Pattern Recognition 85 (2019) 1–12

Contents lists available at ScienceDirect

Pattern Recognition

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

Asymmetric 3D Convolutional Neural Networks for action recognition

Hao Yang[a,c], Chunfeng Yuan[a,*], Bing Li[a], Yang Du[a,c], Junliang Xing[a], Weiming Hu[a,b,c], Stephen J. Maybank[d]

[a] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, PR China
[b] CAS Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Beijing 100190, PR China
[c] University of Chinese Academy of Sciences, Beijing 100190, PR China
[d] Department of Computer Science and Information Systems, Birkbeck College, London WC1E 7HX, United Kingdom

ARTICLE INFO

Article history:
Received 1 October 2017
Revised 2 June 2018
Accepted 22 July 2018
Available online 24 July 2018

Keywords:
Asymmetric 3D convolution
MicroNets
3D-CNN
Action recognition

ABSTRACT

Convolutional Neural Network based action recognition methods have achieved significant improvements in recent years. The 3D convolution extends the 2D convolution to the spatial-temporal domain for better analysis of human activities in videos. The 3D convolution, however, involves many more parameters than the 2D convolution. Thus, it is much more expensive on computation, costly on storage, and difficult to learn. This work proposes efficient asymmetric one-directional 3D convolutions to approximate the traditional 3D convolution. To improve the feature learning capacity of asymmetric 3D convolutions, a set of local 3D convolutional networks, called *MicroNets*, are proposed by incorporating multi-scale 3D convolution branches. Then, an asymmetric 3D-CNN deep model is constructed by *MicroNets* for the action recognition task. Moreover, to avoid training two networks on the RGB and Flow frames separately as most works do, a simple but effective multi-source enhanced input is proposed, which fuses useful information of the RGB and Flow frame at the pre-processing stage.

The asymmetric 3D-CNN model is evaluated on two of the most challenging action recognition benchmarks, UCF-101 and HMDB-51. The asymmetric 3D-CNN model outperforms all the traditional 3D-CNN models in both effectiveness and efficiency, and its performance is comparable with that of recent state-of-the-art action recognition methods on both benchmarks.

1. Introduction

In recent years, Convolutional Neural Networks (CNNs) have achieved great success and become the mainstream method in many computer vision tasks, such as image classification [1–4], object detection [5–7], semantic segmentation [8–10], and human action recognition [11–14]. In these great improvements, several practices have been drummed in designing deep convolutional networks. First, information bottlenecks should be avoided by increasing the number of feature channels with the depth of the network. Second, the receptive fields at the end of the network should be large enough, so that the processing units can base their operations on larger regions of the input. Large receptive fields can be achieved by stacking many small filters or by using large filters. The stacking of small filters can be implemented with fewer parameters and operations, and more complex non-linearities can be included. Third, dimension reduction by aggregating filter is supported by the fact that outputs of neighboring filters are highly correlated.

To accelerate the training and inference of 2D convolutional networks, many methods [15–20] have been proposed in recent years. The linear structure is exploited to approximate the convolutional filters [15–17]. These methods [15,18] flatten 2D convolutional filters into a sequence of one-dimensional filters across the spatial domain and channels. In [19], the 2D convolution is divided into two phases, namely in-channel convolution and across-channel linear projection. Moreover, the sparse regularity is introduced in training by Liu et al. [20] to remain the sparsity in convolutional filters. This sparsity decreases the computational cost of the 2D convolution.

The 3D convolutional networks [22,23] naturally extend the 2D convolutional network to the 3D spatial-temporal domain, in order to better analyze human activities in videos. The traditional 3D convolution is illustrated in Fig. 1(a). However, the 3D convolution is very expensive to compute, because a 3D convolution with k pa-

* Corresponding author.
E-mail addresses: hao.yang@nlpr.ia.ac.cn (H. Yang), cfyuan@nlpr.ia.ac.cn (C. Yuan), bli@nlpr.ia.ac.cn (B. Li), duyang2014@ia.ac.cn (Y. Du), jlxing@nlpr.ia.ac.cn (J. Xing), wmhu@nlpr.ia.ac.cn (W. Hu), sjmaybank@dcs.bbk.ac.uk (S.J. Maybank).

https://doi.org/10.1016/j.patcog.2018.07.028
0031-3203/

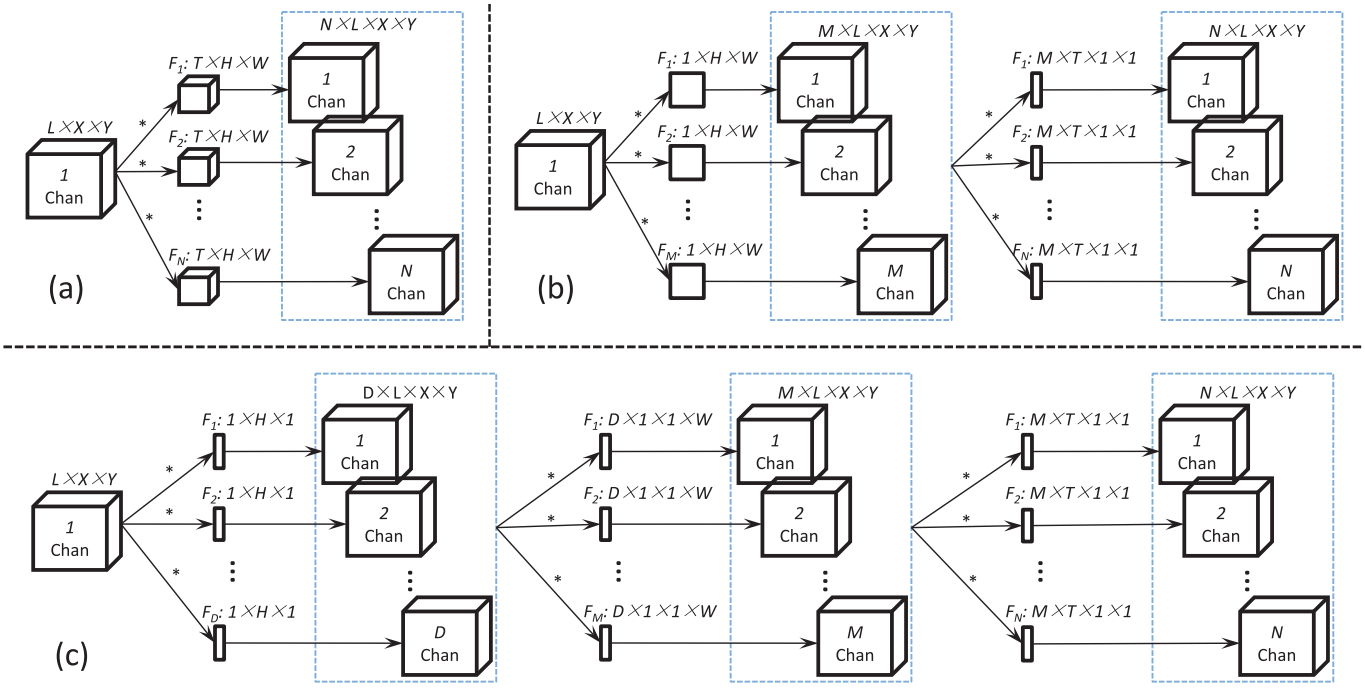


Fig. 1. Illustration of three types of 3D convolutional operations: (a) the traditional 3D convolution, (b) the factorized spatial-temporal convolution proposed in FstCN [21], and (c) our asymmetric 3D convolutions. Our asymmetric 3D convolutions have fewer parameters and operations than (a) and (b).

rameters in each direction requires one order more weights to be learned than a 2D convolution (k^3 VS k^2). Additionally, a 3D convolutional deep model requires much more training data than a 2D convolutional deep model. The annotations of video data are much more costly than that of images. Last but not least, the 3D convolutional networks cannot be fine-tuned from a model pre-trained on the large-scale ImageNet dataset [24] as 2D-CNN based action recognition methods [11,25,26]. To decrease the number of parameters and the computational cost of the 3D convolution, FstCN [21] approximates a 3D convolutional layer by several 2D convolutional layers. The former 2D convolutional layers operate on the spatial domain and the last one operates on the temporal domain, as simply illustrated in Fig. 1(b). The FstCN model is the first attempt to accelerate the traditional 3D convolution. It reduces the number of parameters and the computational cost of traditional 3D convolution from cubic to quadratic ($2k^2$ VS k^3). But there is still room to reduce the number of parameters and the computational cost of the traditional 3D convolution further.

This paper, inspired by Denton et al. [15] and Jin et al. [18], exploits three cascaded asymmetric one-directional 3D convolutions to approximate a traditional 3D convolution with the same size of receptive field and to accelerate the traditional 3D convolution further, as shown in Fig. 1(c). These asymmetric 3D convolutions decrease the number of parameters and the computational cost significantly ($3k$ VS k^3). Moreover, several local 3D convolutional networks referred to as *MicroNets* are proposed. These *MicroNets* incorporate the asymmetric 3D convolutional layers with traditional 3D convolutional layers in multi-scale branches, to improve the representational ability of the asymmetric 3D convolutional layers without increasing the computational cost. Finally, following the practices in designing deep networks, an efficient and effective 3D-CNN deep model is constructed by stacking several *MicroNets*. The asymmetric 3D-CNN deep model has fewer weights, lower computational complexity, and stronger representative ability than traditional 3D-CNN models. Thus, it is more easily trained on video datasets which are usually too small for training traditional 3D-CNN models.

In addition, an effective multi-source enhanced input is proposed for action recognition. Previous 2D-CNN based action recognition methods [11,12,26,27] usually train two deep convolutional networks individually: the SpatialNet is trained on RGB frame and the TemporalNet is trained on stacked optical flow fields (Flow). The softmax scores of the two deep convolutional networks are fused to achieve a better classification performance. However, training two convolutional deep models individually not only is costly in terms of computation, but also it does not allow end-to-end training of the whole model to better exploit the correlations between the appearance features and motion features. To overcome these limitations, a enhanced input is proposed by incorporating useful information of the RGB and Flow frames. It is used to decrease the computational cost by avoiding training two networks separately. It improves classification performance significantly from the two individual networks fed with RGB and Flow frames respectively and achieves a comparable performance with that obtained by fusing the outputs of SpatialNet and TemporalNet.

The main contributions of this work are summarized as follows:

- Asymmetric 3D convolutions are proposed to approximate the traditional 3D convolution. The asymmetric 3D convolutions have fewer parameters and a reduced computational cost.
- To improve the feature learning capacity of asymmetric 3D convolutional layers, several local asymmetric 3D convolutional *MicroNets* are proposed by incorporating multi-scale convolutional features.
- The asymmetric 3D convolutional deep model is constructed by stacking multiple *MicroNets*, which outperforms the traditional 3D-CNN models on both effectiveness and efficiency.
- A multi-source enhanced input is proposed to decrease the computational cost further by avoiding training two deep networks individually.

2. Related works

The classification of actions in trimmed videos is a very challenging task and has long been an active research topic

in computer vision, with many applications such as intelligent surveillance, human-computer interaction, robotics, etc. In recent decades, researchers have proposed many methods to classify human actions. In particular, the deep learning based methods have outperformed the traditional action recognition methods by a wide margin in recent years. The traditional action recognition methods and the deep learning based action recognition methods are briefly discussed in this section.

2.1. Traditional methods for action recognition

In the past decades, researchers have designed many elaborate handcrafted features to represent videos or actions [28–33]. The Spatial-Temporal Interest Points (STIPs) [28] extend the Harris corner detector [34] to 3D spatial-temporal domain, in order to search for interest points in videos. The Histogram of Gradients (HOG) [35] and Histogram of Optical Flows (HOF) [36] features are widely used to describe the STIPs. The Sparse Spatio-Temporal Features [29] improve the 3D Harris detector by applying Gabor filtering in the spatial and temporal dimensions separately. The scale-invariant STIPs [30] generalize the SURF descriptors [37] by computing a weighted sum of space-time Haar-wavelets in grid cells. Similarly, the 3D-SIFT descriptor [31] is the spatial-temporal extension of the SIFT [38] for human action recognition. The dense trajectories methods [32,33] track densely sampled interest points through video sequences. The resulting trajectories and the aligned space-time volumes are used to represent the videos. These methods, combined with local HOG, HOF and MBH descriptors have achieved the best performance among handcrafted features. However, it is difficult to transfer these handcrafted features from the original training dataset to another dataset.

2.2. CNN based methods for action recognition

Inspired by the great success of deep convolutional models in many computer vision tasks [3,4,6,8], many CNN based methods have been proposed for action recognition [11–13,39]. The Slow Fusion model [39] fuses spatial and temporal information at multiple semantic levels. Although this model is fed with multiple consecutive RGB frames, it cannot learn the motion features, because the temporal information collapses after the first 2D convolutional layer according to [23]. The Two-stream model [11] learns the appearance and motion features using two individual networks, namely SpatialNet, which is trained on single RGB frame to extract appearance features, and TemporalNet, which is trained on ten consecutive Flow frames to extract motion features. The confidence scores of SpatialNet and TemporalNet are fused to classify actions. The Fusion Two-stream model [27] demonstrates that fusing the appearance and motion features after the last convolutional layer achieves the best performance. The Deeper Two-stream model [25] exploits a very deep convolutional network, *i.e.,* VGG-16 [2], as the backbone of SpatialNet and TemporalNet, to improve the performance of action classification. Based on the Deeper Two-stream model [25], the Temporal Segment Network (TSN) [12] splits each input video into three segments in the temporal domain and trains a very deep SpatialNet and a TemporalNet on each segment. The final fusing result from the three segments achieves the current state-of-the-art performance in action recognition. Trajectory-Pooled Deep-Convolutional Descriptors (TDD) [40] combine deep convolutional features extracted from the Two-stream model [11] with dense trajectory features using a trajectory centered pooling method. Although these 2D-CNN models are good at extracting spatial appearance features from subjects and backgrounds, it is difficult to learn the motion features required for action recognition.

2.3. RNN based methods for action recognition

Recurrent Neural Network (RNN) models [41,42] are effective in capturing temporal information because the current prediction is not only based on the current observation but also on the past information stored in hidden states. For this reason, RNN models are widely applied in action recognition to model the motion features in videos. The general pipeline for RNN based action recognition methods [26,43–45] begins with the extraction of frame-wise features using a CNN model. Then the frame-wise features are fed to LSTM layers to model temporal dependency. Following this pipeline, Baccouche et al. [44] utilize a 3D-CNN model to extract spatial-temporal features. Beyond Short Snippets [45] models the full length content of the videos to produce large performance improvements over previously results [26,43,44]. The Two-stream LSTM model [46] stacks multiple LSTM layers to capture dynamic information in a hierarchical manner. The two feature streams, *i.e.*, a convolutional feature stream and a pooled feature stream, communicate with each other in training. LSTM based attention models [47,48] can focus on a region in each frame of a video sequence that is most discriminative for action recognition. A soft attention mechanism is utilized in [47]. The focus of attention varies throughout the video sequence. Learning such attention weights through back-propagation is a computationally demanding task, because all possible combinations of input and output have to be checked.

2.4. 3D-CNN based methods for action recognition

The 3D-CNN model [22] extends a 2D convolution to the temporal domain, to extract spatial-temporal features for action recognition. The 3D-CNN model abstracts spatial-temporal information naturally at multiple semantic levels from videos. The C3D model [23] is pre-trained on a large-scale video dataset to learn general features which are used to train a linear SVM for action classification. The T-CNN model [49] extends the R-CNN model [5] from detecting objects in images to detecting actions in videos. It replaces the last Max-pooling layer of the C3D model [23] with a TOI pooling layer. These 3D convolutional deep models are typically learned within a short snippet of the video, so they fail to model actions over their full temporal extent. The LTC-CNN model [50] operates on longer temporal extents of videos in order to improve the accuracy of action recognition. I3D [51] proposes a very deep Inflated 3D-CNN model by extending the Inception model [3] to 3D to extract spatial-temporal features of actions. The I3D model is pre-trained on the very large and well-trimmed Kinetics video dataset and achieves a great improvement for action recognition. To avoid using computationally expensive 3D convolutions, the Factorized spatial-temporal Convolutional Network (FstCN) [21] extracts spatial-temporal features by introducing a Transformation-Permutation layer to convert the 3D convolutional layer into several 2D convolutional layers operating in spatial domain followed by one 2D convolutional layer operating in temporal domain. However, the FstCN model may have difficulty learning effective spatial-temporal features, given that there is only one temporal convolutional layer following several spatial convolutional layers.

3. Proposed 3D convolutional model

In this section, a video-friendly asymmetric 3D convolutional deep model is proposed. Firstly, an asymmetric one-directional 3D convolutional layer is introduced. It is compared with traditional 3D convolutional layer taking into account the number of parameters and the computational cost. Then the efficient asymmetric 3D convolutional layers are used to construct local 3D convolutional

networks which are the building blocks of the asymmetric 3D-CNN deep model. Finally a multi-source enhanced input is proposed, to train the 3D-CNN deep model easily.

3.1. Asymmetric 3D convolution

The 3D convolution is very effective in extracting spatial-temporal features from videos for action recognition [22,23,44,50,52]. The weights of a 3D convolution are denoted as 5-dimensional filters: $F \in \mathcal{R}^{N\times C\times T\times H\times W}$, where C is the number of input channels, T, H and W are the temporal length, height and width of the 3D convolutional kernel respectively, and N is the number of filters or output channels. The input video volume or internal feature volume is denoted as $V \in \mathcal{R}^{C\times L\times X\times Y}$, where L, X and Y are the temporal length and spatial height and width of the volume. The operation of each 3D convolutional filter $F_f \in \mathcal{R}^{C\times T\times H\times W}$, $f = 1, \ldots, N$ is formulated as:

$$\Phi_f(l,x,y) = V * F_f \tag{1}$$

$$= \sum_{c=1}^{C}\sum_{t=1}^{T}\sum_{h=1}^{H}\sum_{w=1}^{W} V(c, l-t, x-h, y-w) F_f(c,t,h,w), \tag{2}$$

where $l = 1, \ldots, L$, $x = 1, \ldots, X$ and $y = 1, \ldots, Y$, as shown in Fig. 1(a). Without loss of generality, the output feature volume is denoted as $V' \in \mathcal{R}^{N\times L\times X\times Y}$. So the number of parameters of the traditional 3D convolutional filters is $C(THW)N$ and the number of multiplications is $C(THW)N(LXY)$. Both numbers are dramatically larger than the corresponding numbers of 2D convolutional filters. As a result, 3D convolutional filters have a much higher computational cost and require many more videos for training than 2D convolutional filters.

In order to alleviate the drawbacks of traditional 3D convolution, an efficient asymmetric 3D convolution is proposed. Each traditional 3D convolutional filter is approximated by three cascaded asymmetric 3D convolutional filters operating on three different directions, as shown in Fig. 1(c). Corresponding to Eq. (2), the operations of asymmetric 3D convolutional filters are formulated as:

$$\Phi_{\alpha_f}(l,x,y) = V * F_{\alpha_f} = \sum_{c=1}^{C}\sum_{h=1}^{H} V(c, l, x-h, y) F_{\alpha_f}(c,1,h,1), \tag{3}$$

$$\Phi_{\beta_f}(l,x,y) = \Phi_{\alpha_f} * F_{\beta_f} = \sum_{\alpha_f=1}^{D}\sum_{w=1}^{W} \Phi_{\alpha_f}(l,x,y-w) F_{\beta_f}(\alpha_f,1,1,w), \tag{4}$$

$$\Phi_{\gamma_f}(l,x,y) = \Phi_{\beta_f} * F_{\gamma_f} = \sum_{\beta_f=1}^{M}\sum_{t=1}^{T} \Phi_{\beta_f}(l-t,x,y) F_{\gamma_f}(\beta_f,t,1,1), \tag{5}$$

$$\widehat{\Phi}_f(l,x,y) = V * \widehat{F}_f = ((V * F_{\alpha_f}) * F_{\beta_f}) * F_{\gamma_f}, \tag{6}$$

where $\alpha_f = 1, \ldots, D$, $\beta_f = 1, \ldots, M$ and $\gamma_f = 1, \ldots, N$. In Eq. (6), $\widehat{\Phi}_f(l,x,y)$ denotes the output feature volume of the approximated 3D convolution. Eqs. (3)–(5) define the asymmetric 3D convolutional filters operating on the height, width and temporal directions respectively. The numbers of parameters and multiplications in the approximated 3D convolution are obtained by summing the corresponding quantities of the three asymmetric 3D convolutions defined in Eqs. (3), (4) and (5), respectively. The sums are $CHD + DWM + MTN$ convolutional parameters and $(CHD + DWM + MTN)(LXY)$ multiplications. If $D = M = N$, then the total number of convolutional parameters is $C(H + W + T)N$ and the number of multiplications is $C(T + H + W)N(LXY)$, which are reduced by two orders, compared with the totals $C(THW)N$ and $C(THW)N(LXY)$ for the traditional 3D convolution.

More specifically, the traditional $3 \times 3 \times 3$ 3D convolutional layer which is widely used in the previous 3D-CNN models [22,23,50] is approximated using the proposed asymmetric 3D convolutions, *i.e.*, three cascaded asymmetric 3D convolutional layers with kernel sizes of $1 \times 3 \times 1$, $1 \times 1 \times 3$ and $3 \times 1 \times 1$. As illustrated in Fig. 2(a), the three cascaded asymmetric 3D convolutional layers have same size of receptive field as the traditional 3D convolutional layer, but the cascaded layers are more efficient, as they have fewer parameters.

Then, if each $3 \times 3 \times 3$ 3D convolutional layer in a traditional 3D-CNN deep model is approximated by three asymmetric 3D convolutional layers in the same way, the resulting C3D model will be equivalent to a very deep 3D convolutional network with 24 asymmetric 3D convolutional layers. The computational cost of it is relatively low. However, it is difficult to train the very deep asymmetric 3D convolutional network. In this paper, pairs of adjacent two traditional $3 \times 3 \times 3$ 3D convolutional layers are approximated by three asymmetric 3D convolutional layers with kernel sizes of $1 \times 5 \times 1$, $1 \times 1 \times 5$ and $3 \times 1 \times 1$ respectively. From Fig. 2(b), the pairs of two traditional 3D convolutional layers and the three asymmetric 3D convolutional layers have the same size of receptive field.

Why not use a $5 \times 1 \times 1$ asymmetric 3D convolutional layer in the temporal domain? Generally, the receptive field of the last layer of a 3D convolutional network is expected as large as possible. To compare with 3D-CNN models [21,23,49] fairly, a clip of 16 frames is fed to the deep model. Using the kernel sizes of $5 \times 1 \times 1$ and $3 \times 1 \times 1$, the receptive fields at the end of the 3D deep network are both larger than 16 in the temporal domain, but the $5 \times 1 \times 1$ kernel has more parameters and has more computational cost. In the experiments, the asymmetric 3D convolutional layers with kernel sizes of $1 \times 5 \times 1$, $1 \times 1 \times 5$ and $3 \times 1 \times 1$ are appropriate to an input video volume in which the spatial dimensions are much larger than the number of frames.

3.2. 3D convolutional MicroNets

Inspired by the Inception architecture [3], several local asymmetric 3D convolutional networks, *i.e., MicroNets*, are designed to enhance the effectiveness of the asymmetric 3D convolutional layers. These local networks concatenate multi-scale 3D convolution paths to handle the different scales of spatial-temporal features in videos. Four variants of *MicroNets* are shown in Fig. 3. The *MicroNet-M1* is the base model of local 3D convolutional networks. It only involves traditional 3D convolutional layers following 3D layers with a kernel size of $1 \times 1 \times 1$. These $1 \times 1 \times 1$ 3D convolutional layers decrease the dimension of the last concatenated layer without introducing any representational bottlenecks. The *MicroNet-M1* is more powerfully representative without increasing computational cost compared with the traditional 3D convolutional layer.

Other variants of local 3D convolutional *MicroNets* are designed by incorporating the proposed asymmetric 3D convolutional layers to improve the effectiveness and efficiency, as shown in Fig. 3. *MicroNet-M2* replaces the two traditional $3 \times 3 \times 3$ 3D convolutional layers in the left column of *MicroNet-M1* by three cascaded asymmetric 3D convolutional layers with kernel sizes of $1 \times 5 \times 1$, $1 \times 1 \times 5$ and $3 \times 1 \times 1$. *MicroNet-M3* replaces the traditional $3 \times 3 \times 3$ 3D convolutional layer in the middle column of *MicroNet-M1* by three asymmetric 3D convolutional layers with kernel sizes of $1 \times 3 \times 1$, $1 \times 1 \times 3$ and $3 \times 1 \times 1$. *MicroNet-M4* re-

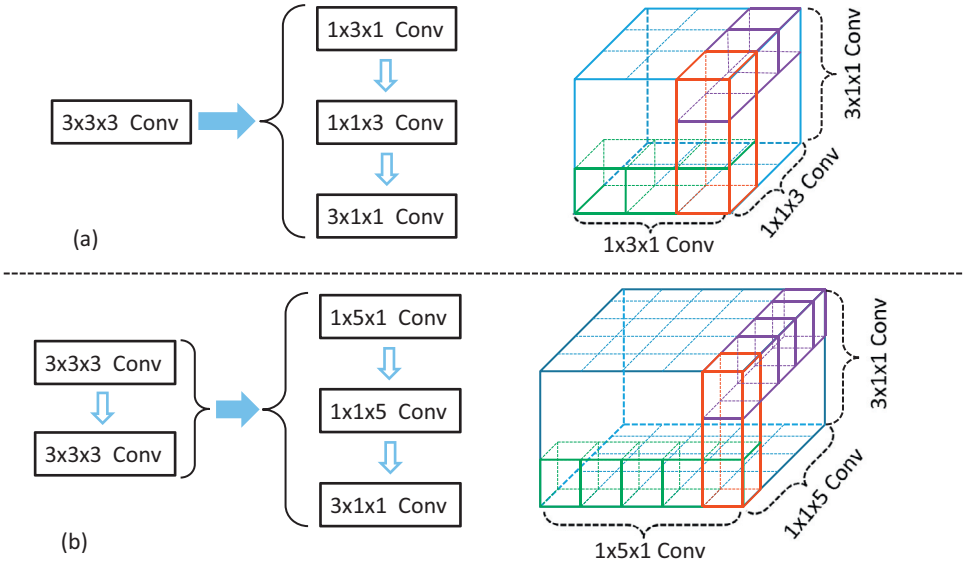


Fig. 2. (a): Approximation of a $3 \times 3 \times 3$ 3D convolutional layer by three asymmetric 3D convolutional layers with same size of receptive field. (b): Factorization of two $3 \times 3 \times 3$ 3D convolutional layers into three asymmetric 3D convolutional layers without reducing the receptive field of the 3D-CNN deep model.

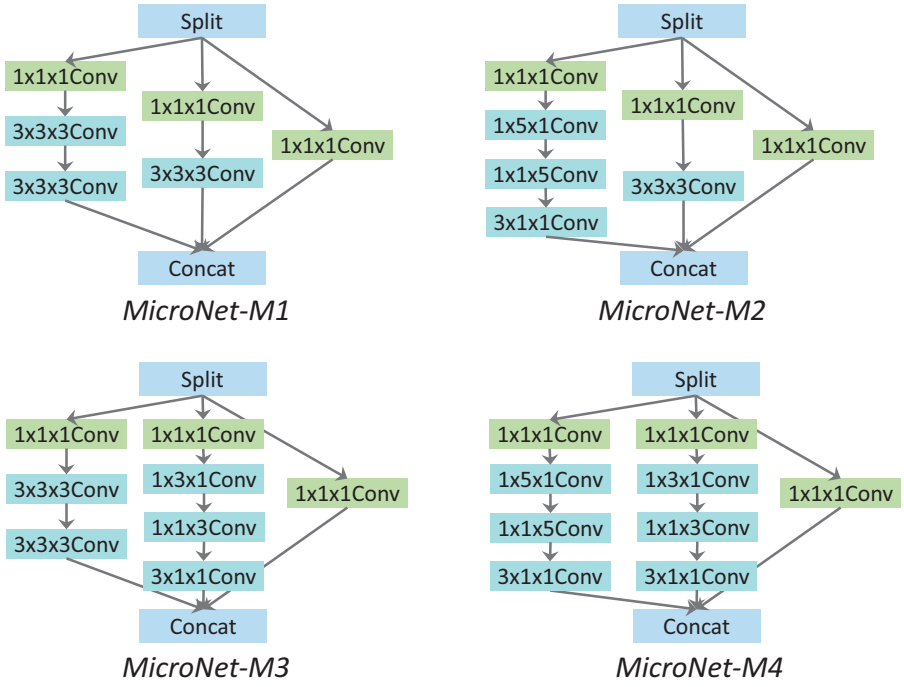


Fig. 3. The four variants of local 3D convolutional *MicroNets*.

places both the left and middle columns of *MicroNet-M1* with asymmetric 3D convolutional layers. These *MicroNets* are used as the building blocks of our 3D convolutional deep model.

3.3. Asymmetric 3D convolutional deep model

The asymmetric 3D convolutional *MicroNets* are used to construct the asymmetric 3D-CNN model. Several *MicroNets* are stacked on two traditional 3D convolutional layers, and then followed by two fully connected layers. The asymmetric 3D-CNN deep architecture is shown in Fig. 4.

As shown in Table 1, the number of parameters in the *Conv1* and *Conv2* layers is only 0.82% of the number in all the convolutional layers and the number of multiplications of the two layers is about 23% of the number of multiplications in all the convolutional layers. So the parameters and multiplications of the first

Table 1
The numbers of parameters and multiplications in each convolutional layer of the C3D [23] and its proportion in all the 3D convolutional layers.

Layers	Parameters/proportion	Multiplications/proportion
Conv1	0.005M (0.02%)	1.04B (1.96%)
Conv2	0.22M (0.80%)	11.09B (20.93%)
Conv3	2.65M (9.48%)	31.12B (58.75%)
Conv4	10.62M (38.41%)	8.33B (15.73%)
Conv5	14.16M (51.21%)	1.39B (2.62%)
Total	27.65M	52.97B

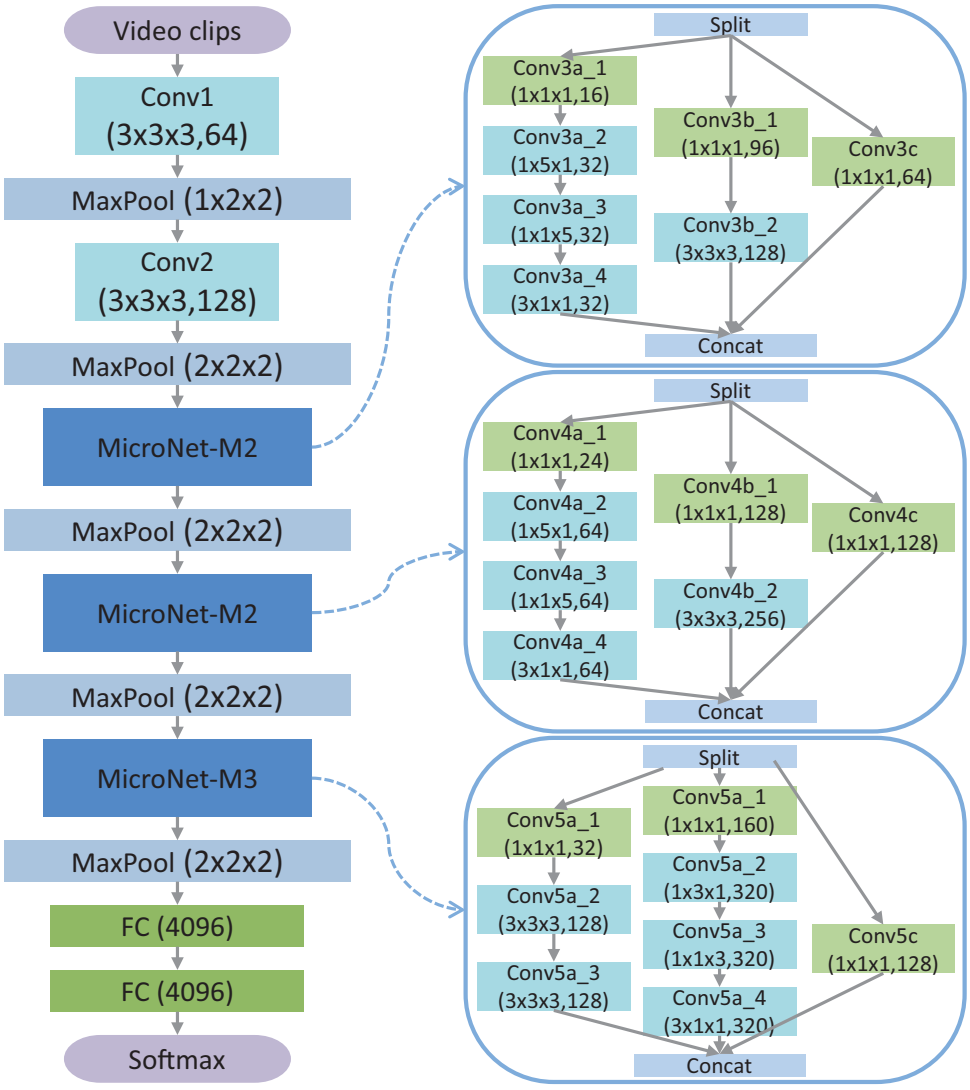


Fig. 4. The layout of the asymmetric 3D-CNN deep architecture. The details of *MicroNets* structure and parameters are shown on the right column.

two layers only form a small part of the 3D-CNN deep model. To achieve a good trade-off between accuracy and complexity, two traditional $3 \times 3 \times 3$ 3D convolutional layers with the input size of $16 \times 112 \times 112$ are stacked in front of the asymmetric 3D convolutional deep network. The first two layers are followed by two local 3D convolutional *MicroNet-M2* with input sizes of $8 \times 28 \times 28$ and $4 \times 14 \times 14$ respectively. The final local convolutional network is *MicroNet-M3* with the input size of $2 \times 7 \times 7$. Following each group of 3D convolutional layers, a 3D Max-pooling layer is used to decrease the spatial-temporal resolution of feature volumes. The number of feature volumes is doubled, to avoid introducing any representational bottleneck in the deep network. Following the last Max-pooling layer, two fully connected layers and one *Softmax* layer are employed for prediction.

The number of output channels and the kernel size of convolutional layers are shown in Fig. 4. The non-linear activation function is set as the Rectified Linear Unit (ReLU) in the deep architecture. The stride of each convolutional layer in the model is set as 1, and an appropriate padding is used for each 3D convolutional layer to keep the output size of each convolutional layer the same as the input size. The kernel sizes and strides of all 3D Max-pooling layers are set as $2 \times 2 \times 2$ except the first Max-pooling layer which has the kernel size and stride of $1 \times 2 \times 2$.

3.4. Multi-source enhanced input

In order to effectively model the dynamic information of actions and to avoid training two deep networks on RGB and Flow frames separately, a multi-source enhanced representation of each video frame, namely RGBF is proposed. It fuses the useful information in the RGB and Flow frames. Firstly, the Flow frame is generated from successive RGB frames by the method [53] in the pre-processing. The Flow is recorded in the form of traditional images, using the horizontal and vertical components of Flow as the first two channels of the frame and the magnitude of Flow as the third channel of the Flow frame. Then, the Flow magnitude is scaled to a range of [0, 1], to generate a movement confidence map. If the magnitudes of the scaled Flow in a region are near to 1, then there is a high probability that movement occurred, and vice versa. The RGBF frame is generated by multiplying each channel of a RGB frame with the corresponding movement confidence map. Formally, the RGBF input at the pixel (x, y) in a frame is computed as:

$$RGBF_{x,y,c} = RGB_{x,y,c} \times \frac{|F|_{x,y} - |F|_{min}}{|F|_{max} - |F|_{min}}, \tag{7}$$

where $c = 1, 2, 3$. $|F|_{x,y}$ denotes the Flow magnitude at the pixel (x, y). $|F|_{max}$ and $|F|_{min}$ denote the maximum and minimum of Flow magnitudes in the frame.

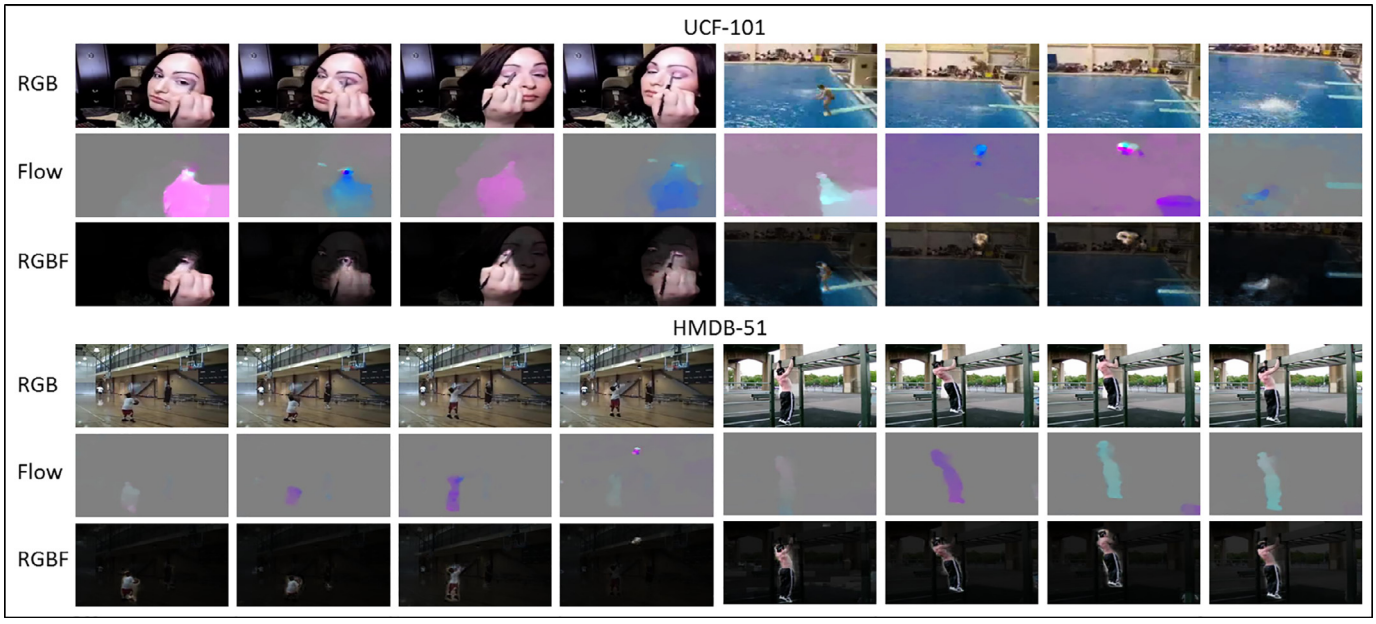


Fig. 5. The comparison of the RGB, Flow and enhanced RGBF frames from the UCF-101 and HMDB-51 datasets.

Several examples of the RGB, Flow, and enhanced RGBF frames in the UCF-101 and HMDB-51 benchmarks are shown in Fig. 5. For each dataset, three rows display the RGB, Flow and RGBF frames from top to bottom, respectively. It can be seen that the RGBF frame highlights the motion related parts and reduces the redundant information compared with the RGB frame. The RGBF frame also adds useful appearance information in motion regions compared with the Flow frame. The enhanced RGBF frames enable the 3D-CNN deep model to learn easily effective spatial-temporal features for action recognition.

4. Experiments

4.1. Datasets

The asymmetric 3D-CNN deep models are tested on two of the most challenging datasets, UCF-101 and HMDB-51. UCF-101 [54] is a dataset of realistic action videos, collected from YouTube, with 101 action categories and 13,320 videos (27 h in total). The UCF-101 dataset has the largest diversity in terms of actions and variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, illumination conditions, etc. The average accuracy is reported for the three standard splits provided in [54]. The HMDB-51 dataset [55] is a large realistic collection of videos from movies and the web. It contains 6849 clips divided into 51 action categories. The average accuracy is reported for the three splits provided by Kuehne et al. [55]. In addition, the 3D-CNN deep model is initialized by pre-training on a large-scale dataset, namely FCVID [56], which contains 91,223 web videos annotated manually into 239 categories. Some categories, such as places, animals and scenes, which do not involve obvious movements are discarded. About 75K videos distributing over 170 categories are used in the pre-training.

4.2. Baselines

Two 3D-CNN deep models which are constructed with only traditional 3D convolutional layers are used as the baselines for comparison with the asymmetric 3D-CNN models. The first baseline is the C3D model [23], which involves 8 3D convolutional layers, 5 Max-pooling layers, and 2 fully connected layers, referred as *c3d-b8*. The architecture and hyper-parameter of the *c3d-b8* are shown in Fig. 6(a). The second baseline is a reduced 3D-CNN model obtained by deleting the *conv3b, conv4b* and *conv5b* convolutional layers from the C3D model and halving the number of channels of the last two convolutional layers and the fully connected layers. The reduced baseline model is denoted as *c3d-b5*. Its architecture and hyper-parameters are shown in Fig. 6(b).

4.3. Experimental settings

The asymmetric 3D-CNN deep models are trained using randomly selected clips with 16 frames in each clip. Each frame in the clip is resized to 128×171 and is cropped into 112×112 spatially. Horizontal flipping and corner cropping are used to prevent over-fitting. The models are trained by Stochastic Gradient Descent (SGD) with a batch size of 16. The momentum and weight-decay are set as 0.9 and 0.0005 respectively. In pre-training on the FCVID video dataset, all frames are cropped and resized to 256×320 spatial size. The base learning rate is set as 0.001 and it is divided by 5 for every 15 epochs. The training is stopped at 50 epochs. In fine-tuning on the UCF-101 and HMDB-51 datasets, the base learning rate is set as 0.0001 and it is divided by 10 for every 8 epochs. The deep models are trained for 20 epochs in total.

In tests, firstly, 25 test clips are sampled from each test video with equal intervals. Each clip consists of continuous 16 frames. Each frame in the clip is resized to 128×171 and then is cropped to obtain the ten standard 112×112 crops, *i.e.,* one center and four corners with horizontal flippings, in the spatial domain. Finally, the average score of all the 250 crops from one video is used to predict the action label.

4.4. Evaluation of asymmetric 3D convolution

Two groups of experiments are designed to evaluate the effectiveness and efficiency of the asymmetric 3D convolution. In the first group of experiments, the third, fourth and fifth 3D convolutional layers of the baseline *c3d-b5* are converted to three cascaded asymmetric 3D convolutional layers, *i.e.,* $1 \times 3 \times 1$, $1 \times 1 \times 3$ and $3 \times 1 \times 1$, to obtain seven asymmetric 3D-CNN variants. The

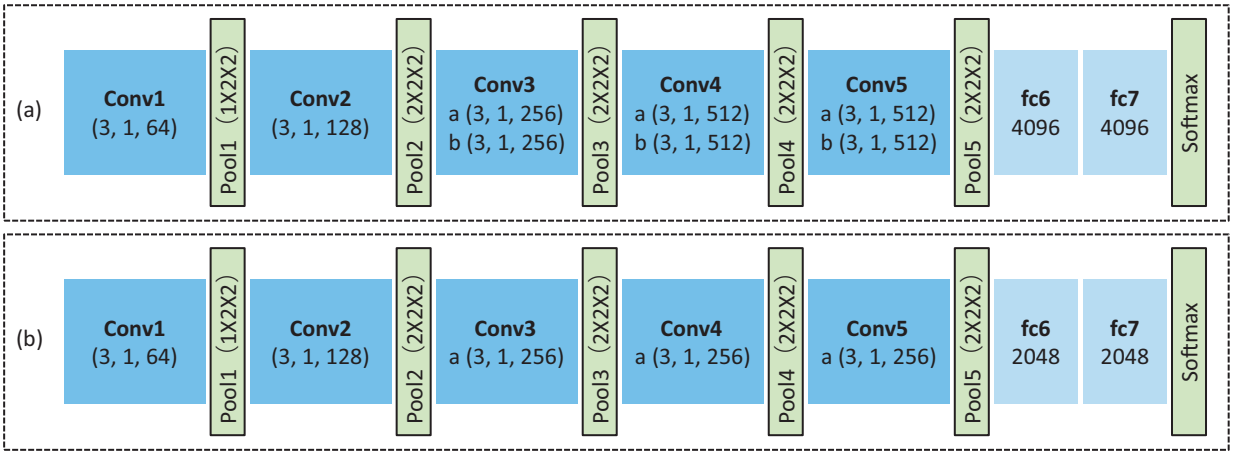


Fig. 6. (a) The 3D-CNN baseline model *c3d-b8*; (b) the 3D-CNN baseline model *c3d-b5*.

Table 2
Two groups of comparison results between the traditional 3D-CNN with their four asymmetric 3D-CNN variants on the UCF-101 dataset.

Models	Parameter numbers	Accuracy	Speed (*s*/*iter*)
c3d-b5	17.44M	45.4%	1.10
b5-asyConv3	17.05M	46.7%	0.85
b5-asyConv4	16.26M	46.5%	0.94
b5-asyConv5	16.26M	**47.2**%	0.94
b5-asyConv34	15.87M	45.7%	0.82
b5-asyConv35	15.87M	46.2%	0.82
b5-asyConv45	15.08M	46.9%	0.83
b5-asyConv345	**14.69M**	45.2%	**0.81**
c3d-b8	78.40M	42.3%	1.28
b8-asyConv3	76.43M	44.3%	0.99
b8-asyConv4	70.53M	44.7%	1.00
b8-asyConv5	67.65M	**45.3**%	1.03
b8-asyConv34	68.56M	42.9%	0.95
b8-asyConv35	65.68M	43.2%	0.97
b8-asyConv45	59.78M	44.1%	0.97
b8-asyConv345	**57.82M**	42.3%	**0.93**

Table 3
Evaluation of the proposed four 3D convolutional *MicroNets* on the UCF-101 dataset.

Models	Paramter numbers	Accuracy
c3d-b5	17.43M	45.7%
b5-asyConv5	16.26M	47.2%
b5-M1	16.07M	46.6%
b5-M2	16.04M	**48.1**%
b5-M3	15.87M	47.5%
b5-M4	**15.84M**	45.8%

names of these asymmetric 3D-CNN variants are prefixed with "b5". The rightmost numbers in the model names denote the traditional 3D convolutional layers which are replaced by the asymmetric 3D convolutional layers, as reported in Table 2. For example, the "*b5-asyConv34*" denotes the network in which the third and fourth 3D convolutional layers in the baseline *c3d-b5* are replaced by the asymmetric 3D convolutional layers. Similarly, to compare with the deeper 3D-CNN baseline model *c3d-b8*, the third, fourth and fifth pairs of 3D convolutional layers of the *c3d-b8* are replaced with three cascaded asymmetric 3D convolutional layers, *i.e.*, $1 \times 5 \times 1$, $1 \times 1 \times 5$ and $3 \times 1 \times 1$. As reported in Table 2, there are seven asymmetric 3D-CNN models extended from the baseline *c3d-b8*. The names of these seven asymmetric 3D-CNN variants are prefixed with "b8". The rightmost numbers in the model names denote the traditional 3D convolutional layers which are replaced by asymmetric 3D convolutional layers.

All the models above are trained from scratch on the UCF-101 dataset. The classification accuracy, the number of parameters and the training speed of the models are reported in Table 2. (1) The proposed asymmetric 3D-CNN models outperform their baseline *c3d-b5* and *c3d-b8* models. In particular, the *b5-asyConv5* asymmetric 3D-CNN model outperforms the *c3d-b5* model by 1.8% and the *b8-asyConv5* achieves the improvement of 2% over the *c3d-b8* model. These results indicate that replacing the traditional 3D convolutional layers with the proposed asymmetric 3D convolutional layers is effective for action recognition. (2) The models using asymmetric 3D convolutional layers in the higher layers are usually better than the models that use the asymmetric 3D convolutional layers in the lower layers. For example, the *b5-asyConv5* outperforms the *b5-asyConv3* and *b5-asyConv4* models, and the *b8-asyConv5* outperforms the *b8-asyConv3* and *b8-asyConv4* models. 3) Increasing the number of asymmetric 3D convolutional layers in the 3D-CNN models does not improve the performance of the asymmetric 3D-CNN further. For example, the *b5-asyConv345* model does not outperform its baseline model. It is probably because that using more asymmetric 3D convolutional layers will increase the depth of the models. The deeper models are more difficult to train from scratch.

The average training times for each iteration of the models are reported in Table 2. All models are trained with the same GPU and batch size. The only difference is the model architecture. The training speed of the asymmetric 3D-CNN models is much higher than that of the baseline models. For example, the *b5-asyConv3* is faster than the *c3d-b5* model by 29% and the best performing *b8-asyConv5* model improves the speed of the baseline model by 25%.

4.5. Evaluation of asymmetric 3D convolutional micronets

To evaluate the effectiveness of the proposed four local 3D convolutional *MicroNets*, four 3D-CNN deep models are constructed by converting the *conv5* layer of the *c3d-b5* model to each of the *MicroNets*. The resulting asymmetric 3D convolutional networks are denoted as *b5-M1, b5-M2, b5-M3* and *b5-M4* respectively. All the models are trained from scratch on the UCF-101 dataset. As shown in Table 3, the four asymmetric 3D-CNN models outperform the baseline *c3d-b5* model. In particular, the *b5-M2* model achieves the best performance among the four 3D-CNN variants. It outperforms the *c3d-b5* model by over 2%. Therefore, adding the 3D convolutional *MicroNets* to the baseline model yields an obvious improvement. In addition, the *b5-M2* and *b5-M3* models both outperform the *b5-asyConv5* model, as shown in Table 3. The *MicroNets*

Table 4
Evaluation of three asymmetric 3D-CNN variants, finetuned on the UCF-101 dataset.

Models	Accuracy
c3d-b8	84.6%
Asymmetric 3D-CNN (M2)	86.2%
Asymmetric 3D-CNN (M3)	85.1%
Asymmetric 3D-CNN	**86.4**%

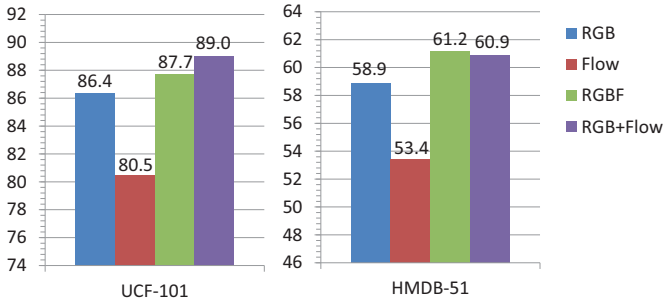


Fig. 7. Evaluating the performance of the asymmetric 3D-CNN deep model fine-tuned on UCF-101 and HMDB-51 datasets with three kinds of inputs.

Table 5
Comparison with current state-of-the-art methods on UCF-101 dataset.

IDT [33]	85.9%
IDT (higher-dimension) [57]	87.9%
MIFS (L = 3) [58]	89.1%
Slow Fusion [39]	65.4%
VGG16+Images on Web [59]	83.5%
Two-stream (fusion by averaging) [11]	86.9%
Two-stream (fusion by SVM) [11]	88.0%
Fusion Two-stream [27]	91.8%
Two-stream (VGG-16) [25]	91.4%
LRCN (weighted average) [26]	82.9%
C3D (1 net+SVM) [23]	82.3%
C3D (3 net+SVM) [23]	85.2%
C3D+IDT [23]	90.4%
T-CNN [49]	87.5%
FstCN (averaging fusion) [21]	87.9%
Asymmetric 3D-CNN (RGBF)	87.7%
Asymmetric 3D-CNN (RGB+RGBF)	89.5%
Asymmetric 3D-CNN (RGB+RGBF+IDT)	92.6%

are thus more effective than a simple cascaded of asymmetric 3D convolutional layers. The *MicroNet-M2* and *MicroNet-M3* local networks are used later in the asymmetric 3D-CNN deep models.

4.6. Evaluation of the 3D-CNN model and the RGBF input

To avoid over-fitting of the 3D-CNN deep models caused by training on the limited quantity of videos, the 3D-CNN models are pre-trained on the large-scale FCVID video dataset. Subsequently, all the layers are fine-tuned on the target dataset with a ten times higher learning rate for the last fully connected layer.

The *Asymmetric 3D-CNN*, as described in Section 3 and Fig. 4, is compared with two variants of the asymmetric 3D-CNN deep models, *i.e., Asymmetric 3D-CNN (M2)* and *Asymmetric 3D-CNN (M3)*, which are similar to the *Asymmetric 3D-CNN* architecture. The *Asymmetric 3D-CNN (M2)* model only uses the local 3D convolutional *MicroNet-M2* and the *Asymmetric 3D-CNN (M3)* model only uses the *MicroNet-M3*. As shown in Table 4, the *Asymmetric 3D-CNN (M2)* deep model achieves better performance than the *Asymmetric 3D-CNN (M3)*. The *Asymmetric 3D-CNN* outperforms both the *Asymmetric 3D-CNN(M2)* and *Asymmetric 3D-CNN(M3)* models. This demonstrates that diverse local networks allow the 3D-CNN deep model to learn complementary spatial-temporal features from videos for action recognition.

In addition, the *Asymmetric 3D-CNN* model is evaluated on two of the most challenging benchmarks. The model is fed with three kinds of inputs, *i.e.,* the RGB, Flow and enhanced RGBF frames. The results of the experiments are shown in Fig. 7. The performance with the enhanced RGBF frames is better than the performances obtained with the RGB and Flow inputs separately. It even outperforms the fusion result of the two networks fed with RGB and Flow frames separately in the HMDB-51 dataset. This indicates that the simply enhanced input is effective. The results obtained from the Flow input are much worse than those obtained from the RGB input. This is different from the results obtained in 2D-CNN based action recognition methods [11,25,26]. It indicates that the 3D-CNN is more appropriate for extracting spatial-temporal features from raw videos compared with 2D-CNN. Moreover, the *c3d-b8* baseline model, fed with the RGB frames and pre-trained on the FCVID dataset, achieves 84.6% and 56.7% accuracy on the UCF-101 and HMDB-51 datasets respectively. The *Asymmetric 3D-CNN* deep model outperforms the *c3d-b8* model by 1.8% and 2.2% respectively, which demonstrates the effectiveness of the *Asymmetric 3D-CNN* model.

4.7. Comparison with the state-of-the-arts

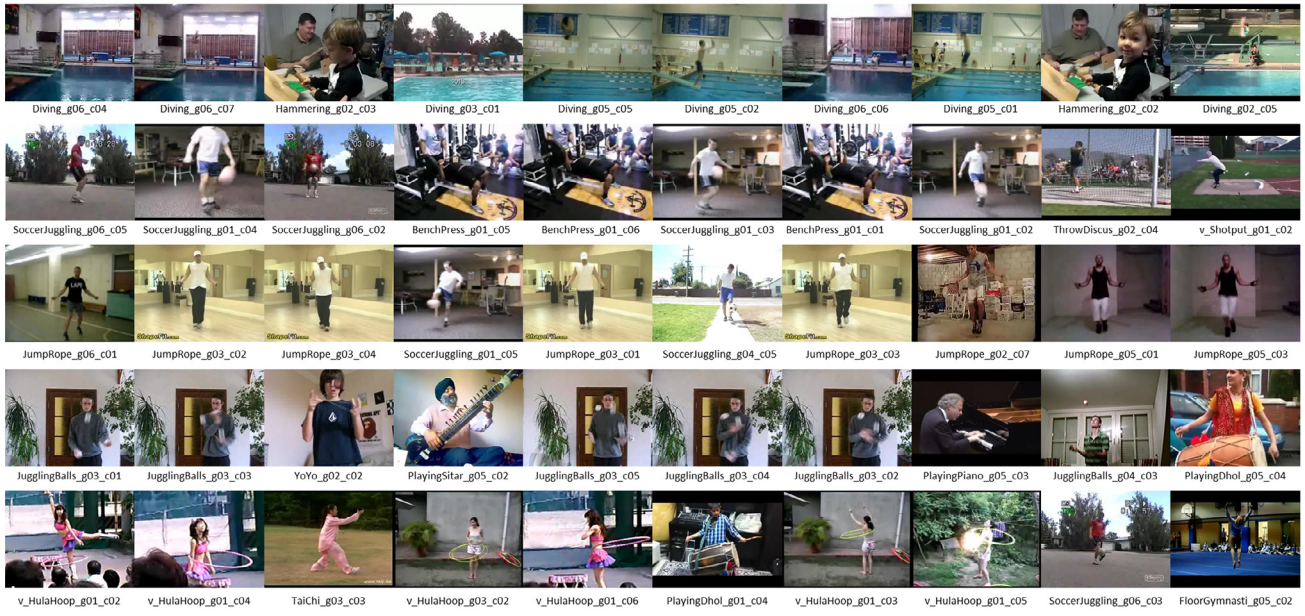
The asymmetric 3D-CNN models are compared with current state-of-the-art methods on the UCF-101 dataset in Table 5. The *Asymmetric 3D-CNN* model fed with RGBF frames achieves a better performance than the most current state-of-the-art models, even though many of them fuse two networks of SpatialNet and TemporalNet [11,26]. Fusing the *Softmax* scores of the two networks fed with RGB and RGBF frames respectively can further improve the performance of the 3D-CNN deep model. The *Asymmetric 3D-CNN (RGB+RGBF)* model outperforms all of the traditional methods, such as the Improved Dense Trajectories (IDT) [33] and the Multi-skIp Feature Stacking (MIFS) [58]. Compared with 2D-CNN based methods, the *Asymmetric 3D-CNN (RGB+RGBF)* model outperforms Slow Fusion [39] by over 24% and outperforms Two-stream (fusion by averaging) [11] by 2.6%. Compared with other 3D-CNN based action recognition models, the asymmetric 3D-CNN model outperforms the C3D model [23] by over 4% and outperforms the FstCN model [21] by 1.6%. Moreover, it outperforms the T-CNN model [49] by 2.0%. In addition, the deep features that are extracted by the 3D-CNN deep models fed with RGB and RGBF individually are combined with the widely used traditional IDT [33] features. Then the actions are classified by a multi-class linear SVM. The resulting *Asymmetric 3D-CNN (RGB+RGBF+IDT)* outperforms the newest state-of-the-art methods [25,27] on the UCF-101 dataset.

In Table 6, the *Asymmetric 3D-CNN* model is compared with current state-of-the-art methods on the HMDB-51 dataset. The *Asymmetric 3D-CNN* model fed with the enhanced RGBF frames outperforms most current state-of-the-art methods. The fusion model, *Asymmetric 3D-CNN (RGB+RGBF)*, fed with RGB and RGBF frames outperforms Two-stream [11] by 4.1% and outperforms FstCN [21] by 4.4%. Finally, the deep features which are extracted from RGB and RGBF frames are combined with the traditional IDT [33] features, which are used to train a linear SVM to classify actions. The resulting *Asymmetric 3D-CNN (RGB+RGBF+IDT)* outperforms the newest state-of-the-art methods [27,60,61] on the HMDB-51 dataset.

4.8. Visualization of model learning results

To get an intuitive understanding on what is learnt by the *Asymmetric 3D-CNN* model, the learned convolutional features and

(a) Visualization of multiple convolutional layers on the HMDB-51 dataset.



(b) Visualization of the last convolutional layer on the UCF-101 dataset.

Fig. 8. Visualization of what is learned by the asymmetric 3D-CNN model from the UCF-101 and HMDB-51 datasets.

Table 6
Comparison with current state-of-the-art methods on HMDB-51 dataset.

IDT [33]	57.2%
IDT (higher-dimension) [57]	61.1%
MIFS (L = 3) [58]	65.1%
TDD [40]	63.2%
KVMF [60]	63.3%
Two-stream (fusion by SVM) [11]	59.4%
Fusion Two-stream [27]	64.6%
Action-Transformations [61]	63.4%
FstCN (averaging fusion) [21]	58.6%
FstCN (SCI fusion) [21]	59.1%
Asymmetric 3D-CNN (RGBF)	61.2%
Asymmetric 3D-CNN (RGB+RGBF)	63.5%
Asymmetric 3D-CNN (RGB+RGBF+IDT)	**65.4**%

the clustering ability of the last convolutional layer are shown in Fig. 8. Firstly, the *Asymmetric 3D-CNN* deep model is pre-trained on the large-scale FCVID dataset to get a better initialization. Then, it is finetuned on the UCF-101 or HMDB-51 datasets respectively.

Fig. 8(a) shows the learned features of several convolutional layers of the *Asymmetric 3D-CNN* deep model which is finetuned on the HMDB-51 dataset. Specifically, three examples are chosen at random from the *test* set of the HMDB-51 dataset and fed into the 3D-CNN model, to extract the learned features at multiple convolutional layers. One channel of feature maps is presented for the *Conv1, Conv2* and *Conv3a_4* convolutional layers for each example. From bottom to top (by depth order) of each example, the *Asymmetric 3D-CNN* model focuses on the appearance at first, then the appearance becomes obscure, and later the motion regions are highlighted.

To visualize the clustering ability of the *Asymmetric 3D-CNN* deep model, one clip is chosen from each video in the UCF-101 *test* set. All the clips are fed to the*Asymmetric 3D-CNN* model to compute the activation of the last convolutional layer. A specific neuron is singled out in the last convolutional layer and the clips are sorted by the neuron activation from highest to the lowest. The top 10 clips of five neurons are presented in Fig. 8(b). Each row corresponds to one neuron. It can be seen that each particular neuron is only fired by some similar actions. In the top row, most clips are "Diving" actions from different subjects and scenes, but two clips of "Hammering" action are included. In fact, the configuration of the person is like a hammer and the "Diving" action shows a person jumping from a diving board, which is like a hammer "jumping" from a board. In the second row, a soccer running up and down in the "SoccerJuggling" action has a similar configure and motion with a bench running up and down in the "BenchPress" action. In the third row, the actions for "JumpRope" and "SoccerJuggling" are very similar. In the last two rows, most entries show the same or similar actions carried out by a number of different actors in different scenes.

5. Conclusion

This paper has proposed an efficient 3D convolution method by approximating the traditional 3D convolution with three cascaded one-directional asymmetric 3D convolutions. Then the local asymmetric 3D convolutional *MicroNets* have improved the effectiveness of the asymmetric 3D convolutional layers by incorporating multi-scale spatial-temporal features. Finally, an asymmetric 3D-CNN deep model has been built by stacking the asymmetric 3D convolutional *MicroNets*. Additionally, the proposed multi-source enhanced RGBF input has decreased the computational cost further by avoiding training two networks on RGB and Flow inputs individually. The asymmetric 3D-CNN model has outperformed the most comparable 3D-CNN models and many action recognition state-of-the-art methods on two challenging datasets. In the future work, we expect to design residual asymmetric 3D convolutional *MicroNets* by introducing the shortcut connection in the local networks and increase the depth of our 3D-CNN model by stacking more 3D convolutional *MicroNets* to further improve the performance of action recognition.

References

[1] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[2] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: International Conference on Learning Representations, 2015, pp. 1–14.
[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.
[4] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[5] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.
[6] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems, 2015, pp. 91–99.
[7] E. Ohn-Bar, M.M. Trivedi, Multi-scale volumes for deep object detection and localization, Pattern Recognit. 61 (2017) 557–572.
[8] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.
[9] Y. Wang, J. Liu, Y. Li, J. Fu, M. Xu, H. Lu, Hierarchically supervised deconvolutional network for semantic video segmentation, Pattern Recognit. 64 (2017) 437–445.
[10] R. Hou, C. Chen, M. Shah, An end-to-end 3d convolutional neural network for action detection and segmentation in videos, arXiv preprint arXiv:1712.01111 (2017).
[11] K. Simonyan, A. Zisserman, Two-stream convolutional networks for action recognition in videos, in: Advances in Neural Information Processing Systems, 2014, pp. 568–576.
[12] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, L. Van Gool, Temporal segment networks: towards good practices for deep action recognition, in: Springer European Conference on Computer Vision, 2016, pp. 20–36.
[13] E.P. Ijjina, K.M. Chalavadi, Human action recognition using genetic algorithms and convolutional neural networks, Pattern Recognit. 59 (2016) 199–212.
[14] M. Ma, N. Marturi, Y. Li, A. Leonardis, R. Stolkin, Region-sequence based six-stream CNN features for general and fine-grained human action recognition in videos, Pattern Recognit. 76 (2018) 506–521.
[15] E.L. Denton, W. Zaremba, J. Bruna, Y. LeCun, R. Fergus, Exploiting linear structure within convolutional networks for efficient evaluation, in: Advances in Neural Information Processing Systems, 2014, pp. 1269–1277.
[16] M. Jaderberg, A. Vedaldi, A. Zisserman, Speeding up convolutional neural networks with low rank expansions, in: British Machine Vision Conference, 2014, pp. 1–13.
[17] Y. Ioannou, D. Robertson, J. Shotton, R. Cipolla, A. Criminisi, Training CNNs with low-rank filters for efficient image classification, J. Asian Stud. 62 (3) (2015) 952–953.
[18] J. Jin, A. Dundar, E. Culurciello, Flattened convolutional neural networks for feedforward acceleration, arXiv preprint arXiv:1412.5474 (2014).
[19] W. Min, L. Baoyuan, F. Hassan, Factorized convolutional neural networks, arXiv preprint arXiv:1606.0433 (2016).
[20] B. Liu, M. Wang, H. Foroosh, M. Tappen, M. Penksy, Sparse convolutional neural networks, in: IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 806–814.
[21] L. Sun, K. Jia, D.-Y. Yeung, B.E. Shi, Human action recognition using factorized spatio-temporal convolutional networks, in: IEEE International Conference on Computer Vision, 2015, pp. 4597–4605.
[22] 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.
[23] T. Du, L. Bourdev, R. Fergus, L. Torresani, Learning spatiotemporal features with 3D convolutional networks, in: IEEE International Conference on Computer Vision, 2015, pp. 4489–4497.
[24] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, F.-F. Li, ImageNet: A large-scale hierarchical image database, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.
[25] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, Towards good practices for very deep two-stream convnets, arXiv preprint arXiv:1507.02159 (2015).
[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: IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2625–2634.

[27] C. Feichtenhofer, A. Pinz, A. Zisserman, Convolutional two-stream network fusion for video action recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1933–1941.
[28] I. Laptev, On space-time interest points, Int. J. Comput. Vis. 64 (2–3) (2005) 107–123.
[29] P. Dollár, V. Rabaud, G. Cottrell, S. Belongie, Behavior recognition via sparse spatio-temporal features, in: IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005, pp. 65–72.
[30] G. Willems, T. Tuytelaars, L. Van Gool, An efficient dense and scale-invariant spatio-temporal interest point detector, in: Springer European Conference on Computer Vision, 2008, pp. 650–663.
[31] P. Scovanner, S. Ali, M. Shah, A 3-dimensional SIFT descriptor and its application to action recognition, in: ACM International Conference on Multimedia, 2007, pp. 357–360.
[32] H. Wang, A. Kläser, C. Schmid, C.-L. Liu, Action recognition by dense trajectories, in: IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 3169–3176.
[33] H. Wang, C. Schmid, Action recognition with improved trajectories, in: IEEE International Conference on Computer Vision, 2013, pp. 3551–3558.
[34] C. Harris, A combined corner and edge detector (1988), in: Proc Alvey Vision Conf, 3, 1988, pp. 147–151.
[35] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: IEEE Conference on Computer Vision Pattern Recognition, 2013, pp. 886–893.
[36] N. Dalal, B. Triggs, C. Schmid, Human detection using oriented histograms of flow and appearance, in: Springer European Conference on Computer Vision, 2006, pp. 428–441.
[37] H. Bay, T. Tuytelaars, L. Van Gool, Surf: Speeded up robust features, in: Springer European conference on computer vision, 2006, pp. 404–417.
[38] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110.
[39] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, F.-F. Li, Large-scale video classification with convolutional neural networks, in: IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1725–1732.
[40] L. Wang, Y. Qiao, X. Tang, Action recognition with trajectory-pooled deep-convolutional descriptors, in: IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4305–4314.
[41] J.L. Elman, Finding structure in time, Cogn. Sci. 14 (2) (1990) 179–211.
[42] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[43] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, A. Baskurt, Action classification in soccer videos with long short-term memory recurrent neural networks, in: Springer International Conference on Artificial Neural Networks, 2010, pp. 154–159.
[44] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, A. Baskurt, Sequential deep learning for human action recognition, in: Springer International Workshop on Human Behavior Understanding, 2011, pp. 29–39.
[45] Y.-H.N. Joe, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, G. Toderici, Beyond short snippets: deep networks for video classification, in: IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4694–4702.
[46] H. Gammulle, S. Denman, S. Sridharan, C. Fookes, Two stream LSTM: a deep fusion framework for human action recognition, in: IEEE Winter Conference on Applications of Computer Vision, 2017, pp. 177–186.
[47] S. Sharma, R. Kiros, R. Salakhutdinov, Action recognition using visual attention, arXiv preprint arXiv:1511.04119 (2015).
[48] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, A. Courville, Describing videos by exploiting temporal structure, in: IEEE international conference on computer vision, 2015, pp. 4507–4515.
[49] R. Hou, C. Chen, M. Shah, Tube convolutional neural network (T-CNN) for action detection in videos, arXiv preprint arXiv:1703.10664 (2017).
[50] G. Varol, I. Laptev, C. Schmid, Long-term temporal convolutions for action recognition, IEEE Trans. Pattern Anal. Mach. Intell. PP (99) (2017). 1–1
[51] J. Carreira, A. Zisserman, Quo vadis, action recognition? A new model and the kinetics dataset, arXiv preprint arXiv:1705.07750 (2017).
[52] Z. Liu, C. Zhang, Y. Tian, 3D-based deep convolutional neural network for action recognition with depth sequences, Image Vis. Comput. 55 (2016) 93–100.
[53] T. Brox, A. Bruhn, N. Papenberg, J. Weickert, High accuracy optical flow estimation based on a theory for warping, in: Springer European Conference on Computer Vision, 2004, pp. 25–36.
[54] 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 (2012).
[55] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, T. Serre, HMDB: a large video database for human motion recognition, in: International Conference on Computer Vision, 2011, pp. 2556–2563.
[56] Y.-G. Jiang, Z. Wu, J. Wang, X. Xue, S.-F. Chang, Exploiting feature and class relationships in video categorization with regularized deep neural networks, IEEE Trans. Pattern Anal. Mach. Intell. 6 (1) (2017) 1–14.
[57] X. Peng, L. Wang, X. Wang, Y. Qiao, Bag of visual words and fusion methods for action recognition: comprehensive study and good practice, Comput. Vis. Image Underst. 150 (2016) 109–125.
[58] Z. Lan, M. Lin, X. Li, A.G. Hauptmann, B. Raj, Beyond Gaussian pyramid: multi-skip feature stacking for action recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 204–212.
[59] S. Ma, S.A. Bargal, J. Zhang, L. Sigal, S. Sclaroff, Do less and achieve more: training cnns for action recognition utilizing action images from the web, Pattern Recognit. 68 (2017) 334–345.
[60] W. Zhu, J. Hu, G. Sun, X. Cao, Y. Qiao, A key volume mining deep framework for action recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1991–1999.
[61] X. wang, F. Ali, G. Abhinav, Actions t̃ransformations, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2658–2667.

Hao Yang received the B.S. degree from China University of Petroleum, China, in 2014. Currently, he is a Ph.D. student training in the Institute of Automation, Chinese Academy of Sciences. His research interests include motion analyses and action recognition.

Chunfeng Yuan received the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences (CASIA), Beijing, China, in 2010. She was a visiting scholar at University of Adelaide, Australia in 2010, and in the Internet Media Group and the Media Computing Group at Microsoft Research Asia in 2016. She is currently a associate professor at the CASIA. Her research interests and publications range from statistics to computer vision, including sparse representation, deep learning, action recognition, and event detection.

Bing Li received the Ph.D. degree from the Department of Computer Science and Engineering, Beijing Jiaotong University, China, in 2009. He is currently an Associate Professor in the Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include video understanding, color constancy, visual saliency, and web content mining.

Yang Du received the B.S. degree from Beijing Jiaotong University, China, in 2014. He is currently a Ph.D. student training in the Institute of Automation, Chinese Academy of Sciences. His research interests include Self Organizing Maps (SOM) and action recognition.

Junliang Xing received the Ph.D. degree in computer science from Tsinghua University, Beijing, China, in 2012. He is currently an associate professor in the Institute of Automation,Chinese Academy of Sciences, Beijing, China. Dr. Xing was the recipient of Google Ph.D. Fellowship 2011, the Excellent Student Scholarships at Xi'an Jiaotong University from 2004 to 2007 and at Tsinghua University from 2009 to 2011. He has published more than 50 papers on inter- national journals and conferences. His current research interests mainly focus on computer vision problems related to faces and humans.

Weiming Hu received the Ph.D. degree from the Department of Computer Science and Engineering, Zhejiang University in 1998. From 1998 to 2000, he was a postdoctoral research fellow with the Institute of Computer Science and Technology, Peking University. Currently, he is a full professor in the Institute of Automation, Chinese Academy of Sciences. He has published more than 200 papers on international journals and conferences. His research interests include visual motion analysis and recognition of web objectionable information.

Stephen J. Maybank received the BA degree in mathematics from Kings College Cambridge in 1976, and the Ph.D. degree in computer science from Birkbeck College, University of London in 1988. He is currently a professor in the Department of Computer Science and Information Systems, Birkbeck College. His research interests include the geometry of multiple images, camera calibration, visual surveillance, etc. He is a fellow of the IEEE.