Adaptive Structured Pooling for Action Recognition

Svebor Karaman[1]
svebor.karaman@unifi.it

Lorenzo Seidenari[1]
lorenzo.seidenari@unifi.it

Shugao Ma[2]
shugaoma@bu.edu

Alberto Del Bimbo[1]
alberto.delbimbo@unifi.it

Stan Sclaroff[2]
sclaroff@bu.edu

[1] MICC (Media Integration and
Communication Center)
University of Florence
Florence, Italy

[2] Boston University
Boston, USA

Abstract

We propose an adaptive structured pooling strategy to solve the action recognition problem in videos. Our method aims at individuating several spatio-temporal pooling regions each corresponding to a consistent spatial and temporal subset of the video. Each subset of the video gives a pooling weight map and is represented as a Fisher vector computed from the soft weighted contributions of all dense trajectories evolving in it. We further represent each video through a graph structure, defined over multiple granularities of spatio-temporal subsets. The graph structures extracted from all videos are finally compared with an efficient graph matching kernel. Our approach does not rely on a fixed partitioning of the video. Moreover, the graph structure depicts both spatial and temporal relationships between the spatio-temporal subsets. Experiments on the UCF Sports and the HighFive datasets show performance above the state-of-the-art.

1 Introduction

Automatic human action recognition in videos is an important and popular research topic in computer vision, with potential applications in video analytics, video retrieval and video surveillance. While near perfect performance has been achieved in simplistic lab video datasets [7, 25], recognizing human action in realistic videos such as sports videos and TV programs is still quite challenging due to camera and object motion, background distraction, occlusion and viewpoint variation. To tackle these issues, several local space-time features have been proposed, e.g. space-time interest points [12] and dense trajectories [34].

The most common approach to exploit these features is to build a Bag-of-Words [27] like representation of the whole video. These representations rely on the definition of a vocabulary of local space-time features. Then the local features are encoded with respect to the vocabulary. The Fisher encoding method [20] has proven to be very powerful when applied on

© 2014. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

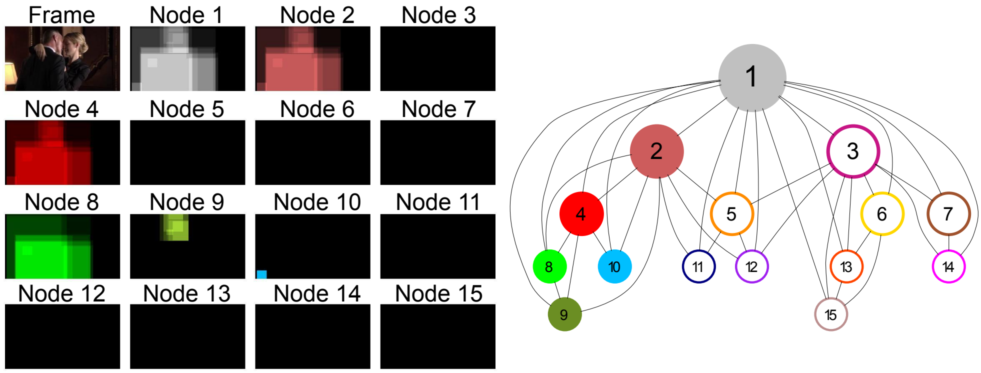

Figure 1: An overview of our method. Top left: a frame of the "kiss" action of the HighFive dataset. Right: the video structure graph where nodes are pooling regions at different granularities (the bigger the node, the coarser the granularity). All the remaining plots are the pooling map of each node where the color correspond to the node color in the graph. Active nodes on this frame are depicted with filled circles. Note how node 9 selects both actors faces. Note that pooling regions are spatio-temporal regions, a video illustrating the pooling process on the whole video is available in the supplementary material.

images and recently has been applied successfully in videos. All top performers [8, 16, 32] of the recent THUMOS Action Recognition challenge used Fisher encoding.

However, many action recognition approaches discard the space, time and hierarchical relationships among the local video subvolumes that these space-time features represent. These overlooked relationships may constitute discriminative structures of local space-time features, which could be very useful for correctly recognizing human actions, e.g. complex but structural movements of different body parts in sports such as bench swing or an interaction between two persons. Indeed, a video contains highly dynamic content, and the visual content corresponding to semantic concepts of the target class may appear at different position, time and speed for different videos of the same class. Hence, the computation of a single global representation may be harmed by some noisy elements in the surrounding (both spatially and temporally) of the event of interest.

In this work, we aim at exploiting multiple dynamic pooling regions that adapt to the spatial layout and dynamics of each video. Each pooling region is represented through a soft Fisher encoding. The relationships between the pooling regions are captured in a graph structure that represents the whole video. Our method is illustrated in Fig. 1. The video structures are efficiently comparable with the state-of-the-art GraphHopper kernel [4]. Our adaptive structured pooling shows an improvement over the state of the art for action recognition.

2 Related works

The most powerful video representation is based on local spatio-temporal features either sampled on a regular grid [33] or along trajectories [34]. Typically local features are encoded following a bag-of-words approach such as [12, 14, 31, 34]. Although simple and relatively successful, these methods largely left the relationships among local features unexplored, and thus potentially omitted discriminative information contained in such relationships.

To better exploit local features and their relational information, many works have sought to encode or model space-time structures of local features. Some works explicitly model local feature neighborhoods, such as [6, 10, 15]. Kovashka *et al.* [10] construct higher-level features composed of local features and their neighborhoods. Class-specific distance functions are learned for defining neighborhoods so that they cover the most informative configurations. Gilbert et al. [6] group simple 2D corners spatially and temporally using a hierarchical process, and discriminative features are then mined at each stage of that hierarchy. In [15] quantized local features are augmented with relative space-time relationships between pairs of features. More recently, Ma *et al.* [14] proposed the hierarchical space-time segments (HSTS) that can extract both static and non-static relevant human body regions and exclude irrelevant background regions.

Another major branch of works, such as [17, 22, 29, 30, 37] learn structural models for human actions. For instance, Wang *et al.* [37] use a hidden conditional random field to model a human action as a constellation of parts, where the model parameters are learned in a max-margin framework. Raptis *et al.* [22] cluster dense trajectories and learn a graphical model per action where the nodes correspond to latent variables selecting appropriate dense trajectory clusters for an action. Some recent works propose to represent a video as a graph [2, 35] of local features and use graph kernel or graph matching techniques for action classification. For example, in [35] the nodes are clusters of dense trajectories and the edges represent temporal relationships, and the random walk graph kernel is used for training action classifier. In [2] the nodes are space-time subvolumes of video and the edges depict the space, time and hierarchical relationships among the subvolumes, and action classification is cast as a graph matching problem.

Approaches that use local features for action recognition typically extract a set of features from a video, while the feature set cardinalities can vary from video to video. To get fixed size feature vectors from videos to ease the learning step, e.g. SVM classifier learning, a feature pooling step is usually used. The straight forward approach is to pool features from the whole video [14, 31, 34], but the space time configuration of the local features is then lost. To encode the space and time configuration information among the local features, one popular approach is to pool features within pre-defined fixed sized space-time grids [12, 24]. In [12] space-time interest point descriptors are pooled within space-time grids that are arranged as a space-time pyramid, and in [24] action detector responses are pooled in grids of a similar pyramid. Simple temporal grids are also used [17, 29], where local features are pooled in temporal segments of a video. More recently, Fisher vector encoding method has been effectively applied as a feature encoding method for action recognition [8, 16, 18, 28, 31, 32, 36], but all these methods use fixed grids (or the whole video) for pooling the encoded Fisher vectors. Obviously, such fixed grids do not adapt to the content of the video, especially the spatial-temporal layout of the regions that contain the human action, so it is quite possible that significantly different feature vectors may be produced through feature pooling of videos that contain the same action.

Instead of using fixed grids, some methods perform feature pooling within space-time clusters of local features such as [5, 22]. In [22] dense trajectories are grouped according to their space-time overlaps and then feature pooling is performed on feature descriptors of group members. In [5] a hierarchical clustering is carried out on tracklets in a video and feature pooling is done in every cluster in the cluster tree to make a Bag-of-Words tree. Our pooling method is closely related to this line of research, but with the following three important distinctions. First, we cluster HSTS [14], which also preserves both the moving and static relevant regions within a video, while the clusters in [5, 22] are based on trajectories that focus on moving regions and relevant static regions may be missing. Secondly, in [5, 22] each feature is weighted equally during pooling, while we use HSTS to compute a pooling map that emphasizes the regions of human action and the features are weighted according to the map. In this way, the pooling result may better represent the human action contained in the video. Finally, our structured representation is not limited to be a tree and the node representation is not limited to histograms as in [5]. This flexibility allows us to model different kind of relationship among video regions. Moreover our approach does not need to learn a specific graph representation for each action as [17, 22, 29, 30, 37].

3 Adaptive structured pooling

In this section, we first formalize how to apply Fisher encoding with a weighting of the local features in 3.1. We then detail our strategy to obtain pooling regions and weights in 3.2.1. We rely on HSTS to define spatio-temporal pooling regions (STPR). Each pooling region defines a weighting map for all local features to be encoded in a video. We further define several granularities of STPR and a graph on top of them, thus yielding a structural representation of the whole video.

3.1 Fisher encoding with soft pooling

A Fisher vector representation requires a generative model for local features in a video. In our case we fit a Gaussian Mixture Model u_λ on a set of randomly sampled local features from videos. The Fisher kernel between two sets of local features X and Y is defined as $K_{\text{FV}} = G_\lambda^{X^\top} F_\lambda^{-1} G_\lambda^Y$, where F_λ is the Fisher information matrix of u_λ and G_λ^X is the gradient of the log-likelihood of the data X with respect to the parameters λ of the generative model $G_\lambda^X = \nabla_\lambda \log u_\lambda(X)$.

To obtain the Fisher vector representation one can apply the Cholesky factorization of $F_\lambda^{-1} = L_\lambda^\top L_\lambda$ and defining $\mathcal{G}_\lambda^X = L_\lambda G_\lambda^X$ we can write $K_{\text{FV}}(X,Y) = \mathcal{G}_\lambda^{X^\top} \mathcal{G}_\lambda^Y$. Assuming that descriptors in X are independent, the Fisher vector of a video X is a normalized sum of gradients at each point $\mathbf{x} \in X$ with respect to the model parameters λ:

$$\mathcal{G}_\lambda^X = \sum_{\mathbf{x}\in X} L_\lambda \nabla_\lambda \log u_\lambda(\mathbf{x}) \tag{1}$$

Fisher vectors are usually computed for a set of features from a whole video or image [18, 21]. The ideas of spatial pyramid (SPM) and spatio-temporal pyramid matching (STPM) can be easily adapted simply computing Fisher encoding over the subset of features that have coordinates inside the boundary of the pyramid sub-region. In [3] a weighted Fisher encoding approach has been proposed for object detection in image, where the weights are based on multiple segmentation outputs. In this work we formalize a flexible way of pooling for Fisher vectors that adapts to arbitrary spatio-temporal regions in video.

We obtain soft-pooling by computing a weight w_m for each feature $x_m \in X$ to encode. Given the GMM $u_\lambda = \sum_{n=1}^N \omega_n u_n(x;\mu_n,\sigma_n)$ and the M features of X, we compute for each component u_n the mean $\mathcal{G}_n^{\mu}(X)$ and covariance elements $\mathcal{G}_n^{\sigma}(X)$ of a Fisher vector as:

$$\mathcal{G}_n^{\mu}(X) = \frac{1}{\sqrt{\omega_n}} \sum_{m=1}^{M} w_m \gamma_n(x_m) \left(\frac{x_m - \mu_n}{\sigma_n} \right), \tag{2}$$

$$\mathcal{G}_n^{\sigma}(X) = \frac{1}{\sqrt{2\omega_n}} \sum_{m=1}^{M} w_m \gamma_n(x_m) \left(\frac{(x_m - \mu_n)^2}{\sigma_n^2} - 1 \right), \tag{3}$$

where $\gamma_n(x_m)$ is the posterior probability of the feature x_m for the component n of the GMM:

$$\gamma_n(x_m) = \frac{\omega_n u_n(x_m)}{\sum_{j=1}^{N} \omega_j u_j(x_m)}. \tag{4}$$

3.2 Spatially and temporally structured pooling of a video

In this section, we first present how we define pooling regions and corresponding weights from a set of HSTS in 3.2.1. We then detail our approach to obtain a structured representation of the video as a graph linking local pooling regions in 3.2.2.

3.2.1 Soft weighted pooling regions

Given a video, we extract a set of HSTS $\mathcal{S}$ via the method in [14], which comprises two major steps: hierarchical video frame segment extraction and video frame segment tracking.

In the first step, a hierarchical segmentation [1] is produced for each video frame using both motion and color channels [13]. After some initial pruning of oversized or undersized segments, a set of segment trees $\mathcal{T}^t$ is extracted from each video frame (t is the frame index). Each segment tree $\mathcal{T}_i^t \in \mathcal{T}^t$ is considered as a candidate segment tree of human body and we denote $\mathcal{T}_i^t = \{s_{ij}^t\}$ where each s_{ij}^t is a video frame segment. $\mathcal{T}^t$ is then pruned using shape, motion and color cues. Note that each $\mathcal{T}_i^t \in \mathcal{T}^t$ is either pruned altogether or retained with all its segments. For example, all segments in $\mathcal{T}_i^t = \{s_{ij}^t\}$ will be preserved as long as at least one segment s_{ij}^t contains motion. In this way, relevant but static segments can be maintained, e.g. the torso in a hand shaking action.

In a second step, every segment s_{ij}^t of all surviving segment trees of a video frame is tracked separately both forward and backward in time in the video sequence based on its motion and color. Finally, the space-time segment $\mathbf{s}_{ij}^t$ is the set of bounding boxes obtained from the tracking process. Since the number of segments may be large and many of them cover non-static body parts, the tracking procedure is designed to be efficient and allow non-rigid deformation of the target segment. Due to the extraction algorithm described above, the resulting HSTS set $\mathcal{S} = \{\mathbf{s}_{ij}^t \mid \forall\, t,\ i,\ j\}$ is dense with many overlapping space-time segments.

Given a set of HSTS $\mathcal{S}_k \subseteq \mathcal{S}$ we can define a weighted pooling map M_k by accumulating how many segments of $\mathcal{S}_k$ are present in each frame at each position. Formally, let us denote $\mathcal{S}_k^t$ all segments of $\mathcal{S}_k$ existing at frame t. For every pixel $p = (x,y)$ of frame t, we compute the corresponding pooling map value $M_k^t(p)$ as the count of segments enclosing this position:

$$M_k^t = \sum_{s \in \mathcal{S}_k^t} \Psi_s \tag{5}$$

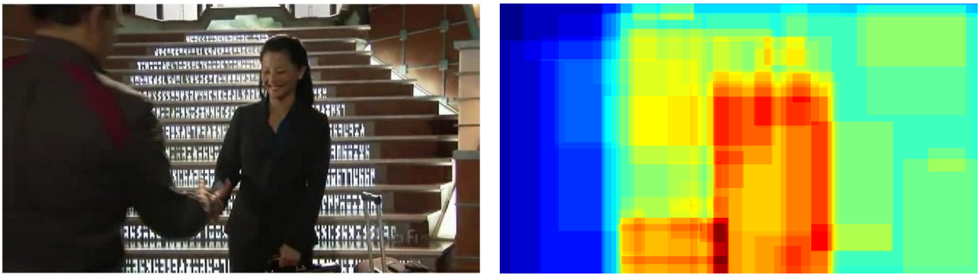

Figure 2: A frame of one HandShake action video from the HighFive dataset and its corresponding pooling map using all segments.

where for each segment $s \in \mathcal{S}_k^t$ we define the function of pixels p in an image

$$\Psi_s(p) \quad = \quad \begin{cases} 1 & \text{if } p \in s \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

The pooling map M_k^t is further normalized by the total number of segments in the frame and square-rooted. This pooling maps represent at any moment of the video, how much each pixel is relevant with respect to the set $\mathcal{S}_k$. The more segments overlap in one position the more likely this pixel is significant for the action taking place. A pooling map with all segments of the corresponding frame is depicted in Fig. 2. Finally, for a video with T frames we define the spatio-temporal pooling map as

$$M_k(x,y,t) = \left\{ M_k^1(x,y) \dots M_k^T(x,y) \right\} \tag{7}$$

For each local feature x_m to be encoded, we estimate the weight w_m^k with respect to set $\mathcal{S}_k$ as a small local integral of the pooling map M_k around its centroid. The parameters (v_x, v_y, v_t) are respectively the horizontal, vertical and temporal span of integration. That is for each $x_m \in X$ with the spatio-temporal coordinates of its centroid being $(x_{x_m}, y_{x_m}, t_{x_m})$, w_m^k is estimated as:

$$w_m^k \quad = \quad \int_{x_{x_m}-v_x}^{x_{x_m}+v_x} \int_{y_{x_m}-v_y}^{y_{x_m}+v_y} \int_{t_{x_m}-v_t}^{t_{x_m}+v_t} M_k(x,y,t) \, \mathrm{d}x \, \mathrm{d}y \, \mathrm{d}t \tag{8}$$

Finally, all weights of a pooling region are normalized to sum to one in order to have a comparable representation no matter how many features are present within the region.

3.2.2 Structured representation of videos

The previous section detailed how to obtain weights to encode local features from a given set of HSTS. We want to build a structured representation of each video. We therefore need to divide the whole set of segments into meaningful subsets. We propose to find coherent subsets by grouping together segments according to their overlap. This will create a set of local (both spatially and temporally) pooling regions.

We first compute an affinity matrix A of all segments $\mathcal{S}$ of a video. The affinity of two segments s_i (alive from frame t_{is} to t_{ie}) and s_j (alive from frame t_{js} to t_{je}) is computed as:

$$A(s_i, s_j) \quad = \frac{1}{\min(t_{ie}-t_{is}, t_{je}-t_{js})} \sum_{t \in [\max(t_{is}, t_{js}), \min(t_{ie}, t_{je})]} \frac{s_i^t \cap s_j^t}{s_i^t \cup s_j^t}. \tag{9}$$

The affinity between two segments will be equal to one only if a segment is temporally fully included in the other and with a perfect spatial overlap in every frame of coexistence. The affinity is zero if two segments do not overlap in any frame. Given this affinity matrix we run the normalized cuts algorithm [26] to obtain the subsets of segments.

Instead of choosing one fixed number of subsets, we propose to use multiple increasing sizes that will each provide a set of finer local representations of the video. We hence run the normalized cuts algorithm with different number of clusters (namely 4 and 16 subsets) to obtain multiple levels of representation. This setting is inspired by the spatial pyramid usually defined with corresponding numbers of cells. Given all the HSTS clusters obtained from the previous step (including the complete set of HSTS as the single cluster of the first level), we can build a graph representing the whole video. We represent each cluster as a node in the graph, and each node attribute is the soft pooling of dense trajectories features weighted by the map computed on all segments of this cluster. We link clusters based on their overlap, we create a link between all clusters that have at least a pair of overlapping segments (even partially). An illustration of one video graph is shown in Fig. 1.

To compare the video graphs we use the efficient GraphHopper kernel from [4]. The GraphHopper kernel is a scalable kernel that can deal with vector values for the nodes attributes. The comparison between two graphs relies on the similarity of the nodes and the computation of shortest paths in each graph. Formally, given two graphs $G = (V,E)$ and $H = (V',E')$ the GraphHopper kernel $K_{gh}(G,H)$ is decomposed as a sum of node kernels:

$$K_{gh}(G,H) = \sum_{v \in V} \sum_{v' \in V'} w(v,v') k_n(v,v') \tag{10}$$

where $w(v,v)$ counts the number of times v and v' appear at the same hop in shortest paths of equal discrete length and $k_n(v,v')$ is a kernel between the node attributes, as proposed [4]. In our experiments, $k_n(v,v')$ is a simple dot product between the soft-pooled Fisher vectors of v and v'.

4 Experiments

4.1 Experimental protocol

We test our approach on two widely-used action recognition benchmark datasets. The UCF Sports Dataset contains 150 videos and 13 classes. Experiments are typically conducted using either the split proposed in [11] or using leave-one-out (LOO). The three "Golfing" classes are merged in a single "Golf" class and the same is done for "Kicking" actions. Considering the smaller size of this dataset, we augmented the training data by flipping training videos as in [34]. Performance is reported as mean per class accuracy.

The HighFive dataset is a challenging dataset of human interactions. The dataset is collected from TV series and contains 4 classes plus a set of negative examples. The dataset contains 300 videos and experiments are performed following the split proposed by [19]. Results are reported in terms of mean average precision on the 4 classes of interest.

We employ features computed along improved dense trajectories [31]. We use Fisher vector coding for HOG, HOF, MBH and Trajectory (TR) features similarly to [18]. We apply PCA to each descriptor vector preserving 64 components for HOG, HOF, MBH and 20 for TR and train a GMM with 256 components on a random set of 200k samples for each of the five descriptors. Each graph node is represented with a soft-pooled Fisher vector, as described in Section 3.1, with L^2 and power normalization [21]. The parameters v_x, v_y and v_t are all set to 7, the rational for v_t being that dense trajectories are defined over 15 frames.

As a feature fusion strategy we adopt a simple kernel fusion approach. For each feature f we compute a graph matching kernel $K_f(G,H)$, comparing graphs G and H. Each kernel is normalized by applying:

$$K'_f(G,H) = \frac{K_f(G,H)}{\sqrt{K_f(G,G)K_f(H,H)}} \tag{11}$$

The final kernel, for an ensemble of features $\mathcal{F}$, is $K(G,H) = \sum_{f\in\mathcal{F}} K'_f(G,H)$. Finally, a one-vs-all SVM is learnt for each class using the fused kernel.

4.2 Results

Table 2 reports results on the UCF Sports dataset. On this easier dataset we get very strong results. Our FV baseline gets 89.4% (88.6% using LOO) accuracy while the structured representation obtains 90.8 % (90.4 % using LOO). Note that our approach does not rely on any person bounding box annotation.

Table 1 reports the results for the HighFive dataset. In this challenging dataset we obtain an improvement of 3% mean average precision (mAP). Note that our Fisher vector baseline (FVB) is very close to previous state-of-the-art but our structured representation improves over our FV baseline by 4%. We include results with our proposed method using only the MBH feature channel to get a fair comparison with [5, 14] that only exploit the MBH feature.

Our method performs better and requires less supervision than [19], which needs discriminatively trained head pose estimators and people detectors. Our approach also has an edge over the methods of [5] and [14], which also extract the spatio-temporal structure in the video. The method of Ma *et al.* [14] exploit space-time segments to sample features but discard spatio-temporal relationships. The video model proposed by Gaidon *et al.* [5] only captures hierarchical relationship in a top-down manner while our method models the relationship between video sub-volumes at different granularity. To the best of our knowledge these are the best results on these datasets.

Fig. 3 shows the top five ranked videos for each action on the HighFive dataset. The videos are represented by their central frame for simplicity. The top two videos are correct for all actions, while the top five videos are correct for both “highFive” and “hug”. The fourth video for the kiss action is actually a video of the hug action, but one can see that these two actions do share similar characteristics. Note also that this central frame may hide ambiguous elements happening before or after that frame within the video.

5 Conclusions

We have introduced a powerful and generic representation of videos for action recognition. Our structured representation is adaptive to the content of the video and does not rely on a fixed partition of neither space nor time. Our method only requires video level label annotation. Indeed we exploit an unsupervised procedure to generate a structured representation of the video. Our representation jointly models the hierarchical and spatio-temporal relationship of videos without imposing a strict hierarchy. Moreover, our method does not require a specific learnt graph model for each action.

Method	mean AP
Our (all features)	**65.4**
Our (MBH only)	62.8
FVB (all features)	61.3
Gaidon *et al.* [5]	62.4
Ma *et al.* [14]	53.3
Wang *et al.* [34]	53.4
Laptev *et al.* [12]	36.9
Patron-Perez *et al.* [19]	42.4

Table 1: Comparison with the state of the art on the HighFive dataset. Results are reported as mean average precision over the 4 classes.

Method	LOO	Split [11]
Our (all features)	**90.4**	**90.8**
Our (MBH only)	88.3	90.0
FVB (all features)	88.6	89.4
Lan *et al.* [11]	83.7	73.1
Kovashka *et al.* [10]	87.3	-
Klaser *et al.* [9]	86.7	-
Wang *et al.* [34]	85.6	-
Yeffet *et al.* [38]	79.3	-
Rodriguez *et al.* [23]	69.2	-
Wang *et al.* [35]	-	85.2
Ma *et al.* [14]	-	81.7
Raptis *et al.* [22]	-	79.4
Tian *et al.* [30]	-	75.2

Table 2: Comparison with the state of the art on the UCF Sports dataset. Results are reported as mean per-class accuracy over the 10 classes.

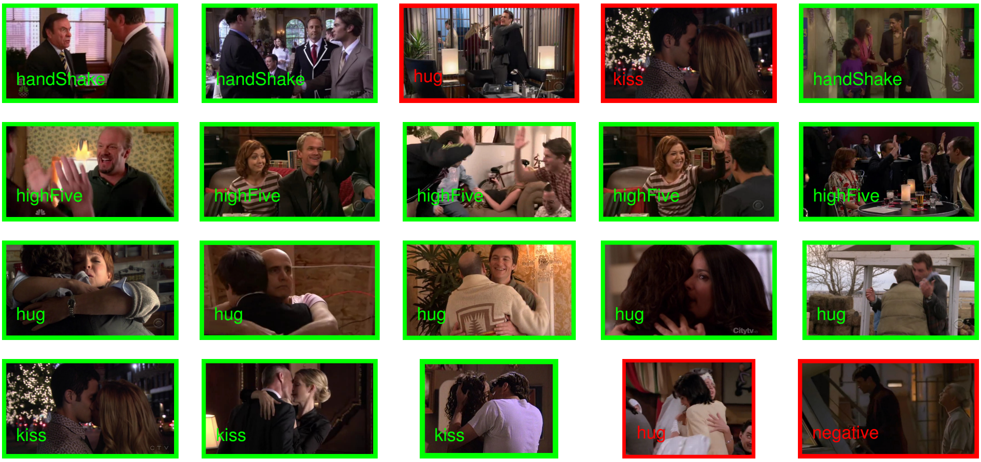

Figure 3: Top five ranked videos (leftmost is higher ranked) by our method for each action (handShake, highFive, hug and kiss) of the HighFive dataset.

Experiments conducted on two standard datasets for action recognition show a significant improvement over the state-of-the-art. In the future, we would like to see if our structured representation could also be used to solve the action localization problem by identifying the paths and/or nodes that are most relevant for the action.

References

[1] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. From contours to regions: An empirical evaluation. In *Proc of. Computer Vision and Pattern Recognition (CVPR)*, pages 2294–2301. IEEE, 2009.

[2] William Brendel and Sinisa Todorovic. Learning spatiotemporal graphs of human activities. In *Proc. of International Conference on Computer Vision (ICCV)*. IEEE, 2011.

[3] Ramazan Gokberk Cinbis, Jakob Verbeek, Cordelia Schmid, et al. Segmentation driven object detection with fisher vectors. In *Proc. of International Conference on Computer Vision (ICCV)*. IEEE, 2013.

[4] Aasa Feragen, Niklas Kasenburg, Jens Petersen, Marleen de Bruijne, and Karsten Borgwardt. Scalable kernels for graphs with continuous attributes. In *Advances in Neural Information Processing Systems*, pages 216–224, 2013.

[5] Adrien Gaidon, Zaid Harchaoui, and Cordelia Schmid. Activity representation with motion hierarchies. *International Journal of Computer Vision*, pages 1–20, 2013.

[6] Andrew Gilbert, John Illingworth, and Richard Bowden. Action recognition using mined hierarchical compound features. *Transactions on Pattern Analysis and Machine Intelligence*, 33(5):883–897, 2011.

[7] Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. *Transactions on Pattern Analysis and Machine Intelligence*, 29 (12):2247–2253, December 2007.

[8] Svebor Karaman, Lorenzo Seidenari, Andrew D Bagdanov, and Alberto Del Bimbo. L1-regularized logistic regression stacking and transductive crf smoothing for action recognition in video. In *THUMOS'13 Action Recognition Challenge*, 2013.

[9] Alexander Kläser. *Learning human actions in video*. PhD thesis, Université de Grenoble, jul 2010. URL http://lear.inrialpes.fr/pubs/2010/Kla10.

[10] Adriana Kovashka and Kristen Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *Proc. of Computer Vision Pattern Recognition (CVPR)*, pages 2046–2053. IEEE, 2010.

[11] Tian Lan, Yang Wang, and Greg Mori. Discriminative figure-centric models for joint action localization and recognition. In *Proc. of International Conference on Computer Vision (ICCV)*, pages 2003–2010. IEEE, 2011.

[12] Ivan Laptev, Marcin Marszałek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *Proc of. Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2008.

[13] Marius Leordeanu, Rahul Sukthankar, and Cristian Sminchisescu. Efficient closed-form solution to generalized boundary detection. In *Proc of. European Conference on Computer Vision (ECCV)*, pages 516–529. Springer, 2012.

[14] Shugao Ma, Jianming Zhang, Nazli Ikizler-Cinbis, and Stan Sclaroff. Action recognition and localization by hierarchical space-time segments. In *Proc. of International Conference on Computer Vision (ICCV)*. IEEE, 2013.

[15] Pyry Matikainen, Martial Hebert, and Rahul Sukthankar. Representing pairwise spatial and temporal relations for action recognition. In *Proc of. European Conference on Computer Vision (ECCV)*, pages 508–521, 2010.

[16] O.V. Ramana Murthy and Roland Goecke. Combined ordered and improved trajectories for large scale human action recognition. In *THUMOS'13 Action Recognition Challenge*, 2013.

[17] Juan Carlos Niebles, Chih-Wei Chen, and Fei-Fei Li. Modeling temporal structure of decomposable motion segments for activity classification. In *Proc of. European Conference on Computer Vision (ECCV)*, pages 392–405, 2010.

[18] Dan Oneata, Jakob Verbeek, and Cordelia Schmid. Action and Event Recognition with Fisher Vectors on a Compact Feature Set. In *Proc. of International Conference on Computer Vision (ICCV)*. IEEE, 2013.

[19] Alonso Patron-Perez, Marcin Marszalek, Ian Reid, and Andrew Zisserman. Structured learning of human interactions in tv shows. *Transactions on Pattern Analysis and Machine Intelligence*, 34(12):2441–2453, 2012.

[20] Florent Perronnin and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. In *Proc of. Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2007.

[21] Florent Perronnin, Jorge Sanchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *Proc of. European Conference on Computer Vision (ECCV)*, 2010.

[22] Michalis Raptis, Iasonas Kokkinos, and Stefano Soatto. Discovering discriminative action parts from mid-level video representations. In *Proc of. Computer Vision and Pattern Recognition (CVPR)*, pages 1242–1249. IEEE, 2012.

[23] Mikel D. Rodriguez, Javed Ahmed, and Mubarak Shah. Action mach: a spatio-temporal maximum average correlation height filter for action recognition. In *Proc of. Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008.

[24] Sreemananath Sadanand and Jason J. Corso. Action bank: A high-level representation of activity in video. In *Proc of. Computer Vision and Pattern Recognition (CVPR)*, pages 1234–1241. IEEE, 2012.

[25] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: A local svm approach. In *Proc. of International Conference on Pattern Recognition (ICPR)*, 2004.

[26] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Transactions on Pattern Analysis and Machine Intelligence*, 22:888–905, 2000.

[27] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proc. of International Conference on Computer Vision (ICCV)*, pages 1470–1477. IEEE, 2003.

[28] Chen Sun and Ram Nevatia. Large-scale web video event classification by use of fisher vectors. In *Proc. of Workshop on Applications of Computer Vision (WACV)*, pages 15–22. IEEE, 2013.

[29] Kevin D. Tang, Fei-Fei Li, and Daphne Koller. Learning latent temporal structure for complex event detection. In *Proc of. Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012.

[30] Yicong Tian, Rahul Sukthankar, and Mubarak Shah. Spatiotemporal deformable part models for action detection. In *Proc of. Computer Vision and Pattern Recognition (CVPR)*, pages 2642–2649. IEEE, 2013.

[31] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proc. of International Conference on Computer Vision*, pages 3551–3558. IEEE, 2013.

[32] Heng Wang and Cordelia Schmid. Lear-inria submission for the thumos workshop. In *THUMOS'13 Action Recognition Challenge*, 2013.

[33] Heng Wang, Muhammad Muneeb Ullah, Alexander Klaser, Ivan Laptev, Cordelia Schmid, et al. Evaluation of local spatio-temporal features for action recognition. In *Proc. of British Machine Vision Conference*, 2009.

[34] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, pages 1–20, 2013.

[35] Ling Wang and Hichem Sahbi. Directed acyclic graph kernels for action recognition. In *Proc. of International Conference on Computer Vision (ICCV)*. IEEE, 2013.

[36] Xingxing Wang, Limin Wang, and Yu Qiao. A comparative study of encoding, pooling and normalization methods for action recognition. In *Proc. of Asian Conference on Computer Vision (ACCV)*, 2012.

[37] Yang Wang and Greg Mori. Hidden part models for human action recognition: Probabilistic versus max margin. *Transactions on Pattern Analysis and Machine Intelligence*, 33(7):1310–1323, 2011.

[38] Lahav Yeffet and Lior Wolf. Local trinary patterns for human action recognition. In *Proc. of International Conference on Computer Vision (ICCV)*, pages 492–497. IEEE, 2009.