Multimed Tools Appl (2018) 77:6955–6968
DOI 10.1007/s11042-017-4614-0

Action recognition based on hierarchical dynamic Bayesian network

Qinkun Xiao[1] · **Ren Song**[1]

Received: 7 July 2016 / Revised: 23 January 2017 / Accepted: 16 March 2017 /
Published online: 6 April 2017

Abstract In this paper, a novel action recognition method is proposed based on hierarchical dynamic Bayesian network (HDBN). The algorithm is divided into system learning stage and action recognition stage. In the stage of system learning, the video features are extracted using deep neural networks firstly, and using hierarchical clustering and assisting manually, a hierarchical action semantic dictionary (HASD) is built. The next, we construct the HDBN graph model to present video sequence. In the stage of recognition, we first get the representative frames of unknown video using deep neural networks. The features are inputted into the HDBN, and the HDBN inference is used to get recognition results. The testing results show the proposed method is promising.

Keywords Action recognition · HDBN · Deep neural networks · HASD · Graph model

1 Introduction

In recent years, human action recognition has become a core issue in field of computer vision. Because of complexity and uncertainly, action identification is still a very challenging subject. Many action recognition methods tend to design descriptors and classifying based feature matching [11, 13]. The previous action recognition methods main include two classes, i.e. feature description and action classification. According to [28], feature representation is always key task for recognizing actions. In general, the feature presentation usually is divided into global representations and local representations. The global feature records total image presentation, however, the global feature is often disturbed by occlusions, viewpoints changing and noises. The global-based feature includes optical flow-based presentation [27], silhouette-based

✉ Qinkun Xiao
xiaoqinkun10000@163.com

[1] Department of Electronic Information Engineering, Xi'an Technological University, Xi'an city, Shaanxi, People's Republic of China 710032

descriptor [3], edge-based features [34], and motion history image (MHI) [2], and so on. The local feature always describes patches independently, and the patches are combined together to build space–time model [17], such as HOG [4] and SURF [1]. The local descriptor presents action video more effectively, especially for noises images and partial occlusions images. However, processing related interest points is high time cost.

In this paper, we present a hierarchical graph model framework for meeting complex videos semantic identification. The highlights of this method include two points, one is hierarchical action semantic dictionary construction, and another is hierarchical dynamic Bayesian network semantic inference model.

The motive of the proposed approach is to recognize actions with the higher accuracy. We consider the problem based 3 points: (1) for reducing video dimension and recognition time cost, we use deep neural networks [19] to extract video features firstly, and the Aligned Cluster Analysis (ACA) [35] is utilized to get representation frames. As we known, the selected discriminating features always are better performance. Based the deep neural network and the ACA, the better discriminating features are got from original action video. (2) For enhancing robustness of recognition, we propose to construct the hierarchical action semantic dictionary (HASD). As we known, high level semantic analysis is always very important for complex and uncertain identification problem, dictionary-based classifier has better performance to recognize action. In this paper, we propose dictionary-based recognition to enhance recognition robustness. (3) At the same time, it is proved that probability graph model is an efficient tool to dig hidden state information [29]. The dynamic Bayesian network (DBN) is promising method to present random time series signals entirely. Hence, we select the HDBN to present action. The HDBN-based signals processing can accomplish 2 tasks: one is to present action entirely and clearly, two is to dig more hidden semantic state information. (4) Based on the above-mentioned, we combine high level semantic analysis and graph model together to obtain effective representation. Based on the HDBN inference and semantic analysis, and the HDBN + HASD-based method can finish recognition task effectively.

The rest of this paper is organized as follows. The related works is described in Section 2, and the proposed approach is described in Section 3. Section 4 compares proposed method with the existing models. Finally, conclusions are given in Section 5.

2 Related works

Many regular methods are utilized to classify human actions. For example, in [11], a multi-categories SVM classifier is proposed, which uses dynamic programming to segment sequences. In [23], local descriptors are combined into SVM for action recognition. In [6], K-nearest neighbor classifiers are used to predict action labels. However, those regular recognition methods hardly capture dynamics space-time sequence information.

Besides of regular classifier approaches, many sequence labeling models based on graph model are used to recognize action, such as condition random fields (CRF) [15], hidden Markov model (HMM) [26], dynamic Bayesian network (DBN) [33], are all good tools to analyze actions. Many graph-based models are utilized in field of pattern recognition, like HMM [26]. In [16], a linear graph sequence model is proposed to build nonlinear mapping of semantic and action recognition. In [5], one-hidden-layer neural network is combined with graph model to model sequence information, the testing results are with lesser computation cost than kernel CRF.

Recently, with deep learning technology developing, many recognition methods combined Convolutional Neural Network (CNN) with graph model are presented. In [12], a novel two-stage hierarchical framework of 3D gesture recognition is proposed. On the first level, a new clustering is presented and a five-dimensional feature vector is used to explore the most relevant body movement. In the second level, two modules are utilized, including motion feature extraction module and action figure module. The testing results at Microsoft Action3D datasets proved that the method is perfect. The recognition rate reaches 95.56%. In [24], a human behavior recognition method based on genetic algorithm (GA) and the CNN is proposed. Gradient descent algorithm is used to train the CNN classifier. Using the global and local search ability of GA, an optimal solution is found by gradient descent algorithm quickly. Testing results suggest that the GA-based classifier can improve recognition performance. The [14] study the behavior recognition method based on RGB-D. They use sequence kernel descriptors to present action scene and build a kernel framework and hierarchical kernel descriptors as higher levels for classification. The [29] propose a 3D human motion recognition method combined the HMM with 3D human body joints movement. Firstly, dynamic instantaneous velocity, direction and 3D trajectory are obtained, and then through unsupervised learning, all information is fused by the self-organizing mapping to generate discrete symbol, the next, the Baum-Welch and Viterbi algorithm are utilized to identify human action. In [32], a new realistic human action recognition approach is proposed. For building a mid-level description, the foreground action is decomposed into several spatio-temporal action parts, and [32] also utilizes a density detector to handle action segmentation. Lastly, a graph model is put into multiple-instance learning framework to identify action. In [7], a simple 2D CNN is extended to a concatenated 3D network. The 3D network model is utilized for content-based video recognition. Experimental results show the proposed model is more general and flexible than other methods. The [25] uses Manifold Regularized Least Square Regression and Semi-supervised Hierarchical Regression Algorithm to recognize action. The [31] proposes a novel graphical structure model to deal with viewpoint changing human action recognition, and unknown viewpoint action can be recognized using improved forward algorithm.

3 Approach

3.1 Overview

Our method can be shown in Fig. 1, the approach is divided into two stages, including system learning and action recognition. In the stage of system learning, firstly, the video key-frame images in training dataset are put into Deep Brief Network to extract the features automatically. Secondly, based on obtained images features, we use the ACA to get representative frames (denoted as rf_i). According to representative frames images, a hierarchical video semantic dictionary (HVSD) is built by clustering and assisting manfully. Based on the HVSD learning, we can obtain parameters of hierarchical dynamic Bayesian networks (HDBN), and the HDBN

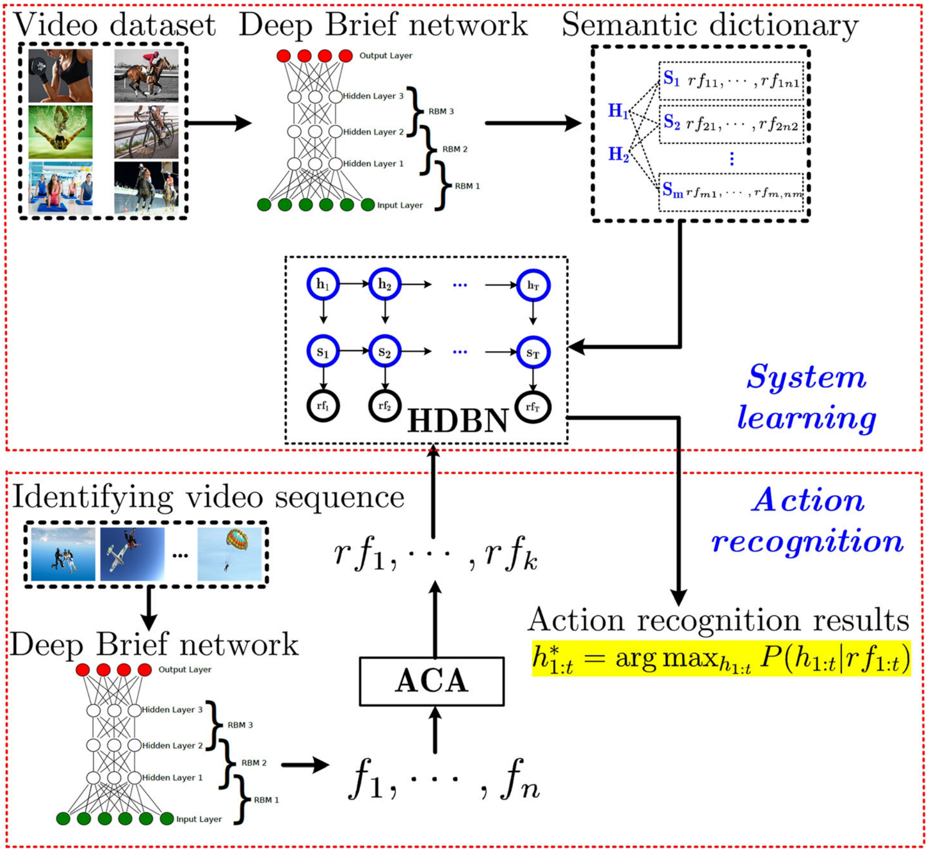


Fig. 1 The HDBN presentation for action recognition

can be as a classifier to identify video semantic. In the stage of action recognition, video features (f_1 , ⋯ ,f_n) are extracted using Deep Brief Network too, then the ACA is used to get representative frames (rf_1 , ⋯ , rf_k) as the HDBN's input, lastly, the semantic classify results is got by the HDBN inference.

3.2 System learning

The aim of system learning is to build action semantic dictionary (ASD) and construct the HDBN model.

(1) *Video information pre-processing*. Firstly, based on motion analysis [30], we get key-frames of training videos, and features of the key-frames are extracted using Deep Brief Networks [19]. Based on obtained features, the ACA [35] is used to obtain the representative frames.

In this paper, we use the Convolutional Deep Brief Networks (CDBN) [19] to extract image features. The CDBN is one kind of deep neural network, similar to Deep Brief Networks (DBN), which is made up of Restricted Boltzmann Machine (RBM) stacked on top of one another, the CDBN consists of several max-pooling and Convolutional Restricted Boltzmann Machine (CRBM). The energy function of the CDBN is sum of energy functions all individual layers. In CDBN training, the greedily trained weights are used as the initialization, and the network training is accomplished with the jointly train all the weights for the entire CDBN, the training method may be a potentially better solution [19].

We use the ACA [35] to get the representative frames. The ACA is a clustering algorithm, as shown in Fig. 2, given a video sequence, in a first level of the hierarchy, the ACA divides each of the actions into action primitives of smaller temporal scale, such as "segmentation 1", "segmentation 2", and so on. The next, some key frames ($rf_1,\ldots,rf_4$) are selected as representative frames in term of the shortest distance to clustering centers.

(2) *Constructing the hierarchical semantic dictionary.* For each action training video, the video is manually divided into smaller scale clips based on the similarity. Let i-th action be $\mathbf{A}_i$, the $\mathbf{A}_i$ is divided into n root level semantic clips, each clip includes only one semantic. If use S_i to denote root level semantic clip, and the $\mathbf{A}_i = C_1 \cup C_2 \cdots \cup C_n$, where $C_i \cap C_j = \varnothing\,(i \neq j)$. The each root level semantic clips are presented by several representative frames.

By unsupervised clustering and manual adjustment, we put the same semantics clips together to establish semantic groups, all root level semantic clips are recombined into k action semantic groups: $\{S^i\}_{i=1:\omega 1}$. Let C_j^k be j-th clip in semantic group S^k, and rf_k^{ij} be k-th representative frame in j-th clip of group S^i. Based on above assumption, the 1st level semantic dictionary is: $\mathbf{ASD}_{level1} = \{S^i\}_{i=1:\omega 1}$, where $S^i = \left\{C_j^i\right\}_{j=1:ni}$, $C_j^i = \left\{rf_k^{ij}\right\}_{k=1:nj}$, as shown in Fig. 2. Based on the 1st level ASD, using hierarchical clustering, we further construct the 2nd level ASD, assuming there are ω_2 high level semantic: $\{H^i\}_{i=1:\omega 2}$, the H^i includes some combinations of the 1st level semantics, as shown in Fig. 2.

(3) *Hierarchical graph model construction.* As shown in Fig. 3, the HDBN is made up of 3 layers, the 1st layer is signals input layer, the 2nd layer is semantic layer, given inputs, the semantics are inferred in 2nd layer using filtering method. The 3rd layer is semantic layer, the higher level abstraction semantic is inferred based on 2nd layer semantic information.

The ASD is used to estimate parameters of the HDBN. Let prior possibility be $\pi = \{P(S^i)\}_{i=1:\omega}$, which is calculated as:

$$P\left(S^i = i\right) \approx \frac{n_{C_t \in S^i}}{n_C} \tag{1}$$

where the $n_{C \in S^i}$ is clip numbers belong to the semantic S^i, and the n_C is the clip numbers of all training videos. Secondly, let $\mathbf{A} = [a_{ij}]_{k \times k}$ be semantic transaction possibility, where the a_{ij} is estimated:

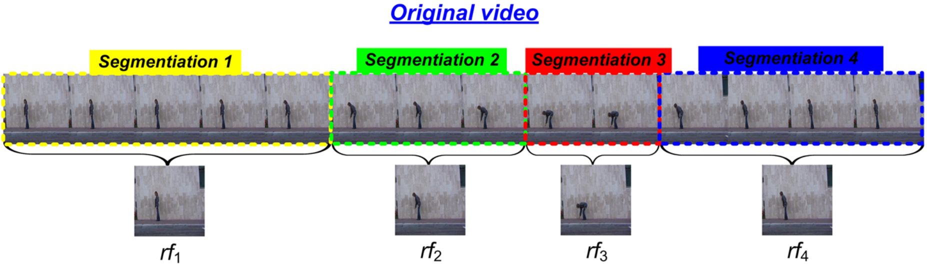


Fig. 2 an example for some representative frames extracted based on ACA clustering

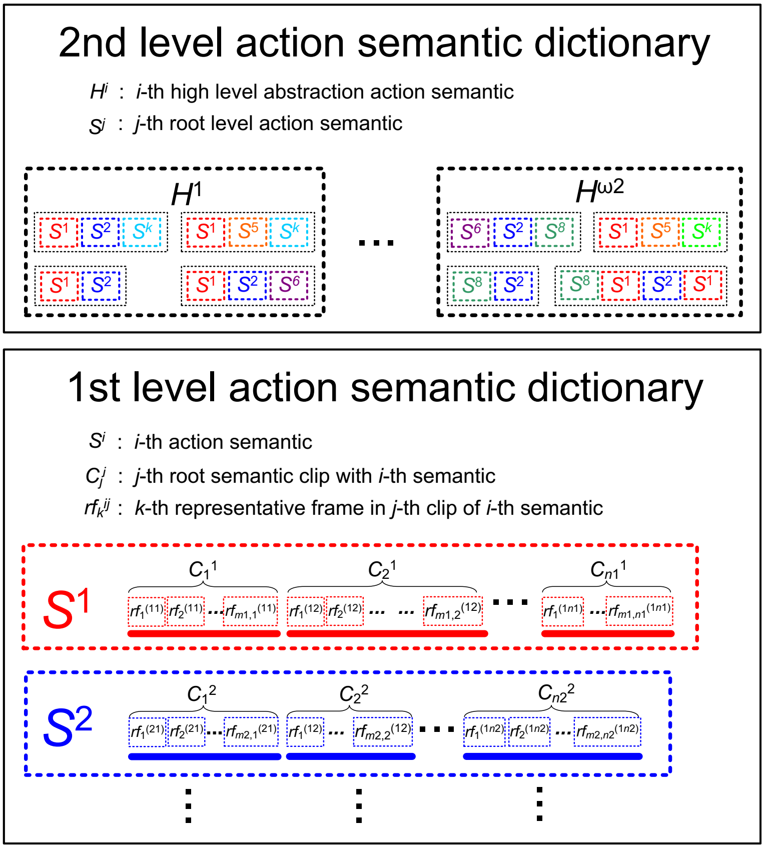

2nd level action semantic dictionary
H^i : i-th high level abstraction action semantic
S^j : j-th root level action semantic
H^1
S^1 S^2 S^k
S^1 S^5 S^k
S^1 S^2
S^1 S^2 S^6
…
H^ω2
S^6 S^2 S^8
S^1 S^5 S^k
S^8 S^2
S^8 S^1 S^2 S^1
1st level action semantic dictionary
S^i : i-th action semantic
C_j^i : j-th root semantic clip with i-th semantic
rf_k^ij : k-th representative frame in j-th clip of i-th semantic
S^1
C_1^1 C_2^1 C_n1^1
rf_1^(11) rf_2^(11) … rf_m1,1^(11)
rf_1^(12) rf_2^(12) … … rf_m1,2^(12)
rf_1^(1n1) … rf_m1,n1^(1n1)
S^2
C_1^2 C_2^2 C_n2^2
rf_1^(21) rf_2^(21) … rf_m2,1^(21)
rf_1^(12) … rf_m2,2^(12)
rf_1^(1n2) rf_2^(1n2) … rf_m2,n2^(1n2)

Fig. 3 Hierarchical action semantic dictionary

$$a_{ij} = P\left(S^j = j|S^i = i\right) \approx \frac{n_{S^j \middle| S^i}}{n_{S \middle| S^i}} \tag{2}$$

where $n_{S^j|S^i}$ denotes adjoining links numbers from clip C_t ($C_t \in S^i$) to clip C_{t+1} ($C_{t+1} \in S^j$) in training videos, the $n_{S|S^i}$ denotes the all possible adjoining links numbers of training videos. Thirdly, let $\mathbf{B} = [b_i]_{i=1:\omega 1}$ be observation possibility matrix, assume the b_i is multi-variable Gaussian distribution, we have:

$$b_i = P\left(rf|S^i = i\right) = \mathbb{N}\left(\mu_{S^i}, \Sigma_{S^i}^{-1}\right) \tag{3}$$

where the parameters μ_{S^i} and $\Sigma_{S^i}^{-1}$ are expectation and variance, respectively, which are learned by maximum likewood (ML) method.

According to above steps, we have got the parameters from 1st layer to 2nd layer, it is written as: $\lambda_{1\to 2} = (\pi, \mathbf{A}, \mathbf{B})$, which is used to infer the 2nd layer information using inputs. Similar to the above steps, parameters of the 2nd layer to the 3rd layer are calculated: $\lambda_{2\to 3} = (\pi_1, \mathbf{A}_1, \mathbf{B}_1)$, where the π_1 is prior distribution. Similar to Eq. 1, we can get$\pi_1 = \{P(H^i)\}_{i=1:\omega 2}$, and $P(H^i) \approx n_{S \in H^i}/n_S$. The $\mathbf{A}_1$ is state translation matrix, similar to Eq. 2, we have $\mathbf{A}_1 = \{a_{ij}\}_{\omega 2 \times \omega 2}$, and $a_{ij} = P(H^i|H^j) \approx n_{H^j|H^i}/n_{H|H^i}$. The $\mathbf{B}_1$ is observation matrix, similar to Eq.3, we have $\mathbf{B}_1 = \{b_i\}_{1\times \omega 2}$, and $b_1^i = P(S|H^i) = \mathbb{N}\left(\mu_{H^i}, \Sigma_{H^i}^{-1}\right)$.

Based on obtained parameters, the HDBN is built in Fig. 4, we use the representative frames $\{rf_{1:T}\}$ as inputs of graph model. The system state is $S = \{s_{1:T}\}$ and $H = \{h_{1:T}\}$.

(4) The HDBN inference. Given inputs $\{rf_{1:T}\}$, to update hidden state signals $S = \{s_{1:T}\}$ and $H = \{h_{1:T}\}$. Based on probability graph model and filtering theory [8], the inference is described as following. Firstly, calculate $P(s_1)$ according to Bayesian rule:

$$P(s_1) = \sum_{s_0=1}^{\omega} P(s_1|s_0)P(s_0) \tag{4}$$

To set s_0 and $P(s_1|\, s_0)$ as initial system input, hence, based on Eq. 4, we can get $P(s_1)$. The next, to update the $P(s_1)$ using newer inputs:

$$P(s_1|rf_1) = \frac{P(rf_1|s_1)P(s_1)}{P(rf_1)} = \alpha P(rf_1|s_1)P(s_1) \tag{5}$$

where the α is confident and assure output within [0,1]. Further, using newer inputs, we have:

$$\begin{aligned} P\left(s_{1+t}|rf_{1:1+t}\right) &= P\left(s_{1+t}|rf_{1+t}, rf_{1:t}\right) \\ &= \alpha P\left(rf_{1+t}|s_{1+t}\right)\int_{rf_t} P(s_{1+t}|rf_{1:t}) \\ &= \alpha P\left(rf_{1+t}|s_{1+t}\right)\sum_{s_t} P(s_{1+t}|s_t)\int_{rf_t} P(s_t|rf_{1:t}) \end{aligned} \tag{6}$$

where $P(rf_t|s_t) = \mathbb{N}\left(\mu_S, \Sigma_S^{-1}\right)\,(rf_t)$. Based on filtering formula Eq. 6, if we use $P(s_{1:t}|\, rf_{1:t})$ to replace $P(s_t|\, rf_{1:t})$ in Eq. 6, according to probability Bayes network filtering theory [9], we have:

$$\max_{s_1,\cdots,s_t} P\left(s_{1:1+t}|rf_{1:1+t}\right) = \alpha P\left(rf_{1+t}|s_{1+t}\right) \max_{s_t} P(s_{1+t}|s_t) \max_{s_1,\cdots,s_{t-1}} P(s_t|rf_{1:t}) \tag{7}$$

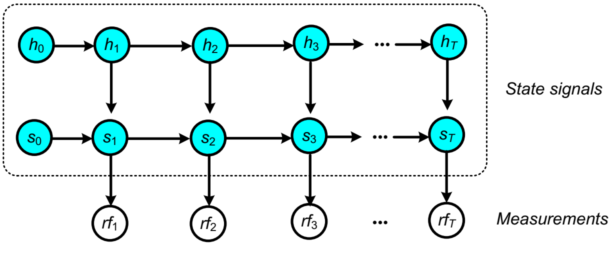
Based on the $\lambda_{1\to 2}$ and eq.4-eq.7, the optimal state $s_{1:T}$ is got. Similar to inference from 1st layer to 2nd layer, if let the $s_{1:T}$ be inputs of the 2nd layer to the 3rd layer, the $h_{1:T}$ is:

$$h^{*}_{1:1+t} = \arg\max_{s_1,\cdots,s_t} P(h_{1:1+t}|s_{1:1+t}) = \alpha P(s_{1+t}|h_{1+t}) \max_{h_t} P(h_{1+t}|h_t) \max_{h_1,\cdots,h_{t-1}} P(h_t|s_{1:t}) \tag{8}$$

3.3 Action recognition

In stage of action recognition, firstly, the features of testing videos are extracted using trained deep neural networks, let features of testing videos be $\{f_{1:n}\}$, using the ACA to obtain representative frames $\{rf_{1:t}\}$. Using the $\{rf_{1:t}\}$ as inputs of the HDBN, we use the above

Fig. 4 The HDBN presentation for action recognition



mentioned steps to get semantic sequence max-possibility states $h^{*}_{1:T} = \text{argmax}_{h1:T} P(h_{1:T}|rf_{1:T})$. The total recognition algorithm is described as:

Algorithm: Action recognition using the ASD and the HDBN.

Input: training video data, checking video data.

Output: recognition results: $h^{*}_{1:T}$.

1.To get all key-frames in training videos dataset;

2.To extract features of key-frames using Deep Brief Network automatically;

3.To obtained representative frames dataset corresponding to training videos;

4. Based representative frames dataset, build hierarchical action semantic dictionary: $\mathbf{ASD}_{level2} = \{H^{i}\}_{i=1:\omega}$, $H^{i} = \{s^{1:T}|s_t = S^{i}, i = 1{:}\omega 1\}$, $\mathbf{ASD}_{level1} = \{S^{i}\}_{i=1:\omega 1}$, $S^{i} = \{C^{i}_{j}\}_{j=1:ni}$, $C^{i}_{j} = \{rf_{k}^{ij}\}_{k=1:nj}$;

5. Base on hierarchical action semantic dictionary, to learn the parameters of the HDBN: $\lambda_{1\to 2}(\pi,\mathbf{A},\mathbf{B})$, $\lambda_{2\to 3}(\pi_1,\mathbf{A}_1,\mathbf{B}_1)$ according to Eq. 1–3;

6. To input identifying video, to extract key-frames and representative frames $\{rf_{1:T}\}$ using Deep Brief Network and the ACA, respectively;

7. Using the $\{rf_{1:T}\}$ as inputs of the HDBN, to infer $\{s_{1:T}\}$ and $\{h_{1:T}\}$ according to Eq. 4–7;

8. To output max-probability sequence as recognition results: $h^{*}_{1:T} = \text{argmax}_{h1:T} P(h_{1:T}|rf_{1:T})$;

4 Experiments

To verify our proposed action recognition method, the experiments are evaluated on Weizmann [8] dataset and Youtube [20] dataset.

4.1 Dataset

The Weizmann dataset includes 93 original action videos that belong to 10 topics, such as running, walking, jumping, and so on. For get the more training samples, we select video frames by equal interval sampling to obtain the more video samples. We set interval numbers is 8, and get total 744 videos samples. For shorten training time of the CDBN, all representative frames images are translated into 50 × 50 pixels images (we use Matlab function "imeasize" to get smaller images), that means, the smaller images are with the lesser training time. The 70% random selected videos are as training data and the 15% remaining data is testing data and the other 15% data is cross validation data. Some examples of Weizmann dataset are shown in Fig. 5.

UCF YouTube action dataset (http://crcv.ucf.edu/data/UCF_YouTube_Action.php) includes 11 action classes, some examples are shown in Fig. 6, such as biking/cycling, basketball shooting, and so on. There are total 1600 videos in UCF11, for balancing each category of data, we select 116 videos from each class, and total 1276 videos to build evaluation dataset, we random select 70% samples as training data, the 15% of remaining data is cross validation data and another 15% data is testing data.

4.2 Performance evaluation

In Weizman dataset, all videos are segmented manually, and in each video, there is only one kind of action. According to this, we can use action recognition accuracy as evaluation criterion [22]:

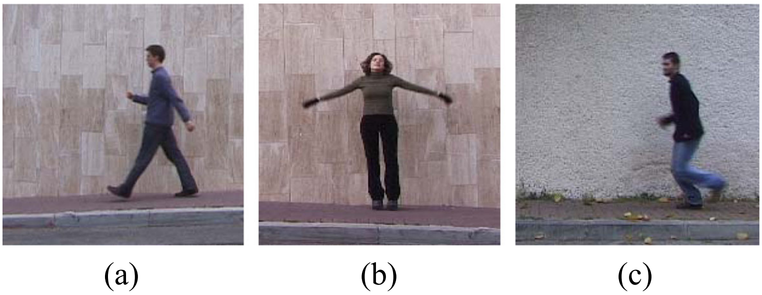


Fig. 5 [24] Some examples of Weizmann dataset. (**a**) walking. (**b**) jumping. (**c**) running

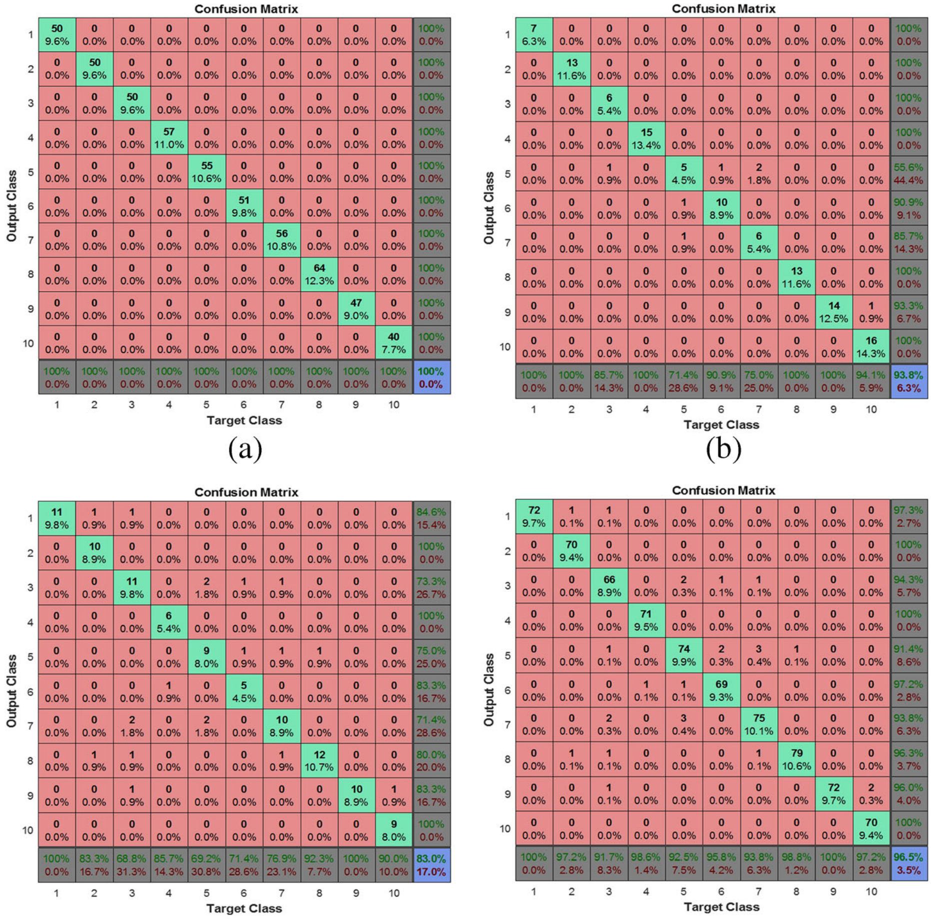
$$\text{Action\_accuracy} = \frac{\#\text{correctly\_recognized\_actions}}{\#\text{segmented\_actions}} \tag{9}$$

The total recognition testing results are shown in Fig. 7, the class names of 1–10 are "bend", "jack", "jump", "pjump", "run", "side", "skip", "walk", "wave1","wave2", respectively. The Fig. 7(a)-(d) are action recognition confusion matrixes. In Fig. 7(a), we know action recognition accuracy is 100%. In Fig. 7(b), action accuracy of testing data is 93.8%, and in Fig. 7(c), action accuracy of validation data is 83%, form the results, the 7th class "skip" have the lowest action accuracy, that is 71.4%, we assume that "skip" may be a compound movement, which contains a variety of other sports, such as "jump"," run", and so on, and the complex action recognition always is more difficult for identifying. In Fig. 7(d), we know total action accuracy is 96.5%.

To further evaluate our model, we also conduct some comparison experiments. Comparison approaches include: (1) conditional random fields CRF [22], which is a sequence model; (2) support vector machine (SVM) [10], one-hidden-layer neural network (NN) [10], both are non-sequence model.

We use a linear kernel SVM model as multi-class classification model, at the same time, we train an NN with one hidden layer. Similar to the CRF, the HDBN is also a sequence model,

Fig. 6 [14] Some examples of UCF dataset



(a)

	1	2	3	4	5	6	7	8	9	10	
1	50 9.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	50 9.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	50 9.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	57 11.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	55 10.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	51 9.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	56 10.8%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	64 12.3%	0 0.0%	0 0.0%	100% 0.0%
9	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	47 9.0%	0 0.0%	100% 0.0%
10	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	40 7.7%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%

(b)

	1	2	3	4	5	6	7	8	9	10	
1	7 6.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	13 11.6%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	6 5.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	15 13.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	1 0.9%	0 0.0%	5 4.5%	1 0.9%	2 1.8%	0 0.0%	0 0.0%	0 0.0%	55.6% 44.4%
6	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.9%	10 8.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	90.9% 9.1%
7	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.9%	0 0.0%	6 5.4%	0 0.0%	0 0.0%	0 0.0%	85.7% 14.3%
8	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	13 11.6%	0 0.0%	0 0.0%	100% 0.0%
9	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	14 12.5%	1 0.9%	93.3% 6.7%
10	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	16 14.3%	100% 0.0%
	100% 0.0%	100% 0.0%	85.7% 14.3%	100% 0.0%	71.4% 28.6%	90.9% 9.1%	75.0% 25.0%	100% 0.0%	100% 0.0%	94.1% 5.9%	93.8% 6.3%

(c)

	1	2	3	4	5	6	7	8	9	10	
1	11 9.8%	1 0.9%	1 0.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	84.6% 15.4%
2	0 0.0%	10 8.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	11 9.8%	0 0.0%	2 1.8%	1 0.9%	1 0.9%	0 0.0%	0 0.0%	0 0.0%	73.3% 26.7%
4	0 0.0%	0 0.0%	0 0.0%	6 5.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	9 8.0%	1 0.9%	1 0.9%	1 0.9%	0 0.0%	0 0.0%	75.0% 25.0%
6	0 0.0%	0 0.0%	0 0.0%	1 0.9%	0 0.0%	5 4.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	83.3% 16.7%
7	0 0.0%	0 0.0%	2 1.8%	0 0.0%	2 1.8%	0 0.0%	10 8.9%	0 0.0%	0 0.0%	0 0.0%	71.4% 28.6%
8	0 0.0%	1 0.9%	1 0.9%	0 0.0%	0 0.0%	0 0.0%	1 0.9%	12 10.7%	0 0.0%	0 0.0%	80.0% 20.0%
9	0 0.0%	0 0.0%	1 0.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	10 8.9%	1 0.9%	83.3% 16.7%
10	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	9 8.0%	100% 0.0%
	100% 0.0%	83.3% 16.7%	68.8% 31.3%	85.7% 14.3%	69.2% 30.8%	71.4% 28.6%	76.9% 23.1%	92.3% 7.7%	100% 0.0%	90.0% 10.0%	83.0% 17.0%

(d)

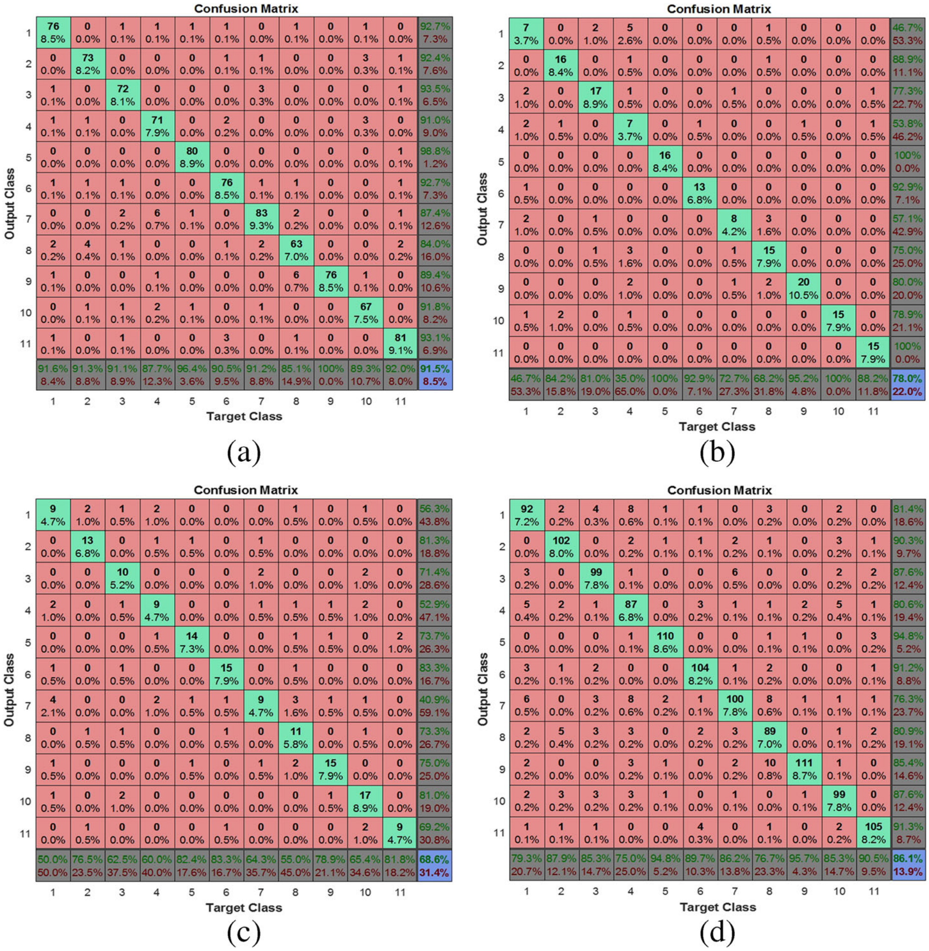
	1	2	3	4	5	6	7	8	9	10	
1	72 9.7%	1 0.1%	1 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	97.3% 2.7%
2	0 0.0%	70 9.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	66 8.9%	0 0.0%	2 0.3%	1 0.1%	1 0.1%	0 0.0%	0 0.0%	0 0.0%	94.3% 5.7%
4	0 0.0%	0 0.0%	0 0.0%	71 9.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
5	0 0.0%	0 0.0%	1 0.1%	0 0.0%	74 9.9%	2 0.3%	3 0.4%	1 0.1%	0 0.0%	0 0.0%	91.4% 8.6%
6	0 0.0%	0 0.0%	0 0.0%	1 0.1%	1 0.1%	69 9.3%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	97.2% 2.8%
7	0 0.0%	0 0.0%	2 0.3%	0 0.0%	3 0.4%	0 0.0%	75 10.1%	0 0.0%	0 0.0%	0 0.0%	93.8% 6.3%
8	0 0.0%	1 0.1%	1 0.1%	0 0.0%	0 0.0%	0 0.0%	1 0.1%	79 10.6%	0 0.0%	0 0.0%	96.3% 3.7%
9	0 0.0%	0 0.0%	1 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	72 9.7%	2 0.3%	96.0% 4.0%
10	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	70 9.4%	100% 0.0%
	100% 0.0%	97.2% 2.8%	91.7% 8.3%	98.6% 1.4%	92.5% 7.5%	95.8% 4.2%	93.8% 6.3%	98.8% 1.2%	100% 0.0%	97.2% 2.8%	96.5% 3.5%

Fig. 7 recognition confusion matrix in Weizman dataset, and class 1–10 are "bend", "jack", "jump", "pjump", "run", "side", "skip", "walk", "wave1","wave2", respectively. **a** The confusion matrix of training data; **b** The confusion matrix of testing data; **c** The confusion matrix of cross-validation data; **d** The confusion matrix of total data

and we can use standard linear-chain CRF as the baseline methods to test the HDBN performance. Table 1 shows the test results, from results, we know: (1) The CRF has better performance than linear SVM, the results denote that sequence structure is always important

Table 1 Comparison of methods on Weizman dataset

Algorithm	Action_accuracy
SVM [10]	91.3%
NN [10]	95.2%
CRF [22]	93.3%
Our method	96.5%



Confusion Matrix

	1	2	3	4	5	6	7	8	9	10	11	
1	76 8.5%	0 0.0%	1 0.1%	1 0.1%	1 0.1%	1 0.1%	0 0.0%	1 0.1%	0 0.0%	1 0.1%	0 0.0%	92.7% 7.3%
2	0 0.0%	73 8.2%	0 0.0%	0 0.0%	0 0.0%	1 0.1%	1 0.1%	0 0.0%	0 0.0%	3 0.3%	1 0.1%	92.4% 7.6%
3	1 0.1%	0 0.0%	72 8.1%	0 0.0%	0 0.0%	0 0.0%	3 0.3%	0 0.0%	0 0.0%	0 0.0%	1 0.1%	93.5% 6.5%
4	1 0.1%	1 0.1%	0 0.0%	71 7.9%	0 0.0%	2 0.2%	0 0.0%	0 0.0%	0 0.0%	3 0.3%	0 0.0%	91.0% 9.0%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	80 8.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.1%	98.8% 1.2%
6	1 0.1%	1 0.1%	1 0.1%	0 0.0%	0 0.0%	76 8.5%	1 0.1%	1 0.1%	0 0.0%	0 0.0%	1 0.1%	92.7% 7.3%
7	0 0.0%	0 0.0%	2 0.2%	6 0.7%	1 0.1%	0 0.0%	83 9.3%	2 0.2%	0 0.0%	0 0.0%	1 0.1%	87.4% 12.6%
8	2 0.2%	4 0.4%	1 0.1%	0 0.0%	0 0.0%	1 0.1%	2 0.2%	63 7.0%	0 0.0%	0 0.0%	2 0.2%	84.0% 16.0%
9	1 0.1%	0 0.0%	0 0.0%	1 0.1%	0 0.0%	0 0.0%	0 0.0%	6 0.7%	76 8.5%	1 0.1%	0 0.0%	89.4% 10.6%
10	0 0.0%	1 0.1%	1 0.1%	2 0.2%	1 0.1%	0 0.0%	1 0.1%	0 0.0%	0 0.0%	67 7.5%	0 0.0%	91.8% 8.2%
11	1 0.1%	0 0.0%	1 0.1%	0 0.0%	0 0.0%	3 0.3%	0 0.0%	1 0.1%	0 0.0%	0 0.0%	81 9.1%	93.1% 6.9%
	91.6% 8.4%	91.3% 8.8%	91.1% 8.9%	87.7% 12.3%	96.4% 3.6%	90.5% 9.5%	91.2% 8.8%	85.1% 14.9%	100% 0.0%	89.3% 10.7%	92.0% 8.0%	91.5% 8.5%

(a)

Confusion Matrix

	1	2	3	4	5	6	7	8	9	10	11	
1	7 3.7%	0 0.0%	2 1.0%	5 2.6%	0 0.0%	0 0.0%	0 0.0%	1 0.5%	0 0.0%	0 0.0%	0 0.0%	46.7% 53.3%
2	0 0.0%	16 8.4%	0 0.0%	1 0.5%	0 0.0%	0 0.0%	0 0.0%	1 0.5%	0 0.0%	0 0.0%	0 0.0%	88.9% 11.1%
3	2 1.0%	0 0.0%	17 8.9%	1 0.5%	0 0.0%	0 0.0%	1 0.5%	0 0.0%	0 0.0%	0 0.0%	1 0.5%	77.3% 22.7%
4	2 1.0%	1 0.5%	0 0.0%	7 3.7%	0 0.0%	1 0.5%	0 0.0%	0 0.0%	1 0.5%	0 0.0%	1 0.5%	53.8% 46.2%
5	0 0.0%	0 0.0%	0 0.0%	0 0.0%	16 8.4%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
6	1 0.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	13 6.8%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	92.9% 7.1%
7	2 1.0%	0 0.0%	1 0.5%	0 0.0%	0 0.0%	0 0.0%	8 4.2%	3 1.6%	0 0.0%	0 0.0%	0 0.0%	57.1% 42.9%
8	0 0.0%	0 0.0%	1 0.5%	3 1.6%	0 0.0%	0 0.0%	1 0.5%	15 7.9%	0 0.0%	0 0.0%	0 0.0%	75.0% 25.0%
9	0 0.0%	0 0.0%	0 0.0%	2 1.0%	0 0.0%	0 0.0%	1 0.5%	2 1.0%	20 10.5%	0 0.0%	0 0.0%	80.0% 20.0%
10	1 0.5%	2 1.0%	0 0.0%	1 0.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	15 7.9%	0 0.0%	78.9% 21.1%
11	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	15 7.9%	100% 0.0%
	46.7% 53.3%	84.2% 15.8%	81.0% 19.0%	35.0% 65.0%	100% 0.0%	92.9% 7.1%	72.7% 27.3%	68.2% 31.8%	95.2% 4.8%	100% 0.0%	88.2% 11.8%	78.0% 22.0%

(b)

Confusion Matrix

	1	2	3	4	5	6	7	8	9	10	11	
1	9 4.7%	2 1.0%	1 0.5%	2 1.0%	0 0.0%	0 0.0%	0 0.0%	1 0.5%	0 0.0%	1 0.5%	0 0.0%	56.3% 43.8%
2	0 0.0%	13 6.8%	0 0.0%	1 0.5%	1 0.5%	0 0.0%	1 0.5%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	81.3% 18.8%
3	0 0.0%	0 0.0%	10 5.2%	0 0.0%	0 0.0%	0 0.0%	2 1.0%	0 0.0%	0 0.0%	2 1.0%	0 0.0%	71.4% 28.6%
4	2 1.0%	0 0.0%	1 0.5%	9 4.7%	0 0.0%	0 0.0%	1 0.5%	1 0.5%	1 0.5%	2 1.0%	0 0.0%	52.9% 47.1%
5	0 0.0%	0 0.0%	0 0.0%	1 0.5%	14 7.3%	0 0.0%	0 0.0%	1 0.5%	1 0.5%	0 0.0%	2 1.0%	73.7% 26.3%
6	1 0.5%	0 0.0%	1 0.5%	0 0.0%	0 0.0%	15 7.9%	0 0.0%	1 0.5%	0 0.0%	0 0.0%	0 0.0%	83.3% 16.7%
7	4 2.1%	0 0.0%	0 0.0%	2 1.0%	1 0.5%	1 0.5%	9 4.7%	3 1.6%	1 0.5%	1 0.5%	0 0.0%	40.9% 59.1%
8	0 0.0%	1 0.5%	1 0.5%	0 0.0%	0 0.0%	1 0.5%	0 0.0%	11 5.8%	0 0.0%	1 0.5%	0 0.0%	73.3% 26.7%
9	1 0.5%	0 0.0%	0 0.0%	0 0.0%	1 0.5%	0 0.0%	1 0.5%	2 1.0%	15 7.9%	0 0.0%	0 0.0%	75.0% 25.0%
10	1 0.5%	0 0.0%	2 1.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.5%	17 8.9%	0 0.0%	81.0% 19.0%
11	0 0.0%	1 0.5%	0 0.0%	0 0.0%	0 0.0%	1 0.5%	0 0.0%	0 0.0%	0 0.0%	2 1.0%	9 4.7%	69.2% 30.8%
	50.0% 50.0%	76.5% 23.5%	62.5% 37.5%	60.0% 40.0%	82.4% 17.6%	83.3% 16.7%	64.3% 35.7%	55.0% 45.0%	78.9% 21.1%	65.4% 34.6%	81.8% 18.2%	68.6% 31.4%

(c)

Confusion Matrix

	1	2	3	4	5	6	7	8	9	10	11	
1	92 7.2%	2 0.2%	4 0.3%	8 0.6%	1 0.1%	1 0.1%	0 0.0%	3 0.2%	0 0.0%	2 0.2%	0 0.0%	81.4% 18.6%
2	0 0.0%	102 8.0%	0 0.0%	2 0.2%	1 0.1%	1 0.1%	2 0.2%	1 0.1%	0 0.0%	3 0.2%	1 0.1%	90.3% 9.7%
3	3 0.2%	0 0.0%	99 7.8%	1 0.1%	0 0.0%	0 0.0%	6 0.5%	0 0.0%	0 0.0%	2 0.2%	2 0.2%	87.6% 12.4%
4	5 0.4%	2 0.2%	1 0.1%	87 6.8%	0 0.0%	3 0.2%	1 0.1%	1 0.1%	2 0.2%	5 0.4%	1 0.1%	80.6% 19.4%
5	0 0.0%	0 0.0%	0 0.0%	1 0.1%	110 8.6%	0 0.0%	0 0.0%	1 0.1%	1 0.1%	0 0.0%	3 0.2%	94.8% 5.2%
6	3 0.2%	1 0.1%	2 0.2%	0 0.0%	0 0.0%	104 8.2%	1 0.1%	2 0.2%	0 0.0%	0 0.0%	1 0.1%	91.2% 8.8%
7	6 0.5%	0 0.0%	3 0.2%	8 0.6%	2 0.2%	1 0.1%	100 7.8%	8 0.6%	1 0.1%	1 0.1%	1 0.1%	76.3% 23.7%
8	2 0.2%	5 0.4%	3 0.2%	3 0.2%	0 0.0%	2 0.2%	3 0.2%	89 7.0%	0 0.0%	1 0.1%	2 0.2%	80.9% 19.1%
9	2 0.2%	0 0.0%	0 0.0%	3 0.2%	1 0.1%	0 0.0%	2 0.2%	10 0.8%	111 8.7%	1 0.1%	0 0.0%	85.4% 14.6%
10	2 0.2%	3 0.2%	3 0.2%	3 0.2%	1 0.1%	0 0.0%	1 0.1%	0 0.0%	1 0.1%	99 7.8%	0 0.0%	87.6% 12.4%
11	1 0.1%	1 0.1%	1 0.1%	0 0.0%	0 0.0%	4 0.3%	0 0.0%	1 0.1%	0 0.0%	2 0.2%	105 8.2%	91.3% 8.7%
	79.3% 20.7%	87.9% 12.1%	85.3% 14.7%	75.0% 25.0%	94.8% 5.2%	89.7% 10.3%	86.2% 13.8%	76.7% 23.3%	95.7% 4.3%	85.3% 14.7%	90.5% 9.5%	86.1% 13.9%

(d)

Fig. 8 Recognition confusion matrix in UCF11 dataset, and the class names of 1–11 are "walk", "jump", "golf", "biking", "diving", "juggle", "riding", "shoot", "spiking", "swing", "tennis", respectively. **a** The confusion matrix of training data; **b** The confusion matrix of testing data; **c** The confusion matrix of cross-validation data; **d** The confusion matrix of total data.

for action recognition. (2) From results, we know the NN model has better performance, the nonlinear models outperform the linear models for action recognition. (3) The HDBN combines sequence structure with non-linear model, therefore, the HDBN achieves the best performance.

Table 2 Comparison with state-of-the-arts methods on Youtube dataset

Algorithm	Action_accuracy
Liu et al. [20]	73.2%
Le et al. [18]	76.3%
Liu et al. [21]	78.2%
Our method	86.1%

In the comparative experiments, in split of the model used, we also noticed that the 7th class motion, the “skip” motion, still has a relatively lower recognition rate, such as, the action accuracy are 70.1%, 73.6% and 69.8% in SVM model, NN model and CRF model, respectively. The results also suggest that the motion “skip” may contain more simple motions, hence, no matter what method we use, the results are not satisfactory. This should be considered to improve in the next step.

The total testing results on the UCF11 dataset are shown in Fig. 8, the Fig. 8(a) and (b) are recognition confusion matrixes. In Fig. 8(b), the total action recognition accuracy is 86.1%, the action recognition accuracy of the “swing” is 76.3%, which is the lowest accuracy. We think there may be too rich scene changing, that mean, there are multiple motion objects appear in scene, at the same time, the backgrounds changes from time to time. In Fig. 8(b), from the cross-validation results, we know that the action “swing” has the lower action accuracy, that are 40.9%. Just as the reason of above analysis, the richer scene changing, and the lower action recognition accuracy.

For further evaluate our model performance, we compare the HDBN performance with some state-of-the-arts approaches. As shown in Table 2. Liu et al. [32] put static descriptors and action features to develop feature description. In [7], a multi-layer of ISA is used to obtain a deep-based description. In [25], a semantic visual vocabulary is learned automatically from mid-level representations. Form results, we know our algorithm is promising.

5 Conclusions

In this paper, a novel the HDBN-based action recognition method is proposed. Our contribution can be described as:

(1) We propose a novel graph-based action recognition model. The model combines the hierarchical action semantic dictionary and Bayesian graph model inference together, and uses recursion-based method to recognize action video data.

(2) Based on some theories, such as Bayesian rules, graph model, the probability-based recursion calculation structure is presented to obtain the higher accuracy of action recognition. Experimental results show that the proposed model has better performance than some existing algorithms.

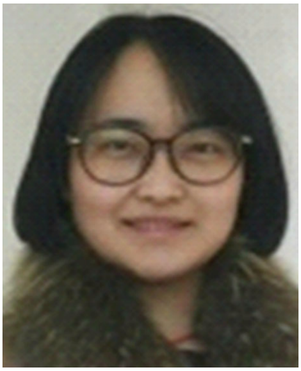
Acknowledgements This research is supported by the project (61271362, 61671362) of the National Natural Science Foundation of China.

References

1. Bay H, Tuytelaars T, Van Gool L (2006) Surf: speeded up robust features. In: ECCV, 404–417
2. Bobick AF, Davis JW (2001) The recognition of human movement using temporal templates. IEEE Trans Pattern Anal Mach Intell 23(3):257–267

3. Chaaraoui AA, Climent-Pérez P, Flórez-Revuelta F (2013) Silhouette-based human action recognition using sequences of key poses. Pattern Recogn Lett 34(15):1799–1807
4. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. CVPR 1:886–893
5. Do T-M-T, Artieres T (2010) Neural conditional random fields. In: Proceedings of the 13th international conference on artificial intelligence and statistics, vol. 9, pp 177–184
6. Efros AA, Berg AC, Mori G, Malik J (2003) Recognizing action at a distance. In: ICCV 2003, Nice, France, 726–733
7. Farzad H, Babette D, Carme T (2016) Action recognition based on efficient deep feature learning in the spatio-temporal domain. IEEE Robotics and Automation Letters 1(2):984–991
8. Gorelick L, Blank M, Shechtman E, Irani M, Basri R (2005) Actions as space–time shapes. In: IEEE ICCV, 1395–1402
9. Gross JL, Yellen J (2011) Graph theory and its applications, 2nd edn. Chapman and Hall/CRC, Boca Raton
10. Haykin S (2009) Neural networks and learning machines, 3rd edn. Prentice Hall, New York
11. Hoai M, Lan Z-Z, Dela Torre F (2011) Joint segmentation and classification of human actions in video. In: CVPR, 3265–3272
12. Hongzhao C, Wang G, Xue J-H, He L (2016) A novel hierarchical framework for human action recognition. Pattern Recogn 55:148–159
13. Jhuang H, Serre T, Wolf L, Poggio T (2007) A biologically inspired system for action recognition. In: ICCV, 1–8
14. Kong Y, Behnam S, Yun F (2016) Learning hierarchical 3D kernel descriptors for RGB-D action recognition. Comput Vision Image Underst 144:14–23
15. Lafferty J, Mccallum A, Pereira F (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: ICML, pp 282-289
16. Lafferty J, Zhu X, Liu Y (2004) Kernel conditional random fields: representation and clique selection. In: ICML
17. Laptev I (2005) On space–time interest points. Int J Comput Vis 64(2–3):107–123
18. Le QV, Zou WY, Yeung SY, Ng AY (2011) Learning hierarchical in variant spatio- temporal features for action recognition within dependent subspace analysis. CVPR, In, pp 3361–3368
19. Lee H, Grosse R, Ranganath R, Ng AY (2011) Unsupervised learning of hierarchical representations with convolutional deep belief networks. Commun ACM 54(10):95–103
20. Liu J, Luo J, Shah M (2009) Recognizing realistic actions from videos in the wild. In: CVPR
21. Liu J, Yang Y, Sha M (2009) Learning semantic visual vocabularies using diffusion distance. In: CVPR, 461–468
22. Liu CH, Liu J, He ZC, Zhai YJ, Hu QH, Huang YL (2016) Convolutional neural random fields for action recognition. Pattern Recognition 59:213–224
23. Ni B, Pei Y, Liang Z, Lin L, Moulin P (2013) Integrating multi-stage depth-induced contextual information for human action recognition and localization. In: IEEE international conference and workshops on automatic face and gesture recognition, pp 1–8
24. Paul IE, Mohan CK (2016) Human action recognition using genetic algorithms and convolutional neural networks. Pattern Recogn 59:199–212
25. Shen H, Yan Y, Xu S, Ballas N, Chen W (2015) Evaluation of semi-supervised learning method on action recognition. Multimedia Tools and Applications 74(2):523–542
26. Sminchisescu C, Kanaujia A, Metaxas D (2006) Conditional models for contextual human motion recognition. Comput Vis Image Underst 104(2–3):210–220
27. Walker J, Gupta A, Hebert M Dense optical flow prediction from a static image, arXiv preprint arXiv: 1505.00295
28. Weinland D, Ronfard R, Boyer E (2011) A survey of vision-based methods for action representation, segmentation and recognition. Comput Vis Image Underst 115(2):224–241
29. Wenwen D, Kai L, Xujia F, Cheng F (2016) Profile HMMs for skeleton-based human action recognition. Signal Process Image Commun 42:109–119
30. Wolf W (1996) Key frame selection bymotion analysis. Atlanta: Rodney Andrew Kennedy Inc 2:1228–1231
31. Xiaofei J, Zhaojie J, Wang C, Wang C (2016) Muti-view transition HMMs based view-invariant human action recognition method. Multimedia Tools and Applications 75(19):11847–11864
32. Yang Y, Lin M (2016) Human action recognition with graph-based multiple-instance learning. Pattern Recognition 53:148–162
33. Yang Q, Xue D-y, Jian-jiang C (2012) Human action recognition using dynamic bayesian network. International Journal of Advancements in Computing Technology 4(12):291–298
34. Zhang Z, Hu Y, Chan S, Chia L-T (2008) Motion context: a new representation for human action recognition. In: ECCV, 817–829
35. Zhou F, De la Torre F, Hodgins JK (2013) Hierarchical aligned cluster analysis for temporal clustering of human motion. IEEE Trans Pattern Anal Mach Intell 35(3):582–596

Qinkun Xiao was born in 1974. He is a Ph.D. and a professor in Xi'an technological University. He obtained doctor degree from Northwestern Polytechnic University in 2007, and from 2007 to 2009, he is postdoctoral in Tsinghua University. His research interests include object recognition and information retrieval, dynamic Bayesian network and image processing. E-mail: xiaoqinkun10000@163.com.



Ren Song was born in 1992 and she is currently a graduated student in Xi'an technological university. Her research interests include motion recognition and video information processing. E-mail: 775082347@qq.com.