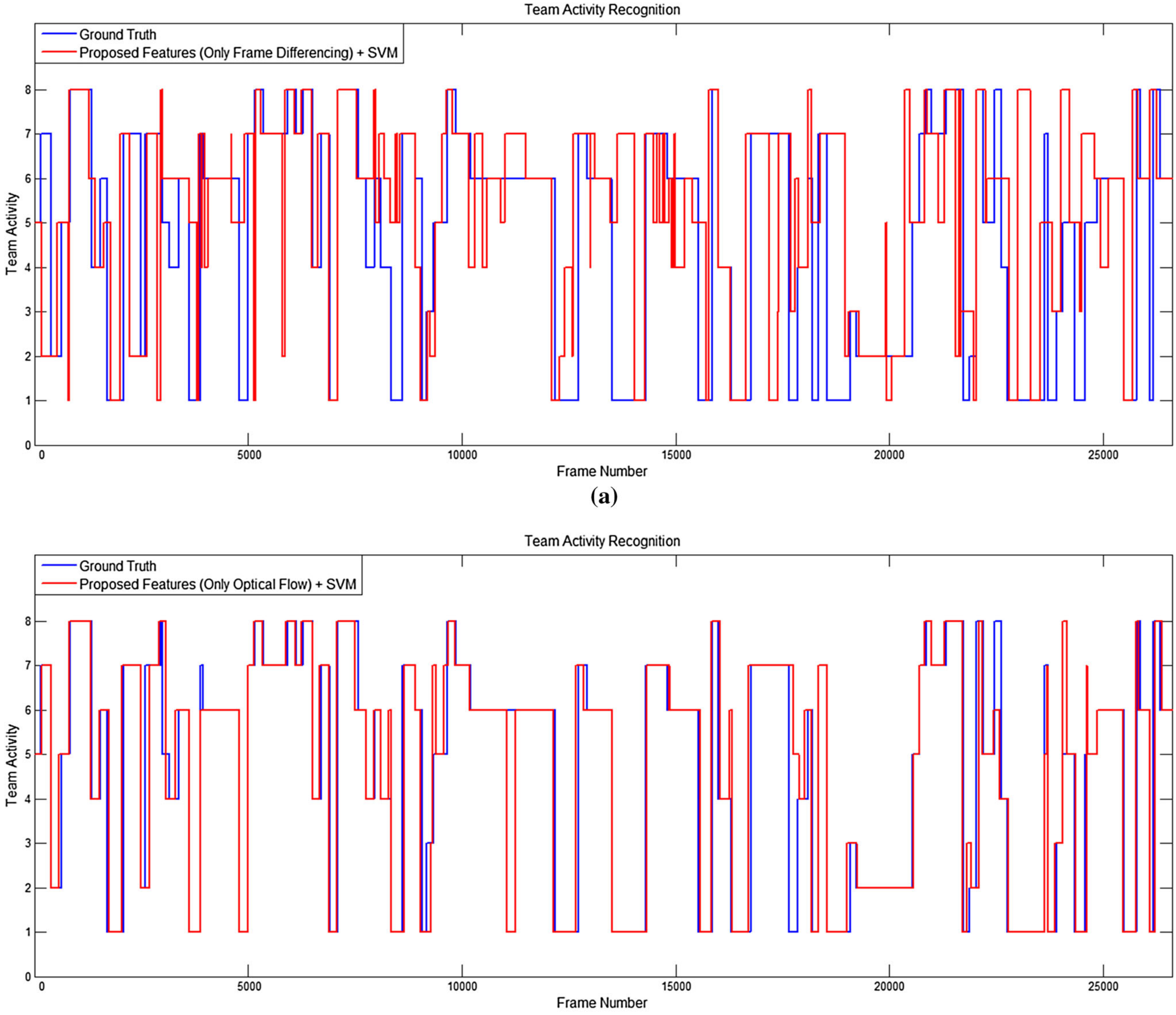


Fig. 10 Classification performances (CCR%) with differing window size for **a** the second half and **b** the first half

Table 25 Optimal window size for each stage of the methods in field hockey dataset

Stages	FET + SVM [12]	CNN + SVM	Proposed features + SVM
Second half	179	193	225
First half	163	201	217



(a)

(b)

Fig. 11 Temporal segmentation and recognition in the second half **a** using only frame differencing and **b** using only optical flow

have two parameters: the cost function parameter (C) and the order of polynomial function (P). The optimal parameter values appear to be $C = 0.02$ and $P = 4$ both for the second-half testing and for the first-half testing, respectively. Finally, in Gaussian radial basis kernel function there are two parameters: the cost function parameter (C) and the Gaussian scale factor (S). The optimal parameter values appear to be $C = 10$ and $S = 2$ both for the second-half testing and for the first-half testing, respectively.

6.2.9 Computational efficiency

Table 28 shows the computational time required for each stage of the methods. Results are obtained using MATLAB 7 on a Windows 7 operating system with Intel Core i3-870, 2.93 GHz and 8 MB RAM. In feature extraction, the FET + SVM method is more efficient than the proposed method and the CNN + SVM. On the other hand, the proposed method (combination) has significantly better classification accuracy in comparison with the other methods.

7 Conclusion

We have presented an approach for temporal segmentation and recognition of team activities in sports based on a new activity feature extraction strategy. Given the positions of team players from a plan view of the playing field at any given time, we solve a particular Poisson equation to generate a position distribution for the team. Computing the position

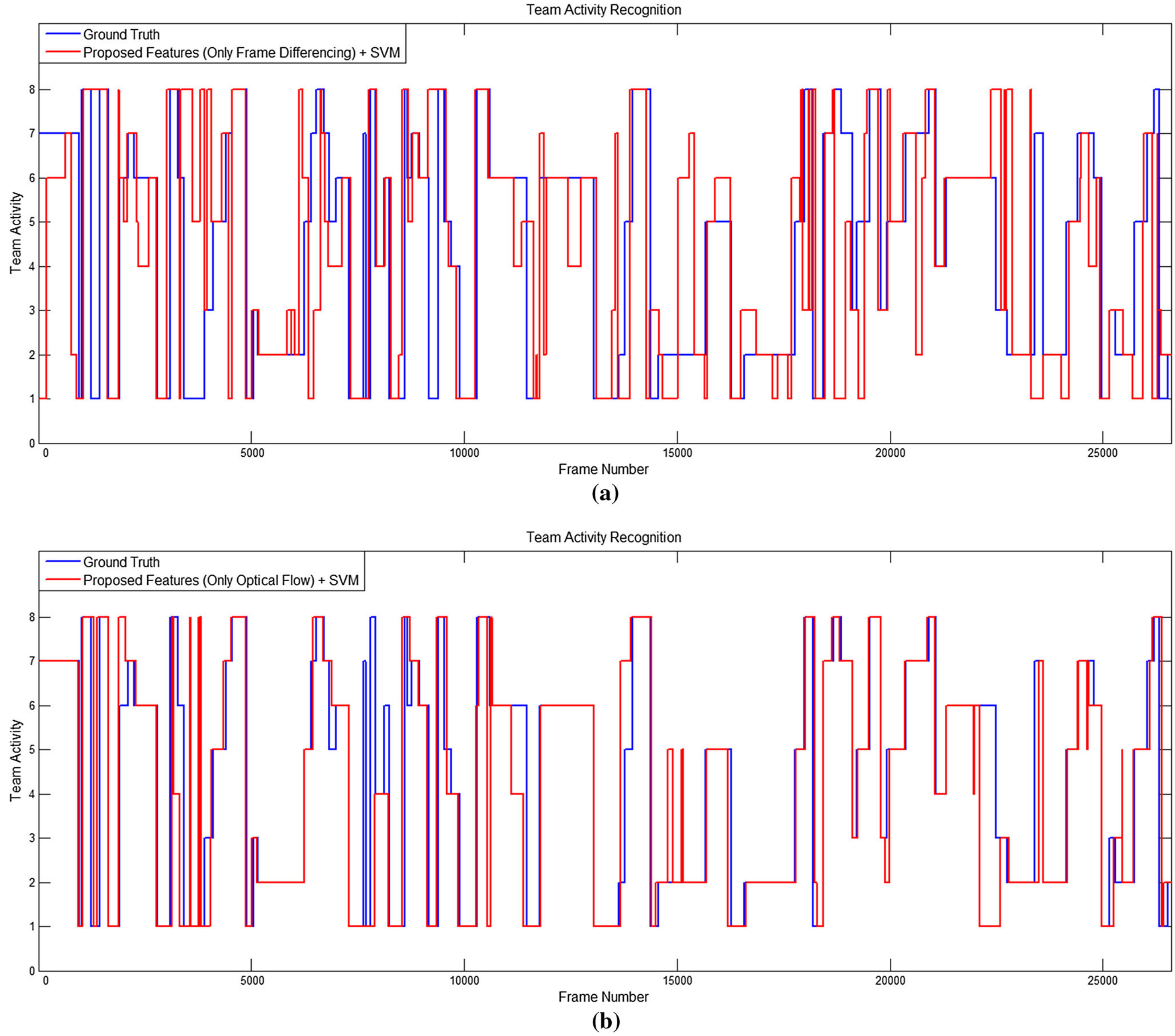


Fig. 12 Temporal segmentation and recognition in the first half **a** using only frame differencing and **b** using only optical flow

Table 26 Effect of motion-information images in the field hockey dataset

Motion-information images	Only frame differencing	Only optical flow	Combination
Second-half performance (CCR%)	53.09	86.73	88.06
First-half performance (CCR%)	54.46	80.37	84.05
Overall performance (CCR%)	53.77	83.55	86.05

Table 27 Evaluation of SVM classifier with respect to different kernels in hockey dataset

SVM kernel function	Second half (CCR%)	First half (CCR%)
Proposed features + SVM (Gaussian radial basis kernel)	88.06	84.05
Proposed features + SVM (polynomial kernel)	84.85	81.38
Proposed features + SVM (linear kernel)	77.04	74.02

Table 28 Computation time for each stage of the methods in field hockey dataset

Stages	FET+SVM [12]	CNN+SVM	Proposed features+SVM (combination)	Proposed features + SVM (frame differencing)	Proposed features + SVM (optical flow)
Feature extraction in whole video	11.05 s	67,654.61 s	65,428.04 s	10,512.47 s	54,932.51 s
Training activities in the first half	65.22 s	55.93 s	53.86 s	67.34 s	60.12 s
Classifying activities in the second half	0.59 s	0.64 s	0.61 s	0.64 s	0.63 s
Training activities in the second half	61.27 s	66.72 s	63.06 s	62.38 s	63.56 s
Classifying activities in the first half	0.61 s	0.66 s	0.63 s	0.59 s	0.64 s

distribution for each frame provides a sequence of distributions, which we process to extract motion features at each frame. Then the motion features are used to classify team activities. Our method is evaluated on two different datasets, i.e., the European handball and the field hockey datasets. Results show that the proposed approach is effective and performs better than the method (FET+SVM) that extracts features from the explicitly defined trajectories, and better than the method (CNN+SVM) that uses a predefined convolutional neural network for feature extraction.

Acknowledgements This work is supported by Science Foundation Ireland under Grant 07/CE/I114. The authors would like to acknowledge Disney Research Pittsburgh for their help in constructing the camera network around the field hockey playground in Ireland. We also would like to thank Statsports for supplying us GPS sensors, and finally, we thank Irish Hockey Association for their collaboration to collect the field hockey dataset.

References

1. Aggarwal, J.K., Ryoo, M.S.: Human activity analysis: a review. ACM Comput. Surv. **43**(3), 16 (2011)
2. Kong, Y., Zhang, X., Wei, Q., Hu, W., Jia, Y.: Group action recognition in soccer videos. In: Proceedings of ICPR, pp. 1–4 (2008)
3. Kong, Y., Hu, W., Zhang, X., Wang, H., Jia, Y.: Learning group activity in soccer videos from local motion. In: Proceedings of ACCV, pp. 103–112 (2009)
4. Wei, Q., Zhang, X., Kong, Y., Hu, W., Ling, H.: Group action recognition using space time interest points. Int. Symp. Adv. Vis. Comput. **2**, 757–766 (2009)
5. Li, R., Chellappa, R., Zhou, S.K.: Learning multi-modal densities on discriminative temporal interaction manifold for group activity recognition. In: CVPR, pp. 2450–2457 (2009)
6. Swears, E., Hoogs, A.: Learning and recognizing complex multi-agent activities with applications to american football plays. In: IEEE Workshop on Applications of Computer Vision (WACV), pp. 409–416 (2012)
7. Ibrahim, M.S., Deng, Z., Muralidharan, S., Vahdat, A., Mori, G.: A hierarchical deep temporal model for group activity recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1971–1980 (2016)
8. Shu, T., Todorovic, S., Zhu, S.-C.: CERN: Confidence-energy recurrent network for group activity recognition. In: IEEE Conference on Computer Vision Pattern Recognition (To appear in proc. of CVPR 2017) (2017)
9. Fani, M., Neher, H., Clausi, D.A., Wong, A., Zelek, J.: Hockey action recognition via integrated stacked hourglass network. In: IEEE International Workshop on Computer Vision in Sports at CVPR (2017)
10. Tsunoda, T., Komori, Y., Matsugu, M., Harada, T.: Football action recognition via hierarchical LSTM. In: IEEE International Workshop on Computer Vision in Sports at CVPR (2017)
11. Intille, S.S., Bobick, A.F.: Recognizing planned, multiperson action. CVIU **81**(3), 414–445 (2001)
12. Blunsden, S., Fisher, R.B., Andrade, E.L.: Recognition of coordinated multi agent activities, the individual vs the group. In: ECCV Workshop on Computer Vision Based Analysis in Sport Environments (CVBASE), pp. 61–70 (2006)
13. Perse, M., Kristan, M., Kovacic, S., Vuckovic, G., Pers, J.: A trajectory-based analysis of coordinated team activity in a basketball game. CVIU **113**(5), 612–621 (2009)
14. Perse, M., Kristan, M., Pers, J., Music, G., Vuckovic, G., Kovacic, S.: Analysis of multi-agent activity using petri nets. Pattern Recognit. **43**(4), 1491–1501 (2010)
15. Hervieu, A., Bouthemy, P., Cadre, J.P.L.: Trajectory-based handball video understanding. In: International Conference on Image and Video Retrieval, vol. 43, pp. 1–8 (2009)
16. Dao, M.S., Masui, K., Babaguchi, N.: Event tactic analysis in sports video using spatio-temporal pattern. In: Proceedings of ICIP, pp. 1497–1500 (2010)
17. Li, R., Chellappa, R.: Recognizing offensive strategies from football videos. In: Proceedings of ICIP, pp. 4585–4588 (2010)
18. Varadarajan, J., Atmosukarto, I., Ahuja, S., Ghanem, B., Ahuja, N.: A topic model approach to represent and classify american football plays. In: British Machine Vision Conference (BMVC) (2013)
19. Montoliu, R., Martin-Felez, R., Torres-Sospedra, J., Rodrguez-Prez, S.: ATM-based analysis and recognition of handball team activities. Neurocomputing **150**, 189–199 (2015)
20. CVBASE'06 dataset, in workshop on computer vision based analysis in sport environments. http://vision.fe.uni-lj.si/cvbase06/downloads.html (2006)
21. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Neural Information Processing Systems Conference, pp. 1097–1105 (2012)
22. Direkoglu, C., O'Connor, N.E.: Team behavior analysis in sports using the Poisson equation. In: IEEE International Conference on Image Processing (ICIP), pp. 2693–2696 (2012)
23. Direkoglu, C., O'Connor, N.E.: Team activity recognition in sports. In: European Conference on Computer Vision (ECCV), vol. 7578, pp. 69–83 (2012)
24. Braun, M.: Differential Equations and Their Applications. Springer, Berlin (1993)
25. Simchony, T., Chellappa, R., Shao, M.: Direct analytical methods for solving Poisson equations in computer vision problems. T-PAMI **12**(5), 435–446 (1990)

26. Horn, B.K.P., Schunck, B.G.: Determining optical flow. Artif. Intell. **17**, 185–203 (1981)
27. Kim, K., Grundmann, M., Shamir, A., Matthews, I., Hodgins, J., Essa, I.: Motion fields to predict play evolution in dynamic sport scenes. In: CVPR, pp. 840–847 (2010)
28. Efros, A.A., Berg, A.C., Berg, E.C., Mori, G., Malik, J.: Recognizing action at a distance. In: Proceedings of ICCV, pp. 726–733 (2003)
29. Janez, M.K., Kovacic, S.: Multiple interacting targets tracking with application to team sports. In: International Symposium on Image and Signal Processing and Analysis, pp. 322–327 (2005)
30. Ward, J.A., Lukowicz, P., Gellersen, H.W.: Performance metrics for activity recognition. ACM Trans. Intell. Syst. Technol. **2**(1), 6 (2011)
31. GPSports Systems Limited, Australia. http://gpsports.com/
32. Statsports Company, Northern Ireland. http://www.statsports.ie/
33. MacLeod, H., Morris, J., Nevill, A., Sunderland, C.: The validity of a non-differential global positioning system for assessing player movement patterns in field hockey. J. Sport Sci. **27**(2), 121–128 (2009)
34. Roweis, S., Saul, L.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**(5500), 2323–2326 (2000)
35. Chen, J., Liu, Y.: Locally linear embedding: a survey. Artif. Intell. Rev. **36**(1), 29–48 (2011)
36. Onderwater, M.: Outlier Detection Matlab Toolbox. http://www.martijn-onderwater.nl/bmi-thesis/

Cem Direkoğlu has expertise in image processing and computer vision. Currently, he is an Assistant Professor at Electrical and Electronics Engineering program of Middle East Technical University Northern Cyprus Campus (METU-NCC). Before joining to METU-NCC, he was a post-doctoral researcher at INSIGHT Centre for Data Analytics (School of Electronic Engineering) in Dublin City University (DCU). Before DCU, he was also a post-doctoral researcher at Graphics, Vision and Visualization group (School of Computer Science and Statistics) in Trinity College Dublin. He has his Ph.D. degree in computer vision from University of Southampton, and he was a member of Information: Signals, Images and Systems research group in School of Electronics and Computer Science. His MSc and BSc degrees are in Electrical and Electronics Engineering from Eastern Mediterranean University. He has been involved in diverse computer vision-based research projects and has many publications in international journals and conferences, including the high-impact ones as a lead author such as in IJCV, PR, MVA, PRL, ECCV, BMVC and ICIP. He is a reviewer for prestigious journals in his field such as Computer Vision and Image Understanding, Pattern Recognition, Image and Vision Computing, and IEEE Transactions on Circuits and Systems for Video Technology.

Noel E. O'Connor obtained his PhD from Dublin City University in 1998 and is currently a Professor in the School of Electronic Engineering at DCU and a Principal Investigator (PI) and Director in INSIGHT Centre for Data Analytics. Prof. O'Connor is also Academic Director of DCU's Research & Enterprise Hub on Information Technology and the Digital Society. Since 1999 Prof. O'Connor has published over 180 peer-reviewed publications, made 11 standards submissions, filed 6 patents and spun off a campus company, AliopeLtd (with Prof. Alan Smeaton). He has graduated 21 PhD students and 5 Masters students. He is an Area Editor for Signal Processing: Image Communication (Elsevier) and an Associate Editor for the Journal of Image and Video Processing (Springer). He has edited 6 journal special issues, including Signal Processing: Image Communication, Multimedia Tools and Applications, the Journal of Web Semantics and the Journal of Embedded Systems. He has acted as Programme Co-Chair for 4 international conferences, CIVR 2009, SAMT 2006, WIAMIS 2007 and CIVR 2004. He was General Chair for WIAMIS 2012. He was Content Track Chair for ICME 2008 and Tutorial/Workshop Chair for SAMT 2009, CBMI 2009 and SAMT 2008. He was a member of the Academic Steering Committee of the ACM Multimedia Grand Challenge series, run in conjunction with ACM Multimedia. He has acted as an expert reviewer for the EU FP6PROFI project (FET Open STREP) and the EU FP7 VITALAS and PrestoPrime Integrated Projects, as well as acting as an expert evaluator for EU FP6 Calls 2 and 4 and FP7 Call 3 & 5. He was awarded the DCU President's Research Award for Science and Engineering in 2010. Also in 2010, he was awarded Enterprise Ireland's National Commercialization Award for ICT. He is a member of the IEEE, Engineers Ireland and the IET.