# RPAN: An End-to-End Recurrent Pose-Attention Network for Action Recognition in Videos

Wenbin Du*[1,2] Yali Wang*[2] Yu Qiao†[2,3]

[1] Shenzhen College of Advanced Technology, University of Chinese Academy of Sciences, China

[2] Guangdong Provincial Key Laboratory of Computer Vision and Virtual Reality Technology,
Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China

[3] The Chinese University of Hong Kong, Hong Kong SAR, China

## Abstract

*Recent studies demonstrate the effectiveness of Recurrent Neural Networks (RNNs) for action recognition in videos. However, previous works mainly utilize video-level category as supervision to train RNNs, which may prohibit RNNs to learn complex motion structures along time. In this paper, we propose a recurrent pose-attention network (RPAN) to address this challenge, where we introduce a novel pose-attention mechanism to adaptively learn pose-related features at every time-step action prediction of RNNs. More specifically, we make three main contributions in this paper. Firstly, unlike previous works on pose-related action recognition, our RPAN is an end-to-end recurrent network which can exploit important spatial-temporal evolutions of human pose to assist action recognition in a unified framework. Secondly, instead of learning individual human-joint features separately, our pose-attention mechanism learns robust human-part features by sharing attention parameters partially on the semantically-related human joints. These human-part features are then fed into the human-part pooling layer to construct a highly-discriminative pose-related representation for temporal action modeling. Thirdly, one important byproduct of our RPAN is pose estimation in videos, which can be used for coarse pose annotation in action videos. We evaluate the proposed RPAN quantitatively and qualitatively on two popular benchmarks, i.e., Sub-JHMDB and PennAction. Experimental results show that RPAN outperforms the recent state-of-the-art methods on these challenging datasets.*

## 1. Introduction

Action recognition in videos has been intensely investigated in computer vision areas, due to its wide applications in video retrieval, human-computer interaction, etc [27]. The challenges of classifying actions in the wild videos mainly come from high dimension of video data, complex motion styles, large inter-category variations, and confused background clutters. With tremendous successes of deep models in image classification, there is a growing interest in developing deep neural networks for action recognition [9, 16, 18, 24, 31, 32, 38].

Recurrent Neural Networks (RNNs) show the power as sequential models for action videos [9, 24, 31]. In most of these works, the inputs to RNN are high-level features extracted from the fully-connected layer of CNNs, which may be limited in describing fine details about action. To alleviate this issue, attention-based models have been proposed [21, 28]. However, most existing attention approaches only utilize video-level category as supervision to train RNNs, which may lack a detailed and dynamical guidance (such as human movement over time), and consequently restrict their capacity of modeling complex motions in videos. Alternatively, human poses have proven useful for action recognition [14, 15, 17, 25, 40]. As shown in Subplots (a-c) of Fig. 1, human poses of different actors are closely related to the saliency regions in the average of convolutional feature maps estimated by CNN, and different joints of human pose can also be highly activated in certain individual feature maps. More importantly, spatial-temporal evolution of human poses in Subplot (d) of Fig. 1 yields a dynamical attention cue, which can guide RNNs to efficiently learn complex motions for action recognition in videos.

Inspired by this analysis, this paper proposes a novel recurrent pose-attention network (RPAN) for action recognition in videos, which can adaptively learn a highly-discriminative pose-related feature for every-step action prediction of LSTM. Specifically, we make three main contributions as follows. **Firstly**, unlike the previous works on pose-related action recognition, our RPAN is an end-to-end recurrent network, which allows to take advantage of dy-

*Equally-contributed first authors ({wb.du, yl.wang}@siat.ac.cn)

†Corresponding author (yu.qiao@siat.ac.cn)

2380-7504/17 $31.00 © 2017 IEEE
DOI 10.1109/ICCV.2017.402

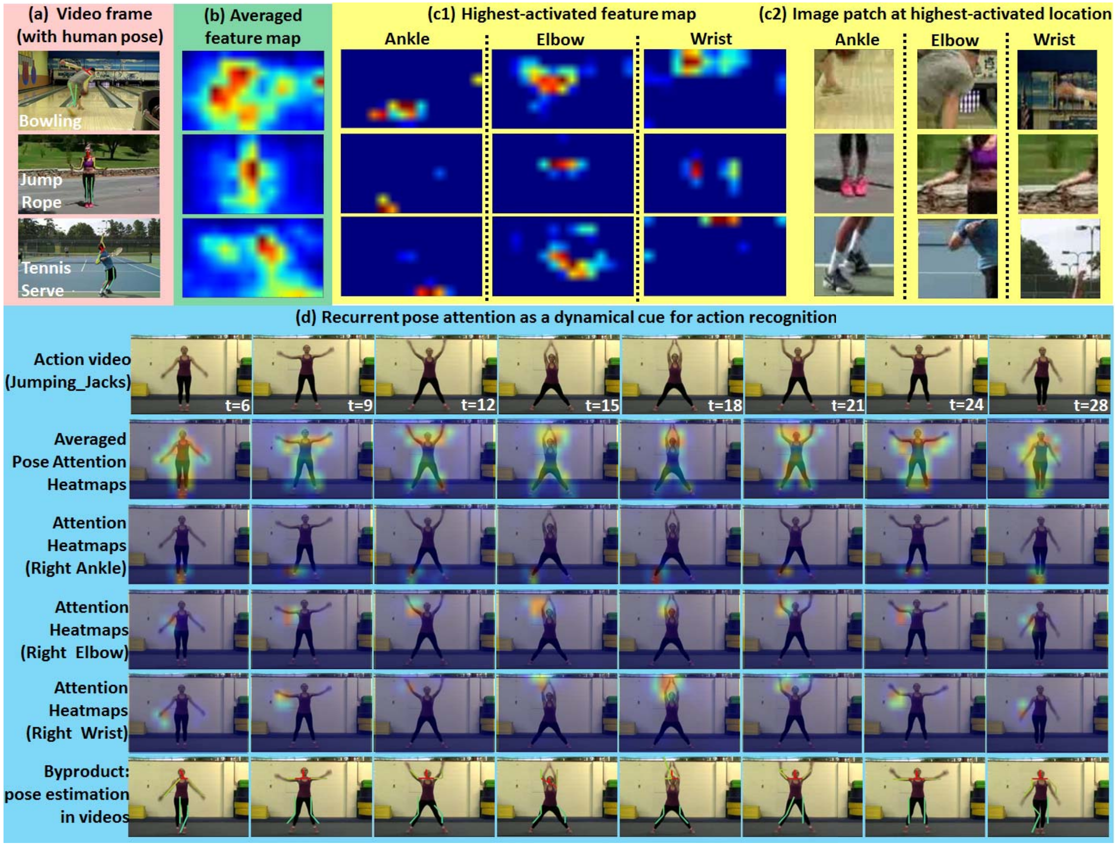
(a) Video frame (with human pose)
Bowling
Jump Rope
Tennis Serve
(b) Averaged feature map
(c1) Highest-activated feature map
Ankle
Elbow
Wrist
(c2) Image patch at highest-activated location
Ankle
Elbow
Wrist
(d) Recurrent pose attention as a dynamical cue for action recognition
Action video (Jumping_Jacks)
t=6
t=9
t=12
t=15
t=18
t=21
t=24
t=28
Averaged Pose Attention Heatmaps
Attention Heatmaps (Right Ankle)
Attention Heatmaps (Right Elbow)
Attention Heatmaps (Right Wrist)
Byproduct: pose estimation in videos

Figure 1. Our motivations. (a) The sampled video frames of different actions in PennAction. The ground truth human poses are annotated in the video frames. (b) Averaged feature map. We generate convolutional cube from the 5a layer ($9 \times 15 \times 1024$) in the spatial-stream of temporal segment net [47], and then sum the convolutional cube over feature channels to obtain this averaged feature map. (c1) The highest-activated feature map for different human joints (Ankle, Elbow, and Wrist). First, the video frame is reshaped to be the same size as the feature map in the convolutional cube. Then, we find the location of each human joint on all the feature maps. Finally, the feature map with the highest-activated value at the joint location is selected as the highest-activated feature map for the corresponding joint. (c2) Image patch at the highest-activated location. The highest-activated feature map is firstly reshaped to be the same size as the video frame. Then we find the image patch ($80 \times 80$) from the video frame, according to the location of the highest-activated value in the resized feature map. (d) The pose-attention-related heat maps and estimated poses of sampled video frames by our recurrent pose attention network (RPAN). One can see that, human pose is a discriminative cue for action recognition (Subplots a-c). More importantly, spatial-temporal evolution of human pose can provide a dynamical guidance to assist recurrent network learning (Subplot d).

namical human pose cues to improve action recognition in a unified framework. **Secondly**, our novel pose-attention mechanism can learn a number of robust human-part features, with guidance of human body joints in videos. By sharing attention parameters partially on the semantically-related joints, human-part features not only represent the distinct joint characteristics, but also preserve rich human-body-structure information which is robust to recognize complex actions. Subsequently, these features are fed into a human-part pooling layer to construct a discriminative pose feature for temporal action modeling. **Thirdly**, one important byproduct of our RPAN is pose estimation in videos, which can be applied to coarse pose annotation in action videos. To show the effectiveness of our RPAN, we conduct extensive experiments on two popular benchmarks (sub-JHMDB and PennAction) in pose-related action recognition. The empirical results show that, the classification accuracy of RPAN outperforms the recent state-of-the-art approaches on these challenging datasets.

## 2. Related Works

**Action Recognition**. Early approaches for action recognition are mainly based on hand-crafted features [20, 41, 42], which represent videos with a number of local descriptors. However, hand-crafted approaches may only capture the local contents and thus lack the discriminative power to recognize complex actions [45]. With significant successes of CNNs in image recognition [12, 19, 30, 33], several works proposed to design effective CNNs for action recognition in videos [16, 18, 29, 32, 38, 46]. One of the most popular approaches is two-stream CNNs [29], where

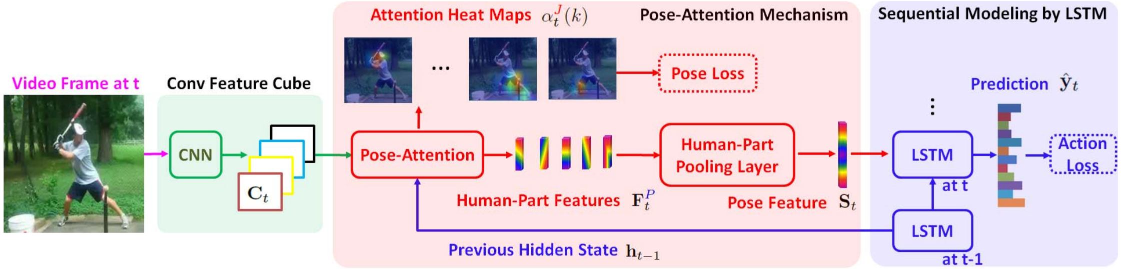
Attention Heat Maps $\alpha_t^J(k)$
Pose-Attention Mechanism
Sequential Modeling by LSTM
Video Frame at t
Conv Feature Cube
...
Pose Loss
Prediction $\hat{\mathbf{y}}_t$
CNN
$\mathbf{C}_t$
Pose-Attention
Human-Part Pooling Layer
LSTM
Action Loss
at t
Human-Part Features $\mathbf{F}_t^P$
Pose Feature $\mathbf{S}_t$
LSTM
Previous Hidden State $\mathbf{h}_{t-1}$
at t-1

Figure 2. Our End-to-End Recurrent Pose-Attention Network (RPAN). At the $t$-th step, the video frame is fed into CNN to generate the convolutional feature cube $\mathbf{C}_t$. Then, with guidance of the previous hidden state $\mathbf{h}_{t-1}$ of LSTM, our pose attention mechanism learns several human-part-related features $\mathbf{F}_t^P$ from $\mathbf{C}_t$. As attention parameters are partially shared on the semantic-related human joints belonging to the same body part, our human-part-related features encode robust body-structure-information to discriminate complex actions. Finally, these features are fed into the human-part pooling layer to produce a highly-discriminative pose-related feature $\mathbf{S}_t$, which is the input to LSTM for action recognition. The whole RPAN can be efficiently trained in an end-to-end fashion, by considering the action loss (prediction $\hat{\mathbf{y}}_t$ vs. action label) and the pose loss (attention heat maps $\alpha_t^J(k)$ vs. pose annotations) together.

spatial and temporal CNNs were designed to process RGB images and optical flows separately. One limitation in this approach is that the stacked optical flows can only capture motion information in short temporal scale. To improve the performance, several extensions have been proposed by designing trajectory-pooled deep descriptors [45], mining key volume of videos [54], fusing two streams [11], introducing temporal segments [47]. Furthermore, the sequential nature of video inspires researchers to learn video representations by RNNs, especially LSTM [9, 24, 31]. However, the inputs to these LSTMs are high-level features obtained from the fully-connected (FC) layer of CNNs, which are limited to represent fine action details in videos [1]. Recently, attention has been incorporated into LSTMs to learn detailed spatial or temporal action cues [21, 28, 51], motivated by its efficiency for image understanding [39, 50]. However, these attention methods only utilize video-level category as supervision, and thus lack the temporal guidance (such as human-pose dynamics) to train LSTMs. This may restrict their capacity of modeling complex motions in the real-world action videos.

**Pose-related Action Recognition**. Human pose has proven highly-discriminative to recognize complex actions [14, 15, 17, 25, 40]. One well-known pose-based representation is poselet [2] which has been applied to action recognition and detection in videos [34, 44, 52]. However, the hand-crafted features in these approaches may lack the discriminative power to represent pose-related complex actions. To improve the performance, several latent structures were proposed by learning meaningful hierarchical pose representations for action recognition [13, 22, 48]. Furthermore, with the recent development of deep models in action recognition [29, 38] and pose estimation [4, 7, 23, 26, 37, 36, 49], pose-related deep approaches [3, 5, 10] have been recently introduced to boost recognition accuracy. However, these approaches are not in an end-to-end learning procedure, since human poses are either given or estimated before action recognition. As a result, spatial-temporal pose evolutions may not effectively apply to action recognition in a unified framework.

Different from the works above, we propose a novel end-to-end recurrent pose-attention network (RPAN) for action recognition in videos. At each time step, our pose attention learns a highly-discriminative pose feature for key action regions, with the guidance of human joints. Subsequently, the resulting pose feature is fed into LSTM for action recognition. In this case, our RPAN naturally takes advantage of human-pose evolutions as a dynamical assistant task for action recognition, and thus it can alleviate the complexity of hand-crafted designs in the previous works.

## 3. Recurrent Pose-Attention Network (RPAN)

In this section, we describe the proposed Recurrent Pose-Attention Network (RPAN), which can dynamically identify the important pose-related feature to enhance every time-step action prediction of LSTM. First, the current video frame is fed into CNN to generate a convolutional feature cube. Then, our pose attention mechanism takes the previous hidden state of LSTM as a guidance to estimate a number of human-part-related features from the current convolutional cube. Our attention parameters are partially shared on semantic-related human joints, hence the learnt human-part features encode rich and robust body-structure-information. Next, these features are fed into a human-part pooling layer to produce a highly-discriminative pose-related feature for temporal action modeling within LSTM. The whole framework is shown in Fig. 2.

### 3.1. Convolutional Feature Cube from CNN

In this work, we use the well-known deep architecture in action recognition, two-steam CNNs [46, 47], to generate the convolutional cubes from spatial (RGB) and temporal (Optical Flow) stream CNNs. Since we follow [46, 47] to process two streams separately, we henceforth describe the

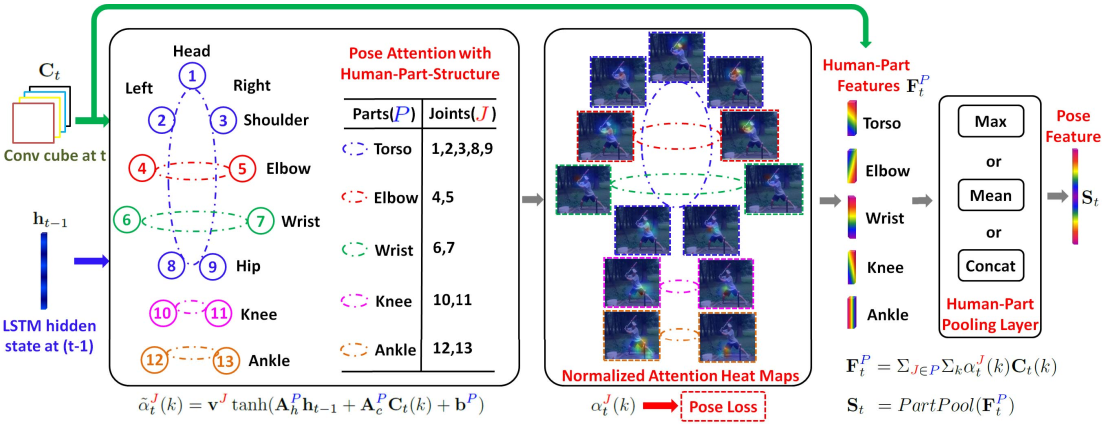
$\mathbf{C}_t$
Conv cube at t
$\mathbf{h}_{t-1}$
LSTM hidden state at (t-1)
Head
Left
Right
Shoulder
Elbow
Wrist
Hip
Knee
Ankle
1
2
3
4
5
6
7
8
9
10
11
12
13
Pose Attention with Human-Part-Structure
Parts($P$)
Joints($J$)
Torso
1,2,3,8,9
Elbow
4,5
Wrist
6,7
Knee
10,11
Ankle
12,13
Normalized Attention Heat Maps
Human-Part Features $\mathbf{F}_t^P$
Torso
Elbow
Wrist
Knee
Ankle
Max
or
Mean
or
Concat
Human-Part Pooling Layer
Pose Feature
$\mathbf{S}_t$
$\tilde{\alpha}_t^J(k) = \mathbf{v}^J \tanh(\mathbf{A}_h^P \mathbf{h}_{t-1} + \mathbf{A}_c^P \mathbf{C}_t(k) + \mathbf{b}^P)$
$\alpha_t^J(k)$
Pose Loss
$\mathbf{F}_t^P = \Sigma_{J\in P}\Sigma_k \alpha_t^J(k)\mathbf{C}_t(k)$
$\mathbf{S}_t = PartPool(\mathbf{F}_t^P)$

Figure 3. Our Pose-Attention Mechanism. We firstly group the semantically-related human joints into a number of body parts. For each body part $P$, we take the previous hidden state $\mathbf{h}_{t-1}$ of LSTM as guidance to generate attention heat maps $\alpha_t^J(k)$ (Eq. 2-3) for each joint $J \in P$. Since attention parameters are partially shared for the joints in $P$, their attention maps not only represent their joint characteristics, but also preserve the important body structure information. Subsequently, we use these attention maps $\alpha_t^J(k)$ in the human part $P$ to learn the human-part feature $\mathbf{F}_t^P$ from the convolutional cube $\mathbf{C}_t$ (Eq. 4). In this case, $\mathbf{F}_t^P$ can contain the robust human-part-information. Finally, we fuse all the human-part features with a human-part-pooling layer, to generate a discriminative pose feature for temporal modeling. More details can be found in Section 3.2.

convolutional feature cube in general to reduce notation redundancy. More details can be found in our experiments.

For the $t$-th video frame ($t = 1, ..., T$), we denote the convolutional cube from CNN as $\mathbf{C}_t \in \mathbb{R}^{K_1 \times K_2 \times d_c}$, which consists of $d_c$ feature maps with size of $K_1 \times K_2$. Furthermore, we denote $\mathbf{C}_t$ as a set of feature vectors at different spatial locations,

$$\mathbf{C}_t = \{\mathbf{C}_t(1), ..., \mathbf{C}_t(K_1 \times K_2)\}, \tag{1}$$

where the feature vector at the $k$-th location is $\mathbf{C}_t(k) \in \mathbb{R}^{d_c}$ and $k = 1, ..., K_1 \times K_2$. Based on the convolutional cube from CNN, we next propose a novel pose-attention mechanism to assist action prediction at each step of LSTM.

## 3.2. Pose Attention Mechanism

After obtaining $\mathbf{C}_t$, we use it for temporal modeling with LSTM. However, LSTM with only action-category supervision often lacks the dynamical guidance (such as human-pose movements over time). This may restrict the capacity of LSTM to learn complex motion structures in the real-world action videos. Motivated by the fact that spatial-temporal evolutions of human poses provide important cues for action recognition [17], we design a novel pose-attention mechanism to learn a discriminative pose feature for LSTM. An illustration of our pose attention is shown in Fig. 3.

**Pose-Attention with Human-Part-Structure**. In fact, human parts (such as Torso in Fig. 3) often contain more robust action information than individual joints (such as Head, Shoulders, and Hips in Fig. 3) [15, 25, 40]. Inspired by this, we propose a novel pose attention mechanism with human part structure.

Firstly, we group semantically-related human joints into a number of body parts in Fig. 3, where $P$ denotes a body part, and $J$ denotes a human joint belonging to $P$. For each body part $P$, we use the previous hidden state $\mathbf{h}_{t-1}$ of LSTM as action guidance, and estimate the importance of convolutional cube $\mathbf{C}_t$ for each joint $J \in P$,

$$\tilde{\alpha}_t^J(k) = \mathbf{v}^J \tanh(\mathbf{A}_h^P \mathbf{h}_{t-1} + \mathbf{A}_c^P \mathbf{C}_t(k) + \mathbf{b}^P), \tag{2}$$

where $\mathbf{C}_t(k)$ is the feature vector of $\mathbf{C}_t$ at the $k$-th spatial location ($k = 1, ..., K_1 \times K_2$), $\tilde{\alpha}_t^J(k)$ is the unnormalized attention score of $\mathbf{C}_t(k)$ for the joint $J$, and $\{\mathbf{v}^J, \mathbf{A}_h^P, \mathbf{A}_c^P, \mathbf{b}^P\}$ are attention parameters. Note that, $\mathbf{v}^J$ is distinct for each joint $J \in P$, while $\{\mathbf{A}_h^P, \mathbf{A}_c^P, \mathbf{b}^P\}$ are shared for all the joints in the body part $P$. *With this partial-parameter-sharing design, each joint heat map $\tilde{\alpha}_t^J(k)$ not only represents distinct joint characteristics, but also preserves rich human-part-structure information.*

Secondly, we normalize $\tilde{\alpha}_t^J(k)$ to the corresponding attention heat map $\alpha_t^J(k)$,

$$\alpha_t^J(k) = \frac{\exp\{\tilde{\alpha}_t^J(k)\}}{\sum_k \exp\{\tilde{\alpha}_t^J(k)\}}. \tag{3}$$

With $\alpha_t^J(k)$ of all the joints in the human part $P$, we can learn the **human-part-related feature** from $\mathbf{C}_t$,

$$\mathbf{F}_t^P = \Sigma_{J\in P}\Sigma_k \alpha_t^J(k)\mathbf{C}_t(k). \tag{4}$$

*Due to the novel human-part-structure design in our pose attention, the learned features $\mathbf{F}_t^P$ can contain body-structure robustness for complex actions.*

**Human-Part Pooling Layer**. To generate a highly dynamical and discriminative **pose-related feature** for temporal modeling, we design a human-part pooling layer to fuse all the human-part-related features,

$$\mathbf{S}_t = PartPool(\mathbf{F}_t^P), \tag{5}$$

where $PartPool$ is investigated with the $max$, $mean$ or $concat$ operations in our experiments.

Note that, our pose attention takes account of occlusions, via the proposed human-part-structure design. First, the human-part feature in Eq. (4) is the attention summarization of all joints belonging to this part. In this case, when some joints are occluded, other joints in the same part may be discriminative for action recognition. Second, the pose feature in Eq. (5) is the part-pooling of all human-part features. In this case, when some parts are occluded, other parts may still yield discriminative features for action recognition. As shown in Fig. 2-3, our approach correctly recognizes Baseball-Swing, even though the upper body of the player is self-occluded.

### 3.3. Sequential Modeling with LSTM

Finally, we feed the dynamical pose feature $\mathbf{S}_t$ into LSTM for temporal modeling,

$$(\mathbf{i}_t, \mathbf{f}_t, \mathbf{o}_t) = \sigma(\mathbf{U}^s_\star \mathbf{S}_t + \mathbf{U}^h_\star \mathbf{h}_{t-1} + \mathbf{b}_\star), \tag{6}$$

$$\mathbf{g}_t = \tanh(\mathbf{U}^s_g \mathbf{S}_t + \mathbf{U}^h_g \mathbf{h}_{t-1} + \mathbf{b}_g), \tag{7}$$

$$\mathbf{r}_t = \mathbf{f}_t \odot \mathbf{r}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t, \tag{8}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{r}_t), \tag{9}$$

$$\hat{\mathbf{y}}_t = softmax(\mathbf{U}^h_y \mathbf{h}_t + \mathbf{b}_y), \tag{10}$$

where $\star$ denotes $i$, $f$ and $o$ for $\mathbf{i}_t$, $\mathbf{f}_t$ and $\mathbf{o}_t$, the sets of $\mathbf{U}$ and $\mathbf{b}$ are the parameters of LSTM, $\sigma(\cdot)$ and $\tanh(\cdot)$ are the sigmoid and tanh functions, $\odot$ is the element-wise multiplication, $\mathbf{i}_t$, $\mathbf{f}_t$ and $\mathbf{o}_t$ are the input, forget and output gates, $\mathbf{g}_t$, $\mathbf{r}_t$ and $\mathbf{h}_t$ are the candidate memory, memory state and hidden state, and $\hat{\mathbf{y}}_t$ is the action prediction vector.

### 3.4. End-to-End Learning

Different from previous approaches in pose-related action recognition [3, 5], the proposed RPAN can be trained in **an end-to-end fashion** with the total loss,

$$\mathcal{L}_{total} = \lambda_{action} \mathcal{L}_{action} + \lambda_{pose} \mathcal{L}_{pose} + \lambda_\Theta \parallel \Theta \parallel_2, \tag{11}$$

where $\mathcal{L}_{action}$ and $\mathcal{L}_{pose}$ are respectively the action and pose losses, $\parallel \Theta \parallel_2$ is the weight decay regularization for all the model parameters, and $\lambda_{action}$,$\lambda_{pose}$,$\lambda_\Theta$ are the coefficients for action, pose losses and weight decay.

Given the training action label $\mathbf{y}_t$ (one-hot label vector), $\mathcal{L}_{action}$ is the cross-entropy loss between $\mathbf{y}_t$ and its prediction $\hat{\mathbf{y}}_t$ in Eq. (10),

$$\mathcal{L}_{action} = -\Sigma_{t=1}^{T} \Sigma_{c=1}^{C} \mathbf{y}_{t,c} \log \hat{\mathbf{y}}_{t,c}, \tag{12}$$

where $C$ is the number of action classes, $T$ is the number of total time steps. Furthermore, given the training pose annotation $\mathbf{M}_t^J$ (heat maps for all the joints), $\mathcal{L}_{pose}$ is the L-2 loss between $\mathbf{M}_t^J$ and the joint attention heat maps $\alpha_t^J$ in Eq. (3),

$$\mathcal{L}_{pose} = \Sigma_J \Sigma_{t=1}^{T} \Sigma_{k=1}^{K_1 \times K_2} (\mathbf{M}_t^J(k) - \alpha_t^J(k))^2, \tag{13}$$

where each training heat map $\mathbf{M}_t^J$ is generated by adding a fixed Gaussian centered at the corresponding joint location. Note that, the heat map loss like Eq. (13) has been widely used in deep networks for pose estimation [26, 36], since the pixel-level supervision yields richer pose representation. *With our end-to-end training procedure, spatial-temporal pose evolutions can be efficiently used as a dynamical guidance of action recognition in a unified framework.*

## 4. Experiments

In this section, we evaluate our recurrent pose-attention network (RPAN) on two popular benchmarks in pose-related action recognition, i.e., Sub-JHMDB [15] and PennAction [53], where Sub-JHMDB / PennAction consists of 316/2,326 videos with 12/15 action classes, and the full-body human joints are annotated for each video. Since videos in both datasets are collected from internet, the complex body occlusions, large appearance and motion variations make both datasets challenging for pose-related action recognition [14, 25]. We use the published evaluation protocol [14, 25] to report classification accuracy for both datasets. Note that, the joint information is only required for training in our RPAN, such information is not required for testing. For a testing frame, we use the estimated heatmaps of all joints (Eq. 2 - 3) to summarize the convolutional cube as a pose feature. This feature is then fed into LSTM for action recognition. All our experiments are performed in this way, without using the joint information in the test set.

### 4.1. Implementation Details

Unless stated otherwise, we perform our RPAN with the following implementation details. Firstly, for Sub-JHMDB / PennAction (size of $240 \times 320$ / $270 \times 480$), the convolutional cube is generated from the convolutional layer (the 5a layer, $8 \times 10 \times 1024$ / $9 \times 15 \times 1024$) of temporal segment net (TSN) [47], due to its good performance on action recognition in videos. Moreover, as TSN is a two-stream deep structure, the convolution cubes for different streams are separately generated by processing the RGB image and stacked optical flow of each video frame respectively. In this case, we perform our RPAN separately on the convolution cubes from different streams, similar to two-stream fashion of TSN. The training data sets for both benchmarks are augmented by the mirror operation. Secondly, human parts are defined as Torso, Elbow, Wrist, Knee and Ankle for both datasets, as shown in Fig. 3. For Sub-JHMDB / PennAction, the dimensions of all hidden variables in LSTM are 512/1024, and the dimensions of attention parameters $\{\mathbf{v}^J, \mathbf{A}_h^P, \mathbf{A}_c^P, \mathbf{b}^P\}$ are $\{1 \times 32, 32 \times 512, 32 \times$

Table 1. Evaluation of the proposed pose-attention mechanism via classification accuracy of our RPAN. 'Without': We take the feature vector from the fully-connected layer of CNN as the input to LSTM, without any attention. 'Share-All': We perform the attention mechanism without guidance of human joints, where the attention parameters $\{\mathbf{v}^J, \mathbf{A}_h^P, \mathbf{A}_c^P, \mathbf{b}^P\}$ in Eq. (2) are changed to be independent of human joints and body parts, i.e., $\{\mathbf{v}, \mathbf{A}_h, \mathbf{A}_c, \mathbf{b}\}$. 'Separate-Joint': We perform the attention mechanism with guidance of separate joints, where $\{\mathbf{v}^J, \mathbf{A}_h^P, \mathbf{A}_c^P, \mathbf{b}^P\}$ is changed for each separate joints without human-part-structure consideration, i.e., $\{\mathbf{v}^J, \mathbf{A}_h^J, \mathbf{A}_c^J, \mathbf{b}^J\}$. 'Human-Part': It is the proposed pose-attention mechanism with human-part-structure in Eq. (2).

| Attention in our RPAN | Sub-JHMDB | PennAction |
|---|---|---|
| Without | 68.5 | 95.0 |
| Share-All | 71.7 | 96.4 |
| Separate-Joint | 78.3 | 96.5 |
| Human-Part | **80.0** | **97.1** |

$1024, 32 \times 1\}$ / $\{1 \times 128, 128 \times 1024, 128 \times 1024, 128 \times 1\}$ respectively. Thirdly, for the training set, we add a fixed Gaussian (std: 5) centered at the joint location. Then, we resize the ground truth heat map of each human joint to be the same size as the attention heat map ($8 \times 10$ / $9 \times 15$ for Sub-JHMDB / PennAction) and normalize it. Finally, we train our RPAN with mini-batch stochastic gradient descent. For both datasets, 16 videos are randomly chosen in each training mini-batch, where 8 frames are randomly sampled from each video with equal interval. 8 frames from each test video with equal interval are selected and the last-frame prediction is used to report test accuracy of our RPAN. The momentum is 0.9, both action and pose coefficients $\lambda_{action}$, $\lambda_{pose}$ are 1, the weight decay coefficient $\lambda_{\Theta}$ is $5 \times 10^{-4}$, the learning rate is set to 0.1 initially, reduced to $10^{-2}$ after 40/50 epochs for Sub-JHMDB / PennAction. The training procedure stops at 100 epochs. We implement our RPAN by Theano [35], with multi-GPU of BPTT [8].

## 4.2. Properties of Our RPAN

To investigate the properties of our RPAN, we evaluate the effectiveness of its key model components on Sub-JHMDB (split one) and PennAction. To be fair, when we explore different strategies of one component in our RPAN, all other components are with the basic strategy, where the convolutional cube is extracted from the temporal-steam of TSN, the pose-attention with human-part-structure refers to Section 3.2, the human-part-pooling layer is based on the concat strategy.

**Pose Attention Mechanism**. We examine our pose-attention mechanism with different settings in Table 1. First, the 'Share-All' attention setting of our RPAN outperforms the 'Without' setting (1024 dimension feature vector of global pool layer in TSN is fed into LSTM without any attention). It illustrates that the visual attention is important for action recognition. Second, the 'Separate-Joint' attention setting outperforms the 'Share-All' setting. It shows that the human joint in each video frame is an effective dynamical guidance of attention. Finally, the 'Human-Part' attention setting, the proposed mechanism in Section 3.2, outperforms the 'Separate-Joint' setting. It demonstrates that,

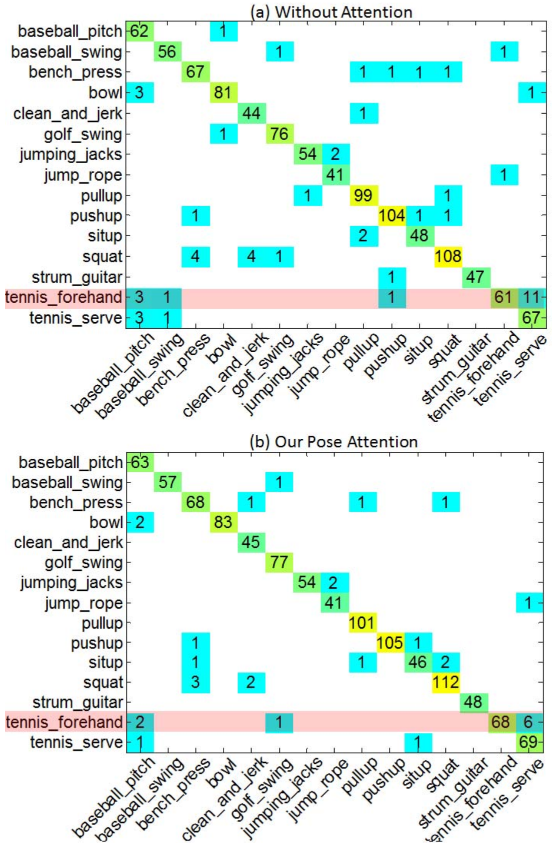

Figure 4. Confusion Matrix Comparison (PennAction). (a) 'Without Attention': the 'Without' setting in Table 1. (b) 'Our Pose Attention': our proposed approach. The values in the matrix for both cases are the number of test videos. First, the confusion matrix of 'Our Pose Attention' is much sparser than the one of 'Without Attention'. It illustrates that more test videos are correctly classified by our approach. Second, we compare two matrices on the 'tennis_forehand' action class, where our approach correctly classifies 7 more videos (68-61=7) than 'Without Attention'. The main difference comes from confusion between 'tennis_forehand' and 'tennis_serve'. In 'Without Attention', these two action classes are largely confused (11 confused test videos). On the contrary, our approach can take spatial-temporal pose evolutions as a dynamical attention cue to reduce confusion between similar actions.

Table 2. Classification accuracy with different strategies of our Human-Part-Pooling Layer.

| Human-Part-Pooling | Sub-JHMDB | PennAction |
|---|---|---|
| Max | **82.6** | 96.6 |
| Mean | 81.5 | 96.2 |
| Concat | 80.0 | **97.1** |

Table 3. Classification accuracy of RPAN with different basic CNNs. We evaluate RPAN, based on two widely-used CNNs in action recognition, i.e., Good-Practice CNN [46] and TSN [47].

| CNNs for RPAN | Sub-JHMDB | PennAction |
|---|---|---|
| Good-Practice CNN | 69.5 | 88.6 |
| TSN | **80.0** | **97.1** |

Table 4. Classification accuracy of RPAN with different streams of TSN. TSN-S: Spatial TSN. TSN-T: Temporal TSN. TSN-(S+T): Prediction score fusion on TSN-S and TSN-T. RPAN-S: RPAN with spatial TSN. RPAN-T: RPAN with temporal TSN. RPAN-(S+T): Prediction score fusion on RPAN-S and RPAN-T.

| Model Variants | Sub-JHMDB | PennAction |
|---|---|---|
| TSN-S | 55.6 | 80.4 |
| TSN-T | 72.2 | 93.3 |
| TSN-(S+T) | 72.2 | 93.8 |
| RPAN-S | 60.0 | 84.8 |
| RPAN-T | 80.0 | 97.1 |
| RPAN-(S+T) | **81.1** | **97.4** |

Table 5. Comparison with state-of-the-art on Sub-JHMDB (average over three splits) and PennAction.

| State-of-the-art | Year | Sub-JHMDB | PennAction |
|---|---|---|---|
| Dense+Pose [15] | 2013 | 52.9 | - |
| STIP [53] | 2013 | - | 82.9 |
| Action Bank [53] | 2013 | - | 83.9 |
| MST [43] | 2014 | 45.3 | 74.0 |
| AOG [25] | 2015 | 61.2 | 85.5 |
| P-CNN [6] | 2015 | 66.8 | - |
| Hierarchical [22] | 2016 | 77.5 | - |
| C3D [3] | 2016 | - | 86.0 |
| JDD [3] | 2016 | 77.7 | 87.4 |
| idt-fv [14] | 2017 | 60.9 | 92.0 |
| Pose+ idt-fv [14] | 2017 | 74.6 | 92.9 |
| Our RPAN | | **78.6** | **97.4** |

the human-part-structure information in our pose attention is robust and discriminative for action recognition.

Furthermore, we analyze classification results with confusion matrix in Fig. 4. The confusion matrix of our approach is much sparser than the one of 'Without Attention', showing that our approach recognizes more test videos correctly. Then, we compare two approaches on the 'tennis_forehand' action class, where the difference between two approaches is the largest with regard to the number of correctly-classified videos (our pose attention vs. without attention: 68 vs. 61). This difference mainly comes from confusion between 'tennis_forehand' and 'tennis_serve'. In the without-attention setting, these two classes are largely confused (11 confused test videos), while our approach takes spatial-temporal pose evolutions as a dynamical attention cue to reduce confusion between similar actions.

**Human-Part-Pooling Layer**. We investigate different strategies for our human-part-pooling layer. As shown in Table 2, the recognition performance is generally robust to different pooling strategies. Hence, in our experiments we use the concat strategy for comparison consistency.

**Choice of Basic CNNs**. We next evaluate different basic CNNs for our RPAN. Hence, we perform our RPAN, based on two widely-used CNNs in the research of action recognition, Good-Practice CNN (built on VGG16) [46] and TSN (built on BN-Inception) [47]. Both CNNs are pretrained on UCF101. For Good-Practice CNN [46], the convolutional feature cube is generated from the convolutional layer in the temporal-stream (the conv5_3 layer, $7 \times 10 \times 512$ / $8 \times 15 \times 512$ for Sub-JHMDB / PennAction). The convolutional cube from the temporal-stream TSN is the same as before. Table 3 shows that our RPAN achieves better performance with TSN. The main reason is that, TSN is a deeper CNN for action recognition, which can generate more powerful convolutional cubes than Good-Practice CNN.

Additionally, we evaluate our RPAN, based on the temporal-stream TSN so far. As TSN is a two-stream CNN architecture for action recognition in videos, we next investigate the performance of our RPAN with different streams of TSN (spatial-stream and temporal-stream). As shown in Table 4, the temporal stream outperforms the spatial stream for both TSN and RPAN, showing the fact that the motion cue is generally more important than the appearance cue for action recognition. The spatial and temporal score fusion (S:T is 1:2) can further improve accuracy due to the complementary properties between them. Finally, RPAN outperforms TSN for all cases of different streams, showing that our recurrent pose-attention is an effective dynamical mechanism for action recognition.

**Parameter Robustness**. In the previous experiments, we use a fixed Gaussian (std: $5$) to generate $\mathbf{M}_t^J$ in Eq. (13). We change this parameter to 1 / 10, and the accuracy of our approach is 78.3 / 79.3 on Sub-JHMDB (split 1). These results are comparable to the one with std=5 ('Human-Part' in Table 1), showing the parameter robustness.

## 4.3. Comparison with the State-of-the-art

We evaluate our RPAN, by comparing it with the recent state-of-the-art approaches in pose-based action recognition. In Table 5, our RPAN outperforms other recent hand-crafted and deep learning approaches on both Sub-JHMDB and PennAction datasets. This is mainly credited to the fact that, our RPAN is an end-to-end recurrent framework, where spatial-temporal evolutions of human pose are exploited as a highly-discriminative attention cue to dynamically assist action recognition in a unified fashion.

## 4.4. Byproduct: Pose Estimation in Videos

One important byproduct of our RPAN is pose estimation in videos, although our main objective is action recog-

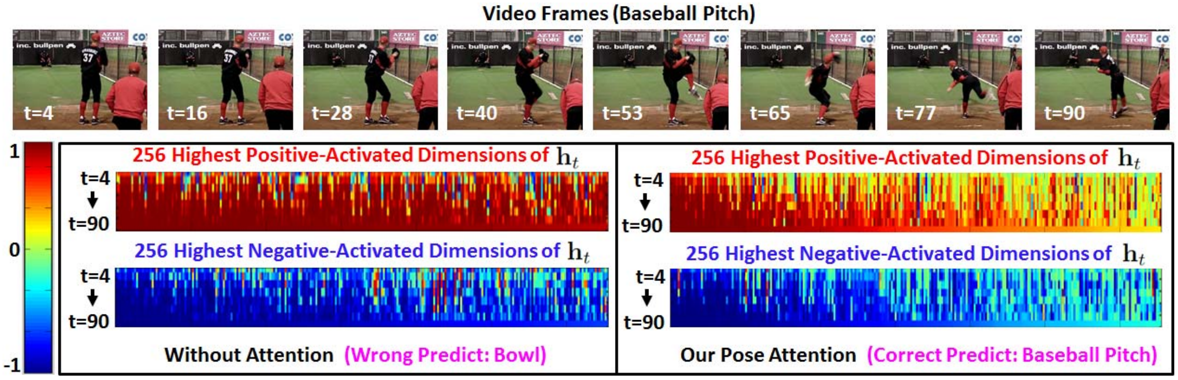

Figure 5. Feature Dynamics of Hidden State $\mathbf{h}_t$ in LSTM (Without Attention vs. Our Pose Attention). For each approach, we show two maps of feature dynamics. One refers to positive-activations of $\mathbf{h}_t$, while the other refers to negative-activations of $\mathbf{h}_t$. For each map, each row refers to 256 highest positive-activated (or negative-activated) dimensions of $\mathbf{h}_t$, where $t$ is the time index of video frames. Note that, 256 highest activated dimensions are sorted in the descending order, according to the activation values in the last video frame. One can see that, maps in our pose attention are more sparsely-activated than the ones in without-attention setting, illustrating that spatial-temporal evolutions of human pose in our approach can assist $\mathbf{h}_t$ to capture motion dynamics in videos. Additionally, as time goes on, we find that more dimensions of $\mathbf{h}_t$ are gradually activated in our approach. This demonstrates that the hidden states at the later steps can effectively integrate important motion information from the previous ones, and improve action recognition in a recurrent manner.

nition in videos. We evaluate this perspective on PennAction. For the attention heat map of each human joint, we find the location with the highest attention score as the estimated joint location. Then we follow the standard evaluation criteria for pose estimation in videos [14, 25], where the threshold is 0.2 for PennAction. The pose estimation accuracy of our RPAN is 0.68, which is comparable to the recent approaches [14, 25]. It illustrates that our RPAN can provide reliable coarse pose annotation in videos.

### 4.5. Visualization

In this section, we qualitatively evaluate our RPAN with the following visualizations. Firstly, we visualize pose-attention heat maps ($\alpha_t^J(k)$ in Section 3.2) of our RPAN in Fig. 1(d). We select an action video (Jumping-Jacks in PennAction) and show different heat maps (averaged, right ankle, right elbow, right wrist) for each sampled video frame. Fig. 1(d) shows that, our pose-attention mechanism can take spatial-temporal evolutions of human pose as a dynamical cue to effectively capture different movements of this action along time, and consequently assist to recognize this action in a recurrent fashion. Additionally, we visualize the important byproduct, i.e., pose estimation in videos, for Jumping-Jacks in Fig. 1(d). One can see that, our RPAN can provide reliable coarse pose annotation in videos.

Secondly, we further examine whether our recurrent pose attention mechanism can provide a dynamical cue to assist the learning of LSTM. We show feature dynamics of hidden state $\mathbf{h}_t$ in LSTM, since $\mathbf{h}_t$ is used to make action prediction for each time step (Eq. 10). We compare our approach to the without-attention setting of Table 1. As shown in Fig. 5, maps of feature dynamics in our pose attention are more sparsely-activated than the ones in without-attention setting, indicating that spatial-temporal pose evolutions in our approach can assist $\mathbf{h}_t$ to capture motion dynamics in videos. Hence, our approach correctly classifies this Baseball-Pitch action video, while the without-attention setting recognizes it as a wrong action (i.e., Bowl). Furthermore, as time goes on, we find that more dimensions of $\mathbf{h}_t$ are gradually activated in our approach. This shows that the hidden states at the later steps can effectively integrate important motion information from the previous ones, and consequently improve action recognition.

## 5. Conclusion

In this paper, we design a novel recurrent pose-attention network (RPAN) for action recognition, with the dynamical guidance of human joints in videos. First, our RPAN is an end-to-end recurrent framework, which takes advantage of spatial-temporal pose evolutions as a dynamical attention cue of action recognition in a unified fashion. Second, our pose-attention can adaptively learn a discriminative pose feature to enhance action prediction at every step of LSTM. Via sharing attention parameters partially on the semantically-related joints, our pose-related representations contain rich and robust human-part-structure information. Finally, an important byproduct of our RPAN is pose estimation in videos, which can be used as a reliable tool for coarse pose annotation in videos. We evaluated our RPAN on two popular benchmarks, i.e., Sub-JHMDB and PennAction. The results demonstrated that our RPAN outperforms the recent state-of-the-art approaches on both datasets.

**Acknowledgements**. This work was supported in part by National High-Tech Research and Development Program of China (2015AA042303) and National Natural Science Foundation of China (61502470, 61502152).

## References

[1] N. Ballas, L. Yao, C. Pal, and A. Courville. Delving deeper into convolutional networks for learning video representations. In *ICLR*, 2016.

[2] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*, 2009.

[3] C. Cao, Y. Zhang, C. Zhang, and H. Lu. Action recognition with joints-pooled 3d deep convolutional descriptors. In *IJCAI*, 2016.

[4] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik. Human pose estimation with iterative error feedback. In *CVPR*, 2015.

[5] G. Chéron, I. Laptev, and C. Schmid. P-cnn: Pose-based cnn features for action recognition. In *ICCV*, 2015.

[6] G. Chéron, I. Laptev, and C. Schmid. P-CNN: Pose-based CNN Features for Action Recognition. In *ICCV*, 2015.

[7] X. Chu, W. Ouyang, H. Li, and X. Wang. Structured feature learning for pose estimation. In *CVPR*, 2016.

[8] W. Ding, R. Wang, F. Mao, and G. Taylor. Theano-based large-scale visual recognition with multiple gpus. In *ICLR Workshop*, 2015.

[9] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015.

[10] Y. Du, W. Wang, and L. Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *CVPR*, 2015.

[11] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, 2016.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[13] N. Hu, G. Englebienne, Z. Lou, and B. Krose. Learning latent structure for activity recognition. In *ICRA*, 2014.

[14] U. Iqbal, M. Garbade, and J. Gall. Pose for action action for pose. In *IEEE FG*, 2017.

[15] H. Jhuang, J. Gall, S. Zuffi, and C. Schmid. Towards understanding action recognition. In *ICCV*, 2013.

[16] S. Ji, W. Xu, M. Yang, and K. Yu. 3D Convolutional Neural Networks for Human Action Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2013.

[17] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception & Psychophysics*, 1973.

[18] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*, 2012.

[20] I. Laptev. On space-time interest points. *IJCV*, 2005.

[21] Z. Li, E. Gavves, M. Jain, and C. G. M. Snoek. VideoLSTM convolves, attends and flows for action recognition. In *ArXiv*, 2016.

[22] I. Lillo, J. C. Niebles, and A. Soto. A hierarchical pose-based approach to complex action understanding using dictionaries of actionlets and motion poselets. In *CVPR*, 2016.

[23] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016.

[24] J. Y. Ng, M. J. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, 2015.

[25] B. X. Nie, C. Xiong, and S.-C. Zhu. Joint action recognition and pose estimation from video. In *CVPR*, 2015.

[26] T. Pfister, J. Charles, and A. Zisserman. Flowing convnets for human pose estimation in videos. In *ICCV*, 2015.

[27] R. Poppe. A survey on vision-based human action recognition. *Image and Vison Computing*, 2010.

[28] S. Sharma, R. Kiros, and R. Salakhutdinov. Action recognition using visual attention. In *ICLR Workshop*, 2016.

[29] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.

[30] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[31] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised Learning of Video Representations using LSTMs. *ICML*, 2015.

[32] L. Sun, K. Jia, D.-Y. Yeung, and B. E. Shi. Human action recognition using factorized spatio-temporal convolutional networks. In *ICCV*, 2015.

[33] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going Deeper with Convolutions. In *CVPR*, 2015.

[34] L. Tao and R. Vidal. Moving poselets: A discriminative and interpretable skeletal motion representation for action recognition. In *ICCV Workshops*, 2015.

[35] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016.

[36] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *NIPS*, 2014.

[37] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In *CVPR*, 2014.

[38] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.

[39] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015.

[40] C. Wang, Y. Wang, and A. L. Yuille. An approach to pose-based action recognition. In *CVPR*, 2013.

[41] H. Wang, A. Kläser, C. Schmid, and C. Liu. Action recognition by dense trajectories. In *CVPR*, 2011.

[42] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, pages 3551–3558, 2013.

[43] J. Wang, X. Nie, Y. Xia, Y. Wu, and S. C. Zhu. Cross-view action modeling, learning and recognition. In *CVPR*, 2014.

[44] L. Wang, Y. Qiao, and X. Tang. Video action detection with relational dynamic-poselets. In *ECCV*, 2014.

[45] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *CVPR*, pages 4305–4314, 2015.

[46] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao. Towards good practices for very deep two-stream convnets. *CoRR*, 2015.

[47] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016.

[48] Y. Wang and G. Mori. Learning a discriminative hidden part model for human action recognition. In *NIPS*, 2008.

[49] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *CVPR*, 2016.

[50] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015.

[51] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and L. Fei-Fei. Every moment counts: Dense detailed labeling of actions in complex videos. *IJCV*, 2017.

[52] M. Zanfir, M. Leordeanu, and C. Sminchisescu. The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. In *ICCV*, 2013.

[53] W. Zhang, M. Zhu, and K. G. Derpanis. From actemes to action: A strongly-supervised representation for detailed action understanding. In *ICCV*, 2013.

[54] W. Zhu, J. Hu, G. Sun, X. Cao, and Y. Qiao. A key volume mining deep framework for action recognition. In *CVPR*, 2016.