P-CNN: Pose-based CNN Features for Action Recognition

Guilhem Chéron, Ivan Laptev, Cordelia Schmid

P-CNN: Pose-based CNN Features for Action Recognition

Guilhem Chéron*† Ivan Laptev* Cordelia Schmid†

INRIA

Abstract

This work targets human action recognition in video. While recent methods typically represent actions by statistics of local video features, here we argue for the importance of a representation derived from human pose. To this end we propose a new Pose-based Convolutional Neural Network descriptor (P-CNN) for action recognition. The descriptor aggregates motion and appearance information along tracks of human body parts. We investigate different schemes of temporal aggregation and experiment with P-CNN features obtained both for automatically estimated and manually annotated human poses. We evaluate our method on the recent and challenging JHMDB and MPII Cooking datasets. For both datasets our method shows consistent improvement over the state of the art.

1. Introduction

Recognition of human actions is an important step toward fully automatic understanding of dynamic scenes. Despite significant progress in recent years, action recognition remains a difficult challenge. Common problems stem from the strong variations of people and scenes in motion and appearance. Other factors include subtle differences of fine-grained actions, for example when manipulating small objects or assessing the quality of sports actions.

The majority of recent methods recognize actions based on statistical representations of local motion descriptors [22, 33, 41]. These approaches are very successful in recognizing coarse action (standing up, hand-shaking, dancing) in challenging scenes with camera motions, occlusions, multiple people, etc. Global approaches, however, are lacking structure and may not be optimal to recognize subtle variations, e.g. to distinguish correct and incorrect golf swings or to recognize fine-grained cooking actions illustrated in Figure 5.

*WILLOW project-team, Département d'Informatique de l'École Normale Supérieure, ENS/Inria/CNRS UMR 8548, Paris, France.

†LEAR project-team, Inria Grenoble Rhône-Alpes, Laboratoire Jean Kuntzmann, CNRS, Univ. Grenoble Alpes, France.

Fine-grained recognition in static images highlights the importance of spatial structure and spatial alignment as a pre-processing step. Examples include alignment of faces for face recognition [3] as well as alignment of body parts for recognizing species of birds [14]. In analogy to this prior work, we believe action recognition will benefit from the spatial and temporal detection and alignment of human poses in videos. In fine-grained action recognition, this will, for example, allow to better differentiate *wash hands* from *wash object* actions.

In this work we design a new action descriptor based on human poses. Provided with tracks of body joints over time, our descriptor combines motion and appearance features for body parts. Given the recent success of Convolutional Neural Networks (CNN) [20, 23], we explore CNN features obtained separately for each body part in each frame. We use appearance and motion-based CNN features computed for each track of body parts, and investigate different schemes of temporal aggregation. The extraction of proposed *Pose-based Convolutional Neural Network* (P-CNN) features is illustrated in Figure 1.

Pose estimation in natural images is still a difficult task [7, 37, 42]. In this paper we investigate P-CNN features both for automatically estimated as well as manually annotated human poses. We report experimental results for two challenging datasets: JHMDB [19], a subset of HMDB [21] for which manual annotation of human pose have been provided by [19], as well as MPII Cooking Activities [29], composed of a set of fine-grained cooking actions. Evaluation of our method on both datasets consistently outperforms the human pose-based descriptor HLPF [19]. Combination of our method with Dense trajectory features [41] improves the state of the art for both datasets.

The rest of the paper is organized as follows. Related work is discussed in Section 2. Section 3 introduces our P-CNN features. We summarize state-of-the-art methods used and compared to in our experiments in Section 4 and present datasets in Section 5. Section 6 evaluates our method and compares it to the state of the art. Section 7 concludes the paper. Our implementation of P-CNN features is available from [1].

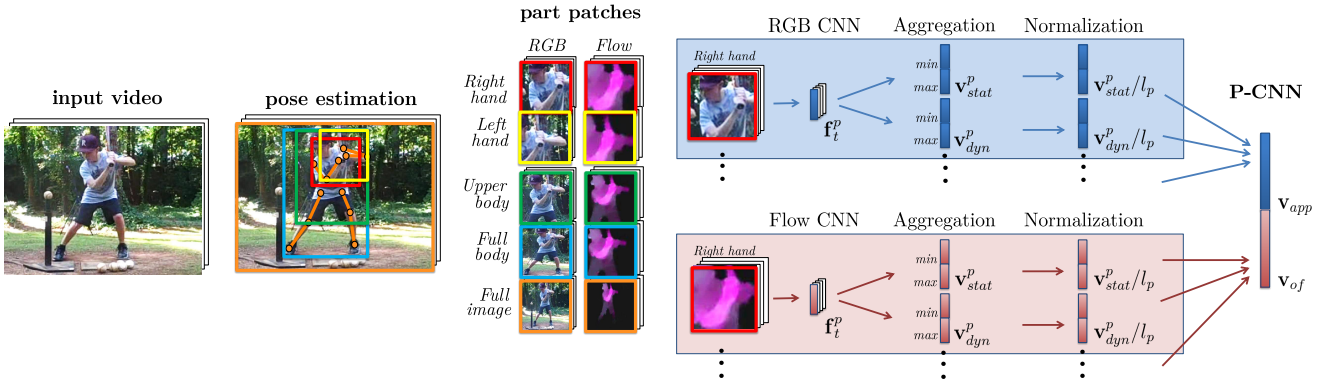

Figure 1: P-CNN features. From left to right: Input video. Human pose and corresponding human body parts for one frame of the video. Patches of appearance (RGB) and optical flow for human body parts. One RGB and one flow CNN descriptor $\mathbf{f}_t^p$ is extracted per frame t and per part p (an example is shown for the human body part *right hand*). Static frame descriptors $\mathbf{f}_t^p$ are aggregated over time using *min* and *max* to obtain the video descriptor $\mathbf{v}_{stat}^p$. Similarly, temporal differences of $\mathbf{f}_t^p$ are aggregated to $\mathbf{v}_{dyn}^p$. Video descriptors are normalized and concatenated over parts p and aggregation schemes into appearance features $\mathbf{v}_{app}$ and flow features $\mathbf{v}_{of}$. The final P-CNN feature is the concatenation of $\mathbf{v}_{app}$ and $\mathbf{v}_{of}$.

2. Related work

Action recognition in the last decade has been dominated by local features [22, 33, 41]. In particular, Dense Trajectory (DT) features [41] combined with Fisher Vector (FV) aggregation [27] have recently shown outstanding results for a number of challenging benchmarks. We use IDT-FV [41] (improved version of DT with FV encoding) as a strong baseline and experimentally demonstrate its complementarity to our method.

Recent advances in Convolutional Neural Networks (CNN) [23] have resulted in significant progress in image classification [20] and other vision tasks [17, 36, 38]. In particular, the transfer of pre-trained network parameters to problems with limited training data has shown success e.g. in [17, 26, 34]. Application of CNNs to action recognition in video, however, has shown only limited improvements so far [34, 43]. We extend previous global CNN methods and address action recognition using CNN descriptors at the local level of human body parts.

Most of the recent methods for action recognition deploy global aggregation of local video descriptors. Such representations provide invariance to numerous variations in the video but may fail to capture important spatio-temporal structure. For fine-grained action recognition, previous methods have represented person-object interactions by joint tracking of hands and objects [24] or, by linking object proposals [44], followed by feature pooling in selected regions. Alternative methods represent actions using positions and temporal evolution of body joints. While reliable human pose estimation is still a challenging task, the recent study [19] reports significant gains provided by dynamic human pose features in cases when reliable pose estimation is available. We extend the work [19] and design a new CNN-based representation for human actions combining positions, appearance and motion of human body parts.

Our work also builds on methods for human pose estimation in images [28, 31, 38, 42] and video sequences [8, 32]. In particular, we build on the method [8] and extract temporally-consistent tracks of body joints from video sequences. While our pose estimator is imperfect, we use it to derive CNN-based pose features providing significant improvements for action recognition for two challenging datasets.

3. P-CNN: Pose-based CNN features

We believe that human pose is essential for action recognition. Here, we use positions of body joints to define informative image regions. We further borrow inspiration from [34] and represent body regions with motion-based and appearance-based CNN descriptors. Such descriptors are extracted at each frame and then aggregated over time to form a video descriptor, see Figure 1 for an overview. The details are explained below.

To construct P-CNN features, we first compute optical flow [4] for each consecutive pair of frames. The method [4] has relatively high speed, good accuracy and has been recently used in other flow-based CNN approaches [18, 34]. Following [18], the values of the motion field v_x, v_y are transformed to the interval $[0, 255]$ by $\tilde{v}_{x|y} = av_{x|y} + b$ with $a = 16$ and $b = 128$. The values below 0 and above 255 are truncated. We save the transformed flow maps as images with three channels corresponding to motion $\tilde{v}_x$, $\tilde{v}_y$ and the flow magnitude.

Given a video frame and the corresponding positions of body joints, we crop RGB image patches and flow patches for *right hand*, *left hand*, *upper body*, *full body* and *full image* as illustrated in Figure 1. Each patch is resized to 224×224 pixels to match the CNN input layer. To represent appearance and motion patches, we use two distinct CNNs with an architecture similar to [20]. Both networks contain 5 convolutional and 3 fully-connected layers. The output of the second fully-connected layer with $k = 4096$ values is used as a frame descriptor ($\mathbf{f}_t^p$). For RGB patches we use the publicly available "VGG-f" network from [6] that has been pre-trained on the ImageNet ILSVRC-2012 challenge dataset [11]. For flow patches, we use the motion network provided by [18] that has been pre-trained for action recognition task on the UCF101 dataset [35].

Given descriptors $\mathbf{f}_t^p$ for each part p and each frame t of the video, we then proceed with the aggregation of $\mathbf{f}_t^p$ over all frames to obtain a fixed-length video descriptor. We consider min and max aggregation by computing minimum and maximum values for each descriptor dimension i over T video frames

$$\begin{aligned} m_i &= \min_{1 \le t \le T} \mathbf{f}_t^p(i), \\ M_i &= \max_{1 \le t \le T} \mathbf{f}_t^p(i). \end{aligned} \tag{1}$$

The *static video descriptor* for part p is defined by the concatenation of time-aggregated frame descriptors as

$$\mathbf{v}_{stat}^p = [m_1, ..., m_k, M_1, ..., M_k]^\top. \tag{2}$$

To capture temporal evolution of per-frame descriptors, we also consider temporal differences of the form $\Delta\mathbf{f}_t^p = \mathbf{f}_{t+\Delta t}^p - \mathbf{f}_t^p$ for $\Delta t = 4$ frames. Similar to (1) we compute minimum Δm_i and maximum ΔM_i aggregations of $\Delta\mathbf{f}_t^p$ and concatenate them into the *dynamic video descriptor*

$$\mathbf{v}_{dyn}^p = [\Delta m_1, ..., \Delta m_k, \Delta M_1, ..., \Delta M_k]^\top. \tag{3}$$

Finally, video descriptors for motion and appearance for all parts and different aggregation schemes are normalized and concatenated into the P-CNN feature vector. The normalization is performed by dividing video descriptors by the average L_2-norm of the $\mathbf{f}_t^p$ from the training set.

In Section 6 we evaluate the effect of different aggregation schemes as well as the contributions of motion and appearance features for action recognition. In particular, we compare "*Max*" vs. "*Max/Min*" aggregation where "*Max*" corresponds to the use of M_i values only while "*Max/Min*" stands for the concatenation of M_i and m_i defined in (2) and (3). *Mean* and *Max* aggregation are widely used methods in CNN video representations. We choose *Max-aggr*, as it outperforms *Mean-aggr* (see Section 6). We also apply *Min* aggregation, which can be interpreted as a "non-detection feature". Additionally, we want to follow the temporal evolution of CNN features in the video by looking at their dynamics (*Dyn*). Dynamic features are again aggregated using *Min* and *Max* to preserve their sign keeping the largest negative and positive differences. The concatenation of static and dynamic descriptors will be denoted by "*Static+Dyn*".

The final dimension of our P-CNN is $(5 \times 4 \times 4K) \times 2 = 160K$, i.e., 5 body parts, 4 different aggregation schemes, 4K-dimensional CNN descriptor for appearance and motion. Note that such a dimensionality is comparable to the size of Fisher vector [5] used to encode dense trajectory features [41]. P-CNN training is performed using a linear SVM.

4. State-of-the-art methods

In this section we present the state-of-the-art methods used and compared to in our experiments. We first present the approach for human pose estimation in videos [8] used in our experiments. We then present state-of-the-art high-level pose features (HLPF) [19] and improved dense trajectories [41].

4.1. Pose estimation

To compute P-CNN features as well as HLPF features, we need to detect and track human poses in videos. We have implemented a video pose estimator based on [8]. We first extract poses for individual frames using the state-of-the-art approach of Yang and Ramanan [42]. Their approach is based on a deformable part model to locate positions of body joints (head, elbow, wrist...). We re-train their model on the FLIC dataset [31].

Following [8], we extract a large set of pose configurations in each frame and link them over time using Dynamic Programming (DP). The poses selected with DP are constrained to have a high score of the pose estimator [42]. At the same time, the motion of joints in a pose sequence is constrained to be consistent with the optical flow extracted at joint positions. In contrast to [8] we do not perform *limb recombination*. See Figure 2 for examples of automatically extracted human poses.

4.2. High-level pose features (HLPF)

High-level pose features (HLPF) encode spatial and temporal relations of body joint positions and were introduced in [19]. Given a sequence of human poses P, positions of body joints are first normalized with respect to the person size. Then, the relative offsets to the head are computed for each pose in P. We have observed that the head is more reliable than the torso used in [19]. Static features are, then, the distances between all pairs of joints, orientations of the vectors connecting pairs of joints and inner angles spanned by vectors connecting all triplets of joints.

Figure 2: Illustration of human pose estimation used in our experiments [8]. Successful examples and failure cases on JHMDB (left two images) and on MPII Cooking Activities (right two images). Only left and right arms are displayed for clarity.

Dynamic features are obtained from trajectories of body joints. HLPF combines temporal differences of some of the static features, i.e., differences in distances between pairs of joints, differences in orientations of lines connecting joint pairs and differences in inner angles. Furthermore, translations of joint positions (dx and dy) and their orientations ($arctan(\frac{dy}{dx})$) are added.

All features are quantized using a separate codebook for each feature dimension (descriptor type), constructed using k-means with $k = 20$. A video sequence is then represented by a histogram of quantized features and the training is performed using an SVM with a χ^2-kernel. More details can be found in [19]. To compute HLPF features we use the publicly available code with minor modifications, i.e., we consider the head instead of the torso center for relative positions. We have also found that converting angles, originally in degrees, to radians and L2 normalizing the HLPF features improves the performance.

4.3. Dense trajectory features

Dense Trajectories (DT) [39] are local video descriptors that have recently shown excellent performance in several action recognition benchmarks [25, 40]. The method first densely samples points which are tracked using optical flow [15]. For each trajectory, 4 descriptors are computed in the aligned spatio-temporal volume: HOG [9], HOF [22] and MBH [10]. A recent approach [41] removes trajectories consistent with the camera motion (estimated computing a homography using optical flow and SURF [2] point matches and RANSAC [16]). Flow descriptors are then computed from optical flow warped according to the estimated homography. We use the publicly available implementation [41] to compute improved version of DT (IDT).

Fisher Vectors (FV) [27] encoding has been shown to outperform the bag-of-word approach [5] resulting in state-of-the-art performance for action recognition in combination with DT features [25]. FV relies on a Gaussian mixture model (GMM) with K Gaussian components, computing first and second order statistics with respect to the GMM. FV encoding is performed separately for the 4 different IDT descriptors (their dimensionality is reduced by the factor of 2 using PCA). Following [27], the performance is improved by passing FV through signed square-rooting and L_2 normalization. As in [25] we use a spatial pyramid representation and a number of $K = 256$ Gaussian components. FV encoding is performed using the Yael library [13] and classification is performed with a linear SVM.

5. Datasets

In our experiments we use two datasets JHMDB [19] and MPII Cooking Activities [29], as well as two subsets of these datasets sub-JHMDB and sub-MPII Cooking. We present them in the following.

JHMDB [19] is a subset of HMDB [21], see Figure 2 (left). It contains 21 human actions, such as *brush hair*, *climb*, *golf*, *run* or *sit*. Video clips are restricted to the duration of the action. There are between 36 and 55 clips per action for a total of 928 clips. Each clip contains between 15 and 40 frames of size 320×240. Human pose is annotated in each of the 31838 frames. There are 3 train/test splits for the JHMDB dataset and evaluation averages the results over these three splits. The metric used is accuracy: each clip is assigned an action label corresponding to the maximum value among the scores returned by the action classifiers.

In our experiments we also use a subset of JHMDB, referred to as **sub-JHMDB[19]**. This subset includes 316 clips distributed over 12 actions in which the human body is fully visible. Again there are 3 train/test splits and the evaluation metric is accuracy.

MPII Cooking Activities [29] contains 64 fine-grained actions and an additional background class, see Figure 2 (right). Actions take place in a kitchen with static background. There are 5609 action clips of frame size 1624×1224. Some actions are very similar, such as *cut dice*, *cut slices*, and *cut stripes* or *wash hands* and *wash objects*. Thus, these activities are qualified as "fine-grained". There are 7 train/test splits and the evaluation is reported in terms of mean Average Precision (mAP) using the code provided with the dataset.

Parts	JHMDB-GT			MPII Cooking-Pose[8]		
	App	OF	App + OF	App	OF	App + OF
Hands	46.3	54.9	57.9	39.9	46.9	51.9
Upper body	52.8	60.9	67.1	32.3	47.6	50.1
Full body	52.2	61.6	66.1	-	-	-
Full image	43.3	55.7	61.0	28.8	56.2	56.5
All	**60.4**	**69.1**	**73.4**	**43.6**	**57.4**	**60.8**

Table 1: Performance of appearance-based (App) and flow-based (OF) P-CNN features. Results are obtained with max-aggregation for JHMDB-GT (% accuracy) and MPII Cooking Activities-Pose [8] (% mAP).

We have also defined a subset of MPII cooking, referred to as **sub-MPII cooking**, with classes *wash hands* and *wash objects*. We have selected these two classes as they are visually very similar and differ mainly in manipulated objects. To analyze the classification performance for these two classes in detail, we have annotated human pose in all frames of sub-MPII cooking. There are 55 and 139 clips for *wash hands* and *wash objects* actions respectively, for a total of $29,997$ frames.

6. Experimental results

This section describes our experimental results and examines the effect of different design choices. First, we evaluate the complementarity of different human parts in Section 6.1. We then compare different variants for aggregating CNN features in Section 6.2. Next, we analyze the robustness of our features to errors in the estimated pose and their ability to classify fine-grained actions in Section 6.3. Finally, we compare our features to the state of the art and show that they are complementary to the popular dense trajectory features in Section 6.4.

6.1. Performance of human part features

Table 1 compares the performance of human part CNN features for both appearance and flow on JHMDB-GT (the JHMDB dataset with ground-truth pose) and MPII Cooking-Pose [8] (the MPII Cooking dataset with pose estimated by [8]). Note, that for MPII Cooking we detect upper-body poses only since full bodies are not visible in most of the frames in this dataset.

Conclusions for both datasets are similar. We can observe that all human parts (hands, upper body, full body) as well as the full image have similar performance and that their combination improves the performance significantly. Removing one part at a time from this combination results in the drop of performance (results not shown here). We therefore use all pose parts together with the full image descriptor in the following evaluation. We can also observe that flow descriptors consistently outperform appearance descriptors by a significant margin for all parts as well as for the overall combination *All*. Furthermore, we can observe that the combination of appearance and flow further improves the performance for all parts including their combination *All*. This is the pose representation used in the rest of the evaluation.

In this section, we have applied the max-aggregation (see Section 3) for aggregating features extracted for individual frames into a video descriptor. Different aggregation schemes will be compared in the next section.

6.2. Aggregating P-CNN features

CNN features $\mathbf{f}_t$ are first extracted for each frame and the following temporal aggregation pools feature values for each feature dimension over time (see Figure 1). Results of max-aggregation for JHMDB-GT are reported in Table 1 and compared with other aggregation schemes in Table 2. Table 2 shows the impact of adding min-aggregation (*Max/Min-aggr*) and the first-order difference between CNN features (*All-Dyn*). Combining per-frame CNN features and their first-order differences using max- and min-aggregation further improves results. Overall, we obtain the best results with *All-(Static+Dyn)(Max/Min-aggr)* for App + OF, i.e., 74.6% accuracy on JHMDB-GT. This represents an improvement over *Max-aggr* by 1.2%. On MPII Cooking-Pose [8] this version of P-CNN achieves 62.3% mAP (as reported in Table 4) leading to an 1.5% improvement over max-aggregation reported in Table 1.

Aggregation scheme	App	OF	App+OF
All(Max-aggr)	60.4	69.1	73.4
All(Max/Min-aggr)	60.6	68.9	73.1
All(Static+Dyn)(Max-aggr)	62.4	**70.6**	74.1
All(Static+Dyn)(Max/Min-aggr)	**62.5**	70.2	**74.6**
All(Mean-aggr)	57.5	69.0	69.4

Table 2: Comparison of different aggregation schemes: *Max*-, *Mean*-, and *Max/Min*-aggregations as well as adding first-order differences (*Dyn*). Results are given for appearance (*App*), optical flow (*OF*) and App + OF on JHMDB-GT (% accuracy).

We have also experimented with second-order differences and other statistics, such as mean-aggregation (last row in Table 2), but this did not improve results. Furthermore,

we have tried temporal aggregation of classification scores obtained for individual frames. This led to a decrease of performance, *e.g*. for *All (App)* on JHMDB-GT score-max-aggregation results in 56.1% accuracy, compared to 60.4% for features-max-aggregation (top row, left column in Table 2). This indicates that early aggregation works significantly better in our setting.

In summary, the best performance is obtained for *Max-aggr* on single-frame features, if only one aggregation scheme is used. Addition of *Min-aggr* and first order differences *Dyn* provides further improvement. In the remaining evaluation we report results for this version of P-CNN, i.e., *All* parts *App+OF* with *(Static+Dyn)(Max/Min-aggr)*.

6.3. Robustness of pose-based features

This section examines the robustness of P-CNN features in the presence of pose estimation errors and compares results with the state-of-the-art pose features HLPF [19]. We report results using the code of [19] with minor modifications described in Section 4.2. Our HLPF results are comparable to [19] in general and are slightly better on JHMDB-GT (77.8% vs. 76.0%). Table 3 evaluates the impact of automatic pose estimation versus ground-truth pose (GT) for sub-JHMDB and JHMDB. We can observe that results for GT pose are comparable on both datasets and for both type of pose features. However, P-CNN is significantly more robust to errors in pose estimation. For automatically estimated poses P-CNN drops only by 5.7% on sub-JHMDB and by 13.5% on JHMDB, whereas HLPF drops by 13.5% and 52.5% respectively. For both descriptors the drop is less significant on sub-JHMDB, as this subset only contains full human poses for which pose is easier to estimate. Overall the performance of P-CNN features for automatically extracted poses is excellent and outperforms HLPF by a very large margin ($+35.8\%$) on JHMDB.

	sub-JHMDB		
	GT	Pose [42]	Diff
P-CNN	72.5	66.8	5.7
HLPF	78.2	51.1	27.1

	JHMDB		
	GT	Pose [8]	Diff
P-CNN	74.6	61.1	13.5
HLPF	77.8	25.3	52.5

Table 3: Impact of automatic pose estimation versus ground-truth pose (GT) for P-CNN features and HLPF [19]. Results are presented for sub-JHMDB and JHMDB (% accuracy).

We now compare and evaluate the robustness of P-CNN and HLPF features on the MPII cooking dataset. To evaluate the impact of ground-truth pose (GT), we have manually annotated two classes "washing hand" and "washing objects", referred to by sub-MPII Cooking. Table 4 compares P-CNN and HLPF for sub-MPII and MPII Cooking. In all cases P-CNN outperforms HLPF significantly. Interestingly, even for ground-truth poses P-CNN performs significantly better than HLPF. This could be explained by the better encoding of image appearance by P-CNN features, especially for object-centered actions such as "washing hands" and "washing objects". We can also observe that the drop due to errors in pose estimation is similar for P-CNN and HLPF. This might be explained by the fact that CNN hand features are quite sensitive to the pose estimation. However, P-CNN still significantly outperforms HLPF for automatic pose. In particular, there is a significant gain of $+29.7\%$ for the full MPII Cooking dataset.

	sub-MPII Cooking		
	GT	Pose [8]	Diff
P-CNN	83.6	67.5	16.1
HLPF	76.2	57.4	18.8

	MPII Cooking
	Pose [8]
P-CNN	62.3
HLPF	32.6

Table 4: Impact of automatic pose estimation versus ground-truth pose (GT) for P-CNN features and HLPF [19]. Results are presented for sub-MPII Cooking and MPII Cooking (% mAP).

6.4. Comparison to the state of the art

In this section we compare to state-of-the-art dense trajectory features [41] encoded by Fisher vectors [25] (IDT-FV), briefly described in Section 4.3. We use the online available code, which we validated on Hollywood2 (65.3% versus 64.3% [41]). Furthermore, we show that our pose features P-CNN and IDT-FV are complementary and compare to other state-of-the-art approaches on JHMDB and MPII Cooking.

Table 5 shows that for ground-truth poses our P-CNN features outperform state-of-the-art IDT-FV descriptors significantly (8.7%). If the pose is extracted automatically both methods are on par. Furthermore, in all cases the combination of P-CNN and IDT-FV obtained by late fusion of the individual classification scores significantly increases the performance over using individual features only. Figure 3 illustrates per-class results for P-CNN and IDT-FV on JHMDB-GT.

Table 6 compares our results to other methods on MPII Cooking. Our approach outperforms the state of the art on

Method	JHMDB		MPII Cook.
	GT	Pose [8]	Pose [8]
P-CNN	74.6	61.1	62.3
IDT-FV	65.9	65.9	67.6
P-CNN + IDT-FV	**79.5**	**72.2**	**71.4**

Table 5: Comparison to IDT-FV on JHMDB (% accuracy) and MPII Cooking Activities (% mAP) for ground-truth (GT) and pose [8]. The combination of P-CNN + IDT-FV performs best.

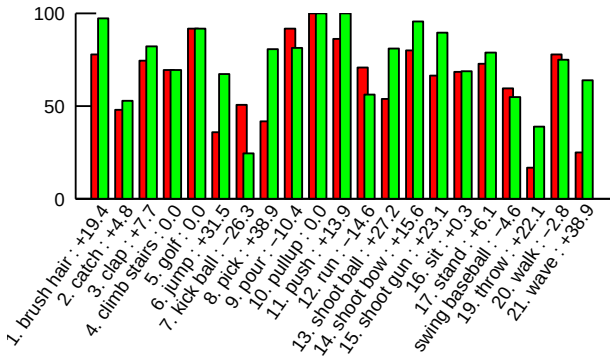

Figure 3: Per class accuracy on JHMDB-GT for P-CNN (green) and IDT-FV (red) methods. Values correspond to the difference in accuracy between P-CNN and IDT-FV (positive values indicate better performance of P-CNN).

this dataset and is on par with the recently published work of [44]. We have compared our method with HLPF [19] on JHMDB in the previous section. P-CNN perform on par with HLPF for GT poses and significantly outperforms HLPF for automatically estimated poses. Combination of P-CNN with IDT-FV improves the performance to 79.5% and 72.2% for GT and automatically estimated poses respectively (see Table 5). This outperforms the state-of-the-art result reported in [19].

Qualitative results comparing P-CNN and IDT-FV are presented in Figure 4 for JHMDB-GT. See Figure 3 for the quantitative comparison. To highlight improvements achieved by the proposed P-CNN descriptor, we show results for classes with a large improvement of P-CNN over IDT-FV, such as *shoot_gun*, *wave*, *throw* and *jump* as well as for a class with a significant drop, namely *kick_ball*. Figure 4 shows two examples for each selected action class with the maximum difference in ranks obtained by P-CNN (green) and IDT-FV (red). For example, the most significant improvement (Figure 4, top left) increases the sample ranking from 211 to 23, when replacing IDT-FV by P-CNN. In particular, the *shoot gun* and *wave* classes involve small localized motion, making classification difficult for IDT-FV while P-CNN benefits from the local human body part information. Similarly, the two samples from the action class *throw* also seem to have restricted and localized motion while the action *jump* is very short in time. In the case of *kick_ball* the significant decrease can be explained by the important dynamics of this action, which is better captured by IDT-FV features. Note that P-CNN only captures motion information between two consecutive frames.

Method	MPII Cook.
Holistic + Pose [29]	57.9
Semantic Features [45]	70.5
Interaction Part Mining [44]	72.4
P-CNN + IDT-FV (our)	71.4

Table 6: State of the art on the MPII Cooking (% mAP).

Figure 5 presents qualitative results for MPII Cooking-Pose [8] showing samples with the maximum difference in ranks over all classes.

7. Conclusion

This paper introduces pose-based convolutional neural network features (P-CNN). Appearance and flow information is extracted at characteristic positions obtained from human pose and aggregated over frames of a video. Our P-CNN description is shown to be significantly more robust to errors in human pose estimation compared to existing pose-based features such as HLPF [19]. In particular, P-CNN significantly outperforms HLPF on the task of fine-grained action recognition in the MPII Cooking Activities dataset. Furthermore, P-CNN features are complementary to the dense trajectory features and significantly improve the current state of the art for action recognition when combined with IDT-FV.

Our study confirms conclusions in [19], namely, that correct estimation of human poses leads to significant improvements in action recognition. This implies that pose is crucial to capture discriminative information of human actions. Pose-based action recognition methods have a promising future due to the recent progress in pose estimation, notably using CNNs [7]. An interesting direction for future work is to adapt CNNs for each P-CNN part (hands, upper body, etc.) by fine-tuning networks for corresponding image areas. Another promising direction is to model temporal evolution of frames using RNNs [12, 30].

Acknowledgements

This work was supported by the MSR-Inria joint lab, a Google research award, the ERC starting grant ACTIVIA and the ERC advanced grant ALLEGRO.

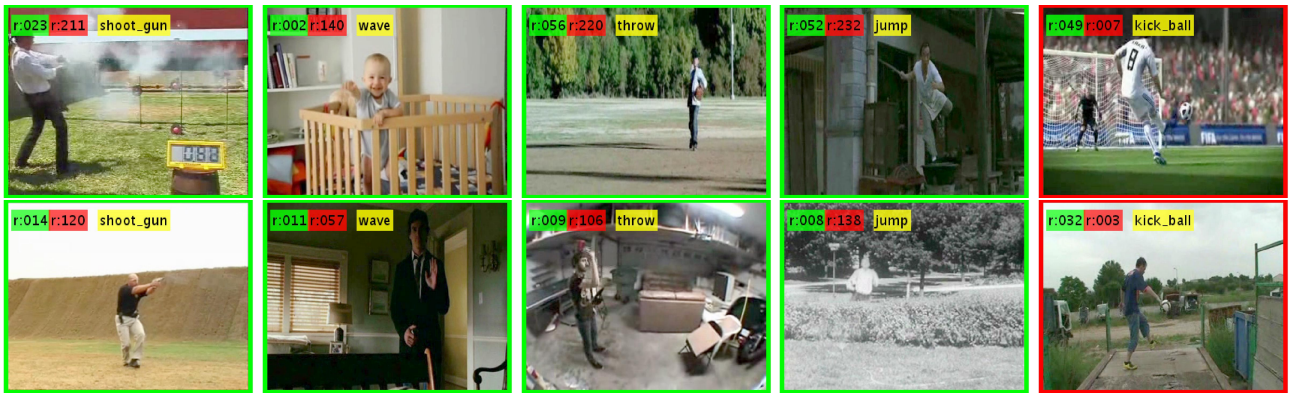

Figure 4: Results on JHMDB-GT (split 1). Each column corresponds to an action class. Video frames on the left (green) illustrate two test samples per action with the largest improvement in ranking when using P-CNN (rank in green) and IDT-FV (rank in red). Examples on the right (red) illustrate samples with the largest decreases in the ranking. Actions with large differences in performance are selected according to Figure 3. Each video sample is represented by its middle frame.

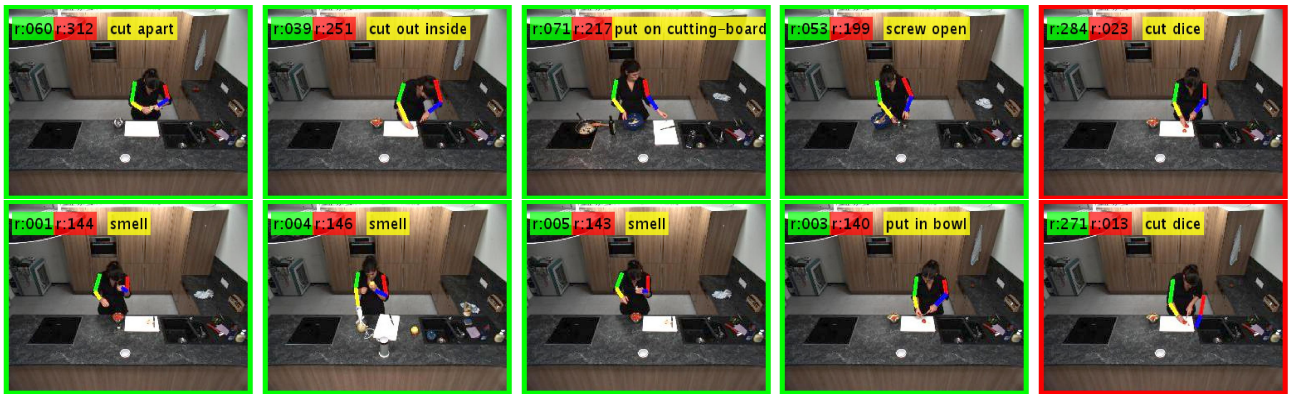

Figure 5: Results on MPII Cooking-Pose [8] (split 1). Examples on the left (green) show the 8 best ranking improvements (over all classes) obtained by using P-CNN (rank in green) instead of IDT-FV (rank in red). Examples on the right (red) illustrate video samples with the largest decrease in the ranking. Each video sample is represented by its middle frame.

References

[1] `http://www.di.ens.fr/willow/research/p-cnn/`.

[2] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Comput. Vis. Image Underst.*, pages 346–359, 2008.

[3] T. Berg and P. N. Belhumeur. Poof: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation. In *CVPR*, 2013.

[4] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, 2004.

[5] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *BMVC*, 2011.

[6] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014.

[7] X. Chen and A. Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *NIPS*, 2014.

[8] A. Cherian, J. Mairal, K. Alahari, and C. Schmid. Mixing body-part sequences for human pose estimation. In *CVPR*, 2014.

[9] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[10] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *ECCV*, 2006.

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *In CVPR*, 2009.

[12] 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.

[13] M. Douze and H. Jégou. The Yael library. In *ACM Multimedia*, 2014.

[14] K. Duan, D. Parikh, D. Crandall, and K. Grauman. Discovering localized attributes for fine-grained recognition. In *CVPR*, 2012.

[15] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. In *SCIA*, 2003.

[16] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 1981.

[17] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[18] G. Gkioxari and J. Malik. Finding action tubes. In *CVPR*, 2015.

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

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012.

[21] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *ICCV*, 2011.

[22] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[23] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

[24] B. Ni, V. R. Paramathayalan, and P. Moulin. Multiple granularity analysis for fine-grained action detection. In *CVPR*, 2014.

[25] D. Oneata, J. Verbeek, and C. Schmid. Action and event recognition with fisher vectors on a compact feature set. In *ICCV*, 2013.

[26] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, 2014.

[27] F. Perronnin, J. Sánchez, and T. Mensink. Improving the Fisher kernel for large-scale image classification. In *ECCV*, 2010.

[28] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele. Poselet conditioned pictorial structures. In *CVPR*, 2013.

[29] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. A database for fine grained activity detection of cooking activities. In *CVPR*, 2012.

[30] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *NATURE*, 323:9, 1986.

[31] B. Sapp and B. Taskar. Modec: Multimodal decomposable models for human pose estimation. In *CVPR*, 2013.

[32] B. Sapp, D. Weiss, and B. Taskar. Parsing human motion with stretchable models. In *CVPR*, 2011.

[33] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. In *ICPR*, 2004.

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

[35] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. In *CRCV-TR-12-01*, 2012.

[36] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, 2014.

[37] 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.

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

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

[40] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, 103(1):60–79, 2013.

[41] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.

[42] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, 2011.

[43] N. J. Yue-Hei, H. Matthew, V. Sudheendra, V. Oriol, M. Rajat, and T. George. Beyond short snippets: Deep networks for video classification. In *CVPR*, 2015.

[44] Y. Zhou, B. Ni, R. Hong, M. Wang, and Q. Tian. Interaction part mining: A mid-level approach for fine-grained action recognition. In *CVPR*, 2015.

[45] Y. Zhou, B. Ni, S. Yan, P. Moulin, and Q. Tian. Pipelining localized semantic features for fine-grained action recognition. In *ECCV*, 2014.