Multimedia Systems (2022) 28:2453–2464
https://doi.org/10.1007/s00530-022-00978-8

REGULAR ARTICLE

ENet: event based highlight generation network for broadcast sports videos

Abdullah Aman Khan[1,2] · Yunbo Rao[1] · Jie Shao[1,2]

Received: 4 May 2022 / Accepted: 2 July 2022 / Published online: 19 July 2022

Abstract
Handcrafting sports video summaries based on highlights and important events from broadcast sports videos is a laborious and time-taking task. Amateur content creators and professional bodies around the world spend hundreds of man-hours to keep the audience up to date with the latest happenings by means of such highlights. In this paper, we present a deep learning-based method capable of automatically generating highlights from a broadcast sports video based on important events and user preferences. Our proposed method classifies the broadcast sports video scene to generate a summary based on highlights or important events. As various sports have different rules, playfield scenarios, and high inter-class similarities, it is quite challenging to devise a generalized method capable of handling different categories of sports. To overcome such problems and to enhance the highlight generation performance, the proposed method internally segregates the sports category and then utilizes various convolution neural network based feature extraction branches to recognize the important events. Additionally, a branch selector mechanism is introduced to select the relevant convolution neural network branch, which predicts the important sports event/activity. We performed extensive experiments using different deep learning architectures. In terms of important event recognition, the results of the experiments show the superiority of our proposed method.

Keywords Sports video analysis · Video summarization · Sports highlights · Broadcast sports

1 Introduction

On daily basis, a lot of videos are posted on the internet. It is time-consuming for the users to browse and watch the full video as well as for the content creators to create such content. Similar is the case with broadcast sports. Broadcast sports videos normally have long durations with only a few exciting events. There exist a lot of redundant scenes/video segments in the original video. Usually, sports fans rely upon sports highlights to keep themselves updated in a short time. For this reason, content creators spend hundreds of hours of manual labor to find the exciting events and crop the exciting video segments to create a highlight (summary) based on important/exciting events.

To save time and other resources, an automatic highlight extractor is required to extract important events from a long duration broadcast sports video. An ideal broadcast sports video highlight generation method must consider the exciting events based on user preferences as different sports fans can have different priorities and preferences for various sports categories. For example, for a soccer game, some sports fans consider the goals as exciting events, whereas some sports fans may consider the corners and penalties as exciting events. This study aims to automatically generate sports video highlights from a long duration sports video without losing the excitement of the sports game and carter user preferences. As there are too many redundant/non-exciting scenes in a sports video, it is quite difficult to judge which kind of event may interest a sports enthusiast (user). For a sports game, every event can be considered a highlight. However, sports enthusiasts prefer to keep themselves updated through highlights that contain important events only. For example, for a soccer game, goals, penalty strikes, corner shots, etc. are considered important events.

✉ Jie Shao
shaojie@uestc.edu.cn

Yunbo Rao
raoyb@uestc.edu.cn

[1] University of Electronic Science and Technology of China, Chengdu, China

[2] Sichuan Artificial Intelligence Research Institute, Yibin, China

As all the sports enthusiasts universally agree on such events are exciting, it is safe to say that such events can be regarded as highlights.

Sports video highlight generation can be considered as a subfield for video summarization and video analysis [1]. Older studies such as [2, 3] focused on non-semantic based summarization of videos. Recently, many researchers diverted their attention to video summarization such as [4–7]. However, in such studies, the main aim is to extract important keyframes. Studies such as [8, 9] focused on generating highlights from different sports video categories. However, some studies focused on a specific category of sports. For example, studies such as [10, 11] fixated on basketball. Moreover, [12, 13] concentrated on tennis. A lot of studies including [14, 15] targeted soccer sports videos. Similarly, some studies [16, 17] related to volleyball and cricket received some attention.

Some studies such as [18] focused on generating sports video highlights based on the contextual information provided by the scorebox. Such methods can accurately capture the exciting events but may miss some of the exciting moments that cannot be represented by a scorebox. For example, for a soccer game, a goal can be detected by tracking the increment in the scorebox. However, events such as a goal miss and penalty kicks may be missed as the scorebox does not cover such information.

Some researchers focused on summarizing sports videos based on scene classification. For example, Yan et al. [19] proposed a method based on the Vgg16 deep learning model for summarization based on sporting events. Minhas et al. [20] proposed a method for sports shot classification based on AlexNet. Similarly, Rafiq et al. [21] presented a scene classification method based on AlexNet for summarizing sports videos. Sanabria et al. [22] proposed a method based on extracting features using GoogleNet (Inception V1) for summarizing soccer sports videos. Similarly, Agyeman et al. [14] proposed a method based on 3D Resnet 34 spatiotemporal deep learning model for summarizing soccer sports videos. Moreover, Turchini et al. [23] presented a method based on Vgg16 with modified dense layers for summarizing and automatically filming soccer games. Datt and Mukhopadhyay [24] proposed a method for summarizing videos based on a convolutional neural network accompanied by LSTM layer.

Different sports types (basketball, soccer, football, etc.) have different ruleset and playfield scenarios. For example, the number of players, playground, and layout for cricket are entirely different from badminton. Most of the previous studies either focus on generating highlights without considering the internal action/event, focus on a single type of sports, use primitive techniques (player tracking, extracting live scores, relying on text streams, etc.), and do not segregate different types of sports categories. For example, for a soccer game, such approaches are not able to detect goal misses, corner shots, outs, etc. Moreover, sports belonging to various sports categories have high inter-class similarities [25]. Previously proposed scene classification based summarization methods are unable to handle high interclass similarities during the prediction process.

In this paper, we address the problem of high inter-class similarity of broadcast sports videos. The proposed method first segregates the sports category then further utilizes multi-branch based multi-task learning (MTL) [26] for accurately predicting important events in a broadcast sports video. Our proposed method automatically extracts highlights (based on user preferences) from different categories of long-duration broadcast sports videos. It not only detects important events but also describes which type of event has occurred. Our contributions are:

- We present a method named ENet that tackles the problem of high inter-class similarities of sports categories. ENet internally segregates the input sports video into the appropriate category and then important events are identified according to user preferences.
- Comprehensive experiments are performed on two new benchmark datasets (SP-2 and C-Sports) to validate the performance of our proposed method. Related material and source code are available[1]

2 Problem formulation

The main aim of this research is to automatically generate a smaller version of a long-duration video that preserves the excitement of the sports. For this purpose, we focus on recognizing important/exciting activities and scene classification. Formally, we have a set of training videos $V_{\{v,i\}}$ with activity labels L_i for each video segment and T_v as the total number of videos segments. Using various feature extraction pipelines, a set of probabilities is generated. Based on these probabilities (P_r) and user preferences U_p, the final set of highlights (H_l) is constructed. The aim of training the learnable is to learn the weights/parameters of the function H_l. The proposed model can be represented by a function, i.e., $H_l = f(V_{\{v,i\}}) = f(V_{\{v,1\}}, V_{\{v,2\}}, V_{\{v,3\}}, \ldots, V_{\{v,T_v\}})$, where v and i represent long-duration broadcast sports video and the i-th video segment respectively. During the training process, the aim is to minimize the classification loss given by:

$$Loss_e(y, \hat{y}) = -\sum_{j=1}^{N} y_j \log\left(\hat{y}_j\right), \tag{1}$$

[1] https://github.com/abdkhanstd/ENet.

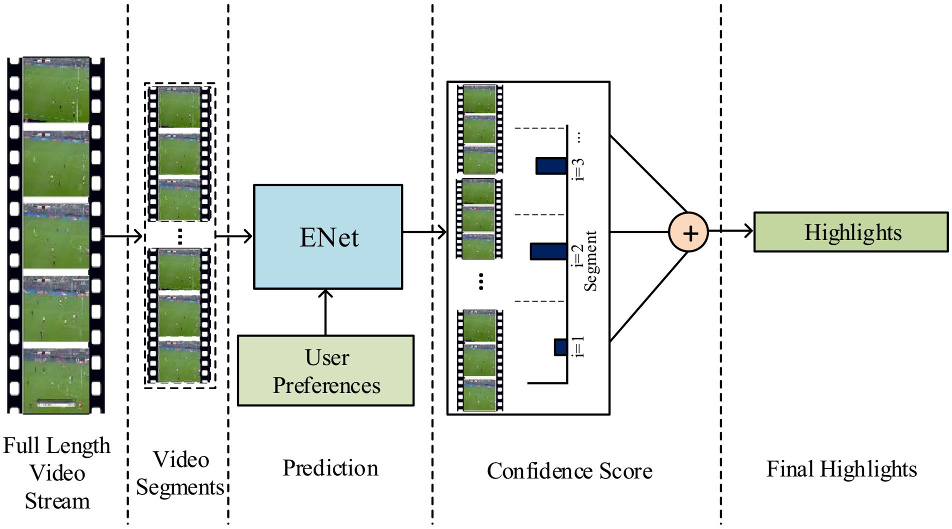

Fig. 1 Overview of the proposed scheme for extracting highlights by detecting the important events in a long-duration video

where y represents the one-hot class vector and ground truth for the test sample. However, the relationship between the video segments is ignored. The loss for scene classification is computed by the average of cross-entropy functions of all video segments:

$$Loss_m = (1-\alpha)\frac{1}{m_i}\sum_{j=1}^{m_i} Loss_{ce}\left(y_j, \hat{y}_j\right) + \alpha Loss_e, \tag{2}$$

where m_i represents the size of the input video segment and α represents the weight for cross entropy, i.e., Eq. 1.

3 Our method

In this section, we elaborate on the details of our proposed method and the implementation details.

3.1 Overview

The main idea of the proposed method is to divide a long video into smaller segments. Afterward, each activity is recognized by the proposed method. If the recognized activity matches the user-provided preferences, the video segment is considered an important event. The general working mechanism of our model is presented in Fig. 1 and the proposed ENet is presented in Fig. 2. ENet first categorizes the type of sports and then handles the input video segment accordingly. To make things simple, we classify the type of sports category by using the RICAPS module [27]. Block-C in Fig. 2 represents the sports categorization module. It should be noted that RICAPS [27] is a baseline method for effective sports video categorization/classiciaftion. Formally, we denote the type of sports as T_s (ground truth).

Different sports categories have various layouts and scenarios. However, there are high inter-class similarities that pose a threat to the performance of activity recognition or scene classification models. To overcome this problem, we first segregate the sports class and utilize independent scene classifications/activity recognition branches as shown in Fig. 2. Based on previous study [27] and our ablation study, we separate the sports into five main branches. The first branch (B1) is associated with sports such as football, ice hockey, baseball, hurling, and tennis. The second branch (B2) handles sports such as handball, volleyball, water polo, and basketball. The third branch (B3) is associated with cricket, soccer, hockey, dodgeball, and snooker. The fourth branch (B4) is associated with sports categories such as rugby, badminton, lacrosse, and table tennis. Moreover, the fifth branch (B5) represents the sports category classification. Each branch is assigned a unique learnable layer (LSTM, RICAPS [27], and dense layers). More information regarding block and branch association is provided in Table 1.

To save computational cost and architecture area, we assign multiple branches to a single deep feature extraction pipeline. This phenomenon is usually referred to as multi-task learning (MTL). In other words, features are extracted using a common feature extraction pipeline and later multiple dense layers (with different classes) are trained using such features. For example, for training B3 and B4, the features are extracted using a single spatiotemporal feature extraction pipeline. Moreover, we train two separate dense layers for B1 and B2. The output probability of each branch B1, B2, B3, and B4 (confidence) is denoted by $P1$, $P2$, $P3$, and $P4$ respectively. Further, we explain the learning flow of the proposed ENet.

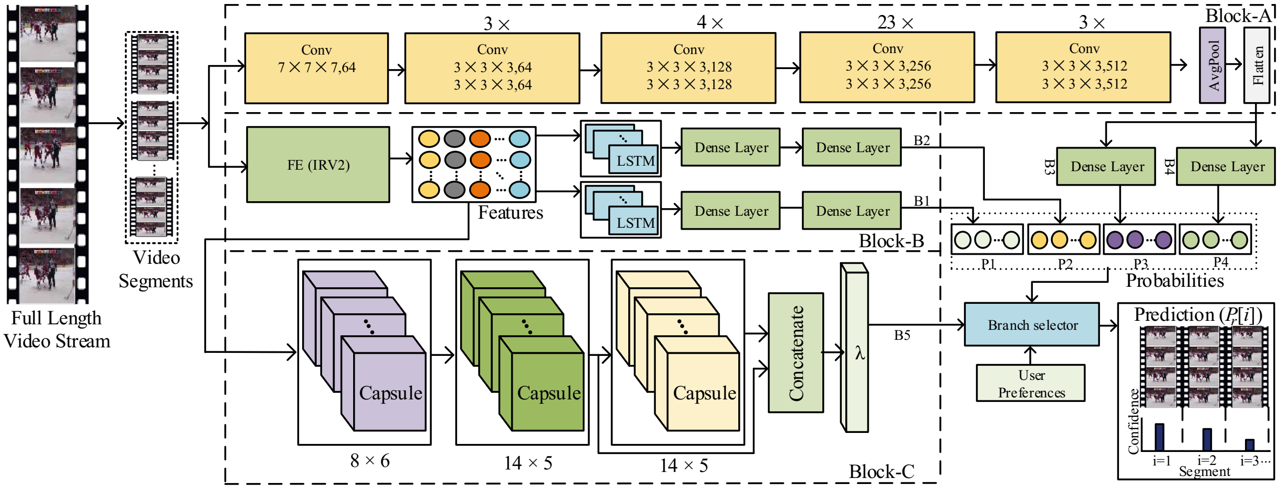

Fig. 2 Overview of the proposed method. FE stands for feature extraction while B1, B2, B3, and B4 represent the four multi-task learning branches associated with various sports categories and feature extraction blocks. Block-A represents the 3D Resnet 101 feature extraction process, Block-B represents the feature extraction process using IRV2, and Block-C shows the sports categorization process. The dashed square represents the blocks

Table 1 Description of the functionality of deep feature extraction blocks and the branches associated with sports categories

Sno	Category/function	Branch	Block	FE
1	Cricket, soccer, hockey, dodgeball, and snooker	B3	Block-A	3D Resnet 101
2	Football, ice hockey, baseball, hurling, and tennis	B1	Block-B	IR2
3	Rugby, badminton, lacrosse, and table tennis	B4	Block-A	3D Resnet 101
4	Handball, volleyball, water polo, and basketball	B2	Block-B	IR2
5	Sports categorization	B5	Block-C	IR2

FE represents the deep feature extraction pipeline associated with the relevant block. It is worth noting that the branches represent the multi-task learning process while the block represents the feature extraction pipelines

3.2 User preferences

As every sports enthusiast may have different priorities for important events, the proposed method predicts important events based on user preferences. For example, for a cricket game, some users might find the boundary events and missed ball events more interesting as compared with outs, run-outs, etc. (despite all being important events). The proposed method is capable of generating user-oriented and customizable highlights according to the needs/requirements of a user. The user preferences are handled by defining a list (i.e., U_p) which consists of the keywords/sub-classes that a user wants to include in the highlights. Our method helps in only selecting the important events provided in the user preference list U_p. For example, if a user wants to generate a customized summary that only contains goals and penalty kicks for a soccer game, the user preferences list will only include “goals” and “penalty kicks” Further details about U_p are provided in Sect. 3.6.

3.3 Sports classification

As mentioned earlier, ENet utilizes RICAPS for automatic sports video classification. RICAPS is represented by Block-C and B5 in Fig. 2. Different sports have different playfield scenarios, scenes, etc. However, there exist some high inter-class similarities of various sports categories. Consequently, different sports can easily be confused [27] even by a human spectator. To avoid such confusion among sports, we first classify the type of sports, and then we attempt to recognize the activity in the video segment accordingly. Based on the video classes, we categorize the sport into four groups and every group is handled by various feature extraction pipelines (branches) explained next.

3.4 Feature extraction

We extract features for B3 and B4 using a modified 3D Resnet 101 [28] feature extraction pipeline (Block-A in Fig. 2). The branches represent dense layers (neural network) associated with MTL. 3D Resnet extracts more valuable spatiotemporal information as compared with its 2D counterpart. A residual block contains two convolution layers, a batch normalization (BN) and ReLU. The top block is connected to ReLu (bypass layer). The first convolution layer with a stride of 64 is followed by three consecutive convolution layers. Four more convolution layers are added with a stride of 128. Next, a block of twenty-three convolution layers with a stride of 256 is added. The resultant is then forwarded to a block of three convolution layers. Average pooling is carried before the dense layers for B3 and B4. Similarly, for B1, B2 and recognizing the sports category, we utilize the features extracted by IRV2 [29] feature extraction pipeline pre-trained on ImageNet. Unlike spatiotemporal architectures (3D), spatial feature extraction pipelines need special treatment for building feature sequences. Features are extracted from each frame while forming a feature tensor or a feature matrix as shown in Block-B of Fig. 2. Instead of processing spatiotemporal information such as 3D Resnet, we build feature sequences by extracting features from individual frames. Each row of the feature matrix represents 1536 features extracted by the IRV2 feature extraction pipeline. The feature matrix represents the feature sequences for the corresponding video frame sequences. Further, these feature sequences are processed as spatiotemporal. The input image size for 3D Resnet 101 is 224×224, and for IRV2 is 299×299 pixels.

We choose 3D Resnet 101 for B3 and B4 as it extracts spatiotemporal features and has exhibited superior performance for action recognition related tasks on various datasets. Moreover, the reason for choosing IR2 as a spatial feature extraction is twofold. First, it has exhibited high classification performance on ImageNet and its pre-trained models are readily available. The second reason is that RICAPS features are also extracted using IR2. Using a common feature extractor optimizes the model (in terms of size and latency) by using a common feature extraction model for B5, B2, and B1. In other words, common features are shared by the learnable layers of B5, B2, and B1 (instead of extracting features using separate feature extraction pipelines).

3.5 Training learnables

This subsection elaborates on the learning flow of the neural network. After extracting features from 3D Resnet 101 and IRV2, we train RICAPS using features extracted from IRV2. The sports categories for B1 and B2 are trained using LSTM with 2048 units with a dropout of 0.5 followed by a dense layer with 512 units, ReLU activation, and a dropout rate of 0.5. As mentioned before, for B3 and B4, the features are extracted using 3D Resnet 101 pipeline. We train two different dense layers for B3 and B4, and both dense layers have 512 units. Probabilities are obtained using dense layers. The top probabilities represent confidence about the importance of the scene.

3.6 Branch selector

We introduce a module branch selector as shown in Fig. 2. The main function of the branch selector is to select the corresponding branch from B1, B2, B3, and B4 according to the sports category (predicted by RICAPS) and determine whether the video segment under consideration is an important (using Boolean) event or not. While determining the importance, the branch selector algorithm also records the confidence score for the video segment under consideration. Later, based on the user's choice, a threshold level for this confidence can be used to select important events. For example, if a user wants an aggressive approach then recognized important events with high confidence scores are chosen only and vice versa. In other words, the branch selector assigns a video segment under consideration to the relevant branch for further processing. The branch selector flags important sports events based on user preferences while recording the confidence score of the segment. For example, a user wants to generate a short video summary of all the goals for the soccer match. The user can pass preferences to the branch selector which in turn determines the importance of the event based on these preferences. Let S_C be the sports category (RICAPS output) and U_P be the set of user preferences. It should be noted that S_c and T_s both represent the sports category. S_c is the sports category determined by RICAPS. However, T_s represents the ground truth sports category. We segregate these two terms here to avoid confusion in our future work. Formally, U_p can be written as $U_p \in A_l$ where A_l is the activity list/labels for an event. Formally, $A_l = \{L_i, L_{i+1}, L_{i+2}, ..., L_{i+n}\}$ where each L_i denotes a sports activity. The working mechanism of the branch selector is provided in Algorithm 1. Moreover, P is defined as $P = \{P1 \cup P2 \cup P3 \cup P4\}$. B_{l1}, B_{l2}, B_{l3}, and B_{l4} are the list for sports categories for B1, B2, B3, and B4 respectively. These lists are hardcoded to the branch selector module and $P_r[i]$ is the probability record for the current segment under consideration.

Fig. 3 Some selected samples from the SP-2 and C-sports datasets. Samples on the top two rows are taken from the C-sports dataset and the bottom three rows from the SP-2 dataset

Algorithm 1 Branch selector algorithm.

```
Input: $S_C$, $U_p$, $A_l$, $P1$, $P2$, $P3$ and $P4$ for $i$-th segment;
Output: Decision and $P_r$;
  if $S_C \in B_{l1}$ then $P_b = P1$;
  end if
  if $S_C \in B_{l2}$ then $P_b = P2$;
  end if
  if $S_C \in B_{l3}$ then $P_b = P3$;
  end if
  if $S_C \in B_{l4}$ then $P_b = P4$;
  end if
  $T=$ Max($P_b$); //(i.e., the top probability in $P_b$)
  $L=$ Class label for corresponding probability in $P_b$;
  if $L \in U_p$ then
      Mark current segment as important event;
      $P_r[i] = T$; //(i.e., represents the confidence of an event)
  else
      $P_r[i] = 0$; //(i.e., not a user preference)
  end if
```

3.7 Highlight generation

After generating the probability/confidence map, using Algorithm 1, P_r holds the probabilities/confidence scores for all the important event. As explained in Sect. 3, we set an average cutoff level to determine the important events as $C_o = average(P_r) \forall P_r > 0$. All corresponding segments having $P_r[i] > C_o$ can be added to the highlights.

4 Experiments and results

In this section, we provide details about the datasets, experimental setup, results, and analysis of the results.

4.1 Datasets

To validate the performance of our proposed method, we performed experiments on two datasets related to broadcast sports. SP-2 [27] and C-Sports [26] are the only two available datasets related to broadcast sports having fine-grained activity annotations.

SP-2 contains 84 activities/actions for 14 sports categories for about twenty-three thousand videos. This dataset contains various lists for fine-grained annotations for broadcast sports categories such as table tennis, badminton, hockey, ice hockey, volleyball, snooker, basketball, handball, tennis, rugby, baseball, soccer, football, and cricket. The SP-2 dataset contains between ten and fourteen groups for each sports category where each group contains about one hundred and fifty videos approximately. SP-2 is the largest available dataset related to broadcast sports video with fine-grained annotations. However, there is a high level of interclass similarity of various sports categories. For example, there exists a huge playfield similarity between hockey and soccer. Statistical and other relevant details about the SP-2 dataset can be found in [27].

C-Sports has five activity annotations for 11 categories of sports for around one thousand videos. These broadcast sports categories include water polo, volleyball, rugby, lacrosse, ice hockey, hurling, handball, soccer, dodgeball, basketball, and football. More information regarding the C-Sports dataset can be found in [26]. Some selected frame samples from these two datasets are shown in Fig. 3.

4.2 Ablation study and baselines

Summarizing broadcast sports video by action or scene classification is a relatively new and under-explored concept. Thus, there are not many methods to compare the proposed ENet with. To validate the performance of ENet, we perform an ablation study and introduce some baseline methods. We present the summarization performance of the baseline methods along with some famous action recognition methods. The details for the baseline methods are provided as follows:

Method 1: For the first baseline method under consideration, we extracted the spatial features for every frame under consideration using Inception Resnet V2 (IRV2) [29]. We modified the original architecture by stripping off the top dense layer. Further, we constructed a feature matrix using the features extracted for each frame by IRV2. A total of 1536 features were extracted for each frame. Further, an LSTM layer with 2048 units has a dropout value of 0.5 followed by two dense layers (neural network) with 512 units and the number of associated classes respectively with a dropout of 0.5.

Method 2: For this method, we utilized an Inception V3 (IV3) [30] feature extraction pipeline to construct feature matrix. From the final average pooling layer, 2048 features were extracted for each frame under consideration. Further, we utilized recurrent layers (LSTM), similar to Method 1, for scene classification.

Method 3: We utilized a modified Inception Resnet V2 (IRV2) deep feature extraction architecture (similar to Method 1) to extract 1536 features for each frame. Further, we utilized two gated recurrent units (GRU) layers with 120 units each followed by two dense layers for scene classification. We used a dropout value of 0.2 for the first GRU layer and 0.5 dropout for the first dense layer.

Method 4: For extracting deep features for every frame, we utilized a Vgg16 [31] architecture. The last three dense layers are stripped off to obtain the required features. Further, we constructed the feature matrix using 25,088 features extracted from each frame. We utilized an LSTM based scene classification mechanism similar to Method 1.

Method 5: Similar to Vgg16, we extracted 25,088 features for each frame using Vgg19 [31] feature extraction pipeline. These features were obtained by removing the last three dense layers from the original architecture. Similar to Method 1, an LSTM based scene classification mechanism is adopted. It should be noted Vgg16 and Vgg19 architectures yield a large number of features as compared with other feature extraction pipelines. Hence, it requires more computational resources to train based on such features.

Method 6: We utilized a Resnet 152 [32] feature extraction pipeline for building feature matrix for the frames under consideration. 2048 features were extracted by stripping off the last dense layer. Further, similar to Method 4, an LSTM based scene classification scheme is used.

Method 7: In this method, we utilized a Densenet169 [33] based feature extraction mechanism. For each frame, 1664 features were extracted from the second last layer of the original model. After constructing the feature matrix, an LSTM based scene classification mechanism was adopted.

Method 8: We built feature matrix using 2048 features extracted from the second last layer of Resnet 50 [32] deep learning model. Further, similar to Method 1, an LSTM based approach is utilized for scene recognition.

4.3 Evaluation

All the experiments were performed on the officially provided test and training lists. For C-Sports, additional testing and training lists were generated. All the experiments were performed on disjoint videos, i.e., the testing and validation videos are unseen to the trained network. The SP-2 dataset is published with raw videos while the C-sports dataset contains only frames. We conducted our experiments by selecting 16 frames for category determination and B1. While for B2 and B3, we use 30 frames for our experiments. The accuracy is calculated as $Acc = (TP + TN)/(TP + TN + FN + FP)$ where TP, TN, FP, and FN are the true positive, true negative, false positive, and false negative respectively. TP represents the correct prediction of important sports activities, whereas TN refers to the correctly predicted segments that are not important activities. Moreover, FP represents the wrongly predicted activities and FN represents the wrongly predicted unimportant activities.

Branch wise results: We performed comprehensive experiments to search for the best feature extraction architecture for each branch based on the SP-2 dataset. To provide a broader picture, the results of each branch (i.e., B1, B2, B3, and B4) are presented in Table 2. Using these results, we selected the best performing architectures (while keeping in view the overall size of the network). We only report some selected results in Table 2. By considering the individual branch performance of each network, corresponding branches in ENet were assigned.

Class wise results: To further study the performance of the proposed model, class-wise (sports category-wise) performance was evaluated on the SP-2 dataset as presented in Table 4. In Table 4, A represents the number of actions associated with the sports class. Moreover, P represents the precision performance obtained over the concerned sports category and Acc represents the accuracy. From Table 4, it can be observed that the proposed Enet performs differently for various sports categories.

Comparison with other methods: Table 3 shows a comparison of our method with other methods that have

Table 2 Branch wise results of the proposed method

Method	BB	B1	B2	B3	B4	Acc
Method 1	IRV2	0.61	0.70	0.64	0.65	0.65
Method 2	IV3	0.60	0.70	0.70	0.59	0.65
Method 3	IRV2	0.61	0.66	0.55	0.45	0.56
Method 4	Vgg16	0.57	0.63	0.61	0.54	0.59
Method 5	Vgg19	0.55	0.62	0.59	0.57	0.58
Method 6	Resnet152	0.59	0.69	0.68	0.66	0.66
Method 7	Densenet169	0.61	0.69	0.68	0.63	0.65
Method 8	Resnet50	0.59	0.66	0.65	0.63	0.63
I3D [34]	Inception 3D	0.61	0.62	0.66	0.58	0.62
3D Resnet 18 [28]	3D Conv	0.61	0.68	0.65	0.62	0.64
3DCNN [35]	3D Conv	0.57	0.61	0.64	0.52	0.59
C3D [36]	3D Conv	0.54	0.65	0.68	0.51	0.60
3D Resnet 50 [28]	3D Conv	0.61	0.67	0.65	0.50	0.61
3D Resnet 101 [28]	3D Conv	0.62	0.62	0.69	0.54	0.62
3D Resnet 151 [28]	3D Conv	0.56	0.63	0.69	0.56	0.61
ENet (ours)	2D + 3D Conv	0.62	0.70	0.69	0.65	**0.67**

B1, B2, B3, and B4 are the branches for different sports categories, whereas Acc is the accuracy computed by the number of videos associated with that particular branch. BB represents the backbone feature extraction pipeline. The bold value shows the superior performance of our proposed method

Table 3 Comparison with other methods having high performance

Method	BB	Size	Acc (SP-2)	Acc (C-S)
C3D [36]	3D Conv	150^2	0.51	0.57
Datt and Mukhopadhyay [24]	LRCN [37]	150^2	0.59	0.62
3DCNN [35]	3D Conv	100^2	0.53	0.58
I3D + LSTM [34]	Inception 3D	224^2	0.62	0.66
Two stream I3D [34]	Inception 3D	224^2	0.59	0.64
3D Resnet 18 [28]	3D Resnet	224^2	0.63	0.68
Agyeman et al. [14]	3D Resnet 34	224^2	0.56	0.62
3D Resnet 50 [28]	3D Resnet	224^2	0.55	0.59
I3D [34]	Inception 3D	224^2	0.55	0.65
3D Resnet 101 [28]	3D Resnet	224^2	0.59	0.66
3D Resnet 151 [28]	3D Resnet	224^2	0.56	0.61
R2P1D BERT [38]	Milti	112^2	0.57	0.60
Yan et al. [19]	Vgg16	224^2	0.60	0.67
Minhas et al. [20]	Alexnet	227^2	0.32	0.45
Rafiq et al. [21]	AlexNet	227^2	0.36	0.46
Sanabria et al. [22]	GoogleNet	224^2	0.38	0.41
Turchini et al. [23]	Vgg16	224^2	0.62	0.68
ENet (ours)	Multi	Multi	**0.67**	**0.74**

The results are obtained over two datasets SP-2 and C-Sports. Acc stands for accuracy and C-S represents the C-Sports dataset. The size refers to the input image size of the feature extraction network. Moreover, BB represents the backbone. The bold values indicate the superior performance of ENet

performed well on other computer vision problems. We trained C3D, 3DCNN, I3D, R2P1D, and 3D Resnet(s) from scratch, while, all other feature extraction networks were pre-trained on ImageNet. In Table 3, it can be seen that our proposed ENet uses multiple resolutions of input video frames (image sizes). In other words, ENet processes the video frame in two different resolutions. The main reason behind this is that ENet relies on two different deep learning architectures having different input layers as explained in Sect. 3. Moreover, we compare our proposed ENet with

Table 4 Class-wise (sports category-wise) performance obtained on the SP-2 dataset

Sports category	*A*	*P*	F1-score	Acc
Football	10	0.58	0.52	0.56
Ice hockey	4	0.85	0.86	0.88
Baseball	7	0.62	0.64	0.66
Tennis	3	0.62	0.64	0.67
Handball	4	0.63	0.65	0.71
Volleyball	5	0.68	0.5	0.57
Basketball	5	0.64	0.59	0.65
Cricket	13	0.45	0.45	0.47
Soccer	8	0.49	0.52	0.6
Hockey	5	0.61	0.59	0.69
Snooker	4	0.76	0.81	0.87
Rugby	8	0.63	0.67	0.73
Badminton	4	0.7	0.68	0.7
Table tennis	4	0.65	0.62	0.64
Average	-	0.63	0.62	0.67

the previous methods proposed for sports summarization. We compare the performance with Yan et al. [19], Minhas et al. [20], Rafiq et al. [21], Sanabria et al. [22], Turchini, et al. [23], Datt and Mukhopadhyay [24], and Agyeman, et al. [14]. The results of these experiments are presented in Table 3.

4.4 Discussion on results

Individual branch results are presented in Table 2. It can be observed in Table 2 that the best performance for branch 1 (B1) is achieved by 3D Resnet 101. Moreover, the best performance for B2 is achieved by IRV2 and IV3. However, the performance of IRV2 for B3 is lower. To reduce the network area, we select IRV2 as a feature extraction network for B2 and B4. 3D Resnet 101 has the satisfactory performance for B1 and B3.

Table 3 presents the overall results of our proposed method compared with other methods. For all other methods, the deep learning networks were trained without splitting the activities. In other words, the networks were trained using all activities/labels together. Table 3 indicates the benefit of splitting into branches (sports category wise). Our proposed method can achieve the best performance in terms of accuracy. The proposed method is able to recognize sports activities with the highest accuracies.

We present a bird-eye view of the confusion matrices obtained using various summarization methods. In Fig. 4, we present confusion maps for only the best-performing methods. From Fig. 4a–c, it can be noticed that there are some lighter heat spots other than the main diagonal. This indicates that there is a large number of misclassifications. However, in Fig. 4d, it can be seen that there are less lighter spots other than the main diagonal. It can be observed that (comparatively) there is a large number of lighter points along the main diagonal of the confusion matrix of our proposed method which indicates that ENet performs better while addressing the inter-class similarities. The results presented in Table 2 and Fig. 4 indicate that ENet has performed better on datasets with high inter-class similarities. Moreover, some selected sample predictions compared with the ground truth samples are presented in Fig. 5. In Fig. 5, a misclassified video segment can be seen in column-C6. Due to high inter-class similarities, it is quite challenging to predict accurately.

From Tables 2 and 3 it can be observed that segregating the sports category enabled us to break the scene classification process and hence to achieve better performance using various methods. For example, in Table 3 it can be seen for I3D we were able to achieve 55% accuracy while by segregating sports category we achieved a performance of 62% approximately (as shown in Table 2).

As mentioned earlier, there exists a high level of inter-class similarities between various sports classes. Other than inter-class similarities, there exist even larger intra-class similarities (within a sports class). These high intra-class similarities still pose a challenge to the performance of any method. In Table 4, it can be observed that the sports categories having more classes have lower performance due to high intra-class similarities. Addressing the high intra-class similarity can further improve the performance and can be considered a possible future research direction.

5 Conclusion

In this paper, we present a deep learning based method (ENet) for generating highlights from long-duration broadcast sports videos. We present an out-of-the-box approach that segregates a broadcast sports video according to its category using RICAPS. As different sports categories have different rules, high inter-class similarities, and playfield scenarios, we assign various sports categories to different deep learning branches. We present a branch selection mechanism to select a sports activity based on user preferences. The proposed method is generalized and can handle different categories of sports. We performed comprehensive experiments on two available datasets SP-2 and C-Sports. The results of the experiments show the superiority of our proposed method (ENet). From the results, it can be seen that ENet is better capable of recognizing various activities events in a broadcast sports video. As ENet can recognize important activities, important video segments containing important/exciting events can be predicted and extracted. Sports video summaries can be generated by

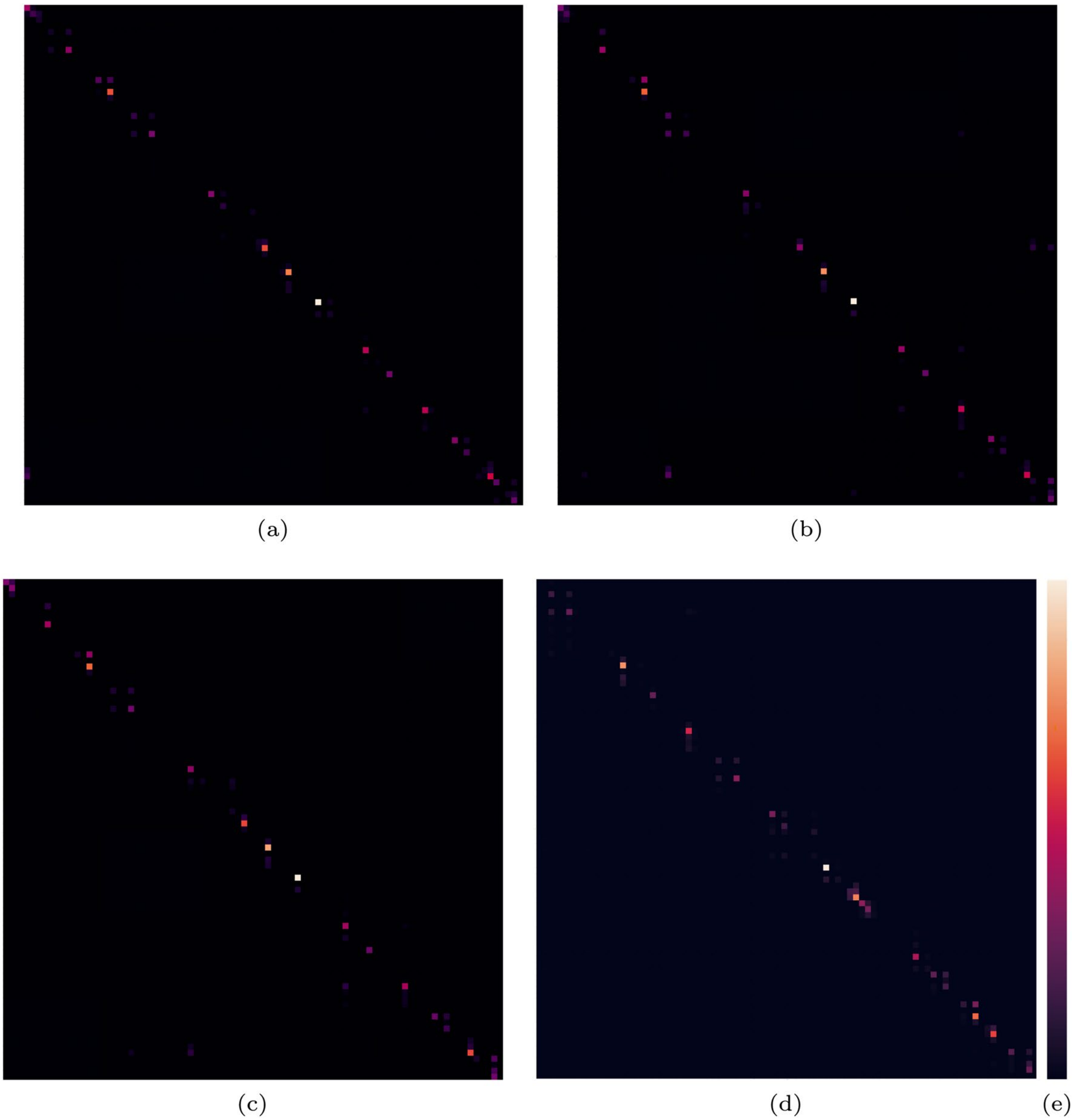
(a)
(b)
(c)
(d)
(e)

Fig. 4 Due to limited space, only a bird-eye view of the confusion matrices are presented. The lighter colors along the diagonal show the true prediction. (a) represents the confusion map for Yan et al. [19] while (b) represents the confusion map for the method of Datt and Mukhopadhyay [24]. Moreover, (c) represents the confusion map obtained using Agyeman et al. [14]. (d) represents the confusion map obtained using ENet. Additionally, (e) represents the heatmap legend where the colors on the top represent higher accuracy in terms of true positives

combining the predicted important events. Moreover, ENet sets a baseline for all the relevant potential future research.

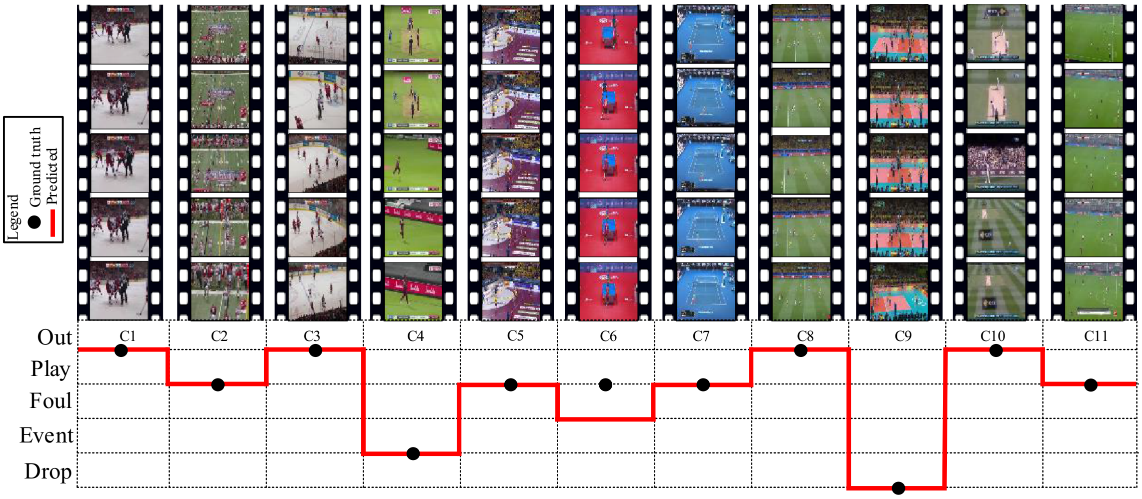

Fig. 5 Some selected sample predictions compared with the ground truth samples

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

References

1. Khan, A.A., 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, p. 115740 (2020)
2. Gong, B., Chao, W., Grauman, K., Sha, F.: Diverse sequential subset selection for supervised video summarization. In: Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, pp. 2069–2077 (2014)
3. Zhao, B., Xing, E.P.: Quasi real-time summarization for consumer videos. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, pp. 2513–2520 (2014)
4. Andonian, A., Fosco, C., Monfort, M., Lee, A., Feris, R., Vondrick, C., Oliva, A.: We have so much in common: Modeling semantic relational set abstractions in videos. In: Computer Vision - ECCV 2020 - 16th European Conference, Proceedings, Part XVIII, pp. 18–34 (2020)
5. Betting, J.L.F., Romano, V., Bosman, L.W.J., Al-Ars, Z., Zeeuw, C.I.D., Strydis, C.: Stairway to abstraction: an iterative algorithm for whisker detection in video frames. In: 11th IEEE Latin American Symposium on Circuits & Systems, LASCAS 2020, pp. 1–4 (2020)
6. Chen, Y., Yuan, H., Li, Y.: Object-oriented state abstraction in reinforcement learning for video games. In: IEEE Conference on Games, CoG 2019, pp. 1–4 (2019)
7. Yamghani, A.R., Zargari, F.: Compressed domain video abstraction based on i-frame of HEVC coded videos. Circ. Syst. Signal Process. **38**(4), 1695–1716 (2019)
8. Islam, M.R., Paul, M., Antolovich, M., Kabir, A.: Sports highlights generation using decomposed audio information. In: IEEE International Conference on Multimedia & Expo Workshops, ICME Workshops 2019, pp. 579–584 (2019)
9. Khan, A.A., Shao, J.: Spnet: A deep network for broadcast sports video highlight generation. Comput. Electr. Eng. **99**, 107779 (2022)
10. Pan, Z., Li, C.: Robust basketball sports recognition by leveraging motion block estimation. Signal Process. Image Commun. **83**, 115784 (2020)
11. Rekik, G., Khacharem, A., Belkhir, Y., Bali, N., Jarraya, M.: The instructional benefits of dynamic visualizations in the acquisition of basketball tactical actions. J. Comput. Assist. Learn. **35**(1), 74–81 (2019)
12. Cai, J., Tang, X.: RGB video based tennis action recognition using a deep weighted long short-term memory. arXiv:1808.00845 (2018)
13. Ghosh, A., Jawahar, C.V.: Smarttennistv: Automatic indexing of tennis videos. In: Computer Vision, Pattern Recognition, Image Processing, and Graphics - 6th National Conference, NCVPRIPG 2017, pp. 24–33 (2017)
14. Agyeman, R., Muhammad, R., Choi, G.S.: Soccer video summarization using deep learning. In: 2nd IEEE Conference on Multimedia Information Processing and Retrieval, MIPR 2019 (2019)
15. Deng, G., Liu, L., Zuo, J.: Scoring framework of soccer matches using possession trajectory data. In: Proceedings of the ACM Turing Celebration Conference - China, ACM TUR-C 2019, pp. 59–1592 (2019)
16. He, D., Li, L., An, L.: Study on sports volleyball tracking technology based on image processing and 3d space matching. IEEE Access **8**, 94258–94267 (2020)
17. Shingrakhia, H., Patel, H.: Emperor penguin optimized event recognition and summarization for cricket highlight generation. Multimed. Syst. **26**(6), 745–759 (2020)
18. Khan, A.A., Shao, J., Ali, W., Tumrani, S.: Content-aware summarization of broadcast sports videos: An audio-visual feature extraction approach. Neural Process. Lett. **52**(3), 1945–1968 (2020)
19. Yan, C., Li, X., Li, G.: A new action recognition framework for video highlights summarization in sporting events. In: 16th International Conference on Computer Science & Education, ICCSE 2021, pp. 653–666 (2021)

20. Minhas, R.A., Javed, A., Irtaza, A., Mahmood, M.T., Joo, Y.B.: Shot classification of field sports videos using alexnet convolutional neural network. Appl. Sci. **9**(3), 483 (2019)
21. Rafiq, M., Rafiq, G., Agyeman, R., Choi, G.S., Jin, S.: Scene classification for sports video summarization using transfer learning. Sensors **20**(6), 1702 (2020)
22. Sanabria, M., Sherly, Precioso, F., Menguy, T.: A deep architecture for multimodal summarization of soccer games. In: Proceedings Proceedings of the 2nd International Workshop on Multimedia Content Analysis in Sports, MMSports@MM 2019, pp. 16–24 (2019)
23. Turchini, F., Seidenari, L., Galteri, L., Ferracani, A., Becchi, G., Bimbo, A.D.: Flexible automatic football filming and summarization. In: Proceedings Proceedings of the 2nd International Workshop on Multimedia Content Analysis in Sports, MMSports@MM 2019, pp. 108–114 (2019)
24. Datt, M., Mukhopadhyay, J.: Content based video summarization: Finding interesting temporal sequences of frames. In: 2018 IEEE International Conference on Image Processing, ICIP 2018, Athens, Greece, October 7–10, 2018, pp. 1268–1272 (2018)
25. Venkataramanan, A., Laviale, M., Figus, C., Usseglio-Polatera, P., Pradalier, C.: Tackling inter-class similarity and intra-class variance for microscopic image-based classification. In: Computer Vision Systems - 13th International Conference, ICVS 2021, pp. 93–103 (2021)
26. Zalluhoglu, C., Ikizler-Cinbis, N.: Collective sports: A multi-task dataset for collective activity recognition. Image Vis. Comput. **94**, 103870 (2020)
27. Khan, A.A., Tumrani, S., Jiang, C., Shao, J.: RICAPS: residual inception and cascaded capsule network for broadcast sports video classification. In: MMAsia 2020: ACM Multimedia Asia, pp. 43–1437 (2020)
28. Hara, K., Kataoka, H., Satoh, Y.: Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, pp. 6546–6555 (2018)
29. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, pp. 4278–4284 (2017)
30. 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, pp. 2818–2826 (2016)
31. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015 (2015)
32. 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, pp. 770–778 (2016)
33. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, pp. 2261–2269 (2017)
34. 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, pp. 4724–4733 (2017)
35. 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, p. 269 (2019)
36. Tran, D., Bourdev, L.D., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: 2015 IEEE International Conference on Computer Vision, ICCV 2015, pp. 4489–4497 (2015)
37. Donahue, J., Hendricks, L.A., 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. **39**(4), 677–691 (2017)
38. Kalfaoglu, M.E., Kalkan, S., Alatan, A.A.: Late temporal modeling in 3d CNN architectures with BERT for action recognition. In: Computer Vision - ECCV 2020 Workshops, Proceedings, Part V, pp. 731–747 (2020)