IET Computer Vision

Research Article

IET Journals
The Institution of Engineering and Technology

Fully convolutional networks for action recognition

ISSN 1751-9632
Received on 3rd January 2017
Revised 7th June 2017
Accepted on 1st July 2017
E-First on 2nd August 2017
doi: 10.1049/iet-cvi.2017.0005
www.ietdl.org

Sheng Yu[1,2,3], Yun Cheng[2], Li Xie[2], Shao-Zi Li[1,3] ✉

[1]*Cognitive Science Department, Xiamen University, Xiamen, Fujian, People's Republic of China*
[2]*School of Information, Hunan University of Humanities, Science and Technology, Loudi, Hunan, People's Republic of China*
[3]*Fujian Key Laboratory of the Brain-like Intelligent Systems, Xiamen, Fujian, People's Republic of China*
✉ *E-mail: szlig@xmu.edu.cn*

Abstract: Human action recognition is an important and challenging topic in computer vision. Recently, convolutional neural networks (CNNs) have established impressive results for many image recognition tasks. The CNNs usually contain million parameters which prone to overfit when training on small datasets. Therefore, the CNNs do not produce superior performance over traditional methods for action recognition. In this study, the authors design a novel two-stream fully convolutional networks architecture for action recognition which can significantly reduce parameters while keeping performance. To utilise the advantage of spatial-temporal features, a linear weighted fusion method is used to fuse two-stream networks' feature maps and a video pooling method is adopted to construct the video-level features. At the meantime, the authors also demonstrate that the improved dense trajectories has significant impact for action recognition. The authors' method can achieve the state-of-the-art performance on two challenging datasets UCF101 (93.0%) and HMDB51 (70.2%).

1 Introduction

Action recognition in video has become a very active research field in computer vision, which has a wide range of applications such as human–computer interaction, intelligent video surveillance, video search, video recommendation and smart home system. Owing to the large intra-class variations, high dimension of video data, varying motion speed, partial occlusion and background clutter, accurate action recognition is still a big challenging.

Traditional action recognition methods are mainly based on hand-crafted features, and can be divided into global and local approaches. Global approaches represent the video clip as a whole which are favouring to capture the general appearance and motion information. Nevertheless, global approaches are very sensitive to occlusion, shift and cluttering which are commonly existed in action recognition [1]. Compared with global approaches, local approaches are much more efficient and robust in real scenes [1, 2]. Currently, the spatial-temporal interest point (STIP) [3] and improved dense trajectories (iDT) [4] are two widely used local features for action recognition. Once local features were computed, an encoding method is used to encode local features to get the video-level representation. The widely used encoding methods are bag-of-words (BoW) [5], fisher vector [6] and vector of locally aggregated descriptors (VLAD) [7]. Unfortunately, the hand-crafted features-based encoding methods are universal visual representations which does not consider much about temporal information for video-based action recognition.

Recently, deep convolutional neural networks (CNNs) have established the state-of-the-art results for many computer vision tasks such as image classification [8, 9], image segmentation [10, 11] and human pose estimation [12]. Although CNNs are very powerful for these tasks; however, the CNNs do not produce superior performance over traditional hand-crafted feature-based methods for action recognition. As we know, the current state-of-the-art performance [13–16] on standard datasets such as UCF101 [17] and HMDB51 [18] are achieved by the combination of CNNs and hand-crafted features. Yet these methods do not superior outperform traditional methods. There are two potential reasons for this phenomenon: (i) CNNs usually contain millions parameters, while current action recognition video datasets are not sufficient enough to train large CNN models. In fact, the widely used UCF101 dataset contains 13,320 video clips with 101 action classes and HMDB51 dataset only has 6766 video clips with 51 actions. Additional, videos are with complex of variations in motion compared with images. (ii) The general architecture of CNNs does not fully utilise the spatial-temporal information of videos. In [19], the first two-stream CNNs model was proposed to fuse temporal and spatial information for action recognition; however, it directly averages the softmax scores of these two stream to get the final results which is not an optimal solution.

To address these issues, we proposed a two-stream fully convolutional networks (FCNs) for action recognition in video. The first stream is a fully convolutional network which focuses on learning appearance feature from RGB video frames. The second stream mainly focus on learning motion feature from optical flow with a same network architecture as first stream. To fully utilise these two type of features, a linear weighted fusion method is used to fuse pixel-wise corresponded appearance and motion features.

Our main contributions of this paper can be summarised as follows:

i. We design a novel two-stream FCNs architecture for action recognition and a max pooling with larger stride is used to compute a frame-level compact feature, which can take advantage of spatial and temporal information.
ii. Our FCNs fuse the pixel-wise corresponded appearance and motion features by a linear weighted fusion method, which can significantly improve the accuracy.
iii. Several video pooling methods have been studied, and we find that temporal pyramid pooling (TPP) is the most suitable pooling method for constructing the video-level features.

The rest of the paper is organised as follows: Section 2 introduces the related works. Section 3 describes the details of our method, namely FCNs for action recognition. In Section 4, the experimental results are given at different conditions and analyses are made, and then the comparison of our method with other approaches. In Section 5, we summarise our paper.

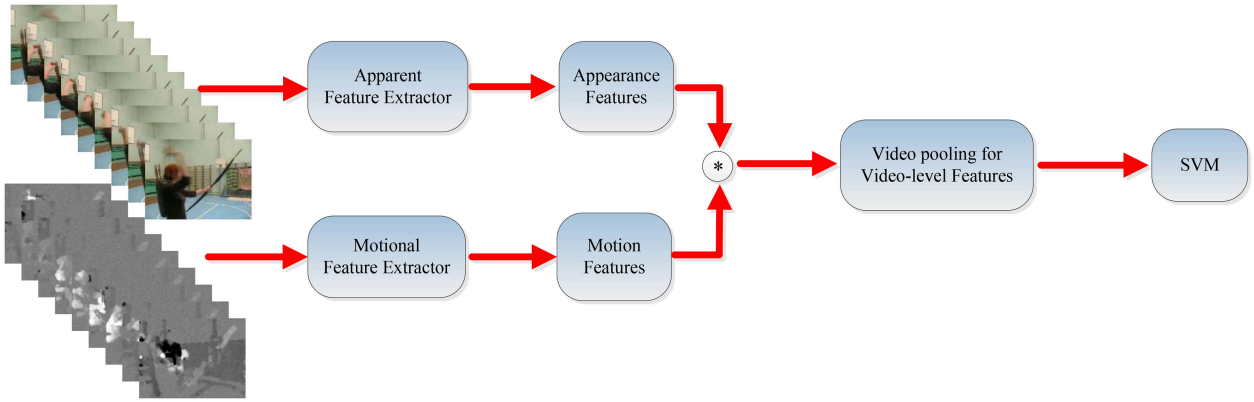

Fig. 1 *FCNs architecture for action recognition*

2 Related works

Over the last decade, action recognition has been dominated by hand-crafted features. In [3], the STIP-based method was proposed by extending Harris corner detectors to 3D. In [20], Klaser *et al.* extended HOG descriptor to histograms of oriented 3D spatial-temporal gradients (HOG3D) for action recognition. Similarly, scale-invariant feature transform (SIFT) [21] is also extended to SIFT-3D [22]. Dollar *et al.* [23] proposed cuboid features for behaviour recognition. Recently, iDT [4], combined with HOG, HOF, MBH feature descriptors, was adopted to human action recognition. Currently, iDTs descriptor is one of the state-of-the-art hand-crafted features. This method begins with densely sampled feature points in video frames and uses optical flows to track them. To improve the performance, Wang *et al.* [4] utilised camera motion estimation to cancel out camera motion from the optical flow. In spite of its good performance, this method is computationally expensive and becomes intractable on large scale datasets. Meanwhile, the performance of this method is limited by the quality of the optical flow available. In [24], a multi-level video representation method has been proposed by stacking the activations from motion features, motion atoms, and motion phrases. The multi-level method can provide complementary information to low-level features and works well for action recognition.

In recent years, deep learning models such as deep belief networks [25], stacked auto-encoders [26], recurrent neural networks (RNNs) [27–30], independent subspace analysis [31, 32], FCNs [33] and CNNs [8] are very popular and have reached the state-of-the-art performance in image domain such as image classification [34, 35], scene recognition [36], re-ranking for objection detection [37] and face recognition [38, 39]. In terms of action recognition, RNNs and CNNs have impressive performance. In [29, 40], the authors combined RNNs and CNNs for action recognition. In detail, they employ a recurrent neural network with long short term memory (LSTM) cells which are connected to the output of the underlying CNNs. These methods have more sophisticated structures; however, only limited improvement has been attained.

Meanwhile, different variants of CNNs for human action recognition in video have been proposed in recent works [13, 15, 19, 41–43]. In [41], Ji *et al.* proposed 3D convolutional neural network architecture, where convolution is implemented in 3D feature maps from both temporal and spatial dimensions. Karpathy *et al.* [42] used a variety of convolutional network architectures to process the task of action recognition on a larger video dataset which consist of 1M videos. In [13], Tran *et al.* proposed convolutional 3D (C3D) method to train CNNs on a limited temporal in terms of using 16 consecutive frames with all convolutional kernels of size $3 \times 3 \times 3$. In [44], Sun *et al.* factorised 3D convolution into a 1D temporal and a 2D spatial convolution. Another type way of learning deep spatial-temporal feature is two-stream CNNs architecture. In [19], Simonyan and Zisserman proposed two-stream CNNs to learn spatial-temporal features and achieve good performance on action recognition. Wang *et al.* [16] proposed a trajectory pooled deep convolutional descriptor (TDD) with two-stream networks to describe the video features. However, the above two methods of action recognition simply average the final outputs from the two networks. Park *et al.* [14] presented two ways of feature amplification and spatially varying multiplicative to fuse multiple sources of knowledge. Wang *et al.* [45] introduced temporal segment networks (TSN) architecture, where sparse temporal sampling strategy is adopt to model long-term temporal structure. In [46], Feichtenhofer *et al.* proposed a novel two-stream convolutional networks fusion method for video action recognition. Compared with previous methods, this method does not increase the number of parameters significantly, and reaches the state-of-the-art performance. In [47], the hybrid fully convolutional network was employed to action detection and generate action bounding box. To estimate the dense map of actionness, the convolutional kernel size is set to 1×1 and stride is set to 1×1. In [33], FCNs was used for the task of semantic segmentation without using any fully connected layer, several convolutional and pooling layers were used compute the semantic probability map at a down-sampled scale, and a final deconvolutional operation was adopt to up-sample the output to original image size. Taking advantages of FCNs, we extend FCNs to two-stream FCNs for action recognition by using different size and number of convolutional kernels. At meantime, we explored a novel strategy for doing fine-tuning which can give good accuracy. To capture temporal information of human actions, the input of the temporal stream is a volume of stacking consecutive optical flow fields instead of just RGB image.

Among these deep learning-based approaches, the two-stream CNNs approaches [19, 46, 47] are most closely related to us. In [19], CNNs are built by convolutional layers and fully connected layers. In contrast, our model only contains the convolutional layers, which decreases the number of parameters significantly. In [46], 3D convolutional fusion is adopt to fuse two streams, but the fully connected layers are also used following by 3D convolutional fusion, which increase much more parameters compared with our method. In [47], hybrid-FCH is used for the task of actionness estimation, but the aim of our method is action recognition. Therefore, the two methods employ different number and size of convolution kernels. Meanwhile, we adopt element-wise add to fuse pixel-wise correspondences spatial and temporal features for action recognition.

3 Our approach

In this section, we first introduce the architecture of the proposed two-stream FCNs and then describe the details of training procedure. As the pipeline shown in Fig. 1, our model contains two sub-streams for extracting the appearance features and the motion features, respectively, followed by a linear weighted method to fuse their pixel-wise correspondence. After the fusion, three different video pooling methods were used to obtain video-level features. Finally, a linear SVM [48] was applied for the classification.

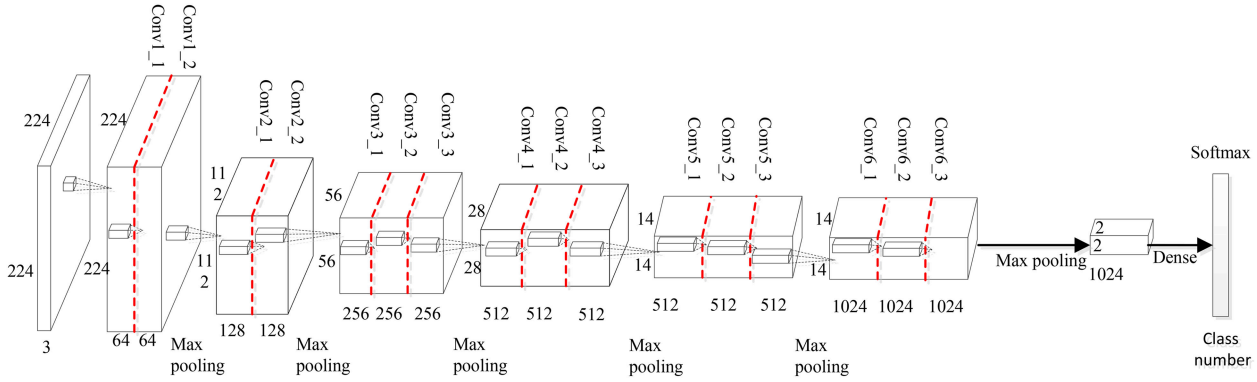

Fig. 2 *Spatial-stream fully convolutional network architecture*

3.1 Fully convolutional networks

Recently, many successful network architectures have been proposed for the ImageNet classification task, such as AlexNet [8], VGGNet [34], GoogLeNet [49] and so on. There are several insights of the network architecture design that get from the evolution of AlexNet to GoogLeNet: using smaller convolutional kernel size, using smaller convolutional strides, and going deeper [50]. For the action recognition tasks, different variants of CNNs also have been propose, such as two-stream convolutional networks [19], 3D CNNs [13] and so on. The main drawback of these CNNs is requiring larger-scale datasets to optimise network parameters. In this paper, we design a two-stream FCNs for action recognition without containing fully connected layers which can significantly reduce the number of parameters.

Our two-stream network architecture contains two separated streams (spatial and temporal). These two streams have the same architecture but with different input data. The details of spatial stream is given in Fig. 2. The network was named FCN-16 which has six convolutional groups with 16 convolutional layers in total. Each convolutional group has two or three convolutional layers. All the convolutional layers have a filter kernel size 3×3 and use the rectified linear unit as activation function. The convolutional group 1 to 5 are followed by a max pooling layer with kernel size 2×2 and stride of 2. The group 6 is followed by a max pooling with kernel size 7×7 and stride 7. The filter number of each convolution layer is showed in Fig. 2. The last layer of the network is a softmax layer, which output the class probabilities.

3.2 Network input and training

The input of spatial stream is a single RGB video frame of size $224 \times 224 \times 3$. Following [16], the volume of stacking optical flow fields is used as input of the temporal stream. The size of volume is $224 \times 224 \times 2\,\mathrm{L}$, where L is the number of stacking flows and is set to 10 in this paper. The optical flow can encode the apparent motion information of objects computed from adjacent frames of videos. To balance the efficiency and accuracy, we use TVL1 method [51] to compute the optical flows. The optical flow fields are discretised into $[0, 255]$ by a linear transformation and then feed to the temporal stream.

For the spatial stream, we use a pre-trained model [16] to initialise parameters of first 5 convolutional groups and randomly initialise the last convolutional group with a standard deviation equal to 0.01. For temporal stream network, we use the initialisation method proposed in [50] which average the ImageNet model filters of the first layer across feature channels and repeat the average filters for $2\,\mathrm{L}$ times.

The modified Caffe toolbox [16] was used to implement the two stream CNNs, and the training set of standard three splits independently for UCF101 and HMDB51 were used to learn the parameters. After initialisation, we fine-tune the parameters on the UCF101 and HMDB51 with an initial learning rate 10^{-5} and decreased to 10^{-6} after $10\,\mathrm{K}$ iterations. The stochastic gradient descent was adopted for optimisation with a mini-batch size 256 and momentum set to 0.9. For the spatial stream, a data augmentation is adopted, each frame was resized to 256×256, and then randomly cropped to several 224×224 patches as well as a probable horizon flip. The corresponded $224 \times 224 \times 2\,\mathrm{L}$ optical flow fields volume is used as input for temporal stream.

3.3 Frame-level features process

In this part, we discuss the frame-level feature extraction procedure and multiple features fusion methods.

3.3.1 Frame-level feature extraction: Once the training is finished, the spatial and temporal FCN model are treated as a deep learning feature extractor to compute the frame-level feature. The feature maps of spatial and temporal network are extracted in a frame-by-frame and volume-by-volume manner. Especially, we pad the optical flow fields of the first $L-1$ frames by duplicated the optical flow field of the first frame to make sure the input to the temporal stream is correct. Finally, the last max pooling layer feature maps is used as the frame-level feature which has a dimension of 4096.

3.3.2 Feature fusion: Given the spatial and temporal stream frame-level features of a video, $\boldsymbol{F}_s \in \mathbb{R}^{H \times W \times T}$, $\boldsymbol{F}_t \in \mathbb{R}^{H \times W \times T}$, two different fusion methods are used to compute output map $\boldsymbol{F}$, where H and W are the height and width of the feature maps, T is the number of frames.

Linear weighted fusion performs a pixel-wise addition between the spatial and temporal feature maps. The weights are used to measure the significance of appearance and motion features

$$\boldsymbol{F} = w_s \boldsymbol{F}_s \oplus w_t \boldsymbol{F}_t \tag{1}$$

where $\boldsymbol{F} \in \mathbb{R}^{H \times W \times T}$, $\oplus$ is matrix addition, w_s and w_t are weights of appearance and motion features, respectively.

Concatenation fusion stacks the two feature maps. We first reshape the spatial and temporal feature maps into vector, and then concatenate these two vectors

$$\boldsymbol{F} = [\boldsymbol{F}_s \quad \boldsymbol{F}_t] \tag{2}$$

where the dimension is $2 \times H \times W \times T$.

3.4 Video pooling

Once the extraction of frame-level features is complete, we need to generate a discriminative video-level descriptor for action recognition. BoW, fisher vector and VLAD are standard feature encoding methods for local features. Max pooling, average pooling, TPP [52] and so on, are standard feature pooling methods. Prior research [53] showed that the fisher vector and VLAD have great advantages over BoW. Among the pooling methods, the TPP gives the best performance. Therefore, we choose fisher vector, VLAD and TPP to obtain video-level feature. In this paper, we regard all these methods as video pooling.

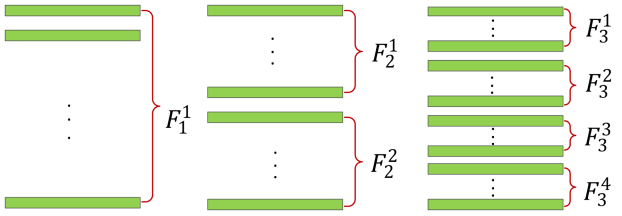

Fig. 3 *Temporal pyramid pooling*

Table 1 Performance comparison of VGG-16 versus FCNs-16 on the UCF101 (mean accuracy over three splits)

	Spatial stream, %	Temporal stream, %	Concatenation fusion, %
VGG-16	75.8	83.7	88.5
FCNs-16	76.8	86.2	89.4

Fisher vector encoding is a supervector based method. The Gaussian mixture model (GMM) is used to describe the distribution over feature space. Generally, the model parameters $\boldsymbol{\theta}$ of a GMM with K components can be denoted as $\boldsymbol{\theta} = \{(\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k), k = 1, 2, \ldots, K\}$, where $\boldsymbol{\mu}_k$ and $\boldsymbol{\sigma}_k$ are mean and variance of the kth component. $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M)$ denotes frame-level features extract from a video, where M is the number of frames, γ_{km} is the weight of frame-level feature $\boldsymbol{x}_m$ for the kth Gaussian. The improved fisher vector can be described as:

$$\boldsymbol{u}_k = \frac{1}{M\sqrt{\pi_k}} \sum_{m=1}^{M} \gamma_{km} \left(\frac{\boldsymbol{x}_m - \boldsymbol{\mu}_k}{\boldsymbol{\sigma}_k} \right) \tag{3}$$

$$\boldsymbol{v}_k = \frac{1}{M\sqrt{2\pi_k}} \sum_{m=1}^{M} \gamma_{km} \left[\left(\frac{\boldsymbol{x}_m - \boldsymbol{\mu}_k}{\boldsymbol{\sigma}_k} \right)^2 - \mathbf{1} \right] \tag{4}$$

$$\gamma_{km} = \frac{\pi_k N(\boldsymbol{x}_m; \boldsymbol{\mu}_k, \boldsymbol{\sigma}_k)}{\sum_{i=1}^{K} \pi_i N(\boldsymbol{x}_m; \boldsymbol{\mu}_i, \boldsymbol{\sigma}_i)} \tag{5}$$

where $N(\boldsymbol{x}_m; \boldsymbol{\mu}_k, \boldsymbol{\sigma}_k)$ is d-dimensional Gaussian distribution, π_k is the mixture weight of the kth component. The final fisher vector $\boldsymbol{F}$ is formed as the concatenation of $\boldsymbol{u}_k$ and $\boldsymbol{v}_k$. The dimension of fisher vector for the video is $2DK$, where D is the dimension of frame-level feature descriptor. To get better result, power normalisation and L_2 normalisation are applied to the fisher vector.

$$\boldsymbol{F} = [\boldsymbol{u}_1, \boldsymbol{v}_1, \ldots, \boldsymbol{u}_K, \boldsymbol{v}_K] \tag{6}$$

VLAD encoding can be regarded as a simple version of fisher vector [2], which only reserves the first order statistic information. Given a collection of frame-level features from an input video and a codebook of centres c_i is generated by K-means, where $i = 1, \ldots, K$, K is the number of the centre. The VLAD descriptor is constructed as follows:

$$\boldsymbol{u} = \left[\sum_{i: c_1 \in rNN(\boldsymbol{x}_i)} (\boldsymbol{x}_i - \boldsymbol{c}_1), \ldots, \sum_{i: c_k \in rNN(\boldsymbol{x}_i)} (\boldsymbol{x}_i - \boldsymbol{c}_k) \right] \tag{7}$$

where $rNN(\boldsymbol{x}_i)$ is r nearest cluster centres. The dimension of VLAD for the video is DK, where D is the dimension of frame-level feature descriptor. We also apply the power and L_2 normalisation to VLAD.

TPP Two level temporal pyramid pooling method has been proposed for video-level representation in [52]. In this paper, we use TPP to process frame-level features; however, it has several significant difference compared with [52]. Our TPP method is illustrated in Fig. 3. The frame-level features of a video are partitioned into a coarse-to-fine type. At the level 1, we treat the whole video frame features as a max pooling segment. While at the level 2 and 3, we divide the video frame features into two and four parts and perform max pooling on each part. The final video-level feature is obtained by concatenate three levels features, $\boldsymbol{F} = [\boldsymbol{F}_1^1, \boldsymbol{F}_2^1, \boldsymbol{F}_2^2, \boldsymbol{F}_3^1, \boldsymbol{F}_3^2, \boldsymbol{F}_3^3, \boldsymbol{F}_3^4]$, which has a dimension $(1 + 2 + 4) \times D$, where D is the dimension of frame-level feature descriptor. Power and L_2 normalisation are applied to this video-level feature.

4 Experiments

In this section, we first give a brief introduction about two challenging human action recognition benchmark dataset UCF101 and HMDB51 in Section 4.1. Experiments conducted on different CNN models are presented in Section 4.2. We then evaluate the performance of different video pooling methods and feature fusion methods in Section 4.3 and 4.4. Experimental results comparison with the state-of-the-art approaches are shown in Section 4.5.

4.1 Dataset

UCF101 is a widely used action recognition benchmark. It contains 13,320 videos clipped from 101 action categories. The videos in 101 action categories are grouped into 25 groups, where each group can consist of four to seven videos of an action. The dataset is divided into a training set containing 9.5 K videos and testing set containing 3.8 K videos.

HMDB51 contains 6766 videos of 51 human actions. The video clips have 320×240 pixels spatial resolution and 30 fps frame rate. Following the evaluation protocol in [18], we use three different training and testing splits in our experiments. For both datasets, we report the mean average accuracy over the three splits.

4.2 Evaluation on different CNNs models

In this section, we compare the performance of the proposed FCNs-16 model with the VGG-16 model. The feature maps of the last fully connected layer FC-4096 of VGG-16 and the last max pooling layer of FCNs-16 are extracted to represent the frame-level feature, respectively. The temporal pyramid pooling is applied to pool frame-level features to obtain the final video feature. We combine two-stream features by concatenation fusion. Results among FCNs-16 model and VGG-16 model on UCF101 over three splits are reported in Table 1. As we can see that, in all cases, the proposed FCNs-16 model consistently outperforms the VGG-16 model in recognition accuracy. In particular, on spatial stream, FCNs-16 outperforms VGG-16 method by 1.0% on the UCF101 (75.8% versus 76.8%). On temporal stream, FCNs-16 outperforms VGG-16 network by 2.5%. Similarly, FCNs-16 achieves a higher accuracy than VGG-16 for two-steam (89.4% versus 88.5%). The results show that our FCNs-16 architecture can achieve higher recognition accuracy with fewer parameters (1.01 to 138 M).

4.3 Evaluation on different video pooling methods

In this section, we conduct experiments to evaluate different video pooling methods, include fisher vector, VLAD and TPP. Following the setting of [4], we randomly sample a subset of 256,000 frame-level features from the training dataset to estimate the GMM. The number of Gaussian is set to $K = 256$, and 256 centres generate by K-means for VLAD. As show in Table 2, the TPP demonstrates great advantages over the fisher vector and VLAD. In linear weighted fusion, the TPP gives 90.2% accuracy, which has significant improvement of 7.2 and 10.0% over fisher vector and VLAD, respectively. Prior research has suggested that fisher vector and VLAD demonstrate better performance than direct pooling method. However, our experiments reach an opposite conclusion. This is may be that fisher vector and VLAD mainly focus on low-level local features, such as HOG and HOF, nevertheless, frame-level features belong to middle-level features. For middle-level features, direct max pooling method can reserve more semantics than conventional encoding methods for action recognition.

Table 2 Results for training/testing split 1 of UCF101 dataset using video pooling methods

	Spatial stream, %	Temporal stream, %	Linear weighted fusion, %
fisher vector	65.0	79.4	83.0
VLAD	62.1	77.3	80.2
TPP	78.4	85.4	90.2

Table 3 Results on UCF101 dataset using feature fusion methods

	Spatial stream, %	Temporal stream, %	Concatenation fusion, %	Linear weighted fusion, %
split 1	78.4	85.4	90.1	90.2
split 2	75.5	86.6	88.9	90.4
split 3	76.6	86.6	89.2	91.0
average	76.8	86.2	89.4	90.5

Table 4 Comparison with the state-of-the-art on the HMDB51 and UCF101 (mean accuracy over three splits)

Feature type	Method	HMDB51	UCF101, %
hand-crafted feature	HOG + fisher vector [4]	40.2%	72.4
	MBH + fisher vector [4]	52.1%	80.8
	iDT + fisher vector [4]	57.2%	84.7
	iDT + HSV [2]	61.1%	87.9
	MoFAP [24]	61.7%	88.3
	MIFS + iDT [54]	65.1%	89.1
deep learning feature |	LRCN [29]	–	82.9
	LSTM [40]	–	88.6
	two-stream network [19]	59.4%	88.0
	C3D [13]	–	85.2
	DANN [43]	63.3%	89.2
	TDD [16]	63.2%	90.3
	3D convolution [46]	64.6%	91.8
	ours (FCNs-16)	63.4%	90.5
two kinds of features	C3D + iDT [13]	–	90.4
	TDD + iDT [16]	65.9%	91.5
	LTC + iDT [15]	67.2%	92.7
	3D convolution + iDT [46]	69.2%	93.5
	TSN (three modalities) [45]	69.4%	94.2
	ours + iDT	70.2%	93.0

4.4 Evaluation on different feature fusion methods

The evaluation of different feature fusion methods are reported in Table 3. We show the recognition accuracy for each split of UCF101 dataset, and the average recognition accuracy over the three splits. Clearly, the performance of the temporal stream is about 10% higher than that of the spatial stream. It indicates that the motion features play an even more important role in the video based action recognition task. Thus, in linear weighted fusion method, we set w_s to 0.2 and w_t to 0.8. Under this setting, the linear weighted fusion improves the accuracy of the concatenation fusion by 1.1%. This enhancement may be due to that the linear weighted fusion method can fuse spatial correspondences between motion and appearance features.

4.5 Comparison with the state-of-the-art methods

In Table 4, we compare our approach to a variety of the state-of-the-art methods over three splits on HMDB51 and UCF101 datasets. We divide these methods into different groups according to the type of the feature being used, include hand-crafted feature, deep-learned feature and the combination of these two kinds of features. During hand-crafted feature based methods, iDT-based features perform well and have competitive in the area of deep learning-based methods, especially with higher-order encodings. In the deep learning methods, our approach obtains an accuracy of 63.4% on the HMDB51 and 90.5% on the UCF101. Compared to the original two-stream method [19], our approach improves the accuracy by 4.0% on the HMDB51 and 2.5% on the UCF101. The 3D convolution [46] is superior to our method ∼1.2% on both datasets, however, our model has much fewer parameters (1.01 versus 97.58 M). Compared with RNN based approaches, our approach outperform long-term recurrent convolutional networks (LRCN) [29] and LSTM [40] model by 7.6 and 1.9% on the UCF101, respectively. Experiments demonstrating that our method has strong discriminative even with fewer parameters. Finally, we explore the benefit of the fusion of iDT feature and deep learning feature. As a result, the TSN with three modalities [45] leads to a performance gain of 1.2% compared with our method on the UCF101. However, our method outperforms TSN by 0.5% on the HMDB51. It shows that our method has superiority on small-scale dataset. On the whole, the combination of our FCNs-16 together with the iDT method provides best results, which obtains the accuracy of 70.2% on the HMDB51 and 93.0% on the UCF101. These state-of-the-art results show that there is a degree of complementary between hand-crafted features and our fully convolutional network approach.

5 Conclusions

In this paper, we propose a new deep fully convolutional network for action recognition, called FCNs-16. The FCNs-16 decreases the number of parameters significantly over previous methods, yet achieves the state-of-the-art performance on standard benchmark datasets. Our network is composed of a spatial stream and a temporal stream, which represents appearance and motion features, respectively. Moreover, we design a linear weighted fusion method to effectively fuse appearance and motion information simultaneously. According to the experimental results, we believe that action recognition will be benefit from the appearance and motion fusion. Experimental results conducted on UFC101 and HMDB51 show that our approach can achieve promising performance and obtain further improvements by combining hand-crafted features.

6 Acknowledgment

This work was supported by the Nature Science Foundation of China (grant nos. 61572409, 61571188, 61402386 and 81230087), a Project Supported by Scientific Research Fund of Hunan Provincial Education Department (grant no. 15C0726), Fujian Province 2011 Collaborative Innovation Center of TCM Health Management and Collaborative Innovation Center of Chinese Oolong Tea Industry – Collaborative Innovation Center (2011) of Fujian Province, the Construct Program of the Key Discipline in Hunan Province, China, and the Aid program for Science and Technology Innovative Research Team in Higher Educational Institute of Hunan Province, China.

7 References

[1] Shao, L., Liu, L., Yu, M.: 'Kernelized multiview projection for robust action recognition', *Int. J. Comput. Vis.*, 2016, **118**, pp. 115–129

[2] Peng, X., Wang, L., Wang, X*., et al.*: 'Bag of visual words and fusion methods for action recognition: comprehensive study and good practice', *Comput. Vis. Image Underst.*, 2016, **150**, pp. 109–125

[3] Laptev, I.: 'On space-time interest points', *Int. J. Comput. Vis.*, 2005, **64**, (2–3), pp. 107–123

[4] Wang, H., Schmid, C.: 'Action recognition with improved trajectories'. , 2013 IEEE Int. Conf. on Computer Vision (ICCV), 2013, pp. 3551–3558

[5] Csurka, G., Dance, C., Fan, L*., et al.*: 'Visual categorization with bags of keypoints'. Workshop on statistical learning in computer vision, ECCV, vol. **1**, 2004, pp. 1–2

[6] Perronnin, F., Dance, C.: 'Fisher kernels on visual vocabularies for image categorization'. 2007 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2007, pp. 1–8

[7] Jegou, H., Perronnin, F., Douze, M., *et al.*: 'Aggregating local image descriptors into compact codes', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2012, **34**, (9), pp. 1704–1716

[8] Krizhevsky, A., Sutskever, I., Hinton, G.E.: 'Imagenet classification with deep convolutional neural networks'. Advances in neural information processing systems (NIPS 2012), 2012, pp. 1097–1105

[9] Bilen, H., Fernando, B., Gavves, E., *et al.*: 'Dynamic image networks for action recognition'. 2016 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 3034–3042

[10] Girshick, R.: 'Fast r-cnn'. 2015 IEEE Conf. on Computer Vision (ICCV), 2015, pp. 1440–1448

[11] Ren, S., He, K., Girshick, R., *et al.*: 'Faster r-cnn: towards real-time object detection with region proposal networks'. Advances in neural information processing systems (NIPS 2015), 2015, pp. 91–99

[12] Tompson, J., Goroshin, R., Jain, A., *et al.*: 'Efficient object localization using convolutional networks'. 2015 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 648–656

[13] Tran, D., Bourdev, L., Fergus, R., *et al.*: 'Learning spatiotemporal features with 3d convolutional networks'. 2015 IEEE Conf. on Computer Vision (ICCV), 2015, pp. 4489–4497

[14] Park, E., Han, X., Berg, T.L., *et al.*: 'Combining multiple sources of knowledge in deep cnns for action recognition'. 2016 IEEE Winter Conf. on Applications of Computer Vision (WACV), 2016, pp. 1–8

[15] Varol, G., Laptev, I., Schmid, C.: 'Long-term temporal convolutions for action recognition', arXiv preprint arXiv:1604.04494, 2016

[16] Wang, L., Qiao, Y., Tang, X.: 'Action recognition with trajectory-pooled deep-convolutional descriptors'. 2015 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 4305–4314

[17] Soomro, K., Zamir, A.R., Shah, M.: 'Ucf101: A dataset of 101 human actions classes from videos in the wild', CRCV-TR-12, 2012

[18] Kuehne, H., Jhuang, H., Stiefelhagen, R., *et al.*: '*Hmdb51: A large video database for human motion recognition*' (High Performance Computing in Science and Engineering 12, Springer, 2013), pp. 571–582

[19] Simonyan, K., Zisserman, A.: 'Two-stream convolutional networks for action recognition in videos'. Advances in Neural Information Processing Systems (NIPS 2014), 2014, pp. 568–576

[20] Klaser, A., Marszałek, M., Schmid, C.: 'A spatio-temporal descriptor based on 3d-gradients'. 2008–19th British Machine Vision Conf. (BMVC), British Machine Vision Association, 2008, pp. 1–10

[21] Lowe, D.G.: 'Distinctive image features from scale-invariant keypoints', *Int. J. Comput. Vis.*, 2004, **60**, (2), pp. 91–110

[22] Scovanner, P., Ali, S., Shah, M.: 'A 3-dimensional sift descriptor and its application to action recognition'. Proc. of the 15th ACM international conference on Multimedia, ACM, 2007, pp. 357–360

[23] Dollár, P., Rabaud, V., Cottrell, G., *et al.*: 'Behavior recognition via sparse spatio-temporal features'. 2005 IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, IEEE, 2005, pp. 65–72

[24] Wang, L., Qiao, Y., Tang, X.: 'Mofap: a multi-level representation for action recognition', *Int. J. Comput. Vis.*, 2016, **119**, (3), pp. 119–254

[25] Lee, H., Grosse, R., Ranganath, R., *et al.*: 'Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations'. Proc. of the 26th Annual Int. Conf. on Machine Learning, ACM, 2009, pp. 609–616

[26] Gehring, J., Miao, Y., Metze, F., *et al.*: 'Extracting deep bottleneck features using stacked auto-encoders'. 2013 IEEE Int. Conf. on Acoustics, Speech and Signal Processing, IEEE, 2013, pp. 3377–3381

[27] Jain, A., Zamir, A.R., Savarese, S., *et al.*: 'Structural-rnn: deep learning on spatio-temporal graphs', arXiv preprint arXiv:1511.05298, 2015

[28] Zaremba, W., Sutskever, I., Vinyals, O.: 'Recurrent neural network regularization', arXiv preprint arXiv:1409.2329, 2014

[29] Donahue, J., Anne Hendricks, L., Guadarrama, S., *et al.*: 'Long-term recurrent convolutional networks for visual recognition and description'. 2015 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 2625–2634

[30] Veeriah, V., Zhuang, N., Qi, G.J.: 'Differential recurrent neural networks for action recognition'. 2015 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 4041–4049

[31] Le, Q.V., Zou, W.Y., Yeung, S.Y., *et al.*: 'Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis'. 2011 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), IEEE, 2011, pp. 3361–3368

[32] Lan, Z., Yu, S.I., Lin, M., *et al.*: 'Handcrafted local features are convolutional neural networks', arXiv preprint arXiv:1511.05045, 2015

[33] Long, J., Shelhamer, E., Darrell, T.: 'Fully convolutional networks for semantic segmentation'. 2015 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3431–3440

[34] Simonyan, K., Zisserman, A.: 'Very deep convolutional networks for large-scale image recognition', arXiv preprint arXiv:1409.1556, 2014

[35] Liu, L., Shen, C., van den Hengel, A.: 'The treasure beneath convolutional layers: cross-convolutional-layer pooling for image classification'. 2015 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 4749–4757

[36] Zhou, B., Lapedriza, A., Xiao, J., *et al.*: 'Learning deep features for scene recognition using places database'. Advances in neural information processing systems (NIPS2014), 2014, pp. 487–495

[37] Zhong, Z., Lei, M., Cao, D., *et al.*: 'Class-specific object proposals re-ranking for object detection in automatic driving', *Neurocomputing*, 2017, **242**, pp. 187–194

[38] Sun, Y., Chen, Y., Wang, X., *et al.*: 'Deep learning face representation by joint identification-verification'. Advances in Neural Information Processing Systems (NIPS 2014), 2014, pp. 1988–1996

[39] Sun, Y., Liang, D., Wang, X., *et al.*: 'Deepid3: face recognition with very deep neural networks', arXiv preprint arXiv:1502.00873, 2015

[40] Yue-Hei Ng, J., Hausknecht, M., Vijayanarasimhan, S., *et al.*: 'Beyond short snippets: deep networks for video classification'. 2015 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 4694–4702

[41] Ji, S., Xu, W., Yang, M., *et al.*: '3d convolutional neural networks for human action recognition', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2013, **35**, (1), pp. 221–231

[42] Karpathy, A., Toderici, G., Shetty, S., *et al.*: 'Large-scale video classification with convolutional neural networks'. 2014 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1725–1732

[43] Wang, J., Wang, W., Wang, R., *et al.*: 'Deep alternative neural network: exploring contexts as early as possible for action recognition'. Advances in Neural Information Processing Systems (NIPS 2016), 2016, pp. 811–819

[44] Sun, L., Jia, K., Yeung, D.Y., *et al.*: 'Human action recognition using factorized spatiotemporal convolutional networks'. 2015 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 4597–4605

[45] Wang, L., Xiong, Y., Wang, Z., *et al.*: 'Temporal segment networks: towards good practices for deep action recognition'. European Conf. on Computer Vision, 2016, pp. 20–36

[46] Feichtenhofer, C., Pinz, A., Zisserman, A.: 'Convolutional two-stream network fusion for video action recognition'. 2016 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1933–1941

[47] Wang L., Qiao, Yu, T.X., Gool, L.V.: 'Actionness estimation using hybrid fully convolutional networks'. 2016 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2708–2717

[48] Fan, R.E., Chang, K.W., Hsieh, C.J., *et al.*: 'Liblinear: a library for large linear classification', *J. Mach. Learn. Res.*, 2008, **9**, pp. 1871–1874

[49] Szegedy, C., Liu, W., Jia, Y., *et al.*: 'Going deeper with convolutions'. , 2015 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1–9

[50] Wang, L., Xiong, Y., Wang, Z., *et al.*: 'Towards good practices for very deep two-stream convnets', arXiv preprint arXiv:1507.02159, 2015

[51] Zach, C., Pock, T., Bischof, H.: 'A duality based approach for realtime tv-l 1 optical flow'. Joint Pattern Recognition Symp., 2007, pp. 214–223

[52] Wang, P., Cao, Y., Shen, C., *et al.*: 'Temporal pyramid pooling based convolutional neural networks for action recognition', arXiv preprint arXiv:1503.01224, 2015

[53] Xu, Z., Yang, Y., Hauptmann, A.G.: 'A discriminative cnn video representation for event detection'. 2015 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1798–1807

[54] Lan, Z., Lin, M., Li, X., *et al.*: 'Beyond Gaussian pyramid: multi-skip feature stacking for action recognition'. 2015 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 204–212