Arabian Journal for Science and Engineering (2018) 43:7777–7788
https://doi.org/10.1007/s13369-018-3189-z

RESEARCH ARTICLE - COMPUTER ENGINEERING AND COMPUTER SCIENCE

Modelling Human Body Pose for Action Recognition Using Deep Neural Networks

Chengyang Li[1] · Ruofeng Tong[1] · Min Tang[1]

Received: 15 November 2017 / Accepted: 19 March 2018 / Published online: 27 March 2018

Abstract

Body pose is an important indicator of human actions. The existing pose-based action recognition approaches are typically designed for individual human bodies and require a fixed-size (e.g., 13 × 2) input vector. This requirement is questionable and may degrade the recognition accuracy, particularly for real-world videos, in which scenes with multiple people or partially visible bodies are common. Inspired by the recent success of convolutional neural networks (CNNs) in various computer vision tasks, we propose an approach based on a deep neural network architecture for 2D pose-based action recognition tasks in this work. To this end, a human pose encoding scheme is designed to eliminate the above requirement and to provide a general representation of 2D human body joints, which can be used as the input for CNNs. We also propose a weighted fusion scheme to integrate RGB and optical flow with human pose features to perform action classification. We evaluate our approach on two real-world datasets and achieve better performances compared to state-of-the-art approaches.

Keywords Pose-based action recognition · Convolutional neural networks · Human pose encoding scheme · 2D human pose · RGB videos

1 Introduction

Human action recognition has received significant attention from the research community due to its various potential applications in video surveillance [1], human–computer interaction [2] and video content analysis [3]. Despite being extensively researched in recent years, human action recognition in real-world monocular videos remains a challenging problem due to large intra-class variation and inter-class similarity.

The existing methods for action recognition typically use either hand-crafted features [4–8] or deep-learned features [9–12] that are derived from RGB and optical flow images. These methods rely on global contextual information and achieve promising results in recognizing coarse actions; however, these methods may fail to distinguish actions with subtle body pose variations, e.g., golf swing and baseball swing.

Human body pose, as a type of high-level semantic feature, has been shown to have an effect on recognizing human actions with discriminative geometric relations between body joints [13–15]. However, the methods that use human poses exhibit two types of defects when applied to unconstrained videos. First, the action recognition accuracy is highly dependent on the precision of the input pose. However, even with the recent developments based on convolutional neural networks (CNNs) [16–18], human pose estimation in monocular videos is still far from perfect due to large pose variations, part occlusions and complex backgrounds. Second, the majority of recognition methods based on human poses are designed for individual action recognition. These methods take a fixed number of body joint coordinates as input. However, in realistic videos taken at public places, such as gymnasiums or bowling alleys, scenes that contain several persons are common, and distinguishing the key individual from others is a non-trivial task [19]. Moreover, human bodies may only be partially visible in real-world videos. For example, some videos of playing guitar only contain upper portions of human bodies. The restriction of using full-body joints for action recognition is problem-

✉ Ruofeng Tong
trf@zju.edu.cn

Chengyang Li
licy_cs@zju.edu.cn

Min Tang
tang_m@zju.edu.cn

[1] State Key Lab of CAD&CG, Zhejiang University, Hangzhou, Zhejiang, China

atic since incorrectly estimated lower-body joints would do more harm than good.

Given the recent success of CNNs in many vision tasks, such as image classification [20–23], pose estimation [16–18] and video-based action recognition [9–12], this paper addresses the 2D pose-based action recognition task using deep networks. The primary difficulty is how to fully utilize human pose information as the input for CNNs. Therefore, a human pose encoding scheme is proposed to remove the fixed-size body joint constraint. The proposed encoding scheme is inspired by Johansson's moving light-spot experiment [24], which showed that the human vision system can distinguish different action patterns by only light spots attached to the human body. Specifically, we decompose human pose information into static and dynamic components, and we encode these components into light-spot images and joint displacement volumes, respectively, which are then taken as the input for networks.

Human pose features and global features derived from RGB images and optical flows have different focuses. We further investigate the effect of combining RGB and optical flow with automatically estimated human poses for action recognition in videos, and we determine that pose features and global features are highly complementary. Combining global features and pose features leads to a substantial performance improvement.

The primary contributions of this paper are threefold: (1) We propose an action recognition framework that integrates human pose features with global RGB and optical flow features. (2) We introduce a general human pose encoding scheme to encode human pose into light-spot images and displacement volumes, which can be directly used as input for deep neural networks. (3) We propose a weighted fusion scheme to adaptively combine pose features with global RGB and optical flow features for robust action recognition. Finally, using the proposed approach, we achieve promising performance on two real-world datasets: the Penn Action dataset and the sub-JHMDB dataset.

The remainder of this paper is organized as follows. We introduce the related work in Sect. 2. The human pose encoding scheme is presented in Sect. 3. The proposed action recognition framework is presented in Sect. 4, followed by experiments in Sect. 5. Finally, we conclude the paper in Sect. 6.

2 Related Work

We briefly review action recognition methods for monocular RGB videos in the literature. These action recognition methods can be roughly grouped into two categories: pose-feature-based methods and video-feature-based methods.

2.1 Pose-Feature-Based Methods

Human body pose is a strong indicator of human actions. Some actions can even be distinguished from a single frame. These methods typically take the estimated body pose as the input for action recognition. Thus, the recognition performance is limited by the pose estimation and human detection techniques. Early work [25,26] performed action classification on simple datasets without severe occlusions or background interference (e.g., the KTH dataset [27] and Weizmann dataset [28]). With the development of pose estimation techniques based on the deformable part model (DPM) [29], several recognition methods that are suitable for more complex scenes have been proposed [14,30–32]. Yao et al. [30] proposed an exemplar-based action classification approach to match estimated body poses with a set of exemplar poses. Wang et al. [13] first applied the contrast mining technique to mine distinctive co-occurring spatial and temporal pose structure and built a feature dictionary, then classified action labels using the bag-of-words model. Jhuang et al. [14] introduced high-level pose features (HLPF) that are manually designed to encode the spatial and temporal relations of human body joint locations. As reported in [13,14], ground-truth pose information significantly improves the performance of action recognition, whereas modestly estimated poses would degrade the improvement. Wang et al. [33] proposed a multi-view action representation in an AND-OR graph structure manner by mining the geometry, appearance and motion patterns in different views. However, 3D skeleton data are required in the training stage. Xu et al. [32] proposed using skeletal joint locations combined with local motion features for action recognition because skeletal pose alone is insufficient for recognizing actions with similar limb configurations. All these methods are designed for individual action recognition and require fixed-size joint coordinates (e.g., a 13 × 2 vector) as input. Researchers have also recently explored the idea of extracting pose information via deep learning architectures. Garbade and Gall [34] introduced a neural network architecture for action recognition based on 2D human poses. They applied a convolution layer on input data, followed by a series of fully connected and max-pooling layers, but their networks were relatively shallow. The pose-based CNN descriptor (P-CNN) [35] utilizes joint locations to crop RGB and optical flow images into part patches as the input for two-stream ConvNets, then used the extracted features to classify the video. Joints-pooled deep convolutional descriptors (JDD) [36] sample discriminative points from feature maps of 3D ConvNets (C3D) according to body joint coordinates, then concatenated the pooled activations for classification. Although both P-CNN and JDD crop images or sample feature maps using body joint locations, after simply concatenating the extracted feature vectors, the spatial relations between different body joints (e.g., left hand above head) are collapsed. Moreover, these three approaches also require individual pose input and are not applicable for multi-body pose input. In this work, we introduce a novel human pose modelling technique that can take unconstrained pose input (e.g., multiple persons or partially visible bodies) and automatically parse the internal structure relations between each joint using CNNs. Recently, RPAN [37] used a pose attention mechanism to learn pose-based features, and they used recurrent networks to model spatial–temporal evolutions. Our approach is simple yet still achieves promising performance.

2.2 Video-Feature-Based Methods

Methods that use video features have become more effective over the past decade since these methods do not explicitly handle human detection or pose estimation. Among these features, improved trajectories [6] have shown their advantages on several challenging datasets. Recently, advances in CNNs have further enhanced the action recognition performance. One of the most popular CNN architectures is two-stream ConvNets proposed in [9], which decomposes video information into spatial and temporal components and takes a single RGB frame and stacked flow frames into separate networks. This architecture is shown to be effective, particularly on datasets with limited training data. An alternative approach is C3D [10], which extends convolution and pooling operations to the time domain. C3D is a natural concept for video modelling; however, its network has considerably more parameters than 2D ConvNets and does not benefit from ImageNet pre-training. Thus, it needs to be trained on a large dataset. Recently, Carreira and Zisserman [38] proposed two-stream inflated 3D ConvNets (I3D), which inflates 2D model weights into 3D as an initialization, thereby making the training of deeper C3D possible. As discussed in Sect. 1, these methods focus less on the human body; thus, distinguishing human actions with similar scene contextual characteristics but subtle body pose variations is challenging. In this work, we experimentally demonstrate the complementarity of pose features and video features, and we show that a comprehensive video descriptor for action recognition can be obtained by combining pose features and video features, integrating the advantages of each.

3 Human Pose Encoding Scheme

When using deep neural networks to model human pose characteristics, the main difficulty is how to encode human pose information as the input for networks, since scenes with multiple persons or partial visible bodies may be encountered in real life. The gap between human pose information and feature representation can be bridged by the proposed human pose encoding scheme.

We manually decompose human pose information into static and dynamic pose components. The static pose component captures the spatial structure of human body joints in each frame, whereas the dynamic pose component characterizes the temporal evolution of human pose configurations. The static and dynamic pose components are encoded as light-spot images and joint displacement volumes, respectively, which are then used as input for the action recognition framework. The encoding schemes for static and dynamic pose components are presented in Sects. 3.1 and 3.2, respectively.

3.1 Static Pose Component

Let $p_1, \ldots, p_N$ be N human bodies in an image, which are acquired by either manual annotation or automatic pose estimation. Each body possesses K joints, and the spatial location of the jth joint of p_i is denoted as $l_{i,j} = \left(x_{i,j}, y_{i,j}\right)$, where $x_{i,j} \in [1, X]$ and $y_{i,j} \in [1, Y]$ are the spatial coordinates in the image and $[X, Y]$ is the spatial size of the input frame.

Body joints can semantically be divided into several body parts according to different semantic levels. For example, we can simply segment a human body as upper body and lower body. A finer-level segmentation can obtain left/right upper/lower limbs. Some actions are only concerned with certain body parts. Therefore, grouping body joints according to semantic categories may be helpful for parsing human actions. Specifically, we discuss four joint grouping schemes here: integral level, half level, limb level and joint level. Let C be the number of joint groups, and let O_c be the cth joint group. Different joint grouping schemes are detailed in the following, and their performances will be evaluated in Sect. 5.3.1.

Integral Level All body joints are grouped into $C = 1$ part.

$$O_1 = \{all\ body\ joints\}.$$

Half Level Joints are classified into $C = 2$ categories: upper-body joints and lower-body joints.

$$O_1 = \{upper\ body\ joints\};$$
$$O_2 = \{lower\ body\ joints\}.$$

Limb Level We classify joints into $C = 5$ categories according to different limbs. The head joint alone is the first category.

$$O_1 = \{head\};$$
$$O_2 = \{left\ upper\ limb\};$$

$O_3 = \{right\ upper\ limb\}$;
$O_4 = \{left\ lower\ limb\}$;
$O_5 = \{right\ lower\ limb\}$.

Joint level Each joint is one category. The number of joint categories is equal to the number of joints. $O_c = \{c_{th}\ joint\}$, where $1 \leqslant c \leqslant K$.

Finally, the static pose input for the CNN is constructed as a light-spot image $S \in \mathbb{R}^{X \times Y \times C}$ based on $l_{i,j}$ and O_c. Each joint is modelled by a Gaussian centred at the joint location, and the values of the light-spot image are normalized to [0, 255]. Formally, a light-spot image is defined as

$$S(x, y, c) = \max_{\substack{1 \leqslant i \leqslant N \\ j \in O_c}} 1_{i,j} f\left(\sqrt{\left(x - x_{i,j}\right)^2 + \left(x - x_{i,j}\right)^2}\right), \tag{1}$$

where $1_{i,j}$ indicates whether the jth joint of p_i exists in the image region, $f(x) = 255e^{-\frac{x^2}{2\sigma^2}}$ is the density function with $\sigma = 0.2L$, and L is the median limb length of each body in each frame. Here, the standard deviation σ is proportional to the median limb length; thus, the sizes of the spots are approximately proportional to the size of the human body. Note that P-CNN [35] requires the body scale input given by the annotation tool, which is unavailable in practice. The median limb length that we use here is computed from the input joint positions and plays a similar role as the annotation scale; thus, additional input is avoided. Figure 1 presents an illustration of different joint grouping schemes. Each grey image corresponds to one channel in a light-spot image. Using skeletal joint positions (annotated in Fig. 1a), an integral-level light-spot image (1 image channel, Fig. 1b), half-level light-spot image (2 image channels, Fig. 1c), limb-level light-spot image (5 image channels, Fig. 1d) and joint-level light-spot image (K image channels, Fig. 1e) are generated. Note that a finer-level grouping may contain richer semantics since the light spots have more certain joint meanings, but such a grouping would slow the training and inference speed due to network computation cost brought by the increased light-spot image channels.

In contrast to existing methods that typically require an individual body pose with complete body joint positions as input, our light-spot images can be derived from more flexible pose inputs. Our consideration is that in real-world videos, scenes with multiple persons or partially visible bodies are common. Hence, methods designed for individual pose input are not suitable for real scenes unless additional pre-processing is performed. Rather than designing sophisticated approaches to identify the key actor in an input image, we use a simple yet effective multi-person input strategy by representing all human bodies as light spots and skipping the invisible body joints. As illustrated in Fig. 2, light-spot images can be generated from multi-person poses or body poses that are partially visible. We will show in Sect. 5.3.2 that this design choice improves the performance of our method.

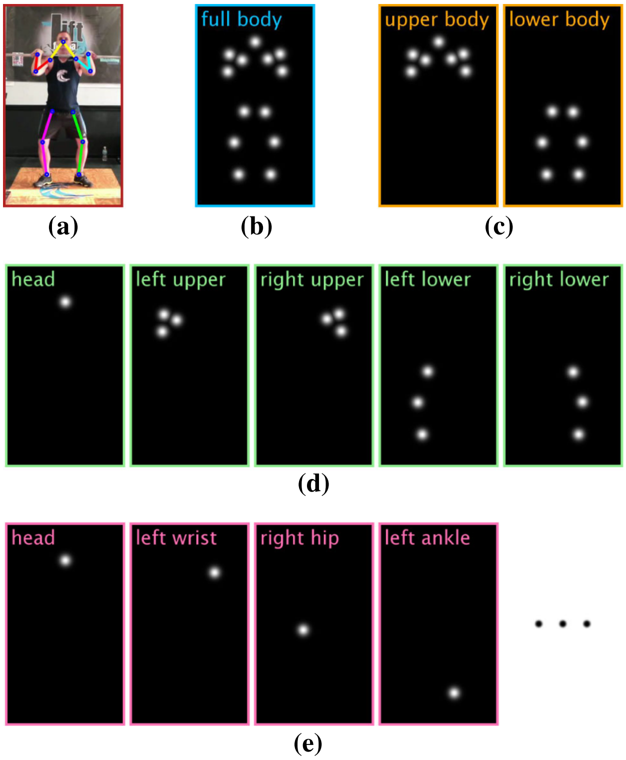

Fig. 1 Light-spot images generated according to different joint grouping schemes. (**a**) RGB image (with annotated skeletal joints), (**b**) integral-level light-spot image, (**c**) half-level light-spot image, (**d**) limb-level light-spot image, and (**e**) joint-level light-spot image

3.2 Dynamic Pose Component

Let $t_1, \ldots, t_m$ be M consecutive time points in a video sequence. The M frames are used as input for dynamic pose encoding. The joint location of the jth joint of p_i at time t_m is denoted as $l_{i,j}^m = \left(x_{i,j}^m, y_{i,j}^m\right)$.

A straightforward approach is to use a $2M$-2 dimensional displacement vector $d_{i,j} = [l_{i,j}^2 - l_{i,j}^1, \ldots, l_{i,j}^M - l_{i,j}^{M-1}] \in \mathbb{R}^{2M-2}$ to derive the temporal evolution of body poses in the given M frames. The values in the displacement vector $d_{i,j}$ that are larger than 20 pixels are clipped and then normalized to [0, 255]. Then, the dynamic pose input for the CNN is constructed as a joint displacement volume $D \in \mathbb{R}^{X \times Y \times 2(M-1)C}$ based on $l_{i,j}^m$, $d_{i,j}$ and O_c. For each joint of each human body, the $2(M-1)C$ channels of the joint displacement volume D around locations $\left[\left(x_{i,j}^1, y_{i,j}^1\right), \ldots, \left(x_{i,j}^{m-1}, y_{i,j}^{m-1}\right)\right]$ within a radius of σ are assigned with the joint displacement vector

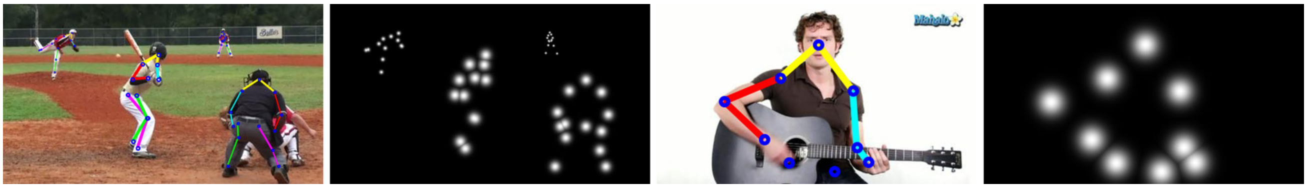

Fig. 2 Light-spot images derived from flexible pose input. Examples of images with multiple persons (left two images) and partially visible body (right two images)

$d_{i,j}$ with respect to each joint group O_c. Formally, a joint displacement volume can be defined as

$$D(x, y, 2(m-1)C + 2c - 1) = 1^m_{i,j,c,x,y}\left(x^{m+1}_{i,j} - x^m_{i,j}\right),$$
$$D(x, y, 2(m-1)C + 2c) = 1^m_{i,j,c,x,y}\left(y^{m+1}_{i,j} - y^m_{i,j}\right), \tag{2}$$

where $1^m_{i,j,c,x,y}$ indicates whether (a) the j_{th} ($j \in O_c$) joint of p_i is visible at both time points m and $m+1$, (b) the location $l^m_{i,j}$ is around (x, y) within a radius of σ, and (c) the distance between location $l^m_{i,j}$ and location (x, y) is shorter than any other joint satisfying (a) and (b). Here, we assign the displacement value to a circular region with a radius of σ rather than a single point to alleviate the sparseness of the joint displacement volume and facilitate stabilization during the model training process. Values in background regions are assigned to 128, which is exactly the same for body joints with no movements; therefore, the obtained displacement volume focuses on the motion of skeletal joints. Figure 3 presents examples of generated displacement images. We can observe that the motion of the guitar player centres on his right wrist, whereas that of the person doing sit-ups centres on his upper body.

Although the above formulation is logical, applying this formulation to real-world video scenarios is challenging for two reasons. First, pose estimation may be inaccurate, and incorrectly estimated joint positions would conflict with the consistency of pose sequences. Second, most pose estimators (including the one used in this paper) are per-frame estimators, and linking the estimated multiple bodies between successive frames is a non-trivial task. Therefore, we propose an alternative encoding formulation that constructs joint displacement volumes aided by optical flow images since optical flow is more stable than automatically estimated pose in terms of the displacement measure. Hence, we will apply this formulation in our experiments.

We pre-compute the optical flow [39] and save the flow fields v_x and v_y as JPEG images. Motion vectors larger than 20 pixels are clipped and then transformed to [0, 255]. Let $U^m \in \mathbb{R}^{\mathbb{X}\times\mathbb{Y}}$ and $V^m \in \mathbb{R}^{\mathbb{X}\times\mathbb{Y}}$ be flow fields of v_x and v_y at time point t_m. Pixel values around each joint position within a radius of σ are sampled from optical flow images to generate the displacement volume. Accordingly, a joint displacement volume can be defined as

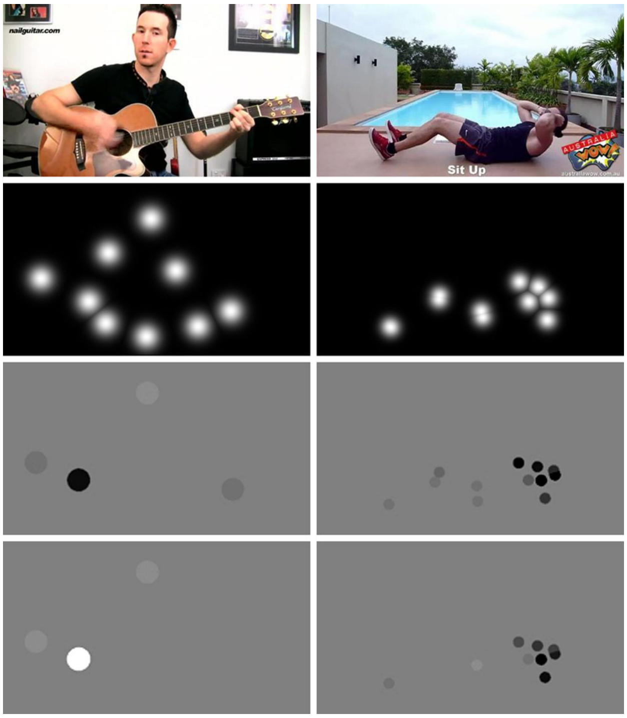

Fig. 3 Illustration of the generated displacement images. Each row corresponds to RGB images, light-spot images as well as horizontal and vertical displacement images

$$D(x, y, 2(m-1)C + 2c - 1) = 1^m_{i,j,c,x,y}U^m(x, y),$$
$$D(x, y, 2(m-1)C + 2c) = 1^m_{i,j,c,x,y}V^m(x, y), \tag{3}$$

where $1^m_{i,j,c,x,y}$ indicates whether the jth ($j \in O_c$) joint of p_i is visible at time point m and the distance to (x, y) is less than σ. Values in background regions are also assigned to 128.

In practice, we also frame-by-frame pre-compute the light-spot images and the displacement volumes with trajectory length $M = 2$ and store them as JPEG images since this would introduce a bottleneck if performed on-the-fly during the training process. Displacement volumes with a trajectory length larger than 2 frames can be obtained by stacking $M-1$ frames of the saved displacement JPEG images.

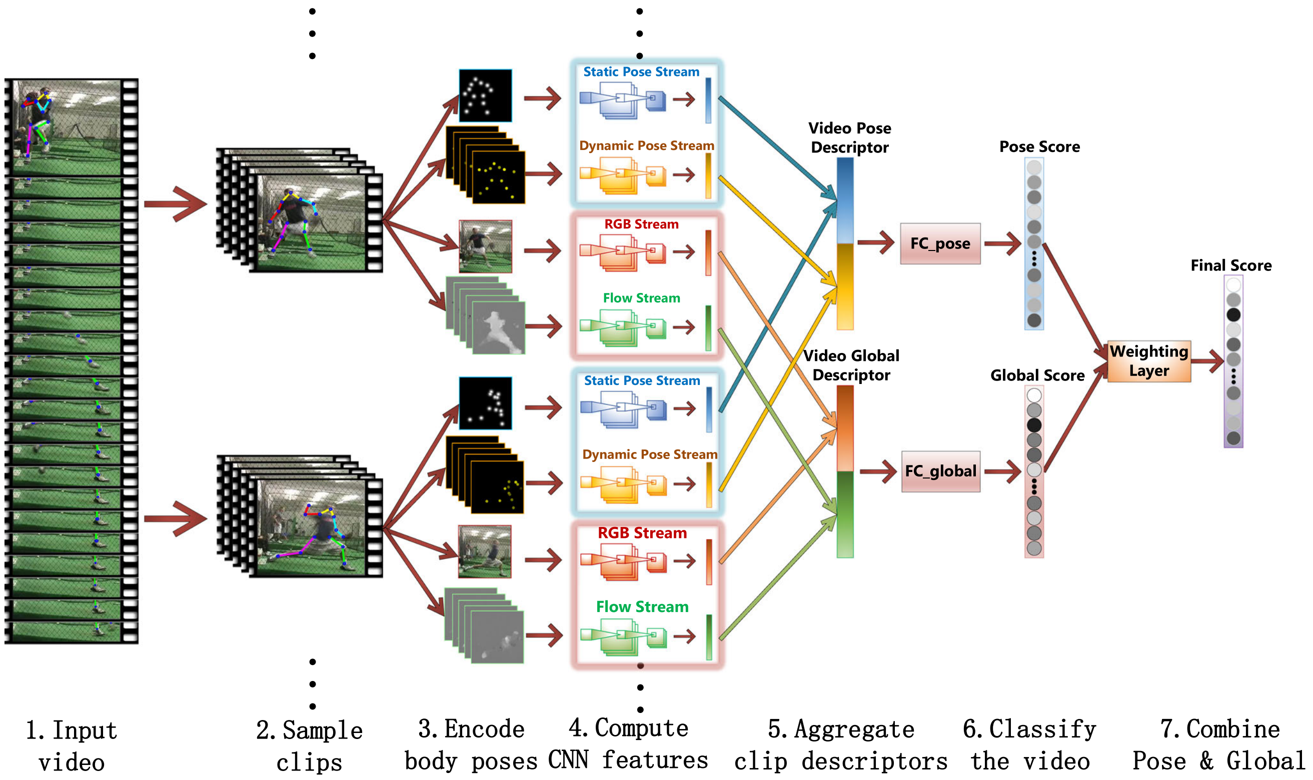

Fig. 4 Pipeline of the proposed approach. Our system (1) takes an input video with (annotated or estimated) body poses, (2) splits the video into clips, (3) encodes body poses as light-spot images and joint displacement volumes, (4) computes features for each clip using convolutional neural networks (CNNs), (5) aggregates clip descriptors into a video descriptor, (6) classifies each video using a softmax classifier, and finally (7) combines pose and global scores via a weighting layer

4 Action Recognition Framework

4.1 Overview

Figure 4 illustrates the overall pipeline of the proposed approach. Given a video sequence, we split it into clips. For each clip, we use separate CNNs to extracted pose features from the light-spot images and displacement volumes, and extract global features from RGB images and optical flow images. After aggregating clip descriptors into video descriptors, video pose descriptors and global pose descriptors are applied separately to predict classification confidence scores, which are trained by minimizing the softmax loss. We further introduce a weighting layer to fuse the pose and global cues effectively for accurate action recognition.

4.2 Clip Feature Extraction

Given a video input, we obtain the body poses in each frame by either manual annotation or automatic pose estimation. We then split the video sequence into clips of 10 frames. For the pose sequences in each split, we encode them into light-spot images and joint displacement volumes using the proposed pose encoding scheme described in Sect. 3. Two separate networks, namely, static pose stream and dynamic pose stream, are applied to extract static and dynamic pose features from the light-spot images and displacement volumes, respectively. Specifically, the sixth frame of the light-spot images in each clip is fed into the static pose stream, and the displacement images of the 10 frames are stacked to form a displacement volume of 20 stacked channels to be fed into the dynamic pose stream. For dynamic stream, we simply use 2D convolutional kernels. For both the static and dynamic streams, we use VGG-M [21] with 5 convolutional and 3 fully connected layers as the base model because it is a smaller network, thus making the extraction of the features more efficient.

To obtain global features, we directly apply a VGG-M model pre-trained on the ImageNet dataset [40] and a VGG-M model pre-trained on the UCF-101 dataset [41] to extract clip features from RGB images and optical flows respectively, in a two-stream fashion [9]. Similarly, the sixth frame of the RGB images in each clip is fed into the RGB stream, and the optical flows of the 10 frames are stacked to form a optical flow volume of 20 stacked channels to be fed into the flow stream.

4.3 Video-Level Classification

To obtain the video descriptor, we first split each video sequence in the training or testing set into clips of 10 frames, with 5 frames overlapping between two consecutive clips.

For each network, the outputs of the FC6 layer with $k = 4K$ values are extracted as clip descriptors, which are then aggregated by max aggregation and normalized to $[-1, 1]$ to obtain a video descriptor. Next, the video descriptors from the static pose stream and dynamic pose stream are concatenated as a comprehensive video pose descriptor, while those from the RGB stream and optical flow stream are concatenated as a comprehensive video global descriptor. Finally, each of video pose descriptor and video global descriptor with $k = 8K$ values is fed into a fully connected layer to predict classification confidence scores, with output $s^p = (s_1^p, \ldots, s_C^p)$ and $s^g = (s_1^g, \ldots, s_C^g)$ over C categories, respectively.

To effectively combine the results of s^p and s^g, we further introduce a weighting layer. Specifically, the fusion weights are learnable parameters defined as $w = (w_1, \ldots, w_C)$, where each value corresponds to a different category. We term $w^p = w = (w_1, \ldots, w_C)$ and $w^g = 1 - w = (1-w_1, \ldots, 1-w_C)$ as the weights for fusing the two scores, where w^p and w^g indicate how confidently we can rely on the pose features and global features, respectively, to predict the final confidence score $s^f = s^p \cdot w^p + s^g \cdot w^g$.

The total loss for the network is thus computed by: $L = \lambda_1 L_{cls}(s^p, \hat{g}) + \lambda_2 L_{cls}(s^g, \hat{g}) + \lambda_3 L_{cls}(s^f, \hat{g})$ where $\hat{g}$ is the ground-truth action label and each component is a cross-entropy loss. In our experiments, we simply set all $\lambda_i = 1$. For $L_{cls}(s^f, \hat{g})$, we only propagate back to the fusion layer, since full back-propagation did not bring an improvement.

5 Experiments

5.1 Datasets

To evaluate the proposed method, we use two challenging datasets for experiments, namely the sub-J-HMDB [14] and Penn-Action dataset [42].

The Penn Action dataset contains 2326 videos of 15 action categories: baseball pitch, clean and jerk, pull ups, strumming guitar, baseball swing, golf swing, push ups, tennis forehand, bench press, jumping jacks, sit-ups, tennis serve, bowling, jump rope, and squats. Each category contains 82–231 videos. This dataset provides 13 human joint annotations for each frame. The 50/50 train/test split provided by the dataset is used for the experiments. The lengths of frames in the videos vary from 18 to 663. Approximately, 662 out of the 2326 videos contain multiple persons.

The sub-JHMDB dataset is a subset of the JHMDB dataset [14] that contains 316 videos distributed over 12 action categories brush hair, catch, clap, climb stairs, golf, jump, kick ball, pick, pour, pull-up, push, run, shoot ball, shoot bow, shoot gun, sit, stand, swing baseball, throw, walk, and wave. Each category contains 19–42 videos. We use the threefold cross validation setting provided by the dataset for the experiments. The lengths of frames in the videos vary from 16 to 40. Approximately, 30 out of the 316 videos contain multiple persons.

5.2 Implementation Details

The clip feature extractors of the static and dynamic pose streams are trained separately. For the Penn Action dataset, the static pose stream and dynamic pose stream are trained using the same settings, which are determined on an 80/20% train/val split. Throughout training, we use stochastic gradient descent (SGD) with a batch size of 128, a momentum of 0.9 and a weight decay of 0.0005. The networks are initialized with the model pre-trained on ImageNet [40] to facilitate training speed. To avoid overfitting, we adopt dropout and data augmentation. Aggressive dropout ratios of 0.9 are used for the first two fully connected layers. We also employ data augmentation in the form of random cropping and horizontal flipping. The learning rate is decreased according to a fixed schedule. We start with a learning rate of 0.003, divide it by 10 at 5 and $7K$ iterations, and terminate training at $8K$ iterations. For the sub-JHMDB dataset, we do not train the pose feature extractor since this dataset has only 316 videos. Rather, we directly apply the pose models trained on the Penn Action dataset to extract pose features from the sub-JHMDB dataset.

For training the video-level classifier, we use batch GD and apply the Adam solver, with momentum parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$. The initial weights w_c in the weighting layer are all set to 0.5. For both the Penn Action dataset and sub-JHMDB dataset, we train $5K$ iterations with learning rate 0.01. For ablation studies, we evaluate the separate pose descriptor or global descriptor by minimizing a single cross-entropy loss $L_{cls}(s^p, \hat{g})$ or $L_{cls}(s^g, \hat{g})$. If pose descriptor or global descriptor only contains appearance or motion component, we duplicate the descriptor vector with $k = 4K$ values by a factor of 2, forming a descriptor vector with $k = 8K$ values. Thus, fair comparison is ensured.

5.3 Ablation Studies

This subsection is devoted to investigating the effectiveness of different design choices of the proposed approach.

Table 1 Comparison of different joint grouping schemes: integral level, half level, limb level and joint level

Grouping scheme	Static	Dynamic
Integral level	86.5	93.5
Half level	90.4	94.0
Limb level	91.4	93.5
Joint level	94.3	94.5

The results are obtained using the ground-truth pose on the Penn Action dataset (% accuracy)

5.3.1 Joint Grouping Scheme

The performances using different joint grouping schemes are compared in Table 1 for both the static and dynamic pose streams on the Penn Action dataset with ground-truth pose input.

For the static pose stream, we observe a stable improvement when the input scheme switches from coarse level to fine level (integral level → half level → limb level → joint level). We obtain the best result of 94.3% with the joint-level input, which corresponds to our expectation that a finer-level input can be more semantically informative.

For the dynamic pose stream, we do not observe remarkable improvement when we switch the joint grouping scheme from coarse level to fine level. The difference between the worst and best performance is only 1%, indicating a bottleneck might be reached.

The performances of both the static and dynamic pose streams demonstrate that the proposed pose features are effective for distinguishing actions with discriminative skeletal structures. In the remainder of our experiments, we use joint-level input for the static pose stream and integral-level input for the dynamic pose stream since this choice is a good trade-off between accuracy and efficiency.

5.3.2 Impact of Ground-Truth and Estimated Pose Input

To evaluate the impact of the pose input quality, we compare the recognition performance using the estimated pose and ground-truth pose. In this part, we focus on pose features and do not consider global features. The pose estimation algorithm in [18] is employed as an off-the-shelf method to estimate body poses for each frame. Although obvious error in the estimated poses may exist, we do not fine-tune the estimated poses. Examples of pose estimation results from both datasets for the successful and failed cases are shown in Fig. 5.

For automatically estimated poses, we also compare two input strategies. One strategy is to use all detected human bodies, which we refer to as the multi-person input strategy (Multi). The other strategy is to retain one individual with the highest confidence, which we refer to as the key-person input strategy (Key). In our experiment, the confidence of each human body is obtained by simply summing up the confidence scores of all its joints.

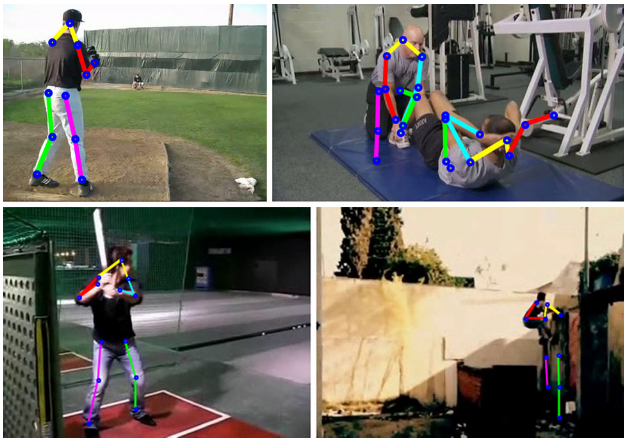

Fig. 5 Illustration of the human pose estimator [18] used in our experiments. Successful examples and failure cases on the Penn Action dataset (top row) and on the sub-JHMDB dataset (bottom row)

Table 2 Impact of the ground-truth pose versus estimated pose on the Penn Action dataset (% accuracy)

Pose input	Static	Dynamic	Static + dynamic
Ground-truth	94.3	93.5	95.4
Estimated (Multi)	90.6	92.4	93.6
Estimated (Key)	86.2	88.1	88.7

The recognition accuracies are compared in Table 2. We observe that the performance with the multi-person input strategy decreases only 1.8% compared with the ground-truth pose input, indicating that our pose representation is robust to errors in pose estimation. The reasons are twofold. First, our pose encoding scheme can accept multi-body input. As shown in Table 2, the multi-person input strategy gains significant improvement compared with the key-person input strategy for both static pose features and dynamic pose features. As described in Sect. 5.1, Penn Action dataset has one fourth of its total videos containing multiple persons. Although the performance of the key-person input strategy can be improved if we use a more sophisticated approach to identify the key actor, we manage to achieve an improvement in classification performance without additional pre-processing. Second, compared with hand-crafted approaches, CNNs parse body poses in a structured perspective. Thus, our approach is less vulnerable to modestly estimated body poses.

5.3.3 Analysis of the Impact of Pose Errors

Here, we investigate the impact of pose input with errors. Usually, accurate body poses are unavailable in practice and the results obtained by automatic pose estimation are imperfect. Even body poses obtained using a motion capture system or through manual annotation may be inaccurate. Hence, a recognition approach needs to be robust to pose errors to be applied in realistic videos.

In general, there are two types of errors. The first type is that some portion of joints are missing, which could occur if a human body is truncated by image boundaries. The other type is that the joint locations are disturbed by noise. Here, we focus on the former type because it is more intuitive. We also consider a special case that only upper-body or lower-body joints are provided and examine which actions are more recognizable under these conditions.

First, we investigate the impact of missing joints. The question is how many skeletal joints are adequate for effective video classification? To this end, we evaluate the recognition performance with $K = 0, 2, 4, 6, 8, 10$ missing joints. To avoid complementation between consecutive frames, we randomly select K joints from a total of 13 joints, and we skip the same K joints in the encoding process for all the frames in the same video. The recognition accuracies with different numbers of missing joints are presented in Fig. 6. We observe a stable performance decrease when the number of missing joints increases. In general, the performance of the dynamic pose stream has a smaller decrease than that of the static pose stream, providing a recognition accuracy of over 85% using only 9 out of 13 joints and an accuracy of over 50% using only 5 out of 13 joints. This indicates that the dynamic pose stream is less vulnerable to missing joints than the dynamic pose stream. Combining the static pose features with dynamic pose features do not witness further improvement when the performance of static pose is relatively too low. Thanks to the architecture of CNNs, we are able to parse human actions with an arbitrary number of joints as input and maintain satisfactory results with a few missing joints. Conversely, previous pose-based approaches, including the recent P-CNN [35] and JDD [36], require a fixed number of joints as input, which hinders their application in real-world scenes.

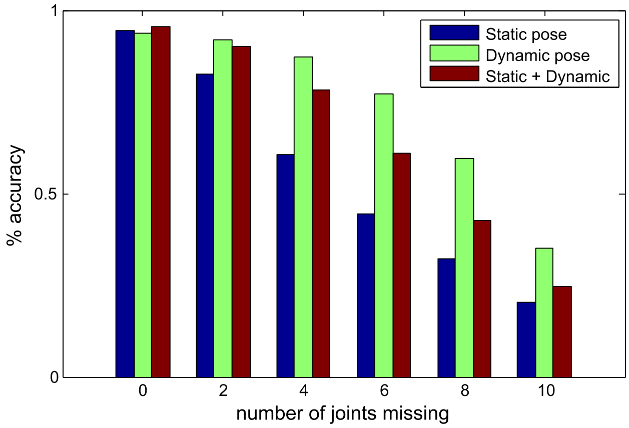

Fig. 6 Action recognition accuracy with different numbers of missing joints

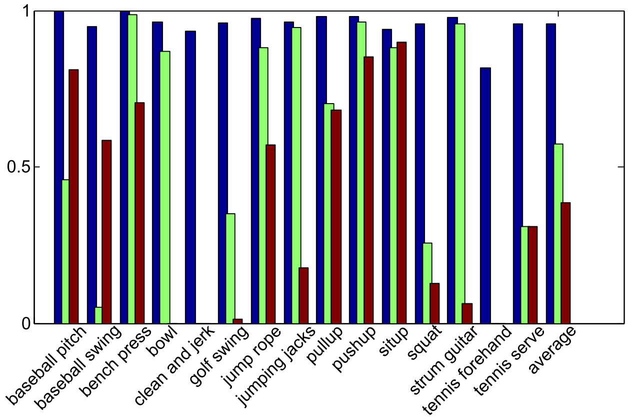

Fig. 7 Per class accuracy on the Penn Action dataset for full-body pose (blue), upper-body pose (green) and lower-body pose (red)

Next, we examine the special condition when only upper-body or lower-body available. The recognition performance is compared in Fig. 7. We can observe that actions like baseball pitch and baseball swing are more recognizable using upper-body pose than lower-body pose, because these actions mainly contain upper-body motion. Conversely, actions like bowling and jumping rope are more discriminative using lower-body pose. This probably because these actions require many feet movements.

5.3.4 Complementarity of Pose and Global Features

Finally, we quantitatively analyse the complementarity between pose features and global features. Our pose features are derived from automatically estimated poses.

Table 3 Performance of pose features, global features and combined features. The results are obtained using the ground-truth pose on the Penn Action dataset (% accuracy)

Model	Accuracy
RGB	81.4
Optical flow	94.4
Static pose	90.6
Dynamic pose	92.4
Global features	95.1
Pose features	93.6
Combined features	98.2

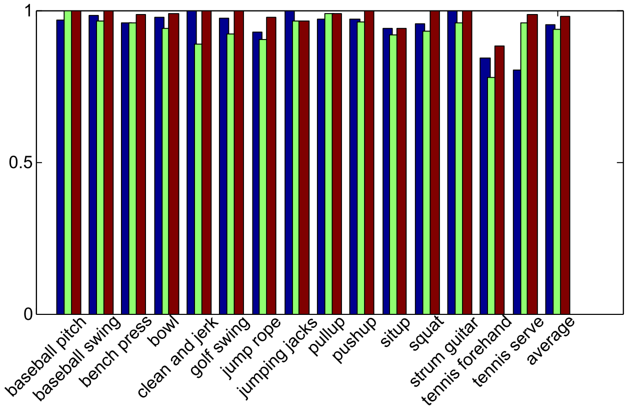

Fig. 8 Per class accuracy on the Penn Action dataset for global features (blue), pose features (green) and combined features (red)

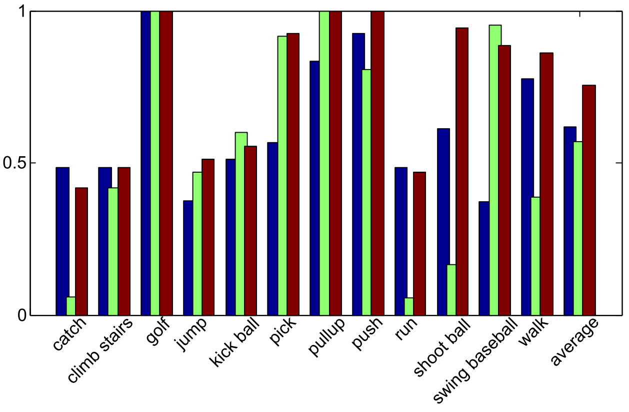

Fig. 9 Per class accuracy on the sub-JHMDB dataset for global features (blue), pose features (green) and combined features (red)

As shown in Table 3, combining both pose and global features obtains an accuracy of 98.2% on Penn Action dataset, which is higher than that obtained using only pose or global features. A detailed comparison on the Penn Action dataset is presented in Fig. 8. Although action recognition using either pose features or global features alone achieves high accuracy, combining both gains an additional performance improvement.

The accuracies on the sub-JHMDB dataset using pose features, global features and combined features are shown in Fig. 9. Global features perform well on recognizing actions with strong scene characteristics, such as basketball shooting (on a basketball court) and pull-up (on a horizontal bar); however, these features may fail to recognize actions such as baseball swinging. Close examination of the confusion matrix (see Figs. 10, 11) reveals that the incorrectly predicted actions share overall similarity with the ground-truth actions. For example, baseball swinging is mainly confused with golf swinging, which also shares a similar hand swinging movement. Our pose features, however, focus on human body structures; thus, they have a high accuracy in distinguishing baseball swinging from golf swinging. Combing pose features and global features obtains a significant improvement in accuracy.

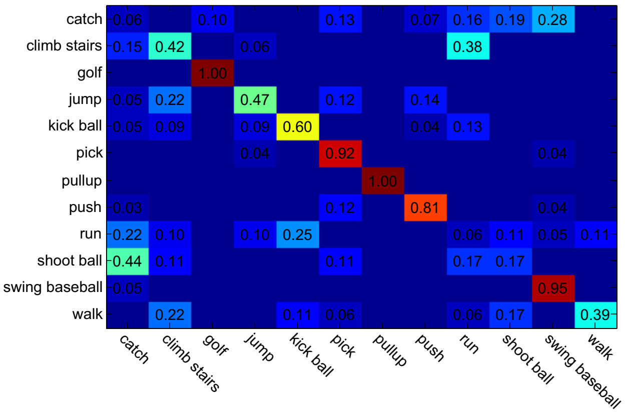

Fig. 10 The confusion matrix on the sub-JHMDB dataset using pose features

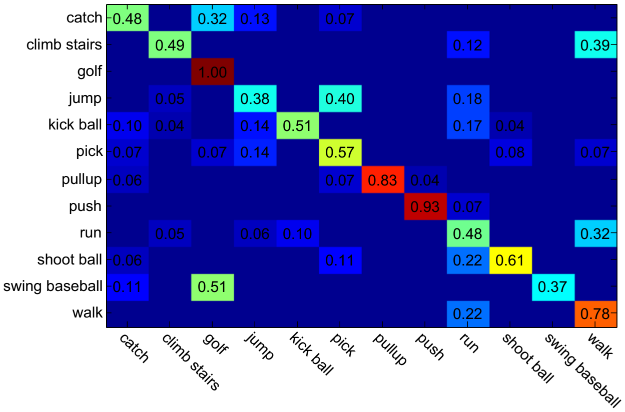

Fig. 11 The confusion matrix on the sub-JHMDB dataset using global features

5.4 Comparison with the State-of-the-Art

In this subsection, we compare our approach to state-of-the-art approaches on the Penn Action dataset and the sub-JHMDB dataset. Among these approaches, Dense Trajectories [5], Action Bank [4] and C3D [10] are video-feature-based approaches. HLPF [14] and Pose [43] are pose-based approach that only uses 2D pose as input. Acteme [42], MST [33], Graph Model [44], RPAN [37], Pose + idt-fv [43], P-CNN [35] and JDD [36] are pose-based approaches aided by features extracted from RGB or optical flow frames. Since some approaches do not use pose input, we use automatically estimated poses in the experiments for a fair comparison.

The recognition performances of our method and other methods on the Penn Action dataset and the sub-JHMDB

Table 4 Recognition accuracy on the sub-JHMDB and the Penn-Action datasets (% accuracy)

Method	Penn Action	sub-JHMDB
Global features only		
Dense [5]	73.4	46.0
Action bank [42]	83.9	–
C3D [36]	86.0	–
Pose features only		
HLPF [14]	–	54.1
Pose [43]	79.0	61.5
Ours	93.6	65.7
Pose + Global features		
Actemes [42]	79.4	–
MST [33]	74.0	45.3
Graph model [44]	85.5	61.2
JDD [36]	87.4	77.7
Pose + idt-fv [43]	92.9	74.6
RPAN [37]	97.4	78.6
P-CNN [35]	–	66.8
Ours	98.2	79.0

dataset are shown in Table 4. RPAN is the best among the existing approaches. On the Penn Action dataset, our method using pose features alone obtains an accuracy of 93.6%, outperforming the other approaches using only pose features. Our method explicitly considers the spatial relations between body joints. We believe that this is the primary reason for our superior performance. This result indicates that human pose can be an effective feature for distinguishing actions defined by specific spatial configurations and motion patterns. If we combine pose features with global features, the accuracy would further increase to 98.2%, outperforming the accuracy of RPAN.

On the sub-JHMDB dataset, our method obtains an accuracy of 79.0%, and it is better than the recent CNN-based methods P-CNN, RPAN and JDD. As a final note, both the Penn Action dataset and the sub-JHMDB dataset are small. We believe that given a larger dataset, we can train a more general model and achieve higher accuracy.

6 Conclusion and Future Work

In this paper, we propose a novel pose-based action representation that can effectively model human actions with flexible 2D body pose input that contains multiple bodies or partial visible bodies. A human pose encoding scheme is designed to encode static and dynamic pose components into sparse light-spot images and joint displacement volumes, respectively, which can be directly used as network input.

We experimentally demonstrate that pose features and global features are highly complementary. Thus, we propose an action recognition framework to perform multi-modal action recognition in monocular videos. Compared with the recent pose-based approaches P-CNN [35] and JDD [36], our approach not only handles more flexible pose input but also relies on overall pose structures; thus, it is more robust to pose errors. Our recognition framework achieves promising classification performance on the Penn Action dataset and the sub-JHMDB dataset.

In the future, we plan to combine pose estimation and action recognition in a unified framework since the two tasks are naturally highly coupled. Although there have been several attempts exploring such a framework [31,44,45], it is still a less explored area in the scope of deep learning architectures. Since our current model is not trained end-to-end, we also plan to address this issue in our future work.

Acknowledgements The research is supported in part by NSFC (61572424) and the People Programme (Marie Curie Actions) of the European Unions Seventh Framework Programme FP7 (2007–2013) under REA Grant Agreement No. 612627-"AniNex". Min Tang is supported in part by NSFC (61572423) and Zhejiang Provincial NSFC (LZ16F020003).

References

1. Cristani, M.; Raghavendra, R.; Del Bue, A.; Murino, V.: Human behavior analysis in video surveillance: a social signal processing perspective. Neurocomputing **100**, 86–97 (2013)
2. Rautaray, S.S.; Agrawal, A.: Vision based hand gesture recognition for human computer interaction: a survey. Artif. Intell. Rev. **43**(1), 1–54 (2015)
3. Papachristou, K.; Nikolaidis, N.; Pitas, I.; Linnemann, A.; Liu, M.; Gerke, S.: Human-centered 2d/3d video content analysis and description. In: International Conference on Electrical and Computer Engineering, pp. 385–388 (2014)
4. Sadanand, S.; Corso, J.J.: Action bank: a high-level representation of activity in video. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1234–1241 (2012)
5. Wang, H.; Kläser, A.; Schmid, C.; Liu, C.L.: Dense trajectories and motion boundary descriptors for action recognition. Int. J. Comput. Vis. **103**(1), 60–79 (2013)
6. Wang, H.; Schmid, C.: Action recognition with improved trajectories. In: IEEE International Conference on Computer Vision, pp. 3551–3558 (2013)
7. Zhu, J.; Wang, B.; Yang, X.; Zhang, W.; Tu, Z.: Action recognition with actons. In: IEEE International Conference on Computer Vision, pp. 3559–3566 (2013)
8. Huang, S.; Ye, J.; Wang, T.; Jiang, L.; Li, Y.; Wu, X.: Extracting discriminative parts with flexible number from low-rank features for human action recognition. Arab. J. Sci. Eng. **41**(8), 2987–3001 (2016)
9. Simonyan, K.; Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Annual Conference on Neural Information Processing Systems, pp. 568–576 (2014)
10. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In:

IEEE International Conference on Computer Vision, pp. 4489–4497 (2015)

11. Wang, X.; Farhadi, A.; Gupta, A.: Actions∼ transformations. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2658–2667 (2016)
12. Feichtenhofer, C.; Pinz, A.; Zisserman, A.: Convolutional two-stream network fusion for video action recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1933–1941 (2016)
13. Wang, C.; Wang, Y.; Yuille, A.L.: An approach to pose-based action recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 915–922 (2013)
14. Jhuang, H.; Gall, J.; Zuffi, S.; Schmid, C.; Black, M.J.: Towards understanding action recognition. In: IEEE International Conference on Computer Vision, pp. 3192–3199 (2013)
15. Moussa, M.M.; Hemayed, E.E.; El Nemr, H.A.; Fayek, M.B.: Human action recognition utilizing variations in skeleton dimensions. Arab. J. Sci. Eng. pp. 1–14 (2017)
16. Bulat, A.; Tzimiropoulos, G.: Human pose estimation via convolutional part heatmap regression. In: European Conference on Computer Vision, pp. 717–732 (2016)
17. Newell, A.; Yang, K.; Deng, J.: Stacked hourglass networks for human pose estimation. In: European Conference on Computer Vision, pp. 483–499 (2016)
18. Cao, Z.; Simon, T.; Wei, S.E.; Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. arXiv preprint arXiv:1611.08050 (2016)
19. Ramanathan, V.; Huang, J.; Abu-El-Haija, S.; Gorban, A.; Murphy, K.; Fei-Fei, L.: Detecting events and key actors in multi-person videos. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3043–3053 (2016)
20. Krizhevsky, A.; Sutskever, I.; Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Annual Conference on Neural Information Processing Systems, pp. 1097–1105 (2012)
21. Chatfield, K.; Simonyan, K.; Vedaldi, A.; Zisserman, A.: Return of the devil in the details: Delving deep into convolutional nets. arXiv preprint arXiv:1405.3531 (2014)
22. Huang, G.; Liu, Z.; Weinberger, K.Q.; van der Maaten, L.: Densely connected convolutional networks. arXiv preprint arXiv:1608.06993 (2016)
23. He, K.; Zhang, X.; Ren, S.; Sun, J.: Deep residual learning for image recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
24. Johansson, G.: Visual perception of biological motion and a model for its analysis. Percept. Psychophys. **14**(2), 201–211 (1973)
25. Feng, X.; Perona, P.: Human action recognition by sequence of movelet codewords. In: Proceedings of First International Symposium on 3D Data Processing Visualization and Transmission, pp. 717–721 (2002)
26. Thurau, C.; Hlavác, V.: Pose primitive based human action recognition in videos or still images. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008)
27. Schuldt, C.; Laptev, I.; Caputo, B.: Recognizing human actions: a local SVM approach. Int. Conf. Pattern Recognit. **3**, 32–36 (2004)
28. Blank, M.; Gorelick, L.; Shechtman, E.; Irani, M.; Basri, R.: Actions as space–time shapes. IEEE Int. Conf. Comput. Vis. **2**, 1395–1402 (2005)
29. Yang, Y.; Ramanan, D.: Articulated pose estimation with flexible mixtures-of-parts. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1385–1392 (2011)
30. Yao, B.; Fei-Fei, L.: Action recognition with exemplar based 2.5 d graph matching. In: European Conference on Computer Vision, pp. 173–186 (2012)
31. Yu, T.H.; Kim, T.K.; Cipolla, R.: Unconstrained monocular 3d human pose estimation by action detection and cross-modality regression forest. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3642–3649 (2013)
32. Xu, R.; Agarwal, P.; Kumar, S.; Krovi, V.; Corso, J.: Combining skeletal pose with local motion for human activity recognition. In: International Conference on Articulated Motion and Deformable Objects, pp. 114–123 (2012)
33. Wang, J.; Nie, X.; Xia, Y.; Wu, Y.; Zhu, S.C.: Cross-view action modeling, learning and recognition. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2649–2656 (2014)
34. Garbade, M.; Gall, J.: Handcrafting vs deep learning: an evaluation of ntraj + features for pose based action recognition. In: Workshop on New Challenges in Neural Computation and Machine Learning (NC^2), pp. 85–92 (2016)
35. Chéron, G.; Laptev, I.; Schmid, C.: P-cnn: Pose-based cnn features for action recognition. In: IEEE International Conference on Computer Vision, pp. 3218–3226 (2015)
36. Cao, C.; Zhang, Y.; Zhang, C.; Lu, H.: Action recognition with joints-pooled 3d deep convolutional descriptors. In: International Joint Conference on Artificial Intelligence, pp. 3324–3330 (2016)
37. Du, W.; Wang, Y.; Qiao, Y.: Rpan: An end-to-end recurrent pose-attention network for action recognition in videos. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3725–3734 (2017)
38. Carreira, J.; Zisserman, A.: Quo vadis, action recognition? A new model and the kinetics dataset. arXiv preprint arXiv:1705.07750 (2017)
39. Brox, T.; Bruhn, A.; Papenberg, N.; Weickert, J.: High accuracy optical flow estimation based on a theory for warping. In: European Conference on Computer Vision, pp. 25–36 (2004)
40. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009)
41. Soomro, K.; Zamir, A.R.; Shah, M.: Ucf101: a dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)
42. Zhang, W.; Zhu, M.; Derpanis, K.G.: From actemes to action: A strongly-supervised representation for detailed action understanding. In: IEEE International Conference on Computer Vision, pp. 2248–2255 (2013)
43. Iqbal, U.; Garbade, M.; Gall, J.: Pose for action-action for pose. In: 12th IEEE International Conference on Automatic Face & Gesture Recognition, pp. 438–445 (2017)
44. Xiaohan Nie, B.; Xiong, C.; Zhu, S.C.: Joint action recognition and pose estimation from video. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1293–1301 (2015)
45. Yao, A.; Gall, J.; Van Gool, L.: Coupled action recognition and pose estimation from multiple views. Int. J. Comput. Vis. **100**(1), 16–37 (2012)