Computers and Electrical Engineering 99 (2022) 107779

Contents lists available at ScienceDirect

Computers and Electrical Engineering

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

SPNet: A deep network for broadcast sports video highlight generation☆

Abdullah Aman Khan, Jie Shao *

Center for Future Media, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
Sichuan Artificial Intelligence Research Institute, Yibin 644000, China

ARTICLE INFO

Keywords:
Sports video
Video analysis
Video summarization
Highlight extraction

ABSTRACT

Professionally broadcasted sports videos usually have long durations but contain only a few exciting events. In general, professional bodies and amateur content creators spend thousands of man-hours to manually crop the exciting video segments from these long-duration videos and generate handcrafted highlights. Sports enthusiasts keep them updated with the latest happening based on such highlights. There exists a need for a method that accurately and automatically recognizes the exciting activities in a sports game. To address this issue, we present a deep learning-based network SPNet that recognizes exciting sports activities by exploiting high-level visual feature sequences and automatically generates highlights. The proposed SPNet utilizes the strength of 3D convolution networks and Inception blocks for accurate activity recognition. We divide the sports video excitement into views, actions, and situations. Moreover, we provide 156 new annotations for about twenty-three thousand videos of the SP-2 dataset. Extensive experiments are conducted using two datasets SP-2 and C-sports, and the results demonstrate the superiority of the proposed SPNet. Our proposed method achieves the highest performance for views, action, and situation activities with an average accuracy of 76% on the SP-2 dataset and 82% on the C-sports dataset.

1. Introduction

The last decade has witnessed a dramatic increase in the number of videos uploaded to the internet, especially, on video-sharing platforms where these videos can exist for a long time. Other than users' videos, such videos include TV programs, drama, sports, talk shows, etc. A large portion of these videos belongs to the category of sports. Usually, a video is accompanied by a user-defined tag(s)/keyword(s), but normally video data is still unstructured and such tags cannot explain what is exactly going on in the video. However, efforts are made to understand the video content [1,2].

Professionally broadcasted sports videos usually have long durations but contain only a few exciting moments [3]. Different sports have different rules and the sports game may continue from one hour to a couple of days. Sports highlights can be considered as a video-based summary that contains only exciting or important events. Highlights deliver the whole excitement of the game in a much shorter period and such highlights are the main source for the sports enthusiasts to keep themselves up to date in a busy lifestyle [4]. Traditionally, such highlights are manually cropped based on human effort alone. Blog writers and other content

☆ This paper is for regular issues of CAEE. Reviews were processed and recommended for publication by Co-Editor in Chief Prof Huimin Lu.

* Corresponding author at: Center for Future Media, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China.
E-mail address: shaojie@uestc.edu.cn (J. Shao).

https://doi.org/10.1016/j.compeleceng.2022.107779
Received 5 May 2021; Received in revised form 17 January 2022; Accepted 30 January 2022
Available online 16 February 2022

creators spend thousands of man-hours to produce such highlights and it is quite challenging to compile a unique set of highlights for different sports videos. There exists a need for automatic sports video summarization methods.

The recent explosion of Artificial Intelligence (AI), especially deep learning, has granted opportunities for advanced visual information processing [5–7]. Deep learning and AI-based techniques have already been quite successfully incorporated in various real-life applications [8,9]. Such techniques can be incorporated to build automatic tools for generating broadcast sports highlights. However, there are a lot of challenges associated with realizing such tools and techniques. For example, different people may have different opinions over an exciting event. For a soccer game, only a goal can be an exciting moment, whereas for some spectators, a missed goal can be an exciting event. This difference in opinion can be resolved by detecting all the exciting events and presenting users with a highlights summary based on users' preferences. The second challenge is the diverse nature of sports, as every kind of sport is different from others having different rules and playfield scenarios [10]. The third challenge is the availability of training data, especially for deep learning-based methods. The fourth and most challenging part is the nature of broadcast sports videos. Unlike user-generated videos, broadcast sports videos are recorded through multiple cameras having different views. Besides, the cameras are switched rapidly as per the instructions of the sports director. These properties of broadcast sports videos have not been adequately acknowledged by previous research.

Automatic sports video summarization is a challenging problem. The previous studies rely on tracking the players' activities [11–13], monitoring the crowd noise [4,14], clustering similar frames of the videos, analyzing player actions, and extracting useful information from the caption regions or user-generated comments [3]. Clustering-based approaches use low-level features to reduce visual redundancy, while other methods focus on extracting semantic features. The unstructured nature of sports video makes video summarization a challenging task.

The previous approaches have high chances of missing exciting events. For example, the caption-based approaches [15,16] depend on the text-based information provided by the broadcaster. Such an approach can exactly track an important event, e.g., goal in a football match. However, this approach cannot be employed to detect goal misses, corner shots, outs, etc. which might seem interesting to a spectator. Tracking players [11–13] in a game has its benefits, but due to the versatile nature of sports games, it is impractical and challenging to track players in different sports categories. Moreover, the players are usually idle for most of the time in sports games such as cricket, baseball, etc. For this reason, this approach has higher chances of missing exciting events. Audio cues themselves (cheering of the crowd) are an important feature for detecting an exciting moment [4]. However, audio cues do not indicate the nature of the event. Besides, it was noticed on several occasions that the crowd cheers without any reason and may lead to false detection of an exciting event. As mentioned above, the rapid camera view change and movement of cameras pose a challenge to the clustering-based methods.

In this paper, we propose a novel approach to recognize the sports activity and robustly describe "what is going on in a video segment". Such descriptions help highlight generation based on spectators' preferences. The proposed approach has practical significance and can help in extracting highlights from various sports categories. First, unlike previous studies, we separate the broadcast sports video scenes based on views, actions, and situations (details are provided in Section 4.1). Second, we propose a deep learning-based network (SPNet) that collectively recognizes exciting events based on spatiotemporal high-level visual features. The proposed network utilizes 3D-ResNet that can directly extract spatiotemporal information using 3D kernels. Moreover, we utilize the Inception V3 block for collectively recognizing views and situations. The inception block stacks 11 inception modules where each module consists of convolution filters, pooling layers, rectified linear units, and filter concatenation. By using every frame under consideration, feature sequences are constructed and further trained using neural networks. Finally, the exciting event is evaluated based on the proposed prediction algorithm. The contributions of this paper can be summarized as follows:

- We propose a deep learning network (SPNet) that exploits high-level visual feature sequences to accurately describe "what is happening" in a broadcast sports video scene and utilizes this information for generating highlights based on spectators'/users' preferences.
- We add fine-grained annotations to the SP-2 dataset[1] and separate the annotations according to view, action, and situation.
- We perform extensive experiments to validate the behavior of our proposed solution and the results of these experiments indicate the superiority of the proposed approach. Relevant data and codes are publicly available.[2]

The rest of the paper is organized as follows: Section 2 gives an overview of the related work. Section 3 presents the proposed method in detail. In Section 4, we demonstrate the results and their discussion followed by a conclusion in Section 5.

2. Related work

In this section, we shed some light on some of the related studies and contributions. Many researchers devoted their time and resources related to the field of video summarization or video abstraction. Sports video highlights generation can be considered as a subclass of video summarization. Some of the studies focused on generation highlights from sports videos such as [4]. Various studies focused on analyzing only a specific category of sports, e.g., basketball [17], tennis [18], soccer [19] and volleyball [20]. However, other studies focused on event recognition in a sports video [21]. Some of these methods attempted to describe the scene or track the movement of the players and gain information from social networks [11,22].

[1] https://github.com/abdkhanstd/Sports2.
[2] https://github.com/abdkhanstd/SPNet.

Some studies focused on gathering exciting events from sports videos using various approaches. As mentioned before, such studies aim to classify the important events based on gathering data game data from social media streams, analyzing text-based data, analyzing visual information, player tracking, and analyzing audiovisual cues. These approaches include methods based on deep learning and shallow learning.

2.1. Player tracking

Some studies focused on tracking players in a sports game. For example, Zhang et al. [11] proposed a robust player tracking framework for basketball videos. Similarly, Host et al. [12] proposed a method for tracking the active player in handball sports game. The study in [13] proposed a method for tracking basketball players in a multi-view environment.

2.2. Text-based approaches

Some of the researchers attempted to exploit the text information available in a sports video stream. For example, the study in [15] exploited the textual information for a basketball game. Moreover, the study in [16] proposed a robust method for extracting text information from long-duration broadcast sports videos. These studies contribute towards detecting important or exciting events in a sports game.

2.3. Audiovisual cues

A broadcast sports video contains various multimodal information including audio and visual cues. Many researchers exploited the audio and visual cues for detecting important events and generation highlights. For example, Khan et al. [4] focused on generating highlights based on audiovisual cues. They distinguished a regular event from an exciting event based on the crowd cheering and the information displayed by the scorebox. Moreover, the study in [14] attempted to close the gap between high-level human perception and low-level video features based on a 2D convolution network and LSTM (Long Short-Term Memory).

2.4. Activity recognition

Agyeman et al. [19] presented a method for soccer video summarization. They recognized each event as a potential highlight and evaluated it against a truth table to determine the importance. They utilized a 3D-ResNet 34 deep learning pipeline for determining the importance of the event. The study in [21] presented a new dataset along with deep learning-based methods for detecting activities in different sports videos. Rafiq et al. [22] presented a deep convolution-based method for summarization of sports video. Some studies such as [23] focused on generating summaries by exploiting deep action features for amateur user-generated sports videos. However, user-generated videos have a different nature as compared with broadcast sports videos [10].

3. Our method

This section first presents the related background knowledge, followed by further details about SPNet.

3.1. Preliminaries

To provide a better understanding of the proposed SPNet, first, we elaborate the preliminary details as follows:

3.1.1. Problem formulation

Given a training set of training videos V_{tvid} with a video-level label Y where $Y \in U_p$ and T_v represents the total activity categories. The goal of the feature extraction process and dense layers is to generate a set of probabilities, i.e., $P = \{P_A, P_S, P_V\}$ (U_p, P_A, P_S and P_V are explained in Section 3.2.5). Based on these probabilities, we decide whether the video segment should be added to the final set of highlights H_l or not. Suppose, a video segment I_j is an important activity recognized according to U_p. Then, H_l is the set of all the recognized important activities, i.e., $H_l = \{I_1, I_2, I_3, \ldots, I_k\}$, where k is the total number of predicted important activities.

3.1.2. Our approach

The previous methods focus on only extracting low and high-level features from sports videos. However, they have not considered spectators' preferences as different spectators have different interests. For example, for a soccer game, some spectators find only goals as interesting, while for some, penalty kick or corner shots might be the interesting part. This difference of opinion poses a challenge to highlight the generation process. A straightforward solution is to detect all the activities (views, actions, and situations) and then generate highlights according to the predefined user preferences.

A highlight is a video segment of a long-duration sports video that seems interesting to a spectator. A set of video segments that contain important events are referred to as sports highlights. Usually, the spectators are interested in important events. However, some spectators find some events/activities interesting that may not seem interesting to other spectators. This paper divides the long-duration broadcast sports video into smaller segments and attempts to recognize the activity in that particular video segment. If the detected activity in that video segment matches the spectators' criteria, the activity is added to the final set of highlights. In other words, the highlights are based on classification of sports video segments.

Fig. 1. Some selected samples from the SP-2 datasets. The first row shows some selected samples from crickets' bowling side view. The second row shows frame samples for a goal kick for soccer. The third row contains samples from baseball showing a boundary view. Moreover, the last row represents around the goal post situation for ice hockey.

3.1.3. Collective activities

In this paper, we refer to the actions, views, and situations as collective activities. The action refers to the action being performed by a player in the respective game. The determination of actions assists in generating highlights effectively. The second type of activity includes views. It should be noted that a view may be associated with an action/situation or it can be independent. For example, in Fig. 1 the first row represents a bowling action with a side view. However, in the third row, the view shows the playfield while no action is being performed. All other activities that do not contain any type of action are referred to as situations.

We denote the set of all views, actions, and situations with V_s, A_s, and S_s respectively. For example, all the class labels of the views belong to V_s while all the action and situation classes belong to A_s and S_s respectively. Formally, V_s can be represented as $V_s = \{L_{V_1}, L_{V_2}, L_{V_3}, \ldots, L_{V_{70}}\}$ where L_{V_i} represents the class label of views. Similarly, A_s can be represented by $A_s = \{L_{A_1}, L_{A_2}, L_{A_3}, \ldots, L_{A_{50}}\}$ where L_{A_i} represents the labels for action classes. Lastly, S_s can be represented by $S_s = \{L_{S_1}, L_{S_2}, L_{S_3}, \ldots, L_{S_{36}}\}$ where L_{S_i} represents the labels for situation classes. In other words, A_s is the set of all the actions, V_s are all the views and S_s are all the situations in the SP-2 dataset. Furthermore, L_{V_i}, L_{A_i}, and L_{S_i} represent an individual element of the sets V_s, A_s, and S_s respectively. More information about the dataset annotations can be found in Section 4.1.

3.1.4. Building feature sequences

Next, we elaborate on how we build feature tensors/matrices using 2D feature extraction pipelines. Typically, 2D-CNN (Convolutional Neural Network) architectures are intended for single image classification. In this paper, we exploit 2D-CNN architectures to extract features frame by frame and further build spatiotemporal feature sequences.

Traditionally, a single two-dimensional CNN layer contains a convolutional and pooling layer. CNN can go deeper by using multiple levels of convolutional and pooling layers. A neuron with value v_{ij}^{xy} having a position (x, y) for the jth feature map of the ith layer can be expressed as follows:

$$v_{ij}^{xy} = g\left(b_{ij} + \sum_{m}\sum_{p=0}^{P_i-1}\sum_{q=0}^{Q_i-1} w_{ijm}^{pq} v_{(i-1)m}^{(x+p)(y+q)}\right), \tag{1}$$

where m represents the feature map computed by the previous layer $(i-1)$. Moreover, w_{ijm}^{p} represents the positional weights of p for the mth feature map. Additionally, P_i and Q_i represent the height and width of the convolution kernel respectively. The bias of the jth feature map of the ith layer is denoted by b_{ij}.

2D-CNN architectures for video understanding do not interrelate spatiotemporal features. Fig. 2 gives an overview of the general working mechanism of our spatial feature extraction network. Our proposed method extracts feature from each frame, and then, we further build a feature matrix of the extracted features. The feature matrix as shown in Fig. 2 represents the features with a size of $\delta \times \alpha$. Each row represents the features extracted from a single video frame. The number of rows δ corresponds to the number of frames under consideration, and α represents the number of features extracted by the feature extraction network. In Fig. 2, we refer to LSTM, GRU (Gated Recurrent Units) and MLP (Multilayer Perceptron) as learning network.

3.1.5. Spatiotemporal features

3D-CNN deep learning methods learn the 3D spatiotemporal features and these methods have shown considerable performance and are superior for some applications. However, in our experience, 3D networks are notoriously hard to train and require a lot of computational resources as they have a large number of parameters. Most of the 3D-CNN networks are shallower compared with their 2D counterpart. Some researchers explored the idea of deeper 3D-CNN to improve performance [24].

A neuron v_{ij}^{xyz} at position (x, y, z) of the jth feature map of the ith layer can be expressed as:

$$v_{ij}^{xyz} = g\left(\sum_{m}\sum_{p=0}^{P_i-1}\sum_{q=0}^{Q_i-1}\sum_{r=0}^{R_i-1} w_{ijm}^{pqr} v_{(i-1)m}^{(x+p)(y+q)(z+r)} + b_{ij}\right). \tag{2}$$

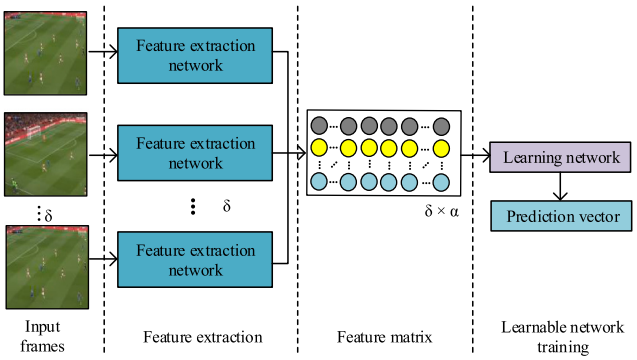

Fig. 2. Overview of the training process of the spatial feature learning branch.

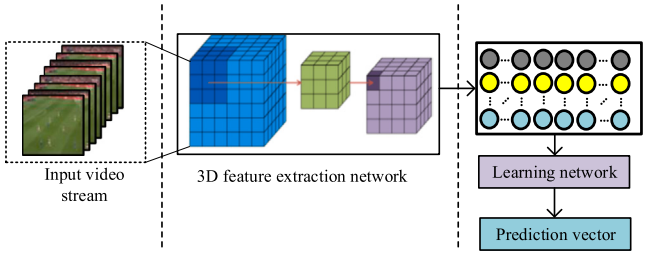

Fig. 3. Overview of the training process of spatiotemporal (3D feature extraction) branch.

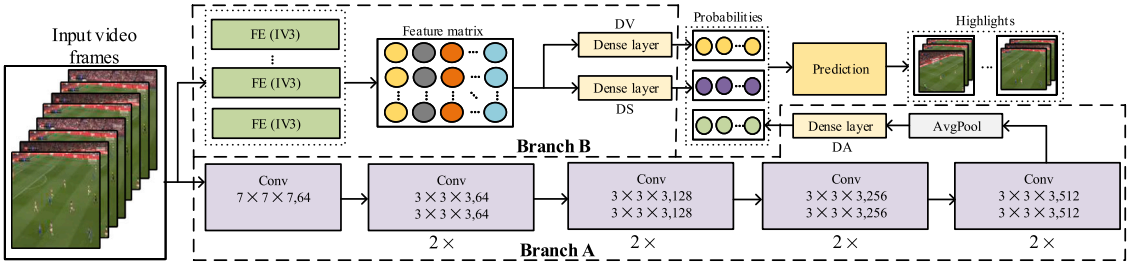

Fig. 4. Overview of the proposed SPNet architecture.

Likewise 2D-CNN, m represents the feature map computed by the previous layer $(i-1)$ associated with the jth feature map. Additionally, P_i, Q_i and R_i represent the height, width, and depth of the convolution kernel respectively. w_{ijm}^{pqr} represents the value at a position (p, q, r) of the mth feature map. Furthermore, the bias of the jth feature map of the ith layer is denoted by b_{ij}.

Fig. 3 gives an overview of the 3D feature extraction network which utilizes 3D convolutions. Such deep learning networks can extract spatiotemporal features across various video frames. In Fig. 3, it can be seen that the 3D feature extraction networks perform operations in a spatiotemporal manner. After extracting the required features, 3D-CNN generally utilizes a learning network to predict based on the extracted features. Similar to Fig. 2, learning network in Fig. 3 refers to LSTM, GRU and MLP.

3.2. Architecture of SPNet

Now we elaborate on the proposed SPNet architecture and its components, and how these components help in recognizing the collective activities.

3.2.1. Dense layers

The proposed method consists of there dense layers, i.e., DV, DS, and DA. These dense layers are shown in Branch A and Branch B of Fig. 4. The dense layers DV and DS are trained based on the feature matrix obtained using the IV3 [25] feature extraction pipeline. Moreover, the dense layer DA is associated with the 3D-ResNet 18 [24] feature extraction pipeline represented by Block A of Fig. 4.

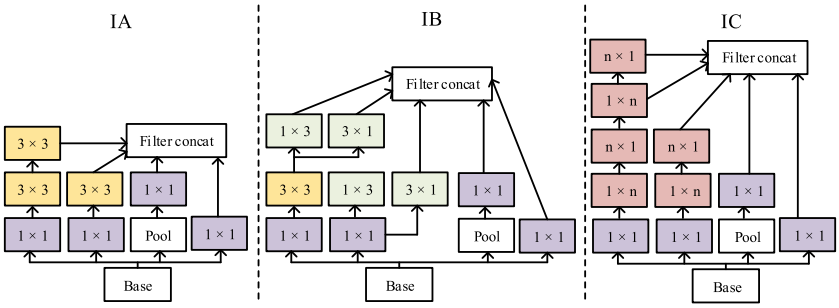

Fig. 5. Overview of the inception modules.

3.2.2. Branch A

We utilize 3D-ResNets [24], with 18 layers, as the action activity recognition branch. ResNets [26] contain a residual bypass connection from one layer to another layer. A connection pass through a batch normalization and ReLu along with a bypass connection from one convolution layer to another convolution layer. This residual connection eases the training of deep convolution networks [24]. Branch A in Fig. 4 represents the 3D-ResNet 18 blocks. The main difference between 3D-ResNets and 2D-ResNets is the number of dimensions of the convolution layers. 3D-ResNet uses a $3 \times 3 \times 3$ convolution block with a temporal stride of 1. The size of the input is $112 \times 112 \times 3 \times 16$. As shown in branch A of Fig. 4, the downsampling is carried out using 3D convolutions with a stride of 2. Zero paddings with adopted identity shortcuts are used where the number of feature maps is increased. This is done to avoid increasing the number of parameters [24].

If, W and H refer to the height and width of the video frame respectively. Moreover, F refers to the number of video frames under consideration with a 3-channel RGB input. Then, z_i is the output tensor computed by the ith convolution block (residual network) for an input video segment V having a size of $3 \times F \times H \times W$. The output of the ith residual block can be represented as:

$$z_i = z_{i-1} + F(z_{i-1}; \theta_i), \tag{3}$$

where $F(z_{i-1}; \theta_i)$ represents the convolution of z_{i-1} having θ_i as weights parameters. In this paper, we utilize the 3D-ResNet 18 extraction pipeline till the final global average pooling layer as shown in Branch A of Fig. 4. An Adams optimizer is utilized with a learning rate of 10^{-5} and a decay of 10^{-6}. Smaller values are used as we trained the network from scratch. Sixteen frames with equal distance temporal positions were selected to train the network.

3.2.3. Branch B

For the views and situation recognition branch, we utilized an Inception V3 (IV3) [25] feature extraction network. The orientation of IV3 is presented in Table 1 and the inception modules are shown in Fig. 5. In Fig. 5, IA represents the module where a 5×5 convolution is replaced by 3×3 convolution. Moreover, IB represents the module with expanded filter bank outputs. IB is used to promote high-dimensional representation. IC represents the inception modules after factorization of the $n \times n$ convolutions. In our case, we choose the default value $n = 7$ for 17×17 grid. IV3 factorizes the traditional 7×7 convolution into 3×3. The inception modules (IA, IB, and IC) have 288 filters each and reduce the features to a 17×17 grid by using 768 filters followed by 5 instances of factorized inception modules. Further, the features are reduced to an $8 \times 8 \times 1280$ grid. At the coarsest level, there are two inception modules, i.e., IC with a concatenated filter bank having a size of 2048 for each tile. To train the IV3 network, we utilized pre-trained weights for the ImageNet dataset. We stripped off the dense layer and fine-tuned using Adams optimizer with a learning rate of 10^{-4} and a decay of 10^{-4}.

The stock IV3 model is intended for a single image classification only. In order to process video sequences, we build a feature matrix by extracting features from every video frame under consideration. Each row in the feature matrix represents the features extracted from a video frame. A graphical overview of the feature matrix building process is shown in Fig. 6.

This branch is represented by Branch B in Fig. 4. In Fig. 4, it can be observed that there are two dense layers (DV and DS) associated with the same feature matrix. DV and DS share IV3 convolution layers. The reason behind using the same feature matrix is that a pre-trained IV3 network is utilized. However, multiple feature-building pipelines will more or less extract the same features while posing a higher computational cost (as the same process is repeated twice). In other words, Branch B follows a multi-task learning approach.

3.2.4. Multiple dense layers

In Fig. 4, it can be noticed that there are three dense layers DS, DV, and DA associated with various feature extraction architectures. These dense layers are MLPs that are trained based on the features extracted by the respective feature extraction pipelines. DS and DV are associated with Branch B of SPNet, while DA is associated with Branch A. DS, DV, and DA are associated with situation, views, and action respectively. These dense layers assist in generating probabilities for input video segments. These probabilities represent the confidence score for an input video segment belonging to a view, action, and situation class.

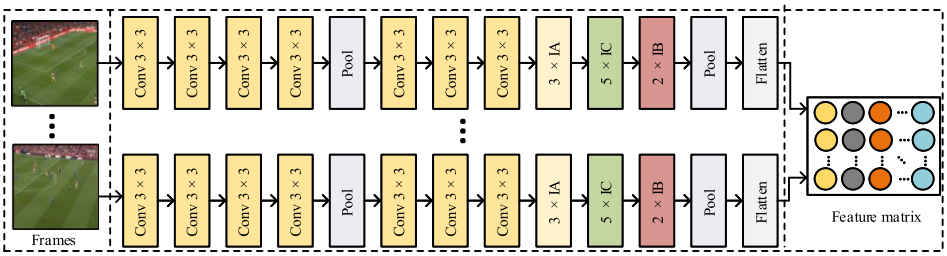

Fig. 6. Graphical overview of our feature matrix building process using IV3 feature extraction pipeline. The dots represent continuity for all the frames under consideration. The detail for patch and stride sizes is presented in Table 1.

Table 1
The orientation of IV3 feature extraction pipeline.

Type	Patch size/stride	Input size
Conv	$3 \times 3/2$	$299 \times 299 \times 3$
Conv	$3 \times 3/1$	$149 \times 149 \times 32$
Conv padded	$3 \times 3/1$	$147 \times 147 \times 32$
Pool	$3 \times 3/2$	$147 \times 147 \times 64$
Conv	$3 \times 3/1$	$73 \times 73 \times 64$
Conv	$3 \times 3/2$	$71 \times 71 \times 80$
Conv	$3 \times 3/1$	$35 \times 35 \times 192$
3 × Inception	IA in Fig. 5	$35 \times 35 \times 288$
5 × Inception	IC in Fig. 5	$17 \times 17 \times 768$
2 × Inception	IB in Fig. 5	$8 \times 8 \times 1280$
Pool	8 × 8	$8 \times 8 \times 2048$
Linear	Logits	$1 \times 1 \times 2048$

3.2.5. Prediction

As mentioned earlier, the main aim is to automatically generate highlights (exciting events) based on user preferences. User preferences may differ from user to user. Formally, we define user preferences as:

$$U_p = \{U_a \cup U_v \cup U_s\}, \tag{4}$$

where U_p is the set of user preferences, U_a is the set of user preferences for the action category, U_v is the set of preferences for views and U_s is the set of users' preferences for situations. Formally, U_a, U_v, and U_s can be written as:

$$U_a \subseteq B \{L_{A_1}, L_{A_2}, L_{A_3}, \ldots, L_{A_{50}}\}, \tag{5}$$

where L_{A_i} represents the class label for the action category. Similarly, L_{V_i} represents the class labels for views and L_{S_i} represents the labels for situations.

$$U_v \subseteq \{L_{V_1}, L_{V_2}, L_{V_3}, \ldots, L_{V_{70}}\}. \tag{6}$$

Similarly, U_s is defined as:

$$U_s \subseteq \{L_{S_1}, L_{S_2}, L_{S_3}, \ldots, L_{V_{36}}\}. \tag{7}$$

We make the final decision based on the activity recognition. Let P_A, P_S, and P_V be the sets of probabilities obtained by the dense layers DA, DS, and DV respectively. We make the final prediction, that whether the activity should be marked as a highlight, based on user preference for the corresponding branch. If a user preference for the corresponding branch exists in the top three probabilities, the activity is marked as an exciting event. Algorithm 1 formally shows the prediction method.

3.2.6. Network flow

Now, we connect all the dots for the proposed SPNet architecture. The main architecture is shown in Fig. 4. In Fig. 4, FE represents feature extraction. We use different branches for action, views, and situation recognition. Overall, all of these can be considered as an activity. The main aim is to recognize the activity and generate highlights based on the type of these activities. As mentioned before, branch A in Fig. 4 represents the action recognition branch trained over the SP-2 dataset with our annotations. It can be observed that we train a separate dense layer for the action, view, and situation branches. Branch A handles spatiotemporal data using 3D convolutions.

For view and situation recognition, features are extracted using the IV3 network. As IV3 is mainly intended for image recognition, features are extracted for each frame under consideration and spatiotemporal feature matrix is constructed. Further, we train two different dense layers for view and situation. Both of the dense layers have 512 hidden units followed by a dropout. Then, the

Algorithm 1 Prediction algorithm.

Input: U_p, P_A, P_V, and P_S
Output: Decision

$Mark\ segment\ as\ non-exciting$;
if $Top_3(P_A) \in U_A$ **then**
  Mark segment as exciting;
end if
if $Top_3(P_V) \in U_V$ **then**
  Mark segment as exciting;
end if
if $Top_3(P_S) \in U_S$ **then**
  Mark segment as exciting;
end if

Table 2
This table presents the total number of annotations per sports category for views, actions, and situations. TV, TA, and TS represent the total number of views, actions, and situations respectively. Moreover, PC represents per category.

Sport category	TV	TA	TS	Total PC
Cricket	7	4	9	20
Table tennis	5	3	1	9
Football	5	7	3	15
Badminton	5	3	1	9
Soccer	4	4	4	12
Snooker	5	2	2	9
Basketball	4	3	2	9
Baseball	6	3	3	12
Rugby	6	6	3	15
Tennis	5	2	2	9
Volleyball	5	4	1	9
Handball	4	2	2	8
Ice hockey	4	3	2	9
Hockey	5	4	1	10
Total	**70**	**50**	**36**	**156**

output of the dropout layer is fed to another dense layer with 512 hidden units and a dropout with 0.5. Based on the probabilities generated by the dense layers, we further predict whether the video segment under consideration should be added to the final set of highlights or not.

4. Experiments and results

We performed comprehensive experiments on the SP-2 dataset to find the best-performing method. In this section, we present details about the dataset, results of the experiments, and some discussion on results.

4.1. Datasets

SP-2: To suit our needs and conduct experiments, we added new annotations to the SP-2 dataset [10]. SP-2 contains about twenty-three thousand sports videos belonging to fourteen categories of broadcast sports. These categories include team sports such as cricket, football, soccer, basketball, baseball, rugby, tennis, handball, snooker, volleyball, ice hockey, hockey, badminton, and table tennis. Originally, the SP-2 dataset contained about 46 action annotations for these sports categories. For each sports category, for the SP-2 dataset, there are about ten to fourteen groups, where each group consists of one hundred and fifty videos on average. Our provided annotations are the first of their kind. These annotations are fine-grained and contain about one hundred and fifty-six annotations in total. We divided the annotations into views, actions, and situations for the sports video. Our new version of annotations contains about seventy different views, fifty action classes, and thirty-six annotations for situations. These details about the number of annotations according to the sports categories are provided in Table 2. Further details about the action, view, and situation annotations are given in Table 3.

The main reason for separating the video into three components is the nature of broadcast sports videos. Broadcast sports videos are evolving, yet, most of the features and camera views remain similar. It means a properly trained deep learning model can generalize well over unseen data. A scene in a broadcast video may contain multiple actions. For example, for a baseball game, during pitching, usually the camera behind the bowler focuses on the bowler and the batsman. Two actions can be associated with such a scene, i.e., the action for the bowler is balling and the batsman is batting. However, the same two actions can be determined

Table 3
Details about views, actions, and situation annotations for each sports category of the SP-2 dataset.

Category	Views	Actions	Situations
Cricket	Bowling front, bowling Side, batsman, players, spectators, arena, boundary	Bowling, single run, fielder throw ball, ball stop	Bowling startup, bouncer, shot, missed, player entering/leaving, six, four, catch, out
Table tennis	Arena, front, top, spectators, players	Game play, drop, service	Short break
Football	Side, boundary, players, spectators, arena	Hike, gameplay, touch down, defense, throw, run, kick	Out, goal, pileup
Badminton	Spectators, front, side, players, top	Drop, game play, service	Short break
Soccer	Side, front, players, spectators	Pass ball, out, long kick, goal kick	Corner kick, out kick/throw, free kick, yellow card
Snooker	Corner, front, side, top view, spectators	Hit, return	Pocket, game start
Basketball	Side, players, front, spectators	Pass, shoot, out	Basket, players idle
Baseball	Pitching/batting, spectators, batsman, players, boundary, arena	Bowling, batting, ball throw	Strike, ball hit, catch
Rugby	Players, spectators, side, pileup, arena, front	Hike, kick, pass, scrum, penalty kick, line out	Out, restart, score
Tennis	Front, players, arena, side, spectators	Service, out	Normal play, out
Volleyball	Side, players, front, spectators, arena	Service, drop, service drop, out	Normal play
Handball	Front, side, players, spectators	Pass ball, yellow card	Goal, out
Ice hockey	Top, players, side, spectators	Pass puck, goal, puck stop	around goal post, out
Hockey	Side, players, spectators, arena, front	Pass ball, goal, out, start after out	Goal attack

using the view only. As it is a standard norm for the broadcast industry, the view across multiple baseball games is more or less the same. Hence, along with an action, we utilize view information for a better description. Moreover, the situation represents the outcome of the action being performed. Consider an example for a cricket scene, where the action performed by the bowler is balling and the batsman hits the ball. As a result of the hit, a boundary score is obtained and we refer to such reaction to the ball hit as a situation. We emphasize the difference of these three subclasses. Some selected samples along with corresponding annotations are presented in Fig. 1.

C-Sports: C-Sports was recently introduced for collective activity recognition. Likewise, C-Sports [21] and SP-2 datasets contain broadcast sports videos. C-Sports contain about eleven sports categories such as American football, basketball, dodge ball, football, handball, hurling, ice hockey, lacrosse, rugby, volleyball, and water polo. There are five different types of annotations for these sports categories including gathering, dismissal, passing, attack, and wandering. C-sports contains about two thousand videos belonging to 11 categories of sports.

4.2. Spatiotemporal (3D networks)

We performed comprehensive experiments based on the state-of-the-art and other spatiotemporal feature extraction networks that performed well for other activity recognition tasks. Details are provided as follows. C3D [27] is a homogeneous 3D convolution based pipeline with small $3 \times 3 \times 3$ convolution kernels. LRCN [28] is spatially and temporally deep, and suitable for sequential learning tasks. We selected another 3D-CNN implementation [29] which is conceptually similar to C3D. Moreover, our experiments include I3D [30] which is inflated inception. The original paper used pre-trained weights on other datasets. In our case, we trained all the 3D networks from scratch. Our experiments include DenseNet-3D and DenseResNet-3D [31]. Additionally, we performed various experiments on the different depths of 3D-ResNets [24]. The results for the experiments based on spatiotemporal deep learning pipelines are presented in Table 4.

4.3. Spatial features

We employed various feature extraction pipelines for our experiment. These feature extraction pipelines are originally designed for image classification. However, we extract spatial features from individual frames and construct spatiotemporal features. Later, we train LSTM and GRU using the extracted spatiotemporal information.

For ResNet 50, we stripped off the top dense layer extracted the features from the last average pooling layer. Further, we train LSTM, GRU, and MLP based on the extracted spatiotemporal features. Similarly, the top layer of a pre-trained Inception V3 [25] was removed. Further, the obtained feature sequences are trained using LSTM, GRU, and MLP. Additionally, pre-trained VGG16 [32] and VGG19 [33] models were used. The last three fully connected layers were removed and 25 088 features were extracted from the flattening layer. Further, LSTM and GRU were trained using the obtained spatiotemporal features.

For IR2 [34], we extracted features using a pre-trained IRV2 network by removing the top layer. Further, LSTM and GRU were trained using the 1536 extracted features obtained from the final average pooling layer. We stripped off the final dense layer from the Resnet152V2 deep learning pipeline and extracted 2048 features from the final average pooling layer. As mentioned before, we trained the obtained feature sequences using LSTM and GRU. Similarly, a spatiotemporal feature sequence was created by obtaining 2048 features from the final average pooling layer in DenseNet169 [35] and Xception [36]. Later, LSTM and GRU were trained using

Table 4
Comparison with state-of-the-art methods with spatiotemporal convolution networks.

Method	Pre-Training	Image size	Views		Action		Situation		Average	
			Acc	T@3	Acc	T@3	Acc	T@3	Acc	T@3
C3D [27]	Scratch	150 × 150	0.64	0.81	0.61	0.91	0.58	0.86	0.61	0.86
LRCN [28]	Scratch	150 × 150	0.73	0.91	0.71	0.92	0.63	0.91	0.69	0.91
3D convolution [29]	Scratch	100 × 100	0.64	0.84	0.54	0.84	0.59	0.81	0.59	0.83
DenseResNet3D [31]	Scratch	112 × 112	0.55	0.78	0.43	0.65	0.55	0.85	0.51	0.76
DenseNet 3D [31]	Scratch	112 × 112	0.36	0.56	0.48	0.79	0.31	0.61	0.39	0.65
I3D [30]	Scratch	224 × 224	0.66	0.84	0.66	0.93	0.53	0.86	0.62	0.88
I3D two stream [30]	Scratch	224 × 224	0.68	0.87	0.69	0.96	0.56	0.87	0.64	0.90
3D-ResNet 18 [24]	Scratch	224 × 224	0.63	0.83	0.73	0.92	0.59	0.87	0.65	0.87
3D-ResNet 34 [24]	Scratch	224 × 224	0.69	0.94	0.68	0.94	0.64	0.86	0.67	0.91
3D-ResNet 50 [24]	Scratch	224 × 224	0.74	0.95	0.66	0.91	0.64	0.86	0.68	0.90
3D-ResNet 101 [24]	Scratch	224 × 224	0.73	0.90	0.70	0.94	0.61	0.88	0.68	0.91
3D-ResNet 151 [24]	Scratch	224 × 224	0.74	0.93	0.70	0.94	0.62	0.86	0.69	0.91
SPNet	Scratch/ImageNet	Dual	0.83	0.96	0.73	0.92	0.73	0.95	0.76	0.94

Table 5
Some selected results for the state-of-the-art spatial feature extraction techniques.

Method	Pre-Training	Image size	Views		Action		Situation		Average	
			Acc	T@3	Acc	T@3	Acc	T@3	Acc	T@3
InceptionV3 + LSTM [25]	ImageNet	299 × 299	0.73	0.96	0.74	0.93	0.69	0.67	0.72	0.85
InceptionV3 + MLP [25]	ImageNet	299 × 299	0.83	0.96	0.69	0.97	0.73	0.95	0.75	0.96
ResNet50 + LSTM [26]	ImageNet	224 × 224	0.77	0.96	0.73	0.98	0.71	0.92	0.74	0.95
ResNet50 + GRU [26]	ImageNet	224 × 224	0.79	0.87	0.68	0.86	0.67	0.79	0.71	0.84
ResNet50 + MLP [26]	ImageNet	224 × 224	0.80	0.94	0.71	0.95	0.68	0.90	0.73	0.93
VGG16 + LSTM [32]	ImageNet	224 × 224	0.73	0.84	0.74	0.91	0.74	0.90	0.73	0.88
VGG16 + GRU [32]	ImageNet	224 × 224	0.76	0.88	0.73	0.98	0.69	0.90	0.73	0.92
IR2 + GRU [34]	ImageNet	299 × 299	0.83	0.95	0.63	0.80	0.64	0.84	0.70	0.86
IR2 + LSTM [34]	ImageNet	299 × 299	0.79	0.95	0.67	0.94	0.68	0.92	0.71	0.94
ResNet152V2 + LSTM [38]	ImageNet	224 × 224	0.82	0.95	0.70	0.95	0.68	0.91	0.73	0.94
ResNet152V2 + GRU [38]	ImageNet	224 × 224	0.77	0.89	0.71	0.92	0.67	0.87	0.72	0.89
DenseNet169 + LSTM [35]	ImageNet	224 × 224	0.80	0.95	0.73	0.97	0.68	0.92	0.74	0.95
DenseNet169 + GRU [35]	ImageNet	224 × 224	0.74	0.86	0.67	0.91	0.65	0.86	0.69	0.88
VGG19 + LSTM [33]	ImageNet	224 × 224	0.79	0.94	0.74	0.97	0.70	0.89	0.75	0.93
VGG19 + GRU [33]	ImageNet	224 × 224	0.73	0.86	0.73	0.88	0.59	0.79	0.68	0.84
Xception + LSTM [36]	ImageNet	299 × 299	0.79	0.96	0.72	0.96	0.65	0.92	0.72	0.95
Xception + GRU [36]	ImageNet	299 × 299	0.77	0.91	0.69	0.91	0.53	0.61	0.66	0.81
EfficientNet (B7) + LSTM [37]	ImageNet	600 × 600	0.35	0.55	0.34	0.57	0.25	0.49	0.31	0.54
EfficientNet (B7) + GRU [37]	ImageNet	600 × 600	0.32	0.51	0.25	0.42	0.23	0.38	0.26	0.44
SPNet	Scratch/ImageNet	Dual	0.83	0.96	0.73	0.92	0.73	0.95	0.76	0.94

the obtained spatiotemporal features. Moreover, we extracted 2560 from the final average pooling layer in EfficientNet (B7) [37]. LSTM and GRU were trained on the obtained spatiotemporal feature sequence. The results for spatial feature extraction methods are presented in Table 5. All the spatial feature extraction networks were pre-trained on the ImageNet dataset.

4.4. Evaluation

We evaluate the performance of the proposed method based on how accurately it detects the activity under consideration. In Tables 5, 4, and 6, Acc represents accuracy. We calculate the accuracy as Acc $= TP + FP/(TP + TN + FP + FN)$, where TP, TN, FP, and FN represent true positive, true negative, false positive, false negative respectively. Moreover, T@3 in Tables 5 and 4 represents the top-3 accuracy. The precision and recall are defined as $TP/TP + FP$ and $TP/TP + FN$ respectively.

We compare the results of our proposed method on two available datasets, i.e., SP-2 and C-Sports. Although C-sports does not contain many videos and does not differentiate between view action and situation, this dataset is the most relevant to our needs. As there are no specified views for C-sports, we only trained two branches for SPNet. For experiments, we consider gathering, dismissal, and pass as actions and attack, and wandering as situations. The comparison and results are presented in Table 6.

4.5. Discussion on results

We performed comprehensive experiments using the SP-2 and C-sports datasets. The results for the 3D convolution based (spatiotemporal) methods are presented in Table 4. 3D deep learning networks have exhibited state-of-the-art performance for other computer vision related tasks. However, in our case, 3D-ResNet 18 exhibited the best performance in recognizing actions. Moreover, 3D-ResNet 151 and 101 exhibited comparable performance for views. From Table 4 it can be seen that our proposed SPNet has clear superiority over other networks in terms of performance.

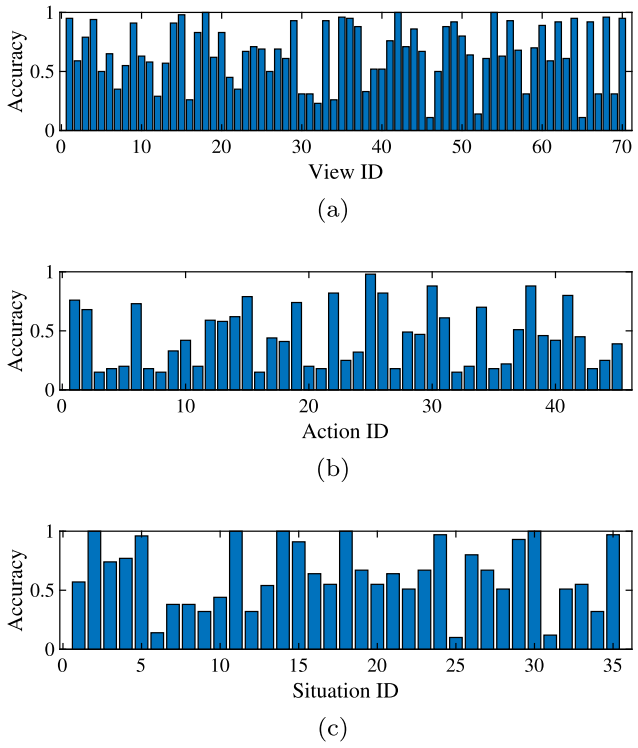

Fig. 7. Accuracy plot of individual classes of view, action, and situation. It should be noted that the results in Tables 4 and 5 show a weighted average for a different number of videos against each label.

Table 5 shows the results of the spatial features extracted using image classifiers. It can be observed that IV3 when combined with MLP exhibited the highest performance for views and situations. This ablation study helps in identifying the best performing deep learning network according to the data under consideration.

As this is the first study that attempts to collectively recognize activities in different broadcast sports videos, we used pre-trained 2D-CNN pipelines to extract the features. By closely inspecting the results presented in Tables 4 and 5, it can be seen that the overall classification accuracies of the spatiotemporal network are lower as compared to feature sequences that we built using pre-trained 2D-CNN pipelines. It should not be considered that the spatiotemporal (3D-CNN) pipelines are less accurate. We believe that the main reason behind the better performance of the 2D-CNN pipeline is that all the feature extraction layers are well trained using a large amount of image data (ImageNet). On the other hand, the spatiotemporal feature extraction networks are trained from scratch.

In Fig. 7, the bar graph (a) shows the per class performance for views, (b) represents the individual performance for actions, and (c) represents the individual performance for situations. From these graphs, it can be observed that some of the accuracies are quite low. In Table 6, we presented a comparison with various methods for the SP-2 and C-Sports datasets. It can be observed that the performance of this method is higher on the C-sports dataset. We believe that the main reason behind this phenomenon is the smaller size and less variation in the dataset. Moreover, it can be observed that our proposed SPNet has the best performance in terms of accuracy.

4.6. Computational complexity

Deep learning-based models require a large number of computational resources to process a large number of parameters. The details for the number of parameters for our proposed method are provided in Table 7. However, it should be noted that the training takes more time as compared with the actual processing of samples. The proposed model processes approximately 400 million parameters for a single video segment. All the experiments were carried out using Ubuntu 18.04 with four GeForce RTX 2080 GPUs and an Intel Xeon processor with 48 cores with a total RAM of 188 GB.

5. Conclusion

Broadcasted sports videos usually have long durations and contain only a few exciting moments. It is not feasible for sports enthusiasts to watch the whole game. For this reason, many professional bodies and amateur content creators manually crop the video segments from long-duration broadcast videos. There exists a need for an automatic method capable of extracting the exciting

Table 6
Comparison of the results obtained over two datasets SP-2 and C-Sports. CS stands for C-Sports dataset.

Method	Image size	Acc (SP-2)	Acc (CS)
C3D [27]	150 × 150	0.61	0.66
LRCN [28]	150 × 150	0.69	0.71
3D convolution [29]	100 × 100	0.59	0.69
I3D [30]	224 × 224	0.62	0.73
I3D two stream [30]	224 × 224	0.64	0.72
3D-ResNet 18 [24]	224 × 224	0.65	0.77
3D-ResNet 34 [24]	224 × 224	0.67	0.71
3D-ResNet 50 [24]	224 × 224	0.68	0.67
3D-ResNet 101 [24]	224 × 224	0.68	0.75
3D-ResNet 151 [24]	224 × 224	0.69	0.70
IV3 + LSTM [25]	299 × 299	0.72	0.74
IR2 + GRU [34]	299 × 299	0.70	0.76
IR2 + LSTM [34]	299 × 299	0.71	0.73
SPNet	Multi	**0.76**	**0.82**

Table 7
Details for the number of parameters processed by the proposed method.

Module	Number of parameters		
	Non-trainable	Trainable	Total
IV3 feature extraction	34,432	21,768,352	21,802,784
DV	0	17,076,294	17,076,294
DS	0	17,058,852	17,058,852
3D-ResNet 18 + DA	7808	33,232,562	33,240,370

moments while keeping in view the different opinions and preferences of sports enthusiasts (user preferences). In this paper, we presented a deep network that collectively recognizes activities to generate highlights while preserving the excitement of the long-duration broadcast sports videos. The proposed network utilizes 3D-ResNet 18 and Inception V3 block that exploits high-level visual feature sequences. We divided the broadcast sports video into views, actions, and situations and provide 156 new annotations for the SP-2 dataset. Moreover, we performed extensive experiments to validate the performance of SPNet. The results of the experiments reveal that the proposed SPNet has the best performance in terms of accuracy and it can generate highlights based on user preference.

CRediT authorship contribution statement

Abdullah Aman Khan: Data curation, Methodology, Writing – original draft. **Jie Shao:** Conceptualization, Supervision, Writing – review & editing.

Declaration of competing interest

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

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61832001).

References

[1] Wei Y, Wang X, Guan W, Nie L, Lin Z, Chen B. Neural multimodal cooperative learning toward micro-video understanding. IEEE Trans Image Process 2020;29:1–14.
[2] Hu Y, Tian Y, Yang W, Wang X, Zhang X. Content to cash: Understanding and improving crowdsourced live video broadcasting services with monetary donations. Comput Netw 2020;178:107281.
[3] Shih H. A survey of content-aware video analysis for sports. IEEE Trans Circuits Syst Video Technol 2018;28(5):1212–31.
[4] Khan AA, Shao J, Ali W, Tumrani S. Content-aware summarization of broadcast sports videos: An audio-visual feature extraction approach. Neural Process Lett 2020;52(3):1945–68.
[5] Lu H, Zhang M, Xu X, Li Y, Shen HT. Deep fuzzy hashing network for efficient image retrieval. IEEE Trans Fuzzy Syst 2021;29(1):166–76.
[6] Ma C, Li X, Li Y, Tian X, Wang Y, Kim H, Serikawa S. Visual information processing for deep-sea visual monitoring system. Cogn Robot 2021;1:3–11.
[7] Nakayama Y, Lu H, Li Y, Kamiya T. Widesegnext: Semantic image segmentation using wide residual network and next dilated unit. IEEE Sens J 2021;21(10):11427–34.
[8] Lu H, Zhang Y, Li Y, Jiang C, Abbas H. User-oriented virtual mobile network resource management for vehicle communications. IEEE Trans Intell Transp Syst 2021;22(6):3521–32.
[9] Lu H, Tang Y, Sun Y. DRRS-BC: decentralized routing registration system based on blockchain. IEEE CAA J Autom Sin 2021;8(12):1868–76.

[10] Khan AA, Tumrani S, Jiang C, Shao J. RICAPS: residual inception and cascaded capsule network for broadcast sports video classification. In: MMAsia 2020: ACM multimedia asia, virtual event / singapore, 7-9 march, 2021. 2020, p. 43:1–7.
[11] Zhang R, Wu L, Yang Y, Wu W, Chen Y, Xu M. Multi-camera multi-player tracking with deep player identification in sports video. Pattern Recognit 2020;102:107260.
[12] Host K, Ivasic-Kos M, Pobar M. Tracking handball players with the deepsort algorithm. In *Proceedings of the 9th international conference on pattern recognition applications and methods, icpram 2020, Valletta, Malta, February 22-24, 2020*, 2020, p. 593–9.
[13] Tanikawa S, Tagawa N. Player tracking using multi-viewpoint images in basketball analysis. In *Proceedings of the 15th international joint conference on computer vision, imaging and computer graphics theory and applications, visigrapp 2020, volume 5: visaPP, Valletta, Malta, February 27-29, 2020*, 2020, p. 813–20.
[14] Lin C, Chen Y. Sports video summarization with limited labeling datasets based on 3D neural networks. In: 16th IEEE international conference on advanced video and signal based surveillance, avss 2019, Taipei, Taiwan, September 18-21, 2019. 2019, p. 1–6.
[15] Miao G, Zhu G, Jiang S, Huang Q, Xu C, Gao W. The demo: A real-time score detection and recognition approach in broadcast basketball sports video. In *Proceedings of the 2007 ieee international conference on multimedia and expo, icme 2007, July 2-5, 2007, Beijing, China*, 2007, p. 1.
[16] Khan AA, Lin H, Tumrani S, Wang Z, Shao J. Detection and localization of scorebox in long duration broadcast sports videos. In *Proceedings of the 5th international symposium on artificial intelligence and robotics isair 2020*, 2020, p. 115740J.
[17] Yoon Y, Hwang H, Choi Y, Joo M, Oh H, Park I, Lee K, Hwang J. Analyzing basketball movements and pass relationships using realtime object tracking techniques based on deep learning. IEEE Access 2019;7:56564–76.
[18] Ghosh A, Jawahar CV. Smarttennistv: Automatic indexing of tennis videos. In: Computer vision, pattern recognition, image processing, and graphics - 6th national conference, ncvpripg 2017, Mandi, India, December 16-19, 2017, revised selected papers. 2017, p. 24–33.
[19] Agyeman R, Muhammad R, Choi GS. Soccer video summarization using deep learning. In: 2nd ieee conference on multimedia information processing and retrieval, mipr 2019, San Jose, CA, USA, March 28-30, 2019. 2019, p. 270–3.
[20] He D, Li L, An L. Study on sports volleyball tracking technology based on image processing and 3D space matching. IEEE Access 2020;8:94258–67.
[21] Zalluhoglu C, Ikizler-Cinbis N. Collective sports: A multi-task dataset for collective activity recognition. Image Vis Comput 2020;94:103870.
[22] Rafiq M, Rafiq G, Agyeman R, Choi GS, Jin S. Scene classification for sports video summarization using transfer learning. Sensors 2020;20(6):1702.
[23] Tejero-de-Pablos A, Nakashima Y, Sato T, Yokoya N, Linna M, Rahtu E. Summarization of user-generated sports video by using deep action recognition features. IEEE Trans Multimedia 2018;20(8):2000–11.
[24] Hara K, Kataoka H, Satoh Y. Learning spatio-temporal features with 3D residual networks for action recognition. In: 2017 ieee international conference on computer vision workshops, iccv workshops 2017, Venice, Italy, October 22-29, 2017. 2017, p. 3154–60.
[25] Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. In: 2016 ieee conference on computer vision and pattern recognition, cvpr 2016, Las Vegas, NV, USA, June 27-30, 2016. 2016, p. 2818–26.
[26] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: 2016 ieee conference on computer vision and pattern recognition, cvpr 2016, Las Vegas, NV, USA, June 27-30, 2016. 2016, p. 770–8.
[27] Tran D, Bourdev LD, Fergus R, Torresani L, Paluri M. Learning spatiotemporal features with 3d convolutional networks. In: 2015 ieee international conference on computer vision, iccv 2015, Santiago, Chile, December 7-13, 2015. 2015, p. 4489–97.
[28] Donahue J, Hendricks LA, Rohrbach M, Venugopalan S, Guadarrama S, Saenko K, Darrell T. Long-term recurrent convolutional networks for visual recognition and description. IEEE Trans Pattern Anal Mach Intell 2017;39(4):677–91.
[29] Weng X, Kitani K. Learning spatio-temporal features with two-stream deep 3D CNNs for lipreading. In: 30th british machine vision conference 2019, bmvc 2019, Cardiff, UK, September 9-12, 2019. 2019, p. 269.
[30] Carreira J, Zisserman A. Quo vadis, action recognition? A new model and the kinetics dataset. In: 2017 ieee conference on computer vision and pattern recognition, cvpr 2017, Honolulu, HI, USA, July 21-26, 2017. 2017, p. 4724–33.
[31] Huang G, Liu Z, Pleiss G, van der Maaten L, Weinberger KQ. Convolutional networks with dense connectivity. IEEE Trans Pattern Anal Mach Intell 2019. http://dx.doi.org/10.1109/TPAMI.2019.2918284.
[32] Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein MS, Berg AC, Li F. ImageNet large scale visual recognition challenge. Int J Comput Vis 2015;115(3):211–52.
[33] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. In: 3rd international conference on learning representations, iclr 2015, San Diego, CA, USA, May 7-9, 2015, conference track proceedings. 2015.
[34] Szegedy C, Ioffe S, Vanhoucke V, Alemi AA. Inception-v4, Inception-ResNet and the impact of residual connections on learning. In *Proceedings of the thirty-first aaai conference on artificial intelligence, February 4-9, 2017, San Francisco, California, USA*, 2017, p. 4278–84.
[35] Huang G, Liu Z, van der Maaten L, Weinberger KQ. Densely connected convolutional networks. In: 2017 ieee conference on computer vision and pattern recognition, cvpr 2017, Honolulu, HI, USA, July 21-26, 2017. 2017, p. 2261–9.
[36] Chollet F. Xception: Deep learning with depthwise separable convolutions. In: 2017 ieee conference on computer vision and pattern recognition, cvpr 2017, Honolulu, HI, USA, July 21-26, 2017. 2017, p. 1800–7.
[37] Tan M, Le QV. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th international conference on machine learning, icml 2019, 9-15 June 2019, Long Beach, California, USA*, 2019, p. 6105–6114.
[38] He K, Zhang X, Ren S, Sun J. Identity mappings in deep residual networks. In: Computer vision - ECCV 2016 - 14th european conference, Amsterdam, the Netherlands, October 11-14, 2016, proceedings, part iv. 2016, p. 630–45.