J. Vis. Commun. Image R. 60 (2019) 170–179

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

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

Region based multi-stream convolutional neural networks for collective activity recognition ☆

Cemil Zalluhoglu *, Nazli Ikizler-Cinbis

Department of Computer Engineering, Hacettepe University, Ankara, Turkey

ARTICLE INFO

Article history:
Received 6 June 2018
Revised 8 February 2019
Accepted 17 February 2019
Available online 20 February 2019

Keywords:
Collective activity recognition
Action recognition

ABSTRACT

Collective activity recognition, which analyses the behavior of groups of people in videos, is an important goal of video surveillance systems. In this paper, we focus on collective activity recognition problem and propose a new multi-stream convolutional neural network architecture that utilizes information extracted from multiple regions. The proposed method is the first work that uses a multi-stream network and multiple regions in this problem. Various strategies to fuse multiple spatial and temporal streams are explored. We evaluate the proposed method on two benchmark datasets, the Collective Activity Dataset and the Volleyball Dataset. Our experimental results show that the proposed method improves collective activity recognition performance when compared to the state-of-the-art approaches.

1. Introduction

In the last decade, the field of computer vision has witnessed a dramatic increase in research regarding human actions and activities. This increase is mostly due to the proliferation of cameras in our everyday lives and the increase in the number of collected images and video data. Understanding what people are doing is critical for surveillance applications, where automatic labeling of day-long videos is necessary.

Sub-topics of human action recognition are becoming progressively active. However, most of the research in this area is directed toward action recognition and detection of individual humans. The aim of such research is to recognize and/or localize the action of individual persons in isolation. There are many situations where the actions of the individuals are not in an isolated setting. In these situations, the actions of individuals are interconnected, resulting in interactions between each other and/or collective activities. In this context, while human interactions can be categorized as pairwise interactions between human-human, human-object and human-scene, collective activities are identified as group activities, involving more than two people and that have a complex structure. Gathering, walking together, queuing of multiple people can be given as examples of collective human activities.

Due to its nature, collective activity recognition has a more complex underlying structure than individual action recognition. In addition to singleton actions, person-person interaction and group-person interactions are also very important in collective activity recognition. These interactions solve the ambiguity present in the representation of a collective activity. For example, consider a person standing still in a frame, as shown in Fig. 1. If we only consider the person, disregarding the environment, the activity of this person is just standing. However, he/she might as well be talking with other people or waiting in a queue, as it is shown in Fig. 1. In order to understand the collective activity, all the persons and the scene information must be considered together to get the idea of the ongoing collective activity. For this reason, detecting and encoding the interactions are very important for collective activity recognition. Furthermore, there may be cases where not every person in the frame is engaged in the collective activity. They may barely be standing, i.e. not doing anything, or do some other activities. These cases introduce noise to the domain and significantly increase the difficulty of the problem to a great extent. The task is to recognize the collective activity in the presence of such distractors. By using the person region that has the maximum score of a group activity, we include additional information that is based on person regions to the recognition process, besides the existing shape and motion information extracted globally from the whole frame. As the experimental results demonstrate, this local information gives further cues about the collective activities.

In recent years, the interest in collective activity recognition has significantly risen. This problem has been the focus of research [1–4]. These works consider the collective activity recognition prob-

☆ This paper has been recommended for acceptance by Zicheng Liu.

* Corresponding author.
E-mail addresses: cemil@cs.hacettepe.edu.tr (C. Zalluhoglu), nazli@cs.hacettepe.edu.tr (N. Ikizler-Cinbis).

https://doi.org/10.1016/j.jvcir.2019.02.016
1047-3203/

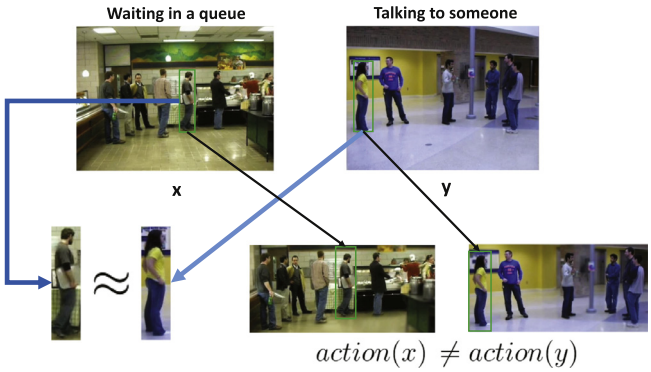

Fig. 1. Illustrating the collective activity recognition problem. While the actions of the individuals can be largely similar, interaction and relation context between individuals makes the collective activities distinctive.

lem by modeling individual person and their relations by using a Long Short Term Memory (LSTM) framework. In contrast, we approach the problem by adopting the successful two-stream convolutional neural network framework that had been originally proposed for action recognition [5], and extending this framework to handle the intrinsic properties inherent to the collective activity recognition problem. Specifically, we propose to use region-based streams in conjunction with the regular RGB and optical flow convolutional neural network(CNN) streams so that the collective structures of activities are more promptly captured.

Our main contributions in this paper are as follows: First, we present a new multi-stream architecture based on person-regions for collective activity recognition. In addition to using the overall image information, our framework analyzes multiple local regions while deciding collective activities. Our study is the first work that uses a multi-stream network and multiple regions in collective activity recognition. Then, we analyze various ways of fusing multiple spatial and temporal streams so that the accuracy of recognition can be improved. Our experimental evaluation on two benchmark datasets demonstrates that our method yields better results compared to the state-of-the-art approaches on collective activity recognition.

The rest of the paper is organized as follows: In Section 2, we first give a brief overview of the related work on human collective activity recognition. Then in Section 3, we introduce the proposed region-based multi-stream CNN approach. Experimental results on two benchmark datasets are presented in Section 4. Finally, in Section 5 we present the conclusions together with possible future research directions.

2. Related work

While there are hundreds of recent works in computer vision literature that try to address the problem of human action recognition (for a recent review on human action recognition, the reader is referred to [6]). The collective activity recognition problem is a relatively newer topic that has been rarely addressed. We should also note that crowd analysis [7], group detection[8] and group activity recognition[9] topics are closely related to the collective activity recognition problem, but due to the variations in problem domains, they are handled separately.

The main approaches to the collective activity recognition problem can be categorized into two main groups. The first approach uses hand-crafted features together with learning frameworks such as graphical models, while the second approach uses the recently introduced powerful deep learning techniques. Our framework follows the second category, where multi-stream CNNs are used. We now review the main works from the above-mentioned approaches.

2.1. Hand-crafted approaches

One of the earliest works in the field of collective activity recognition was presented by Choi et al. [10], where "crowd context" is defined by means of a spatio-temporal descriptor which uses only people's poses and their velocity to extract visual cues about the activities that are performed by individuals in a crowd. Following this work, Lan et al. [11] proposed "action context" which models hierarchical relations from person-level information to group-level interactions using an adaptive latent structure learning algorithm.

In addition to spatio-temporal features, tracking people provides contextual information about the people in a crowd [12,13]. Khamis et al. [13] proposed a unified model which combines frame and track cues for activity recognition. Subsequent works of Choi and Savarese [12,14] merge multiple target tracking and collective activity recognition problems together and propose to use a bottom-up approach. This approach transfers the information about the activity from the estimation of the trajectory of people. First, they obtained the persons pose and atomic actions from the semantic attributes of tracklets then determine the interactions, which are the pairwise relationships between individuals. Each interaction in this method is described by pairs of atomic activities. Finally, they illustrated the collective activity by using collections of interactions.

More recent works attempt to model the relation between the spatio-temporal pattern of persons and their interactions. Antic and Ommer [15] use small semi-local parts extracted from person regions and group related parts based on visual and functional similarities. Max-margin multiple instance learning is then used to classify activities. Tran et al. [16] introduced a top-down approach and represented all detected people in an undirected-weighted graph where each edge describes a social interaction between two people. Similarly, [17] proposed a hierarchical random field (HiRF) to model temporal and frame-wise relations of a video.

Amer et al. [18] detected and localized a wide range of activities by using a three-layer AND-OR Graph model. The AND-OR Graph is a hierarchical model that is recursively defined for effective visual knowledge representation. In this model, the authors use top-down and bottom-up processes together while deciding primitive actions and group activities. In their following work, Amer et al. [19] proposed a spatiotemporal AND-OR Graph. This AND-OR Graph model required a multitude of detectors (object and activity) at different levels. The requirements to apply such approaches are extremely expensive for the problem of collective activity recognition.

Group detection is a problem that is similar to collective activity recognition. Solera et al. [8] built a structural SVM-based framework for the task of group detection. In their proposed model, social features like proxemics and causality were utilized in a supervised hierarchical bottom-up correlation clustering.

2.2. Deep approaches

With the recent development of deep learning architectures, computer vision literature has witnessed significant improvements in a variety of computer vision tasks including image classification [20], human action recognition [5,21–23], and video classification [24,25] to name a few. For action recognition and video classification, early works [5,21] use only the power of Convolutional Neural Networks (CNNs). Some recent studies combine CNNs with recurrent neural network (RNN) models [26]. Another line of approaches uses two-stream convolutional neural networks [5,27,28] by

taking into account the optical flow information together with raw RGB information. Some works [29–31] utilize region information together with two stream approaches, however, these studies use only local information while recognizing the actions.

Various deep learning approaches have been proposed for collective activity recognition [1–4,32,33]. Deng et al. [1] proposed a combination of hierarchical graphical models, where a multi-step message passing approach was used between neural network layers. Following this approach, Deng et al. [32] develop a general framework by integrating graphical models and deep networks with a structure learning. In their study, RNN architecture was used for sequential inference. Hajimirsadeghi et al. [33] introduced Multiple Instance Learning (MIL) approach to recognize activities by embedding cardinality models into a structured kernel, called Cardinality Kernel.

In a recent study, Ibrahim et al. [2] propose a hierarchical model that employed multiple LSTM RNNs. In their framework, the first LSTM is used to recognize individual actions, whereas the other was used to analyze temporal dynamics of the group activities. Shu et al. [3] extended the existing two-stage hierarchical LSTM model of [2] by using an energy layer instead of a common softmax layer. Li and Chuah [4] proposed a semantics-based method that generated a caption for each of the video frames and then recognized the collective activities based on these semantic captions for each video with the two-stage LSTM model. Bagautdinov et al. [34] introduce a unified framework for understanding multi-person social behaviors. Their architecture jointly detects multiple people, infers their social actions, and predicts the collective activities with a single pass through a neural network.

The aforementioned techniques, especially those that are based on deep learning methods, achieve good recognition performance in collective activity recognition. Nevertheless, all studies use the video frames as a whole, while deep learning approaches have focused on the temporal relations of people. In our work, we analyze multiple regions of the images in various ways to capture the spatial relations between people in the group, together with various fusion strategies to merge information coming from multiple channels. Our results demonstrate that this region-based strategy is an effective solution for the collective activity recognition problem.

3. Our approach

In the collective activity recognition problem, multiple people are involved in an activity. Unlike the large body of work that focuses on the activities of a single person, the multiple people regions and their activities should be taken into account. Therefore, apart from analyzing the whole image, we must also consider multiple subregions. We do this by defining multiple CNN streams based on regions.

We propose a multi-stream CNN architecture that uses individual frame RGB and optical flow information together with regional features. This proposed architecture extends the two-stream architecture proposed in [5], such that multiple region information is utilized in an additional stream. In this section, we first review the basic two-stream model of Simonyan and Zisserman [5], and then present our extensions over this model.

In [5], Simonyan and Zisserman propose a two-stream CNN architecture for the purpose of individual action recognition. This architecture utilizes two individual CNNs, namely the spatial and the temporal streams. Basically, the spatial stream operates on RGB input and is intended to carry shape information about the scene. The temporal stream operates over the optical flow input between consecutive frames and carries the motion information. The inputs are passed through standard convolutional, pooling and fully connected layers. Their original network architecture contains 8 layers, 5 of which are convolutional, and the last 3 are fully-connected. The spatial and temporal streams have exactly same layer configurations. In this method, the softmax loss is used as the loss function.

Adapting this idea of using more than one stream of convolutional neural network(CNN) to the problem of collective activity recognition, our proposed architecture has four independent CNN streams, which are referred to as *spatialCNN* for spatial stream, *motionCNN* for optical flow stream extracted from the whole frame, *spatialRCNN* (sRCNN) for RGB information extracted from regions of interest and finally *motionRCNN* (tRCNN) for representing optical flow information extracted from RoIs. The overall proposed method is shown in Fig. 2. The two streams, *i.e.* spatial (top) and temporal (bottom) streams are identical to the two-stream architecture [5], with the exception that we adopt a deeper network architecture, VGG_16 [35]. VGG_16 has 13 convolutional layers, as compared to the 5 convolutional layers of [5]. The spatial stream is intended to extract information from the general scene and holistic shape information, whereas the temporal stream operating over the whole frame extracts information about the motion flow of the whole scene. Our intuition is that paying a closer attention to person regions will be beneficial for collective activity recognition. In this way, our aim is to capture more fine-grained shape and temporal information from the person regions. To this end, we extend the two-stream architecture with two additional streams that operate over person detection regions. Below, we describe each of these additional streams in detail.

3.1. Spatial Region Stream – sRCNN

Spatial Region Stream CNN (sRCNN for short) is a spatial CNN stream that takes individual RGB video frames as input. Different from the spatial stream of [5], it operates over individual person regions, as opposed to whole frames.

Specifically, the initial layers of sRCNN are convolutional layers, same as the original CNN architecture that has been proposed for image classification in [35]. Unlike their standard spatial stream that uses max-pooling after the last convolutional layer, we use a Region-Of-Interest (RoI) pooling layer [36]. For RoIs, we use person detection bounding boxes that are acquired by running the person detector of [36]. Then, the RoI pooling layer extracts a fixed-length feature vector using the output of the convolutional layers on each RoI. Each fixed-length feature vector is then fed into a sequence of fully-connected layers. A softmax layer estimates the classification score for each region. After class scores are computed for each person region, max-operation is applied to select the region with the maximum score.

The RoI Pooling Layer [36] is a type of pooling layer which performs max pooling on feature maps of non-uniform sizes and produces a smaller feature map with a fixed spatial extent of width and height, where height H and width W are predefined network hyper-parameters ($H = W = 7$ for VGG architecture). The RoI is a rectangular window within a convolutional feature map. It can be defined either as (l, c, h, w), where (l, c) is the top-left corner and (h, w) are height and weight. The RoI can also be defined as $(X_{min}, Y_{min}, X_{max}, Y_{max})$ which represent the X and Y coordinates of the top-left and bottom-right corners respectively. The ROI pooling layer divides the $h \times w$ ROIs as H $\times$ W sub-windows, after this operation max-pool each sub-window to get H $\times$ W map to represent the ROIs. In our study, the RoI pooling layer inputs are the convolutional feature map which is produced as the output of the last convolutional layer (Conv5_3) and the bounding box locations of all person detections. The output of this layer is a fixed-size feature

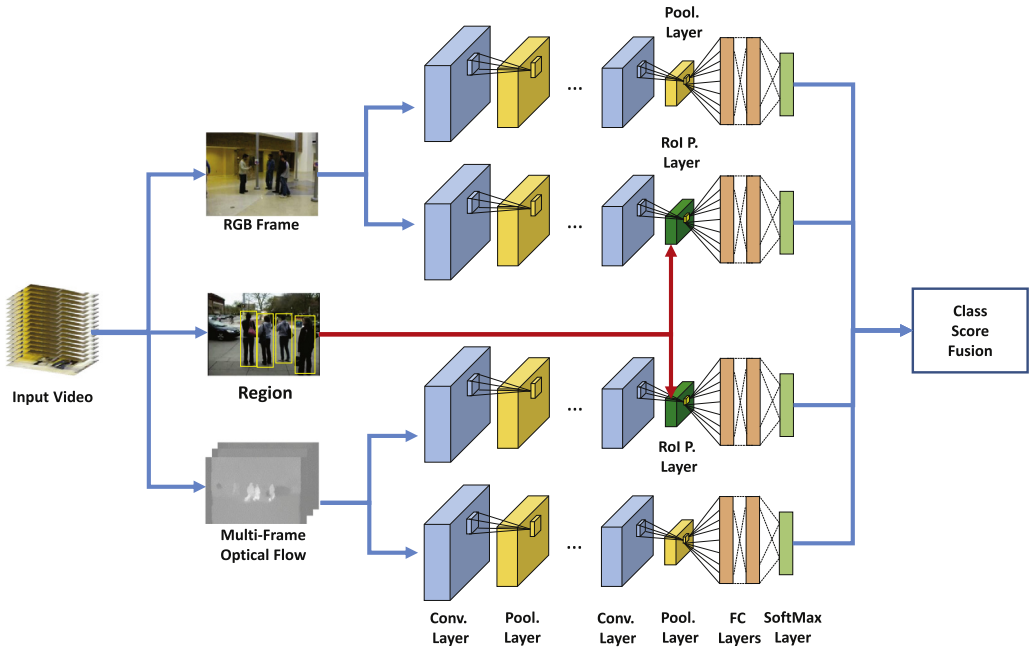

Fig. 2. Our multi region-based multi-stream architecture. Our proposed method has four CNN streams which are from top to the bottom Spatial CNN, Spatial Region CNN, Temporal Region CNN and Temporal CNN.

map for each bounding box. The process of RoI pooling layer is shown in Fig. 3. The operation is as follows in Eq. (1)

$$y_{(r,j)} = x_{i^*(r,j)} \tag{1}$$

in which $i^*(r,j) = argmax_{i^* \in R(r,j)} x_{i^*}$ where x_i is the i-th activation map into the RoI pooling layer and $y_{(r,j)}$ is the j-th output of the r-th RoI. $R(r,j)$ is a set of inputs in the subwindow of the $y_{(r,j)}$ max pools. The RoI pooling layer uses an adaptive max pooling strategy to obtain a fixed sized representation for a variable sized region.

Following [36], we use the function in Eq. (3) to back-propagate through RoI pooling layer. This function computes the partial derivatives of the loss function L given in Eq. (2), which is a standard softmax loss function where o_j is the output class score of the each region.

$$L = -\sum_j y_j \, log\left(\frac{e^{o_j}}{\sum_i e^{o_i}}\right) \tag{2}$$

$$\frac{\partial L}{\partial x_i} = \sum_r \sum_j [i = i^*(r,j)] \frac{\partial L}{\partial y_{(r,j)}} \tag{3}$$

For each pooling output unit of the pooling layer ($y_{(r,j)}$), the gradients are computed and passed to the feature map with the selected activation map which has the maximum score during the forward pass. After the pooling operation, each fixed sized feature map passes through the fully connected (FC) layers and is then classified into one of the classes, as in Fast RCNN [36]. The difference is that while predicting the class label of the video, we use image regions extracted from consecutive frames in the video.

The usage of multiple regions in this manner is similar to the Multiple instance learning (MIL) paradigm [37] that is used frequently in computer vision. MIL provides a framework for training models when full supervision is not available during training. In our study, we choose the bounding box label with the highest score to decide the frame label. This decision making is quite similar to MIL in the sense that we do not have annotations for the bounding boxes, yet, we can leverage regional information by selecting from a bag of regions.

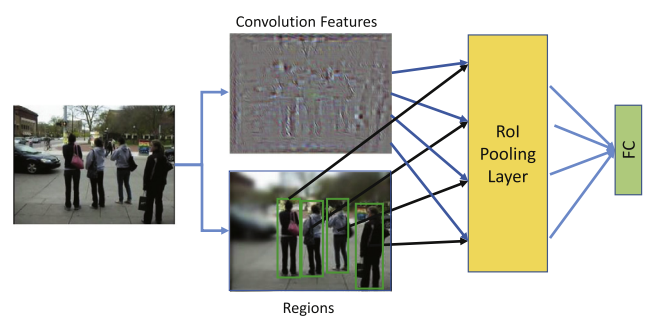

Fig. 3. Person detection region information is fed to the RoI pooling layer together with the CNN features.

3.2. Temporal Region Stream – tRCNN

In the temporal region stream, the network takes stacked optical flows as input instead of the RGB frames. Dense optical flows, which can be seen as a set of displacement vector fields between consecutive frames, are computed by using the method of Brox et al. [38]. The horizontal and vertical components of the displacement vector fields are stored as two optical flow images for a given pair of consecutive frames. Then these components are stacked together for a length of L consecutive frames. The network architecture and training processes are almost identical to the sRCNN, except the input type and the number of channels of the input image. While for the spatial stream the input is $I \in \mathbb{R}^{w \times h \times 3}$, for the temporal stream, it is $I \in \mathbb{R}^{w \times h \times 2L}$. where L is the number of consecutive frames. Similar to sRCNN spatial stream, we replace the last pooling layer with the RoI pooling layer and operate over the person detection regions of the optical flow images. The rest of the training process is the same as the sRCNN stream.

3.3. Fusion strategies

In this section we consider different kinds of fusion strategies for combining the information coming from multiple streams that are identified above. Specifically, we consider *spatial* and *temporal* fusion methods. All streams, including region-based streams, are fused with these fusion methods. We have used similar spatial (sum, max, concatenation) and temporal fusion strategies with [28]. Actually, the spatial fusion methods are used in many studies in the same way. The main difference from [28] is our hybrid fusion method which has been shown to yield effective results in the experiments. Below, we give the details of these fusion strategies.

3.3.1. Spatial fusion

Our intention is to fuse the spatial network and motion context information at the level of the spatial location so that the channel responses at a given pixel position corresponding to each other. For all streams including region based streams, every feature map that is produced by each layer have exact spatial location correlation with other networks due to the networks having the same structure.

A fusion function $f : x_n^s, x_n^t \rightarrow y_n$fuses two feature maps $x_n^s \in \mathbb{R}^{H \times W \times D}$and $x_n^t \in \mathbb{R}^{H' \times W' \times D'}$, spatial and temporal feature maps respectively, at time n, to produce an output map $y_n \in \mathbb{R}^{H'' \times W'' \times D''}$ where $H \rightarrow$ height, $W \rightarrow$ width, and $D \rightarrow$ the number of channels. For simplicity, we assume that $H = H' = H'', W = W' = W''$, $D = D'$. We now discuss different fusion functions:

Sum Fusion: Sum fusion $y^{sum} = f^{sum}(x^s, x^t)$ computes the sum of two feature maps at the same pixel location (i,j) and the same feature channel d:

$$y_{i,j,d}^{sum} = x_{i,j,d}^s + x_{i,j,d}^t \tag{4}$$

where $1 \leqslant i \leqslant H, 1 \leqslant j \leqslant W, 1 \leqslant d \leqslant D$ and $x^a, x^b, y \in \mathbb{R}^{H \times W \times D}$.

Max Fusion: Max fusion $y^{max} = f^{max}(x^s, x^t)$ takes the maximum of the two feature maps:

$$y_{i,j,d}^{max} = max\{x_{i,j,d}^s, x_{i,j,d}^t\} \tag{5}$$

where other variables are the same as the above (4).

Concatenation fusion: Concatenation fusion $y^{cat} = f^{cat}(x^s, x^t)$ stacks the two feature maps at the same pixel location (i,j) across the feature channel d:

$$y_{i,j,2d}^{cat} = [x_{i,j,d}^s \; x_{i,j,d}^t] \tag{6}$$

where $y \in \mathbb{R}^{H \times W \times D_c}, D_c = D + D'$

While the previous fusion methods fused the networks with the same spatial location, in this method does not require to define any correspondence between the networks.

3.3.2. Temporal fusion

In temporal fusion, we combine feature maps x_t over time t, to get an output map y_t (as in [28]). We now consider the input to a temporal pooling layer to be feature maps which are generated by stacking spatial maps across time $t = 1 \ldots T$.

3D Conv + 3D Pooling: 3D Pooling is a simple extension of 2D pooling to the temporal domain. 3D pooling applies max-pooling to the stacked data. In general $3 \times 3 \times 3$ max-pooling is used across the stacked corresponding channels. There is no pooling across different channels.

In this technique 3D pooling is performed after a convolution operation. In that operation, the input is four-dimensional, the filter is $\mathbf{f} \in \mathbb{R}^{W'' \times H'' \times T'' \times D \times D'}$and bias is $\mathbf{b} \in \mathbb{R}^D$.

$$y = x_t * \mathbf{f} + \mathbf{b} \tag{7}$$

This convolution operation is a 3D convolution operation which convolves 3D kernels into a cube formed by stacking multiple adjacent frames together.

3.3.3. Where to fuse the streams

In our work, we use three different strategies for fusing the networks: early fusion, late fusion and hybrid fusion.

Early fusion: The early fusion method, also referred to as feature level fusion, unifies the extracted features from different streams by integrating them into a single stream for training. The way the stream outputs are fused are shown in Fig. 4a. In our study, we fuse networks after the convolution layers. Both temporal and spatial fusion strategies can be used in this method.

Late fusion: Simple late fusion was adopted to combine the softmax scores of two networks by either averaging or using a linear classifier as shown in Fig. 4b. This fusion method is also called decision level fusion or semantic level fusion. This method has been widely used for image and video analysis. Note that, we can use only basic mathematical operations when doing late fusion, since we are operating on softmax scores. To this end, we use spatial fusion techniques identified above.

Hybrid fusion: Hybrid fusion is basically a combination of early and late fusion strategies. In the beginning, the streams are fused using temporal fusion at the last convolutional layer (after ReLU). Then, unlike the previous works which use early fusion, we do not end the temporal stream; this stream continues with fully connected layers, as well as the fused spatiotemporal stream. This process is illustrated in Fig. 5. For the classification decision, we fuse these streams with spatial max fusion at the end just after the softMax layer.

4. Experimental evaluation

In this section, we present the experimental evaluations of our proposed multi-stream approach for collective activity recognition.

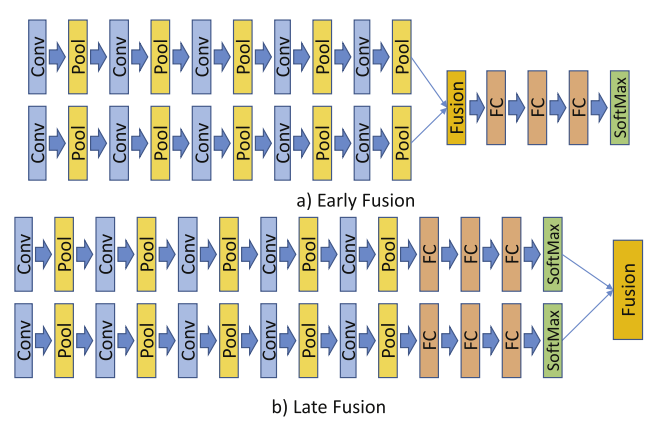

Fig. 4. Different fusion strategies. (a) Early Fusion done after the last pooling layer, (b) Late Fusion is done after the softMax layer.

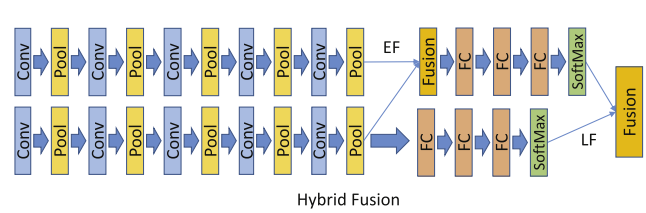

Fig. 5. Hybrid fusion (HF) is a combination of early and late fusion strategies. In the early fusion (EF) phase of HF, streams are fused after the last pooling layer. Besides this fusion, the temporal stream also continues, and later, it is fused with the spatio-temporal stream of EF part after the softMax layer (shown with LF).

Fig. 6. Sample images on Collective Activity Dataset [10].

To this end, we first present the details of the benchmark datasets that are used in the experiments. Then, we present the implementation details and experimental outcomes.

4.1. Dataset

4.1.1. Collective activity dataset [10]

This dataset contains 44 short video sequences, where the videos are 640 × 480 pixels in size and are recorded by a consumer hand-held digital cameras with varying viewpoints. Fig. 6 illustrates sample images from this dataset. There are five collective activity categories in the dataset. These categories are crossing, walking, waiting, talking, and queuing. Every 10th frame in all video sequences were annotated with the locations of people with bounding boxes and an activity label. Each frame is assigned a collective activity label based on the activity of the majority of people. We follow the train/test split provided by Lan et al. [11].

4.1.2. Volleyball dataset [2]

In order to evaluate our method's performance, we also conduct experiments using the more recent Volleyball dataset [2]. This dataset contains 1525 annotated frames that are handpicked from 15 videos with seven player action labels and six team (group) activity labels. The group activities are spiking, setting, passing and the left/right team variants. We use frames from $2/3^{rd}$ of the videos for training, and the remaining $1/3^{rd}$ for testing, following the same setup as in [2].

4.2. Implementation details

In our convolutional streams, we adopt the framework of VGG-16 model [35] that has 13 convolutional and 3 fully-connected layers. Since training network streams from scratch requires an extensive amount of training data and we are very short of such a large scale data for collective activity problem, we use the pre-trained convolutional streams that are originally trained on UCF101 [39] for action recognition [5]. First, we train all streams separately with the same momentum of 0.9 and a weight decay of 0.0005. For training the spatial networks, we use dropout ratios of 0.85 for the first fully-connected layers and the batch size is 256. The learning rate starts from 10^{-2} which is reduced by a factor of 10 as soon as the validation accuracy saturates. For temporal networks, we use optical flow stacking with $L = 10$ frames, the dropout ratio is 0.8, the batch-size is 128 and the learning rate starts from 10^{-3} and is reduced in the same way as in the spatial network. We pre-compute the optical flows before training. In all streams, we use standard softMax loss function given in Eq. (2).

For fusion, the same parameters as the temporal networks are used except for dropout ratios. A dropout ratio of 0.85 is used as in spatial networks.

Table 1
Comparison of the single streams on Collective Activity Dataset. The reported performance measure is per video accuracy.

Architecture	Method	Accuracy
Single stream	Spatial (S)	55.6
	Temporal (T)	77.8
	SpatialRCNN (sRCNN)	61.1
	TemporalRCNN (tRCNN)	77.8

4.3. Results and discussions

4.3.1. Experiments on collective activity dataset

In the following, we first evaluate the base performance of the single streams that we propose. Table 1 shows this evaluation. From these results, we observe that spatial streams when used in isolation yields comparatively worse results than the temporal streams (with or without RoI). Using the RoI pooling layer has made a significant contribution to the results, especially in the spatial stream.

In order to evaluate the effect of the noise of the person detection scheme and its effect on or SRCNN method, we also carry out an experiment using ground truth bounding boxes provided with the dataset. In this experiment, since only every 10th frame is annotated with ground truth person bounding boxes, we use the same subset of the frames. We observe that sRCNN method over the automated person detections yields an accuracy of 38.9%, whereas using groundtruth bounding boxes, it yields an accuracy of 44.4%. This shows that our sRCNN method is likely to benefit further from correct person region detections. Note that this experiment is run using only every 10th frame, not the whole videos, so the accuracies are lower than the case when the whole videos are used.

In Table 2, different late and early fusion strategies are compared. As it can be observed, the late fusion of streams does not yield any improvement of the results, compared to single stream results of Table 1. In late fusion, the best results are achieved when using region-based streams and temporal region stream (tRCNN) produces the best results whether it is used in isolation or in conjunction with other streams via late fusion. We have also looked at the late fusion of multiple streams and even using multiple streams in this fusion does not affect the overall best accuracy of 77.8%. On the other hand, compared to late fusion, early fusion yields better results, especially using concatenation with spatial

Table 2
Comparison of late and early fusion architectures using different strategies on Collective Activity Dataset. The best accuracies are shown in bold.

Architecture	Method	Fusion type	Accuracy
Late fusion (LF)	S + T	Max	66.7
	S + T	Sum	72.2
	sRCNN + tRCNN	Max	77.8
	sRCNN + tRCNN	Sum	77.8
	S + T + sRCNN	Max	66.7
	S + T + sRCNN	Sum	72.2
	S + T + tRCNN	Max	72.2
	S + T + tRCNN	Sum	77.8
	S + T + sRCNN + tRCNN	Max	77.8
	S + T + sRCNN + tRCNN	Sum	77.8
Early fusion (EF)	S + T	Max	66.7
	S + T	Sum	72.2
	S + T	Concat	77.8
	sRCNN + tRCNN	Max	77.8
	sRCNN + tRCNN	Sum	77.8
	sRCNN + tRCNN	Concat	**83.3**
	S + T	3DConv + 3DPool	77.8
	sRCNN + tRCNN	3DConv + 3DPool	**83.3**

Table 3
Comparison of hybrid fusion architecture with different fusion architectures in Collective Activity Dataset. The best accuracies are shown in bold.

Architecture	Method	Accuracy
Late fusion	sRCNN + tRCNN	77.8
Early fusion	sRCNN + tRCNN	83.3
Hybrid fusion	(S + T) (HF) + sRCNN (LF^{sum})	83.3
	(S + T) (HF) + tRCNN (LF^{sum})	**88.9**
	(S + T) (HF) + sRCNN + tRCNN (LF^{sum})	**88.9**

pooling or using temporal pooling. We also observe that for spatial pooling, using concatenation yields a significant improvement in the results (more than 5% improvement in accuracy) compared to late fusion. We have also observed that temporal pooling is effective when used with region-based streams. In Table 2, we have tested different fusion types (sum, max and concat) with early and late fusion architectures. From those results, we observe that 3DConv + 3DPool works best for early fusion (EF) and sum fusion type for late fusion (LF) architectures. Therefore, in the further experiments, we utilize these best performing fusion types when EF or LF architectures are used, respectively.

Table 3 shows the comparison of several architectures that involve hybrid fusion with respect to the late and early fusion. From these results, it can be observed that the proposed hybrid fusion architecture has further improved the results. The best result is obtained when spatial and temporal streams are fused with hybrid fusion architecture and further fused with tRCNN stream by late fusion, shown by `(S + T) (HF) + tRCNN (LF)`. A further combination that we have tried in our experiments is the addition of sRCNN with late fusion method to this architecture. Both of these combinations have shown a remarkable improvement, achieving an accuracy of 88.9%.

Table 4
Comparison of our method with the related work on Collective Activity Dataset. The best accuracies are shown in bold.

Method	Accuracy
Contextual model [40]	79.1
Deep structured model [1]	80.6
Two-stage hierarchical model [2]	81.5
Cardinality model [33]	83.4
SBGAR [4]	86.1
CERN [3]	88.3
Our method	**88.9**

Table 5
The performance comparison of single streams and other fusions on Volleyball dataset. The best accuracies are shown in bold.

Architecture	Method	Accuracy
Single	Spatial (S)	42.5
	Temporal (T)	51.4
	SpatialRCNN (sRCNN)	47.8
	TemporalRCNN (tRCNN)	**64.3**
Late fusion (LF)	S + T	57.6
	sRCNN + tRCNN	**68.3**
Early fusion (EF)	S + T	61.2
	sRCNN + tRCNN	**70.2**
Hybrid fusion (HF)	(S + T) (HF) + tRCNN (LF^{sum})	**72.4**

We compare our proposed approach with the previous state-of-the-art methods for collective activity recognition and Table 4 summarizes the results. For a fair comparison, we use the same train/test split provided by Lan et al. [40] in this comparison. As shown in Table 4, our model outperforms the state-of-the-art methods on the Collective Activity Dataset. In this dataset, our hybrid fusion architecture that utilizes region based streams achieves an accuracy of 88.9%, which is 0.6 % higher than the recent CERN model [3] that is based on a multi-level hierarchy of LSTMs with energy layer and outperforms the Two-Stage Hierarchical Model [2] by a margin of 7.4%. Here, we should note that CERN [3] model has fewer parameters to estimate; our model has $\approx$ 280 Mparameters, whereas CERN[3] model has $\approx$ 210 M parameters.

Some example qualitative results from Collective Activity dataset are given in Fig. 7. These examples show that when the regions are detected correctly, our method gives good results. In this dataset, we observe that when the people are seen from a distance, the person detection may fail to find accurate regions for people, and this may adversely affect the classification performance. In addition, some of the individual actions (such as those actions in waiting and queuing, walking and crossing) are very similar in nature and this may cause misclassification of the corresponding collective activities.

4.3.2. Experiments on volleyball dataset

In Table 5, we present the experimental results on Volleyball dataset [2]. The first part of Table 5 indicates the performance of single stream approaches, namely spatial stream, temporal stream, sRCNN stream with Fast-RCNN results, sRCNN stream with ground truth information of people and tRCNN stream. When we look at

Fig. 7. The qualitative results of the our models on Collective Activity Dataset with person detection results. Green labels are correct classifications and red ones are incorrect. Here, the first row shows *Crossing*, the second row shows *Waiting* and the last row shows *Walking* examples.

the results, a similar pattern is observed in this dataset; the temporal streams yield better recognition performance in general. The spatial streams do not perform as well as the temporal streams, mostly due to the similarity of background context in this dataset, *i.e.* the volleyball videos are acquired in similar settings, where the viewpoints and the contexts are mostly stable.

Table 6
Comparison with related works on Volleyball Dataset. We report per video accuracy on this dataset. The best accuracies are shown in bold.

Method	Accuracy
Two-stage hierarchical model [2]	51.1
SBGAR [4]	66.9
Our method	**72.4**

Fig. 8. Confusion matrix for the Volleyball dataset using hybrid fusion architecture $(S+T)(HF)+tRCNN(LF^{sum})$.

Regarding the experiment with detected person regions versus ground truth bounding boxes, we perform a similar experiment on Volleyball dataset: sRCNN method on detected person regions yields an accuracy of 47.8% whereas, on ground truth bounding boxes, the performance raises up to 50.2%. The results show that if we have perfectly detected all the people on the videos, the accuracy would be increased by 2.4%. This observation also confirms that there is room for improvement for our method with better person detection techniques.

We also observe that the proposed region-based streams yield significant improvements over the standard CNN streams. tRCNN stream achieves a performance improvement $\approx 13\%$ over the regular temporal CNN stream. This is a notable improvement, showing the potential of region-based processing as opposed to working on full frames. Working on regions provides important cues about the ongoing collective activities.

In the second part of Table 5, we present the late and early fusion results of these CNN stream architectures. We use sum fusion method in the late fusion and 3D Conv + 3D Pooling method in the early fusion. As can be seen in the obtained results, the fusion of region-based streams yields a significant improvement when compared to single streams in this dataset. Compared to the late fusion, early fusion yields better results in both stream types. Moreover, we observe that the proposed hybrid fusion architecture improves the accuracy further, as in the case of Collective Activity dataset.

Then, we compare our region-based approach with the previous state-of-the-art results on Volleyball dataset and the results are

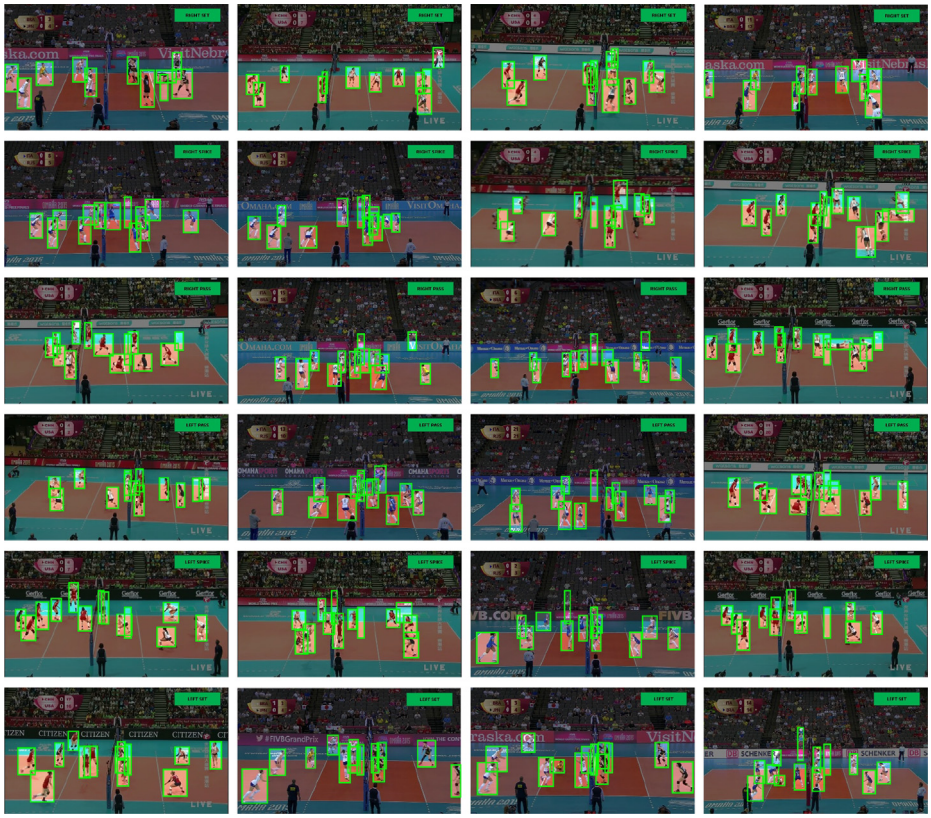

Fig. 9. The correct classification results of our models on Volleyball Dataset with person detection results. Here, the first row shows *Right Set*, the second row shows *Right Spike*, the third row shows *Right Pass*, the fourth row shows *Left Pass*, the fifth row shows *Left Spike* and the last row shows *Left Set* examples.

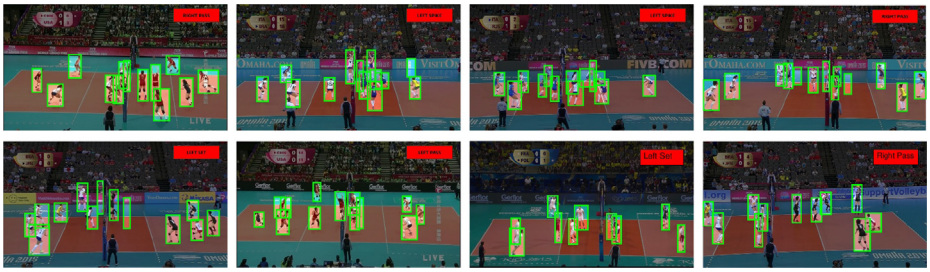

Fig. 10. Some misclassification cases of our hybrid fusion model on Volleyball Dataset with overlaid person detection results. The correct classes for these example frames are *Right Spike*, *Right Pass*, *Left Pass*, *Left Spike* in first row, and *Left Spike*, *Left Set*, *Left Pass*, *Right Set* in second row, respectively.

listed in Table 6. Our method performs better than the previous work, 21.3 % performance increase compared to [2] and 5.5 % accuracy increase compared to [4]. Shu et al. [3] does not report any result on this dataset. Fig. 8 shows the confusion matrix obtained on the volleyball dataset using our best performing hybrid method $(S+T)(HF)+tRCNN(LF^{sum})$. From this confusion matrix, we see that most of the confusion occurs between `set` and `pass` activities and actually, these activities are indeed very similar to each other.

Some qualitative results, where the frames of correctly classified and misclassified videos on Volleyball dataset are shown in Figs. 9 and 10. When we look at these visual examples, we see that person detection works reasonably well in this dataset, even though the person regions are relatively small. On the other hand, some of the confusions occur between categories that are likely to cooccur or follow each other such as `pass` and `spike`, or `pass` and `set`.

5. Conclusion

In this paper, we tackle the problem of collective activity recognition in videos. To this end, we propose a novel multi-stream spatio-temporal architecture with a convolutional fusion. We propose to use RoI pooling layer and form separate *region-based* streams that operate over spatial and temporal information. The proposed architecture fuse these region-based convolutional streams with standard spatio-temporal streams in several ways. The experimental results demonstrate that fusing streams in the early phases of the process and using RoIs produce effective results for recognizing the activities. Compared to the existing state-of-the-art approaches, our proposed approach yields promising recognition accuracies on two benchmark datasets.

Currently, our model does not have any explicit modeling of sequential information within frames, other than the short-term 3D temporal convolutions. In the future, we plan to extend the proposed multi-stream architecture using RNN and LSTM models for capturing the temporal relationships of the sequences.

Acknowledgment

This work was supported in part by the Scientific and Technological Research Council of Turkey (TUBITAK) Research Program (1001), Project No: 116E102.

References

[1] Z. Deng, M. Zhai, L. Chen, Y. Liu, S. Muralidharan, M. Roshtkhari,, G. Mori, Deep structured models for group activity recognition, in: British Machine Vis. Conf. (BMVC), 2015.
[2] M.S. Ibrahim, S. Muralidharan, Z. Deng, A. Vahdat, G. Mori, A hierarchical deep temporal model for group activity recognition., in: Proc. IEEE Comput Soc Conf Comput Vis Pattern Recognit., 2016.
[3] T. Shu, S. Todorovic, S.-C. Zhu, CERN: confidence-energy recurrent network for group activity recognition, in: Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR), 2017.
[4] X. Li, M.C. Chuah, SBGAR: semantics based group activity recognition, in: IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2876–2885.
[5] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: Adv. Neural. Inf. Process. Syst., 2014, pp. 568–576.
[6] S. Herath, M. Harandi, F. Porikli, Going deeper into action recognition: a survey, Image Vis. Comput. 60 (2017) 4–21.
[7] W. Lin, Y. Mi, W. Wang, J. Wu, J. Wang, T. Mei, A diffusion and clustering-based approach for finding coherent motions and understanding crowd scenes, IEEE Trans. Image Process. 25 (4) (2016) 1674–1687.
[8] F. Solera, S. Calderara, R. Cucchiara, Structured learning for detection of social groups in crowd, in: 2013 10th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), IEEE, 2013, pp. 7–12.
[9] W. Lin, H. Chu, J. Wu, B. Sheng, Z. Chen, A heat-map-based algorithm for recognizing group activities in videos, IEEE Trans. Circ. Syst. Video Technol. 23 (11) (2013) 1980–1992.
[10] W. Choi, K. Shahid, S. Savarese, What are they doing? Collective activity classification using spatio-temporal relationship among people, in: EEE Int. Conf. Comput. Vis. Workshops (ICCV Workshops), IEEE, 2009, pp. 1282–1289.
[11] T. Lan, Y. Wang, W. Yang, G. Mori, Beyond actions: discriminative models for contextual group activities, in: Adv. Neural. Inf. Process. Syst., springer, 2010, pp. 1216–1224.
[12] W. Choi, S. Savarese, A unified framework for multi-target tracking and collective activity recognition, in: European Conference on Computer Vision, Springer, 2012, pp. 215–230.
[13] S. Khamis, V.I. Morariu, L.S. Davis, Combining per-frame and per-track cues for multi-person action recognition, in: European Conference on Computer Vision, Springer, 2012, pp. 116–129.
[14] W. Choi, S. Savarese, Understanding collective activities of people from videos, IEEE Trans. Pattern Anal. Mach. Intell 36 (6) (2014) 1242–1257.
[15] B. Antic, B. Ommer, Learning latent constituents for recognition of group activities in video, in: European Conference on Computer Vision, Springer, 2014, pp. 33–47.
[16] K. Tran, A. Gala, I. Kakadiaris, S. Shah, Activity analysis in crowded environments using social cues for group discovery and human interaction modeling, Patt. Recognit. Lett. 44 (2014) 49–57.
[17] M.R. Amer, P. Lei, S. Todorovic, Hirf: Hierarchical random field for collective activity recognition in videos, in: European Conference on Computer Vision, Springer, 2014, pp. 572–585.
[18] M.R. Amer, D. Xie, M. Zhao, S. Todorovic, S.-C. Zhu, Cost-sensitive top-down/bottom-up inference for multiscale activity recognition, in: European Conference on Computer Vision, Springer, 2012, pp. 187–200.
[19] M.R. Amer, S. Todorovic, A. Fern, S.-C. Zhu, Monte carlo tree search for scheduling activity recognition, in: Proc. IEEE Int. Conf. Comput. Vis (ICCV), 2013, pp. 1353–1360.
[20] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in: F. Pereira, C.J.C. Burges, L. Bottou, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 25, Curran Associates, Inc., 2012, pp. 1097–1105.
[21] S. Ji, W. Xu, M. Yang, K. Yu, 3D convolutional neural networks for human action recognition, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2013) 221–231.
[22] W. Lin, Y. Mi, J. Wu, K. Lu, H. Xiong, Action Recognition with Coarse-to-Fine Deep Feature Integration and Asynchronous Fusion, arXiv preprint arXiv:1711.07430.
[23] D. He, F. Li, Q. Zhao, X. Long, Y. Fu, S. Wen, Exploiting spatial-temporal modelling and multi-modal fusion for human action recognition, arXiv preprint arXiv:1806.10319.
[24] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-scale video classification with convolutional neural networks, in: Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern. Recognit., 2014.
[25] S. Xie, C. Sun, J. Huang, Z. Tu, K. Murphy, Rethinking spatiotemporal feature learning: speed-accuracy trade-offs in video classification, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 305–321.

[26] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, T. Darrell, Long-term recurrent convolutional networks for visual recognition and description, in: Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR), 2015, pp. 2625–2634.
[27] L. Wang, Y. Qiao, X. Tang, Action recognition with trajectory-pooled deep-convolutional descriptors, in: Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 2015, pp. 4305–4314.
[28] C. Feichtenhofer, A. Pinz, A. Zisserman, Convolutional two-stream network fusion for video action recognition, in: Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 2016, pp. 1933–1941.
[29] X. Peng, C. Schmid, Multi-region two-stream R-CNN for action detection, in: European Conference on Computer Vision, Springer, 2016, pp. 744–759.
[30] G. Singh, S. Saha, M. Sapienza, P. Torr, F. Cuzzolin, Online real-time multiple spatiotemporal action localisation and prediction, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3637–3646.
[31] S. Saha, G. Singh, M. Sapienza, P.H. Torr, F. Cuzzolin, Deep learning for detecting multiple space-time action tubes in videos, arXiv preprint arXiv:1608.01529.
[32] Z. Deng, A. Vahdat, H. Hu, G. Mori, Structure inference machines: recurrent neural networks for analyzing relations in group activity recognition, in: Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 4772–4781.
[33] H. Hajimirsadeghi, W. Yan, A. Vahdat, G. Mori, Visual recognition by counting instances: a multi-instance cardinality potential kernel, in: Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR), 2015, pp. 2596–2605.
[34] T. Bagautdinov, A. Alahi, F. Fleuret, P. Fua, S. Savarese, Social scene understanding: end-to-end multi-person action localization and collective activity recognition, in: Conference on Computer Vision and Pattern Recognition, vol. 2, 2017.
[35] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: International Conference on Learning Representations, 2015.
[36] R. Girshick, Fast R-CNN, in: IEEE International Conference on Computer Vision (ICCV), 2015.
[37] O. Maron, T. Lozano-Pérez, A framework for multiple-instance learning, in: Adv. Neur. Inform. Process. Syst., 1998, pp. 570–576.
[38] T. Brox, A. Bruhn, N. Papenberg, J. Weickert, High accuracy optical flow estimation based on a theory for warping, in: European Conference on Computer Vision, Springer, 2004, pp. 25–36.
[39] K. Soomro, A.R. Zamir, M. Shah, UCF101: A dataset of 101 human actions classes from videos in the wild, arXiv preprint arXiv:1212.0402.
[40] T. Lan, Y. Wang, W. Yang, S.N. Robinovitch, G. Mori, Discriminative latent models for recognizing contextual group activities, IEEE Trans. Pattern Anal. Mach. Intell 34 (8) (2012) 1549–1562.