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.

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, 27, 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].
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 [43], 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 \( \hat{v}_x, \hat{v}_y \) are transformed to the interval \([0, 255]\) by \( \tilde{v}_x = av_{x,y} + b \) 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 \( \hat{v}_x, \hat{v}_y \) 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 ($f_1^p$). 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 [13] that has been pre-trained for action recognition task on the UCF101 dataset [35].

Given descriptors $f_1^p$ for each part $p$ and each frame $t$ of the video, we then proceed with the aggregation of $f_1^p$ 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

$$m_i = \min_{1 \leq t \leq T} f_1^p(i),$$

$$M_i = \max_{1 \leq t \leq T} f_1^p(i).$$

(1)

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

$$v_{stat}^p = [m_1, ..., m_k, M_1, ..., M_k]^T.$$  

(2)

To capture temporal evolution of per-frame descriptors, we also consider temporal differences of the form $\Delta f_t^p = f_{t+\Delta t}^p - f_t^p$ for $\Delta t = 4$ frames. Similar to [1] we compute minimum $\Delta m_i$ and maximum $\Delta M_i$ aggregations of $\Delta f_t^p$ and concatenate them into the dynamic video descriptor

$$v_{dyn}^p = [\Delta m_1, ..., \Delta m_k, \Delta M_1, ..., \Delta M_k]^T.$$  

(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 $f_t^p$ from the training set.

In Section 3 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 [3] 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.
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, trans-
lations of joint positions (\(dx\) and \(dy\)) and their orientations 
(\(\arctan(dy/dx)\)) 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 repre-
sented by a histogram of quantized features and the training 
is performed using an SVM with a \(\chi^2\)-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, origi-
nally 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 descrip-
tors that have recently shown excellent performance in sev-
eral 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 esti-
mated homography. We use the publicly available implement-
ation \([41]\) 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 \(L_2\) nor-
malization. As in \([25]\) we use a spatial pyramid representa-
tion 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 \(320 \times 240\). 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, re-
ferred 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 back-
ground. There are 5609 action clips of frame size \(1624 \times 
1224\). 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.
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.

<table>
<thead>
<tr>
<th>Parts</th>
<th>JHMDB-GT</th>
<th>MPII Cooking-Pose [8]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>App</td>
<td>OF</td>
</tr>
<tr>
<td>Hands</td>
<td>46.3</td>
<td>54.9</td>
</tr>
<tr>
<td>Upper body</td>
<td>52.8</td>
<td>60.9</td>
</tr>
<tr>
<td>Full body</td>
<td>52.2</td>
<td>61.6</td>
</tr>
<tr>
<td>Full image</td>
<td>43.3</td>
<td>55.7</td>
</tr>
<tr>
<td>All</td>
<td>60.4</td>
<td>69.1</td>
</tr>
</tbody>
</table>

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 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 $f_t$ 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 3) leading to an 1.5% improvement over max-aggregation reported in Table 1.

<table>
<thead>
<tr>
<th>Aggregation scheme</th>
<th>App</th>
<th>OF</th>
<th>App+OF</th>
</tr>
</thead>
<tbody>
<tr>
<td>All(Max-aggr)</td>
<td>60.4</td>
<td>69.1</td>
<td>73.4</td>
</tr>
<tr>
<td>All(Max/Min-aggr)</td>
<td>60.6</td>
<td>68.9</td>
<td>73.1</td>
</tr>
<tr>
<td>All(Static+Dyn)(Max-aggr)</td>
<td>62.4</td>
<td>70.6</td>
<td>74.1</td>
</tr>
<tr>
<td>All(Static+Dyn)(Max/Min-aggr)</td>
<td><strong>62.5</strong></td>
<td>70.2</td>
<td><strong>74.6</strong></td>
</tr>
<tr>
<td>All(Mean-aggr)</td>
<td>57.5</td>
<td>69.0</td>
<td>69.4</td>
</tr>
</tbody>
</table>

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 [7]). 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.

<table>
<thead>
<tr>
<th>sub-JHMDB</th>
<th>GT</th>
<th>Pose [42]</th>
<th>Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>P-CNN</td>
<td>72.5</td>
<td>66.8</td>
<td>5.7</td>
</tr>
<tr>
<td>HLPF</td>
<td>78.2</td>
<td>51.1</td>
<td>27.1</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>sub-MPII Cooking</th>
<th>GT</th>
<th>Pose [8]</th>
<th>Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>P-CNN</td>
<td>83.6</td>
<td>67.5</td>
<td>16.1</td>
</tr>
<tr>
<td>HLPF</td>
<td>76.2</td>
<td>57.4</td>
<td>18.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MPII Cooking</th>
<th>Pose [8]</th>
</tr>
</thead>
<tbody>
<tr>
<td>P-CNN</td>
<td>62.3</td>
</tr>
<tr>
<td>HLPF</td>
<td>32.6</td>
</tr>
</tbody>
</table>

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 5 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

<table>
<thead>
<tr>
<th>sub-MPII Cooking</th>
<th>GT</th>
<th>Pose [8]</th>
<th>Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>P-CNN</td>
<td>83.6</td>
<td>67.5</td>
<td>16.1</td>
</tr>
<tr>
<td>HLPF</td>
<td>76.2</td>
<td>57.4</td>
<td>18.8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MPII Cooking</th>
<th>Pose [8]</th>
</tr>
</thead>
<tbody>
<tr>
<td>P-CNN</td>
<td>62.3</td>
</tr>
<tr>
<td>HLPF</td>
<td>32.6</td>
</tr>
</tbody>
</table>
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.

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


