Abstract

With the improved accessibility to an exploding amount of realistic video data and growing demands in many video analysis applications, video-based large-scale action recognition is becoming an increasingly important task in computer vision. In this notebook paper, we give a brief description on our method for large-scale human action recognition in realistic videos. This method is originally proposed in our ICCV 2013 paper (“Action Recognition with Actons”), which presented a two-layer structure for action recognition to automatically exploit a mid-level “acton” representation. The weakly-supervised actons are learned via a max-margin multi-channel multiple instance learning algorithm (called M^4 IL), which can capture multiple mid-level action concepts simultaneously for producing a discriminative and compact video representation on action recognition. The learned actons (with no requirement for detailed manual annotations) observe the properties of being compact, informative, discriminative, and easy to scale. The experimental results demonstrate the effectiveness of applying the learned actons in our two-layer structure, and show a classification accuracy of 80.9% on the UCF101 dataset.

1. The Method

As presented in [9], our two-layer structure for generating video representation is composed of two components, with the goal of leveraging the benefits of low-level, mid-level as well as the layered representation for knowledge abstraction: The first-layer representation is built by the codes of local spatial-temporal feature points, and the second-layer representation is constructed based on the acton responses of volumes of interest (VOIs). Specifically, the first layer builds a low-level representation based on a classic SPM pipeline [3, 8, 4], while the second layer automatically exploits semantically meaningful mid-level representation (i.e., the acton representation) via a weakly-supervised M^4 IL learning algorithm [9]. The acton representation in the second layer is built directly on top of the first layer, with the goal of efficient knowledge abstraction and aggregation. Besides, the learned actons work as a mid-level dictionary of intermediate concepts to characterize the semantic properties for each VOI. Figure [9] illustrates our two-layer structure on action classification.

As illustrated in Fig. 1, the first layer of our method adopts the linear SPM pipeline [8, 4], which is the state-of-the-art framework in image classification literature, to build the first-layer representation for a video clip. Generally, it includes three common steps: local feature extraction, feature coding and spatial-temporal pooling. For an input video clip, we can detect several local spatial-temporal features (e.g., STIPs [2] or dense trajectory features [7]), and compute corresponding feature descriptors to capture a variety of visual cues (e.g., appearance, motion) from them. Given a visual codebook with \( M \) codewords, each STIP feature is encoded into a \( M \)-dimensional code vector. In our method, we adopt the localized soft-assignment quantization (LSAQ) method [4] for feature coding. It computes a L1-normalized code vector based on the L2-distance between each feature descriptor and its \( k \)-nearest-neighbour codewords. For capturing informative statistics and achieving invariance properties (e.g., transformation invariance) in spatial-temporal domain, a maximum pooling step is exploited to form the video-level representation of whole video clip, via statistical summarization over the codes in a series of subvolumes. We use a volumetric spatial-temporal pyramid for the pooling step as in [7], which includes six different spatial-temporal grids leading to a total of \( L = 24 \) subvolumes. Thus, after the pooling step, we concatenate the code vectors of all subvolumes into a \((M \times L)\)-dimensional single vector as the first-layer representation of a video clip.

Besides the first-layer representation based on local STIPs, we construct a higher-level one by the acton responses of VOIs for action recognition in videos. In the second layer, a video clip is decomposed into a set of VOIs, which
potentially correspond to action parts or relevant objects. Generally, the VOIs can be extracted by saliency detector or densely sampling from a regular grid in spatial-temporal domain. For each VOI, its feature descriptor is computed by pooling the codes of STIPs located in this VOI. Besides, the VOI descriptor is further normalized by its L2-norm. Given $K$ acton models $\{w_1, w_2, \cdots, w_K\}$, we construct the second-layer representation of input video clip via their acton responses on extracted VOIs. Let $w_k$ and $x_j$ respectively denote the $k^{th}$ acton model and the feature descriptor of VOI $j$. The acton responses of VOI, which have a probabilistic value between $[0, 1]$, are computed by the sigmoid transformed scores of acton models on its feature descriptor. That is $r_{j,k} = S(w_k^T x_j)$ ($k = 1, 2, \cdots, K$). The sigmoid function is given by $S(v) = \frac{1}{1+\exp(-\rho v)}$, where $v = w^T x$ denotes a real-valued score and $\rho$ is a saturation parameter for controlling its sharpness. Similar to the first-layer representation, we further pool acton responses for each subvolume of spatial-temporal pyramid, and obtain a $(K \times L)$-dimensional vector as the second-layer representation of video clip. After performing L2-normalization on each of the two-layer representations respectively, we concatenate them into a single vector as our two-layer representation of video clip. Finally, we can apply any off-the-shelf classifier on this two-layer video representation for action classification.

For the second layer, the mid-level intermediate concepts (i.e., actons) are not predefined and learned in a data-driven manner. Although the acton labels of VOIs are unknown in training, the class labels of whole video clip are available. Thus, we learn the linear models of actons via the M^4IL algorithm, which is a new multiple instance learning method proposed in [9]. The details of M^4IL can be found in [9].

2. Experimental Evaluation

We evaluate the recognition performance of our two-layer representation on a large-scale action dataset (i.e. UCF101 [6]). UCF101 is the largest dataset in current human action recognition literature, which includes 101 different human action categories and over 10,000 video clips in total. For each category, the video clips are divided into 25 groups, each of which have 4-7 clips normally sharing common background and actor(s). Following the protocol of competition track in the first ICCV workshop on action recognition with a large number of classes (THUMOS ‘13), we use three predefined training/testing splits released from http://crcv.ucf.edu/ICCV13-Action-Workshop/index.files/UCF101TrainTestSplits-RecognitionTask.zip. For each split the video clips from 7 of the 25 groups are used as testing samples, and the rest for training. The performance measurement is the average classification accuracy over all classes. Specifically, the performance of our method is evaluated by the mean of average accuracies over the three testing splits. In the following, we first introduce the parameter and settings of our method used in experimental evaluation, and then show results afterwards.

For the first-layer representation, we extract local spatial-temporal features by computing the dense trajectories [7] and adopt three types of feature descriptors used in [7]: motion boundary histogram (MBH) [1], histogram of oriented gradients (HOG) [2, 7], and trajectory shape descrip-

---

Figure 1. Illustration of the two-layer structure on action classification [9]. (Best viewed in color)
tor (TrajShape) [7]. In the step of feature coding, we randomly select 100,000 local spatial-temporal features and build a visual dictionary via standard K-means algorithm. The number of visual words is set by $M = 4,000$ for each feature type, leading to a total of $4000 \times 3 \times 24 = 288,000$ dimension for the first-layer representation. The parameter setting of LSAQ coding is the same as [4] (i.e., $\beta = 10$ and $n = 5$).

For the second-layer representation, the VOIs are densely sampled from a regular grid like [5], with a spacing of 50% overlapping in each direction of spatial-temporal domain. There are 8 different configurations of cuboid VOIs used in our method, where the size in $x/y$ direction is 80 or 120 pixels and the size in temporal domain is 40 or 60 frames. We learn the actons in “one-vs-rest” manner. It means that the training video clips of each class are used as positive bags in turn, while the ones of other classes serve as negative bags. After that, we collect all the actons learned from various classes for computing the responses. We train $K = 3$ actons for each action category, such that the dimension of the second-layer representation is $3 \times 101 \times 3 \times 24 = 21,816$. The value of $\rho$ is set by 1.5. Following [8,4], we use the linear SVM classifier in experiments, and the “one-vs-rest” criterion is applied to multi-class classification. The regularization parameter of training SVM is set by 16.

In table [1] we show the classification performance and corresponding dimensions for the representations of individual layers as well as the combined representation of the two layers. When using the first layer only, we obtain the result of 78.8%, and the classification accuracy of our method can boost to 80.9% if we cooperate it with the second-layer representation. It indicates a 2.1% performance improvement for the case of adding the learned acton representation. Moreover, we can see that the performance of using the second layer only (i.e., 78.2%) is merely 0.6% less than the one of using first layer only. Considering there are quite a few (i.e., only 7.6%) dimension used in the second-layer representation relative to the first-layer one, it validates the compactness and effectiveness of our acton representation learned from data.

<table>
<thead>
<tr>
<th></th>
<th>Only 1st</th>
<th>Only 2nd</th>
<th>$1^{st} + 2^{nd}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy (%)</td>
<td>78.8</td>
<td>78.2</td>
<td>80.9</td>
</tr>
<tr>
<td>Number of Dim.</td>
<td>288,000</td>
<td>21,816</td>
<td>309,816</td>
</tr>
</tbody>
</table>

Table 1. The classification performance and the number of dimensions for our two-layer representation. From the second to the fourth columns: using the first layer only, using the second layer only, combination of the two layers.

References