Action Recognition by Hierarchical Mid-level Action Elements Supplementary Material

S1. Action Proposals: Hierarchical Spatiotemporal Segments

In this section, we provide the details of our method for generating a hierarchy of action-related spatiotemporal segments from a video.

A. Generating the Foreground Segments. We initially generate a diverse set of region proposals using the method of [1]. The "region proposal" method works on a single frame of video, and returns a large number of segmentation masks that are likely to contain objects or object parts. For each frame in the video, we generate roughly 1000 region proposals.

In order to select foreground segments among the region proposals, we first use the method of [1] to select an initial set of foreground segments. Each region proposal is scored with both appearance and motion cues, and we look for regions that have generic object-like appearance and distinct motion patterns relative to their surroundings. We refer to [2] for the details of the scoring function. We take the top N highest-scoring region proposals in each frame to form an initial set of foreground segments. Given these candidate segments, we train a linear SVM classifier for each video independently, using these candidate foreground segments as positive examples along with randomly sampled from the background as negative examples. For simplicity, we only use color histogram as the feature descriptor for each segment. The segments with scores above a threshold (-1) are considered as foreground segments.

B. Obtaining the Spatiotemporal Segment Pool. Given the foreground segments for each frame, we seek to compute "tracklets" of these segments over time to construct the spatiotemporal segments. One straightforward way is to perform tracking on the segments. However, these segments may change appearance and shape drastically over time, may be poorly localized or disappear over the course of the video. Tracking algorithms tend to fail in these scenarios. Instead, we perform a spectral clustering on the foreground segments to produce a pool of spatiotemporal segments. We define the similarity between two segments s_i and s_j as:

$$K(s_i, s_j) = \exp(-d_{color}(s_i, s_j) - d_{shape}(s_i, s_j) - d_{xyt}(s_i, s_j)) \quad (1)$$

where $d_{color}(s_i, s_j)$ denotes the χ^2 distance between the color histograms of segments s_i and s_j. $d_{shape}(s_i, s_j) = 1 - \frac{1}{K^2}\sum m(s_i)|m(s_j)$ is the distance between the shape features $m(s_i)$, $m(s_j)$ of these two segments. For a segment s_i, we first extract a squared patch that most tightly bounds the segment and resize the patch to $K \times K$, then the shape feature $m(s_i)$ is defined as the binary mask of the segment in the $K \times K$ patch. Thus the shape feature is invariant to segment size. The distance metric $d_{shape}(s_i, s_j)$ denotes the percentage of non-overlapped area of the shape masks of the segments s_i and s_j. $d_{xyt}(s_i, s_j)$ is the euclidean distance between segments s_i and s_j in both space and time. We normalize each distance by the mean of the distances among all segments in the video.

For each video, we compute the pairwise affinities $K(s_i, s_j)$ between all pairs of segments in the video, to obtain the affinity matrix. Next we perform spectral clustering on the affinity matrix of each video independently to produce the pool of spatiotemporal segments. In order to maintain the purity of each spatiotemporal segment, it is important that we set the number of clusters to a reasonably large number. The pool of spatiotemporal segments correspond to the mid-level action elements (MAEs) at the finest scale.

C. Constructing the Hierarchy. Starting from the initial set of spatiotemporal segments, we agglomeratively group the most similar spatiotemporal segments into super-spatiotemporal segments until only a single super-spatiotemporal segment is left. In this way, we produce a hierarchy of spatiotemporal segments that forms a tree structure: the leaves are the initial set of fine-grained spatiotemporal segments, while the root node corresponds to the super-spatiotemporal segment at the last iteration. The internal nodes are produced by the "merge" operations.

Similar to Eq. (1), we define the similarity between two spatiotemporal segments v_i and v_j in order to decide whether they should be merged. We use the distance metrics d_{color} and d_{xyt} defined in Eq. (1) to measure the appearance and space-time distances between v_i and v_j. Here we

use the color histograms and space-time locations averaged over all regions in the spatiotemporal segment as features. Intuitively, this captures the intuition that the spatiotemporal segments to be merged are similar in appearance and close in space and time. We employ a different shape based distance metric $\hat{d}_{shape}$ to measure how well spatiotemporal segments v_i and v_j fit into each other. This distance metric is defined as: $\hat{d}_{shape} = \frac{1}{N^2}\sum_{(s_i,s_j)\in\{v_j,v_j\}} d_{wh}(s_i, s_j)$, where $d_{wh}(s_i, s_j)$ is the euclidean distance between the width and height of segments s_i and s_j, and N is the total number of segments in the spatiotemporal region merged from v_i and v_j.

The agglomerative clustering process might produce redundant spatiotemporal segments, i.e. the spatiotemporal segment in the parent node may heavily overlap with one in its child node. Thus we trim the tree to remove the redundant nodes: we remove the child node and connect the children of the removed node (if any) to the parent node.

S2. Generating Ground Truth Labels for Action Parsing

In this section, we provide the details of generating ground truth labels for the action parsing experiment on MPI Cooking dataset. Our goal is to generate a label hierarchy which contains actions and mid-level action elements (MAEs) at multiple levels of granularity for each video, see Fig. 1 for an example. In MPI Cooking, the fine-grained action labels along with the temporal extent of each label are provided for each video. We use these labels as the basic MAEs at the bottom level of the label hierarchy (i.e. the MAEs at the finest scale). Then we generate labels at the higher level of the hierarchy by recursively composing the finer-grained labels. For example in Fig. 1, the fine-grained labels "open fridge", "take out food" and "close fridge" are provided by the dataset. Then we build up the label hierarchy by recursively composing the finer-grained labels into higher level labels: e.g. "open fridge and take out food" and "open fridge, take out food and close fridge" are the automatically generated higher-level labels. Due to the large number of action categories, it is impossible to consider all possible combinations of the fine-grained action labels. Instead, we only consider higher action labels with length ranging from 2 to 4, and occurs in the training set for more than 10 times. In this way, we have in total 120 action and MAE labels for parsing evaluation.

The goal of action parsing is to predict the label hierarchy for a given video and localize each instance of the action and MAE labels. The visualizations of action parsing are shown in the paper as well as the supplementary video.

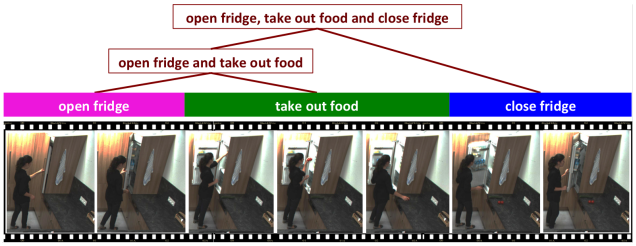

Figure 1. **Action parsing annotations.** This figure shows an example annotation for action parsing in MPI Cooking dataset. The bottom level of the label hierarchy are the fine-grained labels provided by the dataset. These labels are served as the basic action elements which are used to automatically generate the higher level labels in the hierarchy.

References

[1] I. Endres and D. Hoiem. Category independent object proposals. In *Computer Vision–ECCV 2010*, pages 575–588. Springer, 2010. 1

[2] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1995–2002. IEEE, 2011. 1