Unsupervised Human Action Detection by Action Matching

Basura Fernando* Sareh Shirazi† Stephen Gould*
*The Australian National University †Queensland University of Technology
firstname.lastname@anu.edu.au s.shirazi@qut.edu.au

Abstract

We propose a new task of unsupervised action detection by action matching. Given two long videos, the objective is to temporally detect all pairs of matching video segments. A pair of video segments are matched if they share the same human action. The task is category independent—it does not matter what action is being performed—and no supervision is used to discover such video segments. Unsupervised action detection by action matching allows us to align videos in a meaningful manner. As such, it can be used to discover new action categories or as an action proposal technique within, say, an action detection pipeline. Moreover, it is a useful pre-processing step for generating video highlights, e.g., from sports videos.

We present an effective and efficient method for unsupervised action detection. We use an unsupervised temporal encoding method and exploit the temporal consistency in human actions to obtain candidate action segments. We evaluate our method on this challenging task using three activity recognition benchmarks, namely, the MPII Cooking activities dataset, the THUMOS15 action detection benchmark and a new dataset called the IKEA dataset. On the MPII Cooking dataset we detect action segments with a precision of 21.6% and recall of 11.7% over 946 long video pairs and over 5000 ground truth action segments. Similarly, on THUMOS dataset we obtain 18.4% precision and 25.1% recall over 5094 ground truth action segment pairs.

1. Introduction

Recognizing human activities in unconstrained videos is important for many applications including human computer interaction, human robots interaction, sports video analysis, video retrieval, storyline reconstruction and for many other video analysis tasks [23]. However, it is hard to define what a *human action is*. In the current literature, human actions are defined based on tasks such as cutting, washing [25], based on specificity and regularity of human motion such as running, walking, hand waving [27], or based on sports activities such as weight lifting, skying or cricket bowling [30]. Moreover, current methods in human action recognition require a lot of supervised data [18]. Human action detection is the task of temporally localizing a human action within a long video [25]. Obtaining ground truth labels for human actions in video collections is costly and consequently annotated large high quality video datasets are hard to come by. Unlike action classification, which just requires a single label for the entire video sequence, to create an action detection dataset the annotator must watch the entire video and mark the beginning and end of each human action. Such manual annotations could be wrong, subjective and highly ambiguous. But action detection methods require a lot of annotations to supervise training of action detectors. As such, a more efficient method is needed.

Figure 1: Unsupervised action detection by action matching. Two videos share a common human action *throw the ball*. Objective is to temporally localize the common human action units (segments) within a pair of videos.

In this paper, we present a method to discover common action segments from a pair of videos based on human action matching in *unsupervised manner*. We call this novel task *unsupervised action detection by action matching*. We match segments of one video with the other one such that matched segments are from the same human action category. The method must recognize similar human action segments while temporally localizing the actions (i.e., detection) without using any external information (see Figure 1). As the task is unsupervised, our method does not know the type of detected action, only that one is occurring. This task extends the ASLAN [19] challenge where the task was to predict whether a pair of videos contains the same action or not. However, in the proposed task, the pair of videos

may contain a series of human actions and a large number of matched action segments.

Unsupervised action detection by action matching is useful for many applications. Obviously, the output of this task can be used to discover human action categories in unsupervised manner, which could be subsequently labeled. For example, we can cluster the large number of matched human action segments from unsupervised action detection to discover human action categories similar to unsupervised object discovery in static images [33]. The proposed task is also useful for early human action and activity prediction [26]. Imagine a robot that has access to a large video archive containing many human actions and tasks (such as cooking a meal, fixing a table, and cleaning a garage). Now whenever the robot sees a particular sequence of human actions in a live video stream, it is able to align common actions in the live stream with the videos in the archive. This would allow the robot to temporally localize the human activities without supervision. Moreover, the robot could anticipate future human actions by aligning what it has seen with the archive without any annotations. Another application of the proposed task is in video highlights generation of, say, sports activities without human intervention [14]. Imagine a system with access to an archive of sports highlights videos. Given a full sports coverage video, the system can generate new highlights by aligning the highlight videos with full coverage video through detection and alignment of human actions. Since highlights only capture interesting events, we can exploit such prior information to generate new video sports highlights. Yet another application for the proposed task is to generate candidate temporal action proposals similar to object proposal methods that are currently popular for object detection in static images [2].

In this paper we propose a novel unsupervised action detection task and propose a very effective yet simple method to solve the problem by human action matching. The related task of action classification is a well studied with much progress being made over the last decade [8, 16, 20, 32, 35, 28, 22]. However, action detection is a relatively new task that has shown great promise in recent years [25, 21, 29, 36, 24]. Action detection is challenging as the duration of the action is varied and unknown, and the background context can easily confuse the action classifiers. Furthermore, video computation is expensive so very effective temporal encoding methods are needed. In supervised action detection, one can rely on a large source of annotated videos of human actions to obtain discriminative temporal encodings using discriminative sequence encoders [5]. As the proposed task is unsupervised, however, we must rely on unsupervised video temporal dynamic encoding techniques such as unsupervised LSTMs [31] or rank pooling [11]. Due to its demonstrated effectiveness, we use a variant of rank pooling with the addition of temporal consistency for the task of *unsupervised action detection*. The naive approach of simply matching all possible video segments will not produce good results due to large number of false positives and deficiencies in the matching function. At the same time such an approach would not be able to take advantage of temporal consistency and smoothness in the execution of human actions. Moreover, it is highly inefficient. To overcome these drawbacks, we present a simple yet effective algorithm that exploits the temporal smoothness and consistency of human action evolution. Results of our method are reported on three activity recognition benchmarks.

2. Related work

Video alignment and video synchronization is related to the proposed task. However, except one instance we could not find any prior work in video synchronization or alignment that uses semantics such as human actions to align videos [34]. Most prior related work align pair of videos both in temporal and spatial domain without considering the action semantics [4] or human action dynamics. This is mostly done by sequence to sequence matching via frame correspondences. Frames content should be matched and correspondences should be found. Most prior work exploit geometric and photometric properties of the two scenes to find the correspondences in space and time [4]. For example, in Diego *et al*. [4] the paper assumes that the motion in two videos are somewhat similar and there is some overlap between field of view of cameras. Similarly, in Ukrainitz and Irani [34] alignment is performed in space and time by maximizing the local space-time correlations directly using the pixel intensity information. They seek a transformation that minimizes the spatial-temporal displacement of near identical pair of videos. Cross-view action recognition by exploiting the self similarity of videos is also somewhat related to our work [17]. However, compared to other related methods, ours does not rely on geometric or photometric properties of pair of matching videos. We only rely on encoding of human motion dynamics. In contrast to these prior work we exploit the temporal consistency and smoothness of information evolution of human actions. Furthermore, our task is an action detection task. To the best of our knowledge, unsupervised action detection is a novel task.

Supervised action detection is also related to ours [25, 21, 29, 36, 24]. Most of the progress in action detection is thanks to two main stream action detection datasets; the THUMOS challenge [13] and the MPII cooking activity dataset [25]. Rohrbach *et al*. [25] perform action detection using dense trajectory features encoded with bag-of-words and then applying simple temporal pooling method such as sum-pooling followed by SVM classifiers. They use a sliding window method. Joint exploitation of geometrical contextual information among objects, human body parts, body poses is used for action detection using LSTM in Ni *et*

al. [21]. A method for fine-grained action detection in long video sequences based on a multi-stream bi-directional recurrent neural networks was presented in Singh *et al*. [29]. Reinforcement learning based action detection method that utilizes recurrent neural networks has also been studies [36]. Different from all above methods, ours is an unsupervised action detection task where we have to temporally localize similar human actions in two long video sequences.

Recent work on temporal action proposal is also related to our work [6]. However, most of these methods are supervised. Output of our method can be used for temporal action proposals in an unsupervised manner and is agnostic to the action category.

Our method is also related to rank-pooling based action recognition [1, 3, 9, 10, 11, 12]. To the best of our knowledge, we are the first to use rank-pooling based dynamic encoding for an action detection task.

3. Unsupervised action detection

In this section we formalize the action matching problem (§3.1), provide an overview of our proposed solution (§3.2), discuss our method for sequence encoding (§3.3), and give details of how these come together to form a complete algorithm for *unsupervised action detection by action matching* (§3.4 and 3.5). Finally, we present two strong baseline methods (§ 3.6) that we compare against in the experiments.

3.1. Problem formulation

Given a pair of video sequences (X_a, X_b) where $X_a = \langle \boldsymbol{x}_1^a, \boldsymbol{x}_2^a, \ldots, \boldsymbol{x}_n^a \rangle$ and $X_b = \langle \boldsymbol{x}_1^b, \boldsymbol{x}_2^b, \ldots, \boldsymbol{x}_m^b \rangle$, we want to identify common human actions and localize them in each sequence. Many of the video frames may not belong to any human action class and we denote these by the special background label $\diamond$. Let us denote the set of human action categories that we care about by $\mathcal{Y}$. Then for each video $X = \langle \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n \rangle$, there is a corresponding label sequence $Y = \langle y_1, y_2, \ldots, y_n \rangle$ where $y_t \in \mathcal{Y} \cup \{\diamond\}$ is the label for the t-th frame in the sequence. An action unit of video X is a contiguous subsequence of X that contains frames of only a single action class from the action label set $\mathcal{Y}$. An action unit u is called maximal if frames adjacent to those in u take a different action label from $\mathcal{Y} \cup \{\diamond\}$. An action unit u_a from video X_a is matched to action unit u_b from video X_b if the action class label of u_a is same as action class label of u_b. Note that each action unit is valid if the union over intersection between an action unit and any ground truth is greater than some threshold (IoU = 0.5). The goal of our task is to find matching pairs of *valid maximal action units* from a pair of arbitrary long videos.

3.2. Overview of proposed solution

Given a collection of videos, first we extract frame-level CNN features from them. With slight abuse of notation we denote the sequence of vectors for the i-th video as $X_i = \langle \boldsymbol{x}_1^i, \boldsymbol{x}_2^i, \ldots, \boldsymbol{x}_n^i \rangle$. We then sample subsequences of length l_w and stride l_s and apply temporal encoding to each subsequence. For every pair of videos from the collection we construct a similarity matrix between subsequences from the first and second video, respectively. Next, we exploit the temporal consistency of activities to obtain candidate action unit pairs. Last, we apply non-maximal suppression to remove redundant candidate action unit pairs.

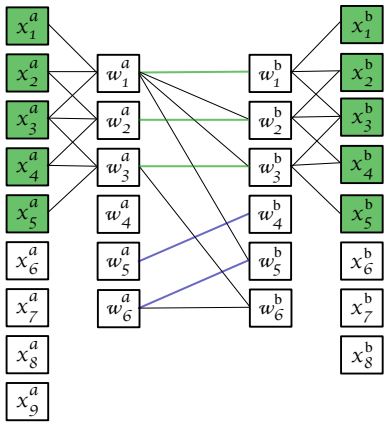


Figure 2: Unsupervised action detection by action matching using temporally constant matching. Frames are denoted by x_t^a for the first video and by x_t^b for the second video. Then subsequences of size three are temporally encoded to obtain vectors w_t^a and w_t^b. Afterwards, these temporal vectors are matched using bipartite graph matching. Matching temporal encodings from two videos are connected with an edge. Temporally consistent edges are shown in green and blue edges while isolated edges are shown in black. Each temporal encoding propagates the matches to frames as shown by the green coloured boxes. This way we can find the matching video segments that share similar temporal evolutions.

3.3. Temporal encoding of video segments

Temporal encoding takes an arbitrary long subsequence and represents it by a fixed length vector. Recently rank pooling was introduced as an effective and efficient method for temporal encoding of human actions [11]. The first step applied by Fernando *et al*. [11] is to smooth the frame-level features by time varying means.

Given an input sequence $X = \langle \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n \rangle$, the time varying mean at frame t is given by $\boldsymbol{m}_t = \frac{1}{t} \sum_{\tau=1}^{t} \boldsymbol{x}_\tau$. The method then normalizes the vector $\boldsymbol{m}_t$ and only looks at the direction of the evolution of the mean. Let us denote this preprocessed sequence by $V = \langle \boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n \rangle$,

where each element $\boldsymbol{v}_t$ is given by

$$\boldsymbol{v}_t = \frac{1}{\|\boldsymbol{m}_t\|}\boldsymbol{m}_t \tag{1}$$

Note that $\boldsymbol{v}_t$ captures only the direction of the unit mean appearance vector at time t. Rank pooling then takes this pre-processed sequence V and models the evolution of data over time using a linear ranking objective [11] as follows:

$$\boldsymbol{w}^\star \in \underset{\boldsymbol{w}}{\operatorname{argmin}} \left\{ \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C}{2}\sum_{t=1}^{J}\left[|t - \boldsymbol{w}^\top \boldsymbol{v}_t| - \epsilon\right]^2_{\geq 0} \right\} \tag{2}$$

As the parameter vector $\boldsymbol{w}^\star$ models the evolution of appearance information within the video, it also captures the temporal structure that can be used to represent the dynamics effectively. Such representations are robust and efficient to compute with stand packages such as LibLinear [7]. In the original rank-pooling method [11] non-linearity is introduced before the temporal encoding and after temporal encoding using a point-wise non-linear function $\Psi(\cdot)$, such as the signed square root. In summary, the steps of temporal encodings proposed by Fernando *et al*. [11] for supervised action classification is shown in the following equation:

$$X \overset{\boldsymbol{m}_t = \frac{1}{t}\sum_{\tau=1}^{t}\boldsymbol{x}_\tau}{\longmapsto} M \overset{\boldsymbol{v}_t = \frac{\boldsymbol{m}_t}{\|\boldsymbol{m}_t\|}}{\longmapsto} V \overset{\Psi(\boldsymbol{v}_t)}{\longmapsto} \widetilde{V} \overset{\Phi(\widetilde{V})}{\longmapsto} \boldsymbol{w} \overset{\Psi(\boldsymbol{w})}{\longmapsto} \widetilde{\boldsymbol{w}} \overset{\frac{\widetilde{\boldsymbol{w}}}{\|\widetilde{\boldsymbol{w}}\|}}{\longmapsto} \widetilde{\boldsymbol{w}^*} \tag{3}$$

To simplify the pre-processing pipeline in this work we remove the non-linear transformations before and after rank pooling. Our pipeline for generating a temporal encoding of a video sequence is then:

$$X \overset{\boldsymbol{m}_t}{\longmapsto} M \overset{\boldsymbol{v}_t = \frac{\boldsymbol{m}_t}{\|\boldsymbol{m}_t\|}}{\longmapsto} V \overset{\Phi(V)}{\longmapsto} \boldsymbol{w} \overset{\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}}{\longmapsto} \boldsymbol{w}^* \tag{4}$$

For long sequences the time varying mean smoothing applied by Fernando *et al*. [11] could be problematic in that smoothing is dominated by early frames. In this paper, we investigate an alternative scheme to obtain the smoothed vector $\boldsymbol{m}_t$. Specifically, we investigate the following ARMA model in addition to the time varying mean,

$$\boldsymbol{m}_t = \alpha \boldsymbol{m}_{t-1} + (1-\alpha)\boldsymbol{x}_t \tag{5}$$

where we set $\boldsymbol{m}_0$ to $\boldsymbol{x}_1$.

3.4. Temporal gram matrix construction

Given a pair of long videos X_a and X_b, we generate subsequences of length l_w with a stride of l_s. Let the number of subsequences be denoted by A and B for X_a and X_b, respectively. To avoid ambiguity in the sequel we refer to the subsequences of video frames as video segments. For each video segment we extract frame-wise features and apply rank pooling to obtain a fixed-length temporal encoding for each of the segments using the process explained in Equation 4. The obtained ordered set of temporally rank pooled vectors is then $\langle \boldsymbol{w}_1^a, \boldsymbol{w}_2^a, \ldots, \boldsymbol{w}_A^a \rangle$ for video X_a and $\langle \boldsymbol{w}_1^b, \boldsymbol{w}_2^b, \ldots, \boldsymbol{w}_B^b \rangle$ for video X_b. We can now construct a similarity matrix (Gram matrix) G of size $A \times B$ by taking the inner-product between each pair of segments from videos X_a and X_b as

$$G_{i,j} = \boldsymbol{w}_i^a \cdot \boldsymbol{w}_j^b, \qquad \begin{matrix} \forall i \in \{1,\ldots,A\} \\ \forall j \in \{1,\ldots,B\} \end{matrix} \tag{6}$$

3.5. Exploiting the temporal consistency

We generate candidate action unit pairs (u_a, u_b) by exploiting the temporal consistency and smoothness properties of the human action evolution. To this end, we use the Gram matrix constructed in the previous step and process it to find the top candidates. We select matched segments which have a similarity score greater than some positive threshold T. In our experiments we set T to the one standard deviation above the mean of elements of G. If T is non-positive we declare no matching segments. We then find the top temporally consistent candidate matches via a simple search algorithm, which we describe below.

Note that the problem of finding the largest common subsequence in two sequences of length A and B has a computational complexity of $O(A^2B^2)$, which is prohibitively expensive. Furthermore, we are interested in finding the top K best matches, not just the single best match. To alleviate the computational cost, we reduce the search space by employing some heuristics. We formalize the problem of identifying action unit pairs as a bipartite graph matching problem with temporal consistency constraints as show in Figure 2. We assume that the graph is sparse (via discarding edges with weight less than our threshold T) and that the matched subsequences of video segments have the same length and satisfy the temporal consistency (i.e., are in the same order). For example, the subsequence $\boldsymbol{w}_i^a, \ldots, \boldsymbol{w}_{i+k}^a$ matches the subsequence $\boldsymbol{w}_j^b, \ldots, \boldsymbol{w}_{j+k}^b$ if and only if the inner-product $\boldsymbol{w}_{i+p}^a \cdot \boldsymbol{w}_{j+p}^b \geq T$ for all $p \in \{0,\ldots,k\}$.

Our proposed algorithm is shown in Algorithm 1. Given sequences of encoded video segments W_a and W_b for videos X_a and X_b, respectively, the algorithm finds all runs of matching action unit candidates containing more than L video segments ($L > 1$). Such a strategy allows us to obtain longer action units, i.e., beyond the size of the original window size l_w, without resorting to a multiple scale strategy (with multiple window sizes) as commonly done in the supervised action detection literature [25, 36].

Once we obtain set of candidates we use non-maximum suppression to get rid of the redundant candidates. Two pairs of candidate matching action units are considered redundant if they overlap with more than 0.5 IoU. To be precise, let u_1^a and u_2^a be two sequence of video segments from video X_a and let u_1^b and u_2^b be two sequences of video segments from video X_b. Let us assume that pairs (u_1^a, u_1^b)

Input: Sequence $W_a = \langle \boldsymbol{w}_1^a, \ldots, \boldsymbol{w}_A^a \rangle$
Input: Sequence $W_b = \langle \boldsymbol{w}_1^b, \ldots, \boldsymbol{w}_B^b \rangle$
Input: Minimum match length L and threshold $T > 0$
Output: Candidate detections J
Construct gram matrix G as $G_{i,j} = \boldsymbol{w}_i^a \cdot \boldsymbol{w}_j^b$;
Initialize candidate detection graph J ($J_{i,j} = 0, \forall i, j$);
for $i \leftarrow 1$ **to** $A - L$ **do**
  // find all $\boldsymbol{w}_j^b$ that match $\boldsymbol{w}_i^a$
  $C_0 = \{j \in \{1, \ldots, B\} \mid G_{i,j} > T\}$;
  for $k \leftarrow 1$ **to** $L - 1$ **do**
    // find all $\boldsymbol{w}_j^b$ that match $\boldsymbol{w}_{i+k}^a$
    $C_k = \{j \in \{1, \ldots, B\} \mid G_{i+k,j} > T\}$;
    // remove temporallly inconsistent matches from C_0
    $C_0 = C_0 \cap (C_k - k)$;
  end
  // update candidate detection graph
  for $k \in \{1, \ldots, L\}$ **and** $c \in C_0$ **do**
    $J(i + k - 1, c + k - 1) = G(i + k - 1, c + k - 1)$;
  end
end

Algorithm 1: Candidate generation of action unit matches with temporal consistency.

and (u_2^a, u_2^b) are matching. Then these two pairs of matching action units are redundant if u_1^a and u_2^a overlap with IoU greater than 0.5 and u_1^b and u_2^b overlap with IoU greater than 0.5. In such situations we keep only the pair with highest matching score. The matching score of a candidate action unit pair (u_*^a, u_*^b) is the sum of scores of all matched segments (of size l_w) within that candidate action unit pair, i.e., $\sum_{\boldsymbol{w}^a, \boldsymbol{w}^b \in (u_*^a, u_*^b)} \boldsymbol{w}^a \cdot \boldsymbol{w}^b$.

3.6. Baselines methods

In this section we present two baseline methods that utilizes rank-pooling for comparison.

Clustering method: Given each video $X = \langle \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n \rangle$, we first generate video segments as above with length l_w with a stride of one. Then we temporally encode each segment using approximate rank pooling [1] without using any non-linear operations on the input data. Approximate rank-pooling is used due to its computational efficiency (it has constant time complexity). Let us denote the sequence of temporally encoded output vectors by $V = \langle \boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{n-l_w+1} \rangle$. To obtain segments with similar dynamics, we cluster each of the sequences V into k clusters using k-means. The goal is to find clusters that are, in fact, temporally meaningful and valid temporal segments. To further enforce this, we use the following simple trick. We modify each element of the sequence $V = \langle \boldsymbol{v}_1, \boldsymbol{v}_2, \ldots \rangle$ such that $\boldsymbol{v}_t^{\text{new}} = (\boldsymbol{v}_t, \beta t)$. This way, we obtain clusters that are correlated in time as well as being dynamically similar. Any cluster that does not have more than some given number of frames (in our case 60) are pruned. Afterwards, we encode each of the valid clusters with temporal rank pooling using Equation 4. Each of the clusters is now time coherent and results in temporally encoded subsequences which we use as candidate action units.

Given a pair of videos, we create such clusterings, one per video and match the clusters of two videos using temporal rank-pooled encoding of segments. The matching is done using cosine similarity. All pairs of clusters having temporal cosine similarity greater than some threshold (0.2 in our experiments) is considered a matching candidate pair for the final evaluation. We keep the top k such detections as candidates.

Rank pooling-based matching: In our second baseline, we select a window size l_w and a stride l_s and temporally rank pool each video segment starting from $t = 1$ up to $t = n, m$ (i.e., to the end of the video). This result in a sequence of rank pooled features $\langle \boldsymbol{w}_1^a, \ldots, \boldsymbol{w}_{n-l_w}^a \rangle$ and $\langle \boldsymbol{w}_1^b, \ldots, \boldsymbol{w}_{m-l_w}^b \rangle$. As before we construct the gram matrix using using Equation 6. We then keep only the pairs of matched sub-sequences having cosine similarity greater than some threshold (again, 0.2 in our experiments) as candidate detections. This method is equivalent to our proposed algorithm when the minimum matched sequence length L is set to zero.

4. Experiments

In this section we report results from extensive experiments on action detection by action matching. We start by outlining our evaluation criteria and datasets used.

4.1. Evaluation criteria

Given a pair of videos from the ground truth temporal annotation we obtain the start and end of each action. Each video may contain more than one instance of human action. Therefore, there can be more than one matching pairs of ground truth action units. For example, let us assume that there are N_a ground truth action units from action class y in video X_a and N_b number of ground truth action units from the same action class in video X_b. Therefore, there are $N_a \times N_b$ matching pairs of ground truth action units. Then a perfect method would be able to detect all of them within a specific IoU threshold (0.5) as explained in Section 3.1. To evaluate algorithms we use precision, recall and F1-score. During evaluations, we ignore all redundant candidates (action unit pairs) and keep only a single best candidate (action unit pair) per ground truth pair. This is done only at the evaluation. Candidate generation algorithm has no access to temporal annotations. For a given pair of videos, and generated candidate pairs of action units, the precision, recall and F1-score is define as follows:

$$P = \frac{\#\text{ of correct candidate action unit pairs}}{\#\text{of candidates generated}} \times 100 \quad (7)$$

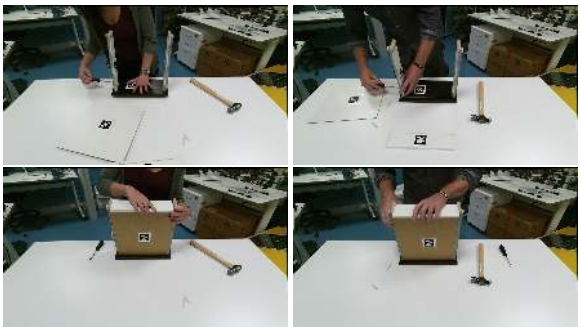


Figure 3: Matched examples from IKEA dataset

$$R = \frac{\#\text{ of correct candidate action unit pairs}}{\#\text{ of ground truth action unit pairs}} \times 100 \quad (8)$$

$$F_1 = \frac{2PR}{P+R} \quad (9)$$

4.2. Datasets

THUMOS dataset [13]: THUMOS'15 dataset includes four parts: training data, validation data, background data and test data. The training data is based on the UCF101 [2] action dataset, where videos are temporally trimmed (each video usually contains one instance of the action without irrelevant frames). A subset of 20 action classes out of 101 is employed for this task. The training videos are not useful for our task. We use the validation and test set of the THUMOS'15 dataset. Both the validation and the test sets consist of temporal annotations (start and end time) of all instances of the actions occurring in the validation videos. Altogether, there are 412 videos belonging to 20 action classes. Some videos contain multiple human action classes. Some pairs of videos do not contain any matching human action. We ignore any such pairs of videos from the evaluations. There are 6325 temporal annotations in this dataset. We ignore any pair of videos that has only a single action unit (ground truth detection). Finally, we end up with 5094 ground truth detection pairs over all 412 videos.

MPII Cooking dataset [25]: This dataset contains 65 different cooking activities, such as cut slices, pour spice, etc., recorded from 12 participants. In total there are 44 videos with a total length of more than 8 hours or 881,755 frames. The dataset contains a total of 5,609 annotations of 65 activity categories. Since each video is very long we use all possible pairs during the evaluation. Therefore, there are 946 total number of video pairs for evaluation.

IKEA dataset: This dataset contains 20 sequences of different people assembling the same IKEA drawer. Each sequence consists of approximately 300-400 frames. The viewpoint of the camera for each sequence is approximately the same. The camera is facing the work bench. The dataset and the annotations we used in our experiments will be available.

4.3. Feature and frame representations

For MPII Cooking activities dataset, we use provided dense trajectory features encoded with bag-of-word. We use HOG, HOF, MBH based trajectory features quantized into 4000 visual words as the frame representations [25]. For THUMOS dataset, we use the residual network features [15] (152-layer network). For IKEA dataset, we evaluate our approach using similar features to the MPII Cooking activities dataset in addition to the 152-layer residual network features.

Baseline 1 details: We apply approximate rank pooling on input sequences with window size (l_w) of 61 (21 for IKEA dataset) and stride 1. Afterwards, we L2 normalize the temporally pooled sequences. We cluster each sequence into 10 clusters. Experimentally we found that 10 clusters is reasonable as it generates roughly 100 candidate action detections. Before clustering step, each temporal pooled vector at time t is concatenated with the time variable such that the new vector $\boldsymbol{v}_t^{\text{new}} = (\boldsymbol{v}_t, \beta t)$ where $\beta = 0.001$. Then we keep candidate temporal segments (clusters) if they are temporally consistent and longer than 60 frames (20 frames for IKEA dataset). Given a pair of videos we have n temporal segments for the first video and m segments for the second and find candidates as explained in section Section 3.6.

4.4. Results

First, we report results for two baseline methods and proposed effective temporal consistency method using Cooking activities dataset in Table 1, IKEA dataset in Table 2 and THUMOS15 dataset in Table 3. For Cooking activities and THUMOS15 datasets, we use window sizes of 61 (stride 10) and minimum match length L of size $L = 10$, and use top 100 candidate detections for evaluation.

Results in Table 1 suggest that the best individual feature is MBH (F1 score of 14.1). Interestingly, the second best feature is the HOG feature. Most interestingly, the temporal consistency method improves over other two baseline by significant margin in terms of F1-score. For MBH features, the clustering method obtains F1 score of 4.4, rank pooling based matching obtains 4.8 while the temporal consistency method improves results to **14.1**. Similar trends can be observed for both IKEA dataset and the challenging THUMOS15 dataset. The temporal consistency method outperforms other two baselines over all three datasets using both trajectory features as well as deep residual network features. In some instances, for MPII Cooking activities dataset, the improvement of the temporal consistency method is more than twice other methods in terms of F1 score. Interestingly, for IKEA dataset the rank pooling matching obtains significant precision values. Perhaps this is because this is

Method	Rc. (%)	Pr. (%)	F1 (%)
HOG			
Cluster method	**17.4**	4.1	6.6
Rank pooling-based matching	3.8	3.1	3.4
Temporal consistency	14.1	**13.7**	**13.9**
HOF			
Cluster method	7.2	2.2	3.4
Rank pooling-based matching	3.2	2.8	3.0
Temporal consistency	**9.5**	**9.6**	**9.6**
MBH			
Cluster method	10.7	2.8	4.4
Rank pooling-based matching	5.3	4.4	4.8
Temporal consistency	**14.2**	**14.0**	**14.1**
Fused			
Cluster method	**13.1**	5.4	7.6
Rank pooling-based matching	7.2	6.4	6.8
Temporal consistency	11.7	**21.6**	**15.1**

Table 1: Unsupervised action detection results for MPII Cooking dataset.

Method	Rc. (%)	Pr. (%)	F1 (%)
HOG			
Cluster method	2.2	2.4	2.1
Rank pooling-based matching	2.1	**57.4**	4.0
Temporal consistency	**14.2**	17.5	**15.7**
HOF			
Cluster method	3.0	5.1	3.6
Rank pooling-based matching	2.1	**58.2**	4.0
Temporal consistency	**22.5**	24.9	**23.6**
MBH			
Cluster method	0.9	1.0	0.9
Rank pooling-based matching	2.1	**58.1**	4.0
Temporal consistency	**17.8**	22.9	**20.0**
Deep			
Cluster method	0.15	0.13	0.14
Rank pooling-based matching	2.7	**51.1**	5.0
Temporal consistency	**19.6**	24.6	**21.1**
Fused			
Cluster method	3.8	3.6	3.5
Rank pooling-based matching	2.1	**58.3**	4.1
Temporal consistency	**25.3**	24.9	**25.1**

Table 2: Unsupervised action detection results for IKEA dataset.

a relatively small dataset.

We also report results by fusing the HOG, HOF and MBH features. HOG, HOF and MBH features are fused using the average gram metric (similar to average Kernel) for Rank pooling-based matching and temporal consistency methods. Other fusion methods such as merging the candidates from different features did not improve our results perhaps because such a strategy would not be able to exploit the advantages of temporal consistency. For clustering method, we use early fusion (concatenation of feature vectors). Even after the fusion, we see that temporal consistency method is effective. Therefore, we conclude that rank pooling based temporal consistency method is useful for unsupervised action detection by action matching task.

Method	Rc. (%)	Pr. (%)	F1 (%)
Cluster method	12.5	**17.6**	14.6
Rank pooling-based matching	11.5	4.8	6.7
Temporal consistency	**24.2**	16.3	**19.5**

Table 3: Unsupervised action detection results for THUMOS15 dataset.

Method	HOG	HOF	MBH	Resnet
IKEA - TVM	15.7	23.6	20.0	21.1
IKEA - ARMA	**18.4**	**26.5**	**21.9**	**21.8**
Cooking - TVM	**13.9**	9.6	14.1	–
Cooking - ARMA	13.7	**10.3**	**14.2**	–
THUMOS15 - TVM	–	–	–	19.5
THUMOS15 - ARMA	–	–	–	**21.2**

Table 4: Impact of ARMA model on unsupervised action detection results.

Class	Recalled % of action segments
GolfSwing	43.0
BaseballPitch	35.2
CleanAndJerk	33.1
SoccerPenalty	25.8
Shotput	22.2

Table 5: Class based action discovery performance for THUMOS dataset.

Class	Recalled % of action segments
take-out-from-drawer	57.0
take-out-from-cupboard	47.7
take-out-from-fridge	34.3
take-&-put-in-cupboard	20.0
take-&-put-in-fridge	16.7

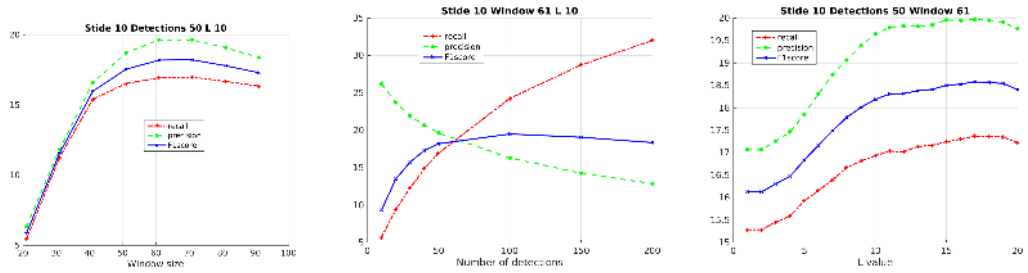
Table 6: Class based action discovery performance for Cooking dataset (Top 5 classes)

4.5. Impact of ARMA model

Next we evaluate the impact of ARMA model on unsupervised action detection task using all three datasets. We only report the F1 score for clarity. Results are reported in Table 4. Results suggest that ARMA smoothing process obtains better results compared to the Time Varying Mean (TVM) [11] except for the HOG features in the Cooking activities dataset. The impact of ARMA model over THUMOS15 dataset is about 1.7%. We conclude that ARMA model is better suited than the TVM for unsupervised action detection task using rank-pooling and temporal consistency algorithm.

4.6. Evaluating temporal consistency parameters

In this section we evaluate the effectiveness of several parameters of the temporal consistency method. We use the THUMOS15 dataset for parameter evaluation. For this ex-



(a) (b) (c)

Figure 4: Effect of several parameter on temporal consistency method. (a) window size, (b) number of candidate detections and (c) maximum temporal segment length (L) is evaluated using THUMOS15 dataset.

Figure 5: Visualizing unsupervised action detection from IKEA dataset

periment we use TVM method (not ARMA). We evaluate the impact of window size l_w, number of candidate detections, and minimum matched sequence length L. Results are shown in Figure 4. As it can be seen from Figure 4 (a), the results improve with bigger window size but results start to decrease after window size of 61. The results in the second plot (Figure 4 (b)) is not surprising. It indicates as the number of detections increases, the precision drops while the recall improves. However, the best F1 score is obtained for 60 detections using a window size of 60 and L value of 10. Next in Figure 4 (c), we see that as the minimum temporal candidate length L increases, the results improve significantly up to about L value of 19. This plot is very interesting and suggests that relatively large temporal candidates are better suited. This is a clear advantage of our rank pooling-based temporal consistency algorithm. Note that our temporal consistency method can be applied over other temporal encoding methods as well (not just limited to rank pooling). The main advantage of our temporal consistency method is that even if one uses a fixed window size, yet able to obtain variable length action unit candidates beyond the window size. Results suggest that such a strategy allows us to improve the detection performance without additional temporal encodings of different window sizes. We conclude that our temporal consistency method is very useful for unsupervised action detection task.

4.7. Class-based analysis and action discovery

In this section we compute the class-based analysis on detected action segments. In the following we show the recall (percentage of discovered action segments) of each action category using our temporal consistency algorithm. We report the best 5 classes for Cooking activities and THUMOS15 datasets in Table 5 and Table 6 respectively. Some of the detected human actions for IKEA dataset is also shown in Figure 3 and Figure 5. We conclude that the top detected action segments from our method are class specific and can be used as an action discovery technique.

5. Conclusion

We have proposed a novel task called unsupervised action detection by action matching. In this task the objective is to find pairs of video segments that share a common human action from a long pair of videos. It is an unsupervised task as the task is agnostic to the action class which make it useful for many real world applications. We have presented an effective and efficient method for discovering such human action pairs. We exploit the temporal consistency and the temporal evolution of videos to discover such pairs of video segments. We obtained promising results on three action detection datasets including MPII Cooking activities dataset and THUMOS15 challenge dataset. We believe in future the proposed task would be evolved to jointly learn video representations, action categories and action detectors in an unsupervised or semi-supervised manner while solving many real world problems.

Acknowledgement : This research was conducted by the Australian Research Council Centre of Excellence for Robotic Vision (project number CE140100016) and was undertaken on the NCI National Facility in Canberra, Australia, which is supported by the Australian Commonwealth Government.

References

[1] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, and S. Gould. Dynamic image networks for action recognition. In *CVPR*, 2016. 3, 5

[2] J. Carreira and C. Sminchisescu. CPMC: Automatic object segmentation using constrained parametric min-cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1312–1328, 2012. 2

[3] A. Cherian, B. Fernando, M. Harandi, and S. Gould. Generalized rank pooling for activity recognition. In *CVPR*, 2017. 3

[4] F. Diego, D. Ponsa, J. Serrat, and A. M. Lopez. Video alignment for change detection. *IEEE Transactions on Image Processing*, 20(7):1858–1869, 2011. 2

[5] J. Donahue, L. Anne 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. 2

[6] V. Escorcia, F. C. Heilbron, J. C. Niebles, and B. Ghanem. Daps: Deep action proposals for action understanding. In *ECCV*, 2016. 3

[7] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *Journal of machine learning research*, 9(Aug):1871–1874, 2008. 4

[8] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, 2016. 2

[9] B. Fernando, P. Anderson, M. Hutter, and S. Gould. Discriminative hierarchical rank pooling for activity recognition. In *CVPR*, 2016. 3

[10] B. Fernando, E. Gavves, J. Oramas, A. Ghodrati, and T. Tuytelaars. Modeling video evolution for action recognition. In *CVPR*, 2015. 3

[11] B. Fernando, E. Gavves, J. Oramas, A. Ghodrati, and T. Tuytelaars. Rank pooling for action recognition. *TPAMI*, PP(99):1–1, 2016. 2, 3, 4, 7

[12] B. Fernando and S. Gould. Learning end-to-end video classification with rank-pooling. In *ICML*, 2016. 3

[13] A. Gorban, H. Idrees, Y.-G. Jiang, A. Roshan Zamir, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. http://www.thumos.info/, 2015. 2, 6

[14] A. Hanjalic. Adaptive extraction of highlights from a sport video based on excitement modeling. *IEEE transactions on Multimedia*, 7(6):1114–1122, 2005. 2

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[16] M. Jain, J. van Gemert, and C. G. M. Snoek. What do 15,000 object categories tell us about classifying and localizing actions? In *CVPR*, 2015. 2

[17] I. N. Junejo, E. Dexter, I. Laptev, and P. Pérez. Cross-view action recognition from temporal self-similarities. In *ECCV*, 2008. 2

[18] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. 1

[19] O. Kliper-Gross, T. Hassner, and L. Wolf. The action similarity labeling challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):615–621, 2012. 1

[20] Z. Lan, M. Lin, X. Li, A. G. Hauptmann, and B. Raj. Beyond gaussian pyramid: Multi-skip feature stacking for action recognition. In *CVPR*, 2015. 2

[21] B. Ni, X. Yang, and S. Gao. Progressively parsing interactional objects for fine grained action detection. In *CVPR*, 2016. 2, 3

[22] X. Peng, C. Zou, Y. Qiao, and Q. Peng. Action recognition with stacked fisher vectors. In *ECCV*, 2014. 2

[23] R. Poppe. A survey on vision-based human action recognition. *Image and vision computing*, 28(6):976–990, 2010. 1

[24] A. Richard and J. Gall. Temporal action detection using a statistical language model. In *CVPR*, 2016. 2

[25] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele. A database for fine grained activity detection of cooking activities. In *CVPR*, 2012. 1, 2, 4, 6

[26] M. S. Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *ICCV*, 2011. 2

[27] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *ICPR*, 2004. 1

[28] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 2

[29] B. Singh, T. K. Marks, M. Jones, O. Tuzel, and M. Shao. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *CVPR*, 2016. 2, 3

[30] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 1

[31] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using lstms. *CoRR, abs/1502.04681*, 2, 2015. 2

[32] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 2

[33] T. Tuytelaars, C. H. Lampert, M. B. Blaschko, and W. Buntine. Unsupervised object discovery: A comparison. *International journal of computer vision*, 88(2):284–302, 2010. 2

[34] Y. Ukrainitz and M. Irani. Aligning sequences and actions by maximizing space-time correlations. In *ECCV*, 2006. 2

[35] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *CVPR*, 2015. 2

[36] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *CVPR*, 2016. 2, 3, 4