

PA3D: Pose-Action 3D Machine for Video Recognition

An Yan^{*1,3} Yali Wang^{*1} Zhifeng Li² Yu Qiao^{†1,4}

¹ Shenzhen Key Lab of Computer Vision and Pattern Recognition, SIAT-SenseTime Joint Lab, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China

² Tencent AI Lab ³ University of California San Diego ⁴ The Chinese University of Hong Kong

Abstract

Recent studies have witnessed the successes of using 3D CNNs for video action recognition. However, most 3D models are built upon RGB and optical flow streams, which may not fully exploit pose dynamics, i.e., an important cue of modeling human actions. To fill this gap, we propose a concise Pose-Action 3D Machine (PA3D), which can effectively encode multiple pose modalities within a unified 3D framework, and consequently learn spatio-temporal pose representations for action recognition. More specifically, we introduce a novel temporal pose convolution to aggregate spatial poses over frames. Unlike the classical temporal convolution, our operation can explicitly learn the pose motions that are discriminative to recognize human actions. Extensive experiments on three popular benchmarks (i.e., JHMDB, HMDB, and Charades) show that, PA3D outperforms the recent pose-based approaches. Furthermore, PA3D is highly complementary to the recent 3D CNNs, e.g., I3D. Multi-stream fusion achieves the state-of-the-art performance on all evaluated data sets.

1. Introduction

Video action recognition has been recently investigated, due to its wide applications in video surveillance, human-computer interaction, etc. The advances in this area are mainly driven by deep learning [2, 24, 35]. In particular, 3D CNNs have proven effective to learn spatio-temporal representations of videos [2, 29, 36]. However, most of existing approaches are mainly built upon two input types, namely RGB and optical flows. This ignores another discriminative action cue, i.e., human pose dynamics.

Alternatively, several pose-based approaches have been developed for action recognition [3, 4, 5, 18], based on the remarkable successes in human pose estimation [1, 38]. One attractive direction is pose dynamics encoding [3, 4],

which aggregates human poses of different frames as spatio-temporal representations for action recognition. However, these approaches mainly depend on two-stream features of the predefined human-pose patches [3] and/or learn pose dynamics with the predefined encoding scheme. In this case, pose representation and action recognition are isolated without adaptive interactions, which may limit the power to understand complex actions in the wild videos. More importantly, the current research of pose-based action recognition lacks a unified framework, i.e., a general semantic stream which is complementary to two-stream 3D CNNs.

To address these difficulties, we propose a novel Pose-Action 3D (PA3D) machine, which provides a seamless workflow to encode spatio-temporal pose representations for video action recognition. Specifically, PA3D consists of three semantic modules, i.e., spatial pose CNN, temporal pose convolution, and action CNN. First, spatial pose CNN can robustly extract different modalities of pose heatmaps (i.e., joints, part affinity fields, and convolutional features) from wild videos. Second, temporal pose convolution can adaptively aggregate spatial pose heatmaps over frames, which generates a spatio-temporal pose representation for each pose modality. Finally, action CNN takes the learned pose representation as input to recognize human actions.

Overall, we make three contributions in this paper. **First**, PA3D is a concise 3D CNN framework, which can achieve the learning efficiency by factorizing semantic task (pose/action), convolution operation (spatial/temporal), pose modality (joints/part affinity fields/convolutional features) within a multi-level fashion. In this case, PA3D can flexibly encode various pose dynamics as a discriminative cue to classify complex actions. **Second**, we propose a novel temporal pose convolution operation, which mainly consists of temporal association and semantic convolution to encode pose motions. Different from the traditional temporal convolution in 3D CNNs, our temporal pose convolution can learn a spatio-temporal semantic representation to explicitly describe pose motions. Moreover, our temporal dilation design allows this convolution to capture complex actions with multi-scale pose dynamics. Hence, it is more

^{*}An Yan and Yali Wang are the equally-contributed first authors (ayan@ucsd.edu, yl.wang@siat.ac.cn).

[†]Yu Qiao is the corresponding author (yu.qiao@siat.ac.cn).

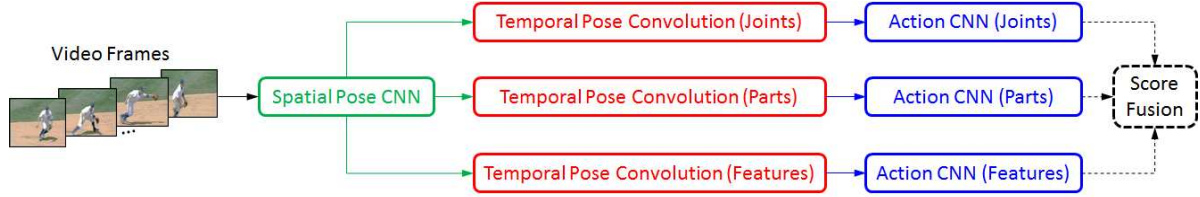


Figure 1. A Generic Framework of Pose-Action 3D Machine (PA3D). Specifically, it consists of **three semantic modules**, i.e., spatial pose CNN, temporal pose convolution, and action CNN. First, spatial pose CNN can robustly extract different modalities of pose heatmaps (i.e., joints, part affinity fields, and convolutional features) for each sampled video frame. Second, temporal pose convolution can adaptively aggregate spatial pose heatmaps over frames, which generates a spatio-temporal pose representation for each pose modality. Finally, action CNN takes the learned pose representation as input to recognize human actions. Since PA3D is built upon a concise spatio-temporal 3D framework, it can be used as another semantic stream for action recognition in videos.

suitable for action recognition in the wild videos. **Finally**, we **conduct extensive experiments** on the popular benchmarks, i.e., JHMDB, HMDB and Charades. The results show that our PA3D outperforms the recent pose encoding approaches on action recognition. Furthermore, it is highly complementary to two-stream 3D CNNs (e.g., I3D), where score fusion leads to the state-of-the-art performance on all evaluated data sets. Hence, our PA3D can be used as another semantic stream for human action recognition.

2. Related Work

Action Recognition. Over the past years, deep learning approaches have significantly boosted the performance of video action recognition [2, 7, 24, 32, 33, 35, 36]. One well-known framework is two-stream CNNs [24], which process RGB and optical flows as two separate streams. Built upon this, a number of variations have been introduced by deep local descriptors [31, 33], two-stream fusion [7, 8], key volume attention and mining [34, 42], temporal segment networks [35], etc. However, 2D CNNs are limited to learn spatio-temporal representations of complex actions. To address this difficulty, 3D CNNs have been highlighted by model inflation [2], spatio-temporal relations [32, 36, 37], factorization [21, 29, 39], etc. However, 3D CNNs often require the large-scale benchmarks (e.g., Sports1M [12] and Kinetics [2, 13]) with costly computation burden. More importantly, these models use RGB and/or optical flows as input, and thus they ignore the pose dynamics which can be discriminative to recognize human actions. To bridge this gap, we propose Pose-Action 3D Machine (PA3D), i.e., a novel 3D CNN for pose-based action recognition.

Pose-based Action Recognition. Human pose provides an important cue to classify complex actions [10, 43]. With the remarkable successes of deep learning in pose estimation [1, 17, 20, 26, 27, 38], there is a growing interest in pose-based action recognition. However, it is often challenging to achieve an effective design, since those pose estimators are not explicitly developed for action recognition in videos. Several attempts have been recently pro-

posed by skeleton representation [6, 40], multi-task learning [18], recurrent pose attention [5], pose dynamics encoding [3, 4, 16], etc. In particular, pose dynamics encoding is an attractive direction by learning spatio-temporal pose representations for action recognition. But these approaches mainly depend on two-stream features of the predefined human-pose patches [3] and/or use the predefined pose encoding scheme [4], which may reduce their capacity of recognizing complex actions in the wild. Furthermore, the current research lacks a unified framework for pose-based action recognition. Motivated by these, we propose a novel factorized 3D CNN (i.e., PA3D), which can effectively learn pose dynamics to classify human actions.

3. Pose-Action 3D Machines (PA3D)

To obtain an effective spatio-temporal pose representation for video action recognition, we introduce Pose-Action 3D Machine (PA3D) in the section. It mainly consists of three semantic modules, i.e., spatial pose CNN, temporal pose convolution, and action CNN. **First**, we use spatial pose CNN to generate human pose features for each video frame. By taking advantage of the state-of-the-art pose estimator (e.g., [1]), our spatial pose heatmaps are robust to occlusion and multi-person cases in the wild. **Second**, we propose a novel temporal pose convolution, which can aggregate spatial poses of different frames semantically into a spatio-temporal pose representation. **Finally**, we feed the resulting pose representations into action CNN, and fuse the prediction scores of different pose modalities to boost action recognition. The generic framework is shown in Fig. 1.

3.1. Spatial Pose CNN

To leverage human pose as an explicit cue of actions in videos, we first use spatial pose CNN to generate pose heatmaps of actors in each frame. Specifically, we choose the widely-used multi-person pose machines [1] as our spatial pose CNN, since it is robust to the cases of multiple people and complex occlusions in the wild. Furthermore, **we feed each video frame into this spatial pose CNN, and**

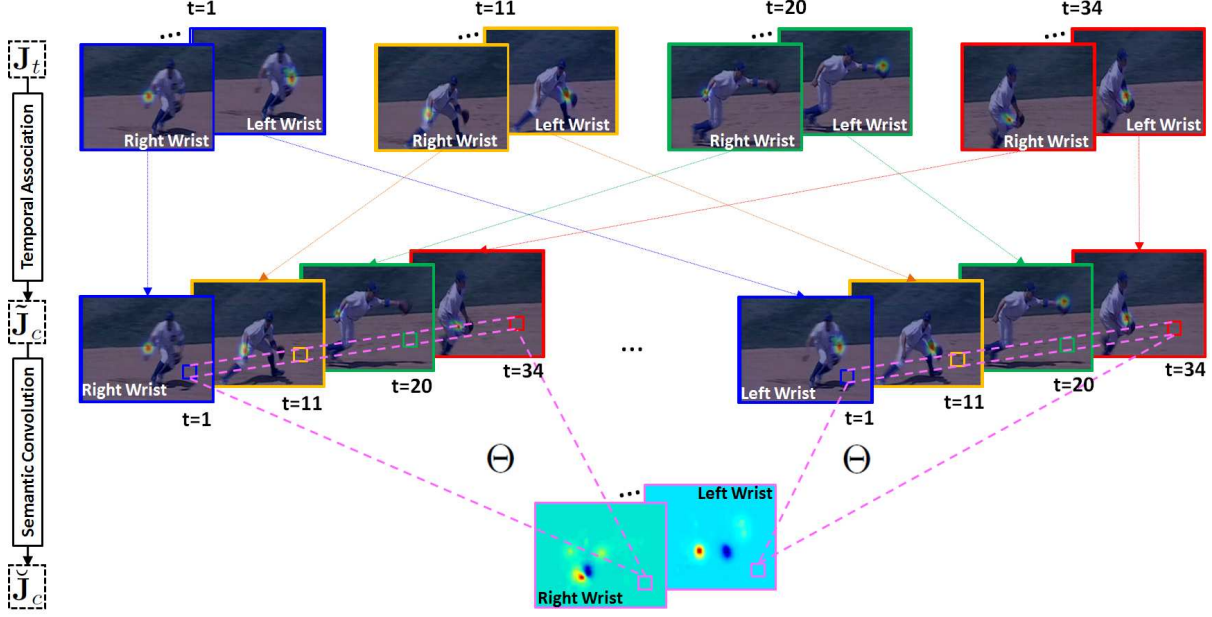


Figure 2. Temporal Pose Convolution. Without loss of generality, we use the joint heatmaps as an illustration. The part affinity fields and convolutional features can be processed in the same manner. Specifically, **temporal pose convolution** consists of two atomic operations. (1) **Temporal association** is used to generate a temporally-ordered cube $\tilde{\mathbf{J}}_c$ for each joint. We achieve it by stacking the heatmaps of all the frames (per joint). (2) **Semantic convolution** is used to generate a spatio-temporal pose representation $\hat{\mathbf{J}}_c$ for each joint. We achieve it by performing 1×1 convolution on $\tilde{\mathbf{J}}_c$ (per joint). To alleviate overfitting, we share the convolutional filter Θ among all the joints.

extract three pose modalities, i.e., joints, part affinity fields, and convolutional features. The modality of **joints** refers to the prediction confidence maps of human joints. The modality of **part affinity fields** refers to the prediction confidence maps which preserve both location and orientation information across the support region of body limb [1]. The modality of **convolutional features** refers to the feature maps from a convolution layer of CNN backbone in [1], e.g., the 10-th layer of VGG19.

Without loss of generality, we use the joint heatmaps as an illustration. The part affinity fields and convolutional features can be processed in the same manner. Specifically, we denote the joint heatmaps of the t -th video frame as $\mathbf{J}_t \in \mathbb{R}^{C \times H \times W}$ ($t = 1, \dots, T$). It consists of C heatmaps with size of $H \times W$, where C is the number of human joints.

3.2. Temporal Pose Convolution

After obtaining spatial pose heatmaps for each frame (e.g., \mathbf{J}_t), we propose a novel temporal pose convolution to encode pose dynamics over frames. As shown in Fig. 2, it mainly consists of two atomic operations, i.e., temporal association and semantic convolution.

Temporal Association. For each joint, we first stack the heatmaps of all the frames along their temporal order. This operation can generate a temporally-associated cube for the c -th joint, i.e., $\tilde{\mathbf{J}}_c \in \mathbb{R}^{T \times H \times W}$, where the t -th channel of $\tilde{\mathbf{J}}_c$ refers to the heatmap of the c -th joint at the t -th temporal

frame, $t = 1, \dots, T$ and $c = 1, \dots, C$.

Semantic Convolution. After obtaining the temporally-associated cube $\tilde{\mathbf{J}}_c$ of the c -th joint, we encode it into a spatio-temporal pose representation over frames. As mentioned before, the channels of $\tilde{\mathbf{J}}_c$ correspond to the temporally-ordered heatmaps of the c -th joint. In this case, we directly perform 1×1 convolution on $\tilde{\mathbf{J}}_c$ to generate the spatio-temporal pose representation $\hat{\mathbf{J}}_c \in \mathbb{R}^{N \times H \times W}$,

$$\hat{\mathbf{J}}_c = \tilde{\mathbf{J}}_c * \Theta. \quad (1)$$

Note that, each of N output channels in $\hat{\mathbf{J}}_c$ is not just an abstract feature map. It semantically encodes the movement of the c -th joint over frames, as shown in Fig. 2. For this reason, we denote the 1×1 convolution as semantic convolution. Moreover, $\Theta \in \mathbb{R}^{N \times T \times 1 \times 1}$ is the convolutional filter. We share it among joints to alleviate overfitting.

Multi-Scale Design via Temporal Dilation. For each joint, semantic convolution is performed over all the frames. As a result, the spatio-temporal representation $\hat{\mathbf{J}}_c$ may lack the ability to describe various scales of pose motions. To address this problem, we introduce temporal dilation convolution,

$$\hat{\mathbf{J}}_c = \tilde{\mathbf{J}}_c * \Phi, \quad (2)$$

where $\Phi \in \mathbb{R}^{M \times (T/d) \times 1 \times 1}$ is the dilated convolutional filter, d is the dilated factor, and M is the number of output heatmaps. As shown in Fig. 3, temporal dilation allows us to perform semantic convolution on those channels of $\tilde{\mathbf{J}}_c$

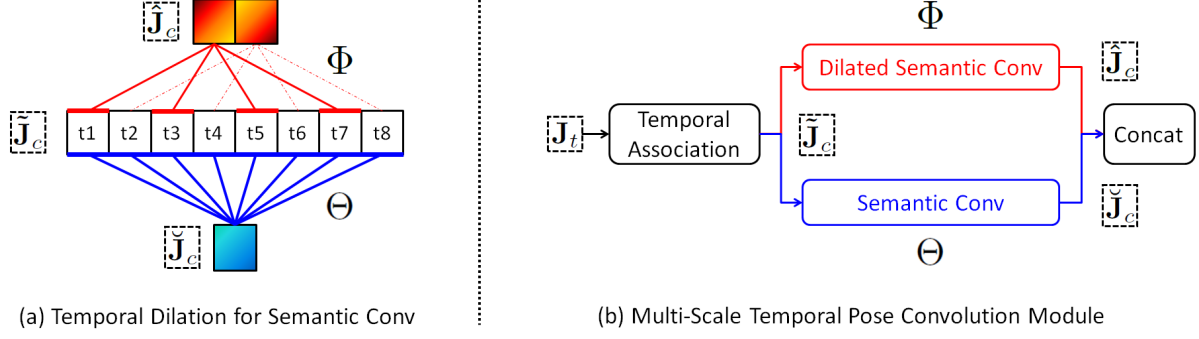


Figure 3. Multi-Scale Temporal Pose Convolution Module. Temporal dilation allows us to perform semantic convolution on those channels with the interval of d time steps (e.g., $d = 2$). Hence, it can learn pose dynamics in a different temporal scale.

with the interval of d time steps (e.g., $d = 2$). Hence, it can learn pose dynamics at different temporal scales. Additionally, temporal dilation is preferable than local convolution, e.g., performing semantic convolution on adjacent 3 frames with stride 1. The main reason is that, temporal dilation can extend temporal receptive fields with different scales, which avoids modeling the noisy pose dynamics in the locally-adjacent frames. Finally, we concatenate $\tilde{\mathbf{J}}_c$ and $\hat{\mathbf{J}}_c$ as a multi-scale spatio-temporal pose representation, and feed it into action CNN for recognition.

Why to Use Temporal Pose Convolution? We mainly explain why our temporal pose convolution is more suitable to learn spatio-temporal pose representation, compared to other temporal approaches [4, 29, 39]. **(1) Temporal Pose Convolution vs. Traditional Temporal Convolution.** First, the traditional temporal convolution [29, 39] can be directly implemented on the joint heatmaps over T frames. For example, in order to produce N output feature maps, temporal convolution has to be formulated as $\Psi \in \mathbb{R}^{N \times C \times T \times 1 \times 1}$, i.e., $T \times 1 \times 1$ with C input channels and N output channels. Apparently, it requires much more parameters than our temporal pose convolution $\Theta \in \mathbb{R}^{N \times T \times 1 \times 1}$. Hence, the traditional temporal convolution often increases the overfitting risk. Second, in the traditional temporal convolution, the output feature map is often a highly-abstract spatio-temporal feature, which may lack the semantic representation of discriminative pose motions. Alternatively, each output of our temporal pose convolution delivers rich semantic descriptions, e.g. the dynamics of a joint over frames. Hence, our convolution is more effective to encode spatio-temporal pose representation. **(2) Temporal Pose Convolution vs. Pose Colorization of PoTion.** First, pose colorization [4] encodes the joint heatmaps of each video frame, according to the relative time of this frame in the clip. Due to the fact that such encoding scheme is predefined as a linear function of time steps, it is often limited to learn the complex pose motions. Alternatively, our temporal pose convolution is trained jointly with action CNN, allowing to capture nonlinear pose dynamics adaptively. Sec-

Layer	Output Size	Action CNN
input	$R \times H \times W$	-
conv1.1	$128 \times H/2 \times W/2$	3×3 , stride 2
conv1.2	$128 \times H/2 \times W/2$	3×3 , stride 1
conv2.1	$256 \times H/4 \times W/4$	3×3 , stride 2
conv2.2	$256 \times H/4 \times W/4$	3×3 , stride 1
conv3.1	$512 \times H/8 \times W/8$	3×3 , stride 2
conv3.2	$512 \times H/8 \times W/8$	3×3 , stride 1
FC-512	$512 \times 1 \times 1$	average pool, dropout
FC-K	$K \times 1 \times 1$	softmax

Table 1. Action CNN. R is the number of feature channels in the spatio-temporal pose representation. Note that, these features are the middle-level semantic representations, which are more sparse than the original images. Hence, we follow the suggestion in [4], and use a light-weight model to recognize K action classes.

ond, PoTion lacks the multi-scale description of pose motions, while our dilated operation can encode various pose dynamics in a unified framework. Third, PoTion only uses joints for pose encoding, while our temporal pose convolution also works for other important pose modalities (e.g., part affinity fields and convolution features). It can further boost our approach. **(3) Temporal Pose Convolution vs. Temporal Segment Network.** Temporal segment networks [35] provide a temporal encoding manner for action recognition, i.e., it averages the scores of sampled frames as video prediction for training. Apparently, one can use it to encode pose heatmaps of different frames, e.g., we feed pose heatmaps of each sampled frame independently into action CNN, and average the scores of sampled frames as video prediction for training. However, this average style may ignore the important pose movement, which can be seen as a discriminative cue for action recognition. Alternatively, our temporal pose convolution can effectively encode semantic pose motions to train action CNN.

3.3. Action CNN

After obtaining spatio-temporal pose representation, we feed it into an action CNN for action recognition in videos.

Temporal Pose Modeling		JHMDB	HMDB
TSN Style	Joints	54.5	42.3
	Parts	58.5	44.0
	Features	38.0	35.5
3DConv	Joints	55.6	45.1
	Parts	54.5	44.8
	Features	40.5	40.0
TempConv	Joints	58.5	45.8
	Parts	51.4	44.7
	Features	38.4	39.3
PoTion	Joints	51.2	43.4
	Parts	50.3	42.6
	Features	38.0	37.8
Our TempPoseConv	Joints	59.3	46.7
	Parts	58.6	47.1
	Features	40.5	40.3

Table 2. Classification accuracy by temporal modeling approaches (JHMDB and HMDB, split one). Specifically, we first use spatial pose CNN to generate the pose features, and then use different temporal modeling approaches to learn spatio-temporal pose representation for action recognition.

It is worth mentioning that, these pose representations are the middle-level features, which are more sparse than the original images. Hence, we follow the suggestion in [4], and use a light-weight action CNN, i.e., six convolutional layers and one fully-connected layer in Table. 1. Furthermore, we train temporal pose convolution and action CNN jointly. It can enhance our PA3D by adaptive interactions between learning pose dynamics and classifying actions. Finally, we fuse the prediction scores of different pose modalities to boost action recognition in the test.

3.4. Further Discussions about PA3D

As shown in Fig. 1, our PA3D can be treated as a **novel spatio-temporal 3D CNN for pose-based action recognition**. To effectively exploit human pose information in the wild videos, we factorize 3D CNN into different semantic levels. **First**, we decouple the target task as spatio-temporal pose encoding and action recognition. In this case, we can explicitly leverage pose dynamics as a discriminative cue to classify human actions. **Second**, we decompose spatio-temporal pose learning via separate spatial and temporal pose convolutions. As illustrated in Fig. 2, this can effectively encode pose motions for each joint. **Finally**, we represent human pose via three modalities, i.e., joints, part affinity fields, and convolutional features. Score fusion can boost pose-based action recognition. Furthermore, our PA3D is built upon pose dynamics. Hence, it can be used as another semantic stream, which is complementary to the popular two streams (i.e., RGB and optical flows).

Types of TempPoseConv	JHMDB	HMDB
Global	59.3	46.7
Local	57.5	44.9
Dilated	58.4	44.8
Global+Local	59.7	44.8
Global+Dilated	60.1	47.8
Global+Dilated+Local	58.5	47.1

Table 3. Types of TempPoseConv (Joints). Specifically, we perform semantic conv on all 8 frames (Global Type) or frames with interval of $d=2$ steps (Dilation Type), as shown in Fig. 3. For comparison, we also design a Local Type of TempPoseConv, e.g., we perform semantic conv on adjacent 3 frames with stride 1 (i.e., t1-t3, t2-t4, t3-t5, t4-t6, t5-t7, t6-t8). + denotes that, we integrate various types of TempPoseConv as a multi-scale module such as Fig. 3. Compared to the Local type, dilation is preferable to model multi-scale pose dynamics with larger temporal receptive fields.

Frames	$T=2$	$T=4$	$T=6$	$T=8$	$T=12$
JHMDB	57.8	56.3	55.3	55.6	50.4
HMDB	44.6	46.1	45.5	46.8	43.2
Outputs	$N=2$	$N=4$	$N=6$	$N=8$	$N=12$
JHMDB	58.2	59.6	56.3	60.1	52.6
HMDB	43.7	46.3	46.1	46.0	45.7

Table 4. Parameters in TempPoseConv (Joints). For $\Theta \in \mathbb{R}^{N \times T \times 1 \times 1}$ in Eq. (1), we evaluate the number of video frames T , and the number of output channels N for each joint. When we change T (or N), we fix $N=6$ (or $T=4$). The results are comparable, showing that TempPoseConv is robust to different parameter choices. More details can be found in Section 4.1.

4. Experiments

Data Sets. Since our goal is pose-based action recognition in videos, we evaluate our PA3D on three popular benchmarks which focus on complex human activities in the wild. Specifically, **JHMDB** [11]/**HMDB** [15] consists of 21/51 action categories with 928/6766 video clips, respectively. They are collected from movies to youtube, which involves daily activities. **Charades** [23] is a recent large-scale video dataset, consisting of 9,848 annotated videos with an average length of 30 seconds. Note that, we choose Charades instead of Kinetics [13] with the following reasons. On one hand, as discussed in [4], it is not suitable to evaluate pose-based action recognition on Kinetics, since humans are poorly visible in many videos of this data. On the other hand, Charades contains activities of 267 different people, and over 15% of this dataset belongs to multi-person scenes. Furthermore, it contains 66,500 activity annotations for 157 action classes. Each video is severely untrimmed and has multiple action labels in the overlapped temporal durations. All these facts make Charades reasonable and challenging for pose-based action recognition.

Implementation Details. Unless stated otherwise, we

Pose Modalities	JHMDB	HMDB
Joint (J)	60.1	47.8
Joint-difference (Jdiff)	52.6	46.4
Part (P)	61.9	48.0
Part-difference (Pdiff)	50.0	42.2
Feature (F)	41.0	40.9
Feature-difference (Fdiff)	35.0	36.2
Fusion Strategies	JHMDB	HMDB
$J \oplus P$	58.7	47.5
$J \cup P$	65.6	50.7
$J \oplus Jdiff$	56.2	45.8
$J \cup Jdiff$	61.2	50.3
$P \cup Pdiff$	64.9	49.1
$F \cup Fdiff$	47.7	45.1
$J \cup P \cup F$	67.5	54.1
$J \cup P \cup F \cup Jdiff \cup Pdiff \cup Fdiff$	69.5	54.7

Table 5. Pose modality fusion. \oplus : We concatenate spatio-temporal representations of different pose modalities, and feed it into an action CNN for action recognition. \cup : We feed spatio-temporal representation of each pose modality into an individual action CNN, and fuse the prediction scores at the test phrase.

perform our PA3D as follows. **First**, we use the official 6-stage multi-person pose CNN [1] to extract spatial pose heatmaps for each sampled frame, i.e., 19 joint heatmaps: joint branch in the last stage, 38 part heatmaps: part affinity field branch in the last stage, 128 feature maps: the 10-th layer of VGG19 which is the backbone of this pose CNN. More specifically, we resize each video frame with a scale of 0.5, 1.0, 1.5 and 2.0, and average their outputs to produce the final pose heatmaps for each frame. **Second**, we set the training batch size as 32/64/256 for JHMDB/HMDB/Charades, under the implementation on PyTorch. We use the standard SGD for training JHMDB/HMDB and the adam optimizer [14] for training Charades. The initial learning rate is set to 0.01, and the training procedure is finished with 150/400/60 epoches for JHMDB/HMDB/Charades. **Third**, each video has a single label in JHMDB and HMDB. Hence, we use cross entropy for training and report the test classification accuracy. Alternatively, each video contains multiple labels in Charades. Hence, we use multi-label loss [23] for training and report the test mean average precision (mAP).

4.1. Ablation Studies

To investigate the properties of our PA3D, we mainly evaluate its key model components on JHMDB and HMDB. For fairness, when we explore different strategies of one component, all other components are set as the basic strategy in the implementation details above.

Does temporal pose convolution help? **First**, we compare temporal pose convolution (TempPoseConv) with

a number of recent temporal modeling approaches, e.g., temporal segment network (TSN) [35], 3D convolution (3DConv) [28], temporal convolution (TempConv) in factorized 3D CNN [29], pose colorization in PoTion [4]. The testing accuracy results of JHMDB and HMDB (split1) are shown in Table 2, where our TempPoseConv outperforms other temporal modeling approaches, w.r.t., all pose modalities. It demonstrates that our TempPoseConv can encode the discriminative pose dynamics for action recognition. **Second**, we evaluate whether temporal dilation is effective to model multi-scale pose dynamics. Hence, we perform the Global/Local/Dilation types of TempPoseConv. Specifically, we perform semantic conv on all 8 frames (Global Type) or frames with interval of $d=2$ steps (Dilation Type), as shown in Fig. 3. For comparison, we also design a Local Type of TempPoseConv, e.g., we perform semantic conv on adjacent 3 frames with stride 1 (i.e., $t1-t3$, $t2-t4$, $t3-t5$, $t4-t6$, $t5-t7$, $t6-t8$). Furthermore, we concatenate different TempPoseConv types as multi-scale module. The results of the joint modality are shown in Table 3, where we keep the same number of output channels to be fair, e.g., $N = 6$ for each joint. One can see that, the Local type does not work well by itself or concatenation, since the locally-adjacent pose dynamics may be noisy. Alternatively, temporal dilation is preferable because it can extend temporal receptive fields with different scales. **Finally**, we evaluate $\Theta \in \mathbb{R}^{N \times T \times 1 \times 1}$ in TempPoseConv. The results for the joint modality are shown in Table 4. When testing, we make prediction over 10 sampled clips of each video, where each clip has T sampled frames. Since videos in JHMDB are truncated into a very short duration, a small T in each clip can be sufficient to capture important pose cues. When T increases in each clip, 10 sampled clips are gradually overlapped for a video in JHMDB. In this case, the complementary properties between different clips are reduced when fusion. Hence, the performance slightly decreases. Additionally, N is the number of output channels in TempPoseConv. When N is small, it may be insufficient to model the discriminative pose dynamics to recognize complex actions. When N is large, the spatio-temporal pose representations may be redundant, which increases the training difficulty of action CNN. Hence, a moderate N is preferable. We choose $T = 4/8$ and $N = 8/6$ for JHMDB/HMDB.

How to fuse different pose modalities? To investigate the best results with various pose modalities, we perform multi-scale TempPoseConv for all the cases in the following. **First**, for each pose modality, we compute the difference between two consecutive frames. For example, Jdiff is $J_t - J_{t-1}$, where J_t is the joint heatmap at t and $t = 2, \dots, T$. As before, we encode this feature as a spatio-temporal representation over all the frames. In Table 5, the pose difference is also effective for action recognition. **Second**, we investigate different fusion strategies, i.e., \oplus de-

Approaches	JHMDB
P-CNN [3]	61.1
Action Tubes [9]	62.5
MR Two-Stream R-CNN [19]	71.1
Chained MultiStream [43]	76.1
PoTion [4]	57.0
I3D [4]	84.1
PoTion+I3D [4]	85.5
our PA3D	69.5
RPAN [†] [5]	83.9
our PA3D + RPAN [†]	86.1

Table 6. State-of-the-art on JHMDB (Acc). [†] denotes our reproduced results. More details can be found in Section 4.2.

notes that we concatenate spatio-temporal representations of different pose modalities together, and feed them into an action CNN. \cup denotes that we feed spatio-temporal representation of each pose modality into an individual action CNN, and fuse their prediction scores at the test phrase. In Table 5, $J \cup P$ outperforms $J \oplus P$, showing that it is more effective to use score fusion between pose modalities. Additionally, $J \cup J_{diff}$ outperforms $J \oplus J_{diff}$. It illustrates that, score fusion is also suitable to integrate a pose modality and its difference. Hence, we use score fusion \cup in the rest. **Finally**, we fuse various pose modalities in different combinations. Fusing all achieves the best result, which shows the complementary characteristics of different pose modalities.

Is it necessary to pretrain action CNN? We evaluate action CNN on HMDB, with two settings (i.e., non-pretraining vs. pretraining on the large-scale Charades). Although the pretrained action CNN tends to converge 1.5x faster than the non-pretrained one, the test accuracy of the pretrained one would be slightly lower (around 1-2%) than that of the non-pretrained one (joint modality: 47.8%). Hence, it may not be necessary to pretrain action CNN, as suggested in [4]. The reason is that the input to action CNN is the sparse pose motion features, which are already the middle-level representations for action recognition. Therefore, we train action CNN from scratch in our experiment.

4.2. Comparison with State-of-the-art

We compare our PA3D with a number of state-of-the-art approaches in Table 6, Table 7 and 8. First, our PA3D significantly outperforms the recent pose encoding approach, e.g., P-CNN [3], PoTion [4]. This indicates that our PA3D can learn the discriminative pose dynamics for action recognition. Second, our PA3D is strongly complementary to other 3D CNNs (e.g., I3D [2] or NL I3D [36]) and two-stream CNN (e.g., RPAN [5]). This shows that our PA3D is an effective semantic stream for human action recognition. Via score fusion, we achieve the state-of-the-art performance on JHMDB, HMDB and Charades.

Local Descriptors	HMDB
IDT [30]	61.7
TDD [33]	63.2
TDD + IDT [33]	65.9
2D Convolution Networks	HMDB
2Stream [24]	59.4
ST-Resnet [7]	66.4
TSN [35]	69.4
Chained MultiStream [43]	69.7
SVMP [31]	72.6
OFF [25]	74.2
3D Convolution Networks	HMDB
C3D [28]	51.6
ARTNet [32]	70.9
S3D [39]	75.9
R(2+1)D [29]	78.7
I3D [2]	80.7
PoTion [4]	43.7
PoTion + I3D [4]	80.9
our PA3D	55.3
our PA3D + I3D	82.1

Table 7. State-of-the-art on HMDB (Acc).

Approaches	Charades
C3D [23]	10.9
2Stream [23]	14.2
Asyn-TF [22]	22.4
Multiscale TRN [41]	25.2
SVMP [31]	26.7
I3D [36]	35.5
GCN [37]	36.2
NL I3D [36]	37.5
(GCN + I3D + NL I3D) [†]	40.7
PoTion [†] [4]	10.3
PoTion [†] + (GCN + I3D + NL I3D) [†]	40.8
our PA3D	13.8
our PA3D + (GCN + I3D + NL I3D) [†]	41.0

Table 8. State-of-the-art on Charades (mAP). [†] denotes our reproduced results. More details can be found in Section 4.2.

4.3. Visualization

In Fig. 4, we visualize PA3D by a *Sword* video of HMDB ($T = 8$ frames). **First**, we use spatial pose CNN to generate the pose heatmaps, e.g., 38 part affinity fields (PAF) heatmaps. For visualization, we perform maxpooling over all the PAF heatmaps, which can describe human poses for each frame. **Second**, we perform temporal pose convolution over frames. For each PAF, we obtain $N = 6$ motion maps, i.e., spatio-temporal pose representation. For visualization, we perform maxpooling over PAFs, which can produce $N = 6$ motion maps of human poses over frames. As

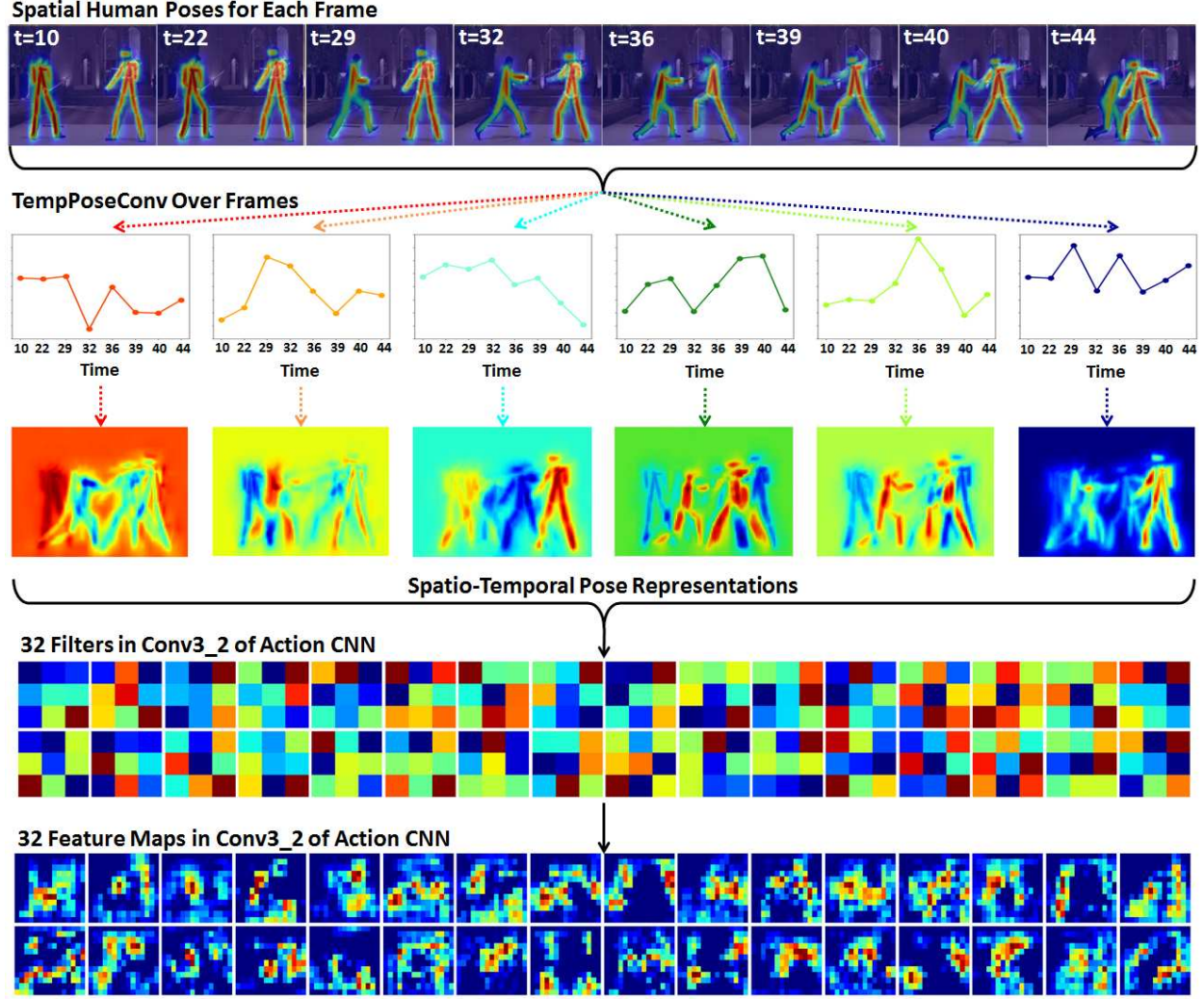


Figure 4. Visualization of our PA3D. One can see that, TempPoseConv can learn various temporal movements and thus represent the diversified human pose motions. Moreover, the features in conv3_2 clearly show that, action CNN can highlight the important pose motions of different actors, and integrate them together as a discriminative representation for action recognition.

shown in Fig. 4, TempPoseConv exhibits various temporal movements, and thus captures the diversified human pose motions. **Finally**, we concatenate all the PAF-motion maps as input to action CNN. We visualize the conv3_2 layer by using 32 convolution filters and the corresponding features. Clearly, action CNN can highlight the important motions of different PAFs, and combine them together as a discriminative representation for action recognition.

5. Conclusion

In this paper, we propose a novel Pose-Action 3D (PA3D) Machine for action recognition. First, it is a concise 3D CNN with multi-level semantic factorization. Second, we introduce a flexible temporal pose convolution, which can explicitly encode spatio-temporal pose representations for action recognition. Finally, we perform extensive ex-

periments on JHMDB, HMDB and Charades, where our PA3D significantly outperforms the recent pose encoding methods. Furthermore, it achieves the state-of-the-art performance via fusion with two-stream 3D CNNs, showing its effectiveness as another semantic stream in general.

Acknowledgements. This work is partially supported by National Natural Science Foundation of China (61876176, U1613211, U1713208), Tencent AI Lab Rhino-Bird Joint Research Program (No. JR201807), Shenzhen Research Program (JCYJ20150925163005055, CXB201104220032A), the Joint Lab of CAS-HK.

References

- [1] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 2017.

- [2] Joao Carreira and Andrew Zisserman. Action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.
- [3] Guilhem Chéron, Ivan Laptev, and Cordelia Schmid. P-cnn: Pose-based cnn features for action recognition. In *ICCV*, 2015.
- [4] Vasileios Choutas, Philippe Weinzaepfel, Jérôme Revaud, and Cordelia Schmid. Potion: Pose motion representation for action recognition. In *CVPR*, 2018.
- [5] Wenbin Du, Yali Wang, and Yu Qiao. Rpan: An end-to-end recurrent pose-attention network for action recognition in videos. In *ICCV*, 2017.
- [6] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *CVPR*, 2015.
- [7] Christoph Feichtenhofer, Axel Pinz, and Richard P. Wildes. Spatiotemporal residual networks for video action recognition. In *NIPS*, 2016.
- [8] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, 2016.
- [9] Georgia Gkioxari and Jitendra Malik. Finding action tubes. In *CVPR*, 2015.
- [10] Umar Iqbal, Martin Garbade, and Juergen Gall. Pose for action action for pose. In *IEEE FG*, 2017.
- [11] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J Black. Towards understanding action recognition. In *ICCV*, 2013.
- [12] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.
- [13] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. In *arXiv:1705.06950*, 2017.
- [14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [15] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *ICCV*, 2011.
- [16] Mengyuan Liu and Junsong Yuan. Recognizing human actions as the evolution of pose estimation maps. In *CVPR*, 2018.
- [17] Yue Luo, Jimmy Ren, Zhouxia Wang, Wenxiu Sun, Jinshan Pan, Jianbo Liu, Jiahao Pang, and Liang Lin. Lstm pose machines. In *CVPR*, 2018.
- [18] Diogo C. Luvizon, David Picard, and Hedi Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. In *CVPR*, 2018.
- [19] Xiaojiang Peng and Cordelia Schmid. Multi-region Two-Stream R-CNN for Action Detection. In *ECCV*, 2016.
- [20] Tomas Pfister, James Charles, and Andrew Zisserman. Flowing convnets for human pose estimation in videos. In *ICCV*, 2015.
- [21] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*, 2017.
- [22] Gunnar A. Sigurdsson, Santosh Divvala, Ali Farhadi, and Abhinav Gupta. Asynchronous temporal fields for action recognition. In *CVPR*, 2017.
- [23] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016.
- [24] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.
- [25] Shuyang Sun, Zhanghui Kuang, Wanli Ouyang, Lu Sheng, and Wei Zhang. Optical flow guided feature: A fast and robust motion representation for video action recognition. In *CVPR*, 2018.
- [26] Jonathan J. Tompson, Arjun Jain, Yann LeCun, and Christoph Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *NIPS*, 2014.
- [27] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *CVPR*, 2014.
- [28] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.
- [29] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018.
- [30] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.
- [31] Jue Wang, Anoop Cherian, Fatih Porikli, and Stephen Gould. Video representation learning using discriminative pooling. In *CVPR*, 2018.
- [32] Limin Wang, Wei Li, Wen Li, and Luc Van Gool. Appearance-and-relation networks for video classification. In *CVPR*, 2018.
- [33] Limin Wang, Yu Qiao, and Xiaoou Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *CVPR*, 2015.
- [34] Limin Wang, Yuanjun Xiong, Dahua Lin, and Luc Van Gool. Untrimmednets for weakly supervised action recognition and detection. In *CVPR*, 2017.
- [35] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016.
- [36] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.
- [37] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. *arXiv preprint arXiv:1806.01810*, 2018.
- [38] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, 2016.
- [39] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning for video understanding. *arXiv:1712.04851*, 2017.

- [40] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 2018.
- [41] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *ECCV*, 2018.
- [42] Wangjiang Zhu, Jie Hu, Gang Sun, Xudong Cao, and Yu Qiao. A key volume mining deep framework for action recognition. In *CVPR*, 2016.
- [43] Mohammadreza Zolfaghari, Gabriel L. Oliveira, Nima Sedaghat, and Thomas Brox. Chained multi-stream networks exploiting pose, motion, and appearance for action classification and detection. In *ICCV*, 2017.