Biased Randomized Spatio-Temporal Pyramids for Egocentric Activity Recognition

Anonymous ICCV submission

Paper ID ****

Abstract

Egocentric video and wearable computing have become increasingly prevalent in the past decade, resulting in a huge explosion in the amount of available video content. In this paper, we present a novel approach for egocentric activity recognition using the UC Irvine ADL (Activities of Daily Living) dataset [11]. Existing work in activity recognition uses predefined binning schemes, which may fail to capture important spatio-temporal relationships between features. We propose to partition video clips into sets of 3-dimensional cuboids based on many different multi-level randomized partitioning schemes, then concatenate object histograms over multiple levels to form feature vectors which we then use to train a pool of weak SVM classifiers. Finally, we use a boosting algorithm to learn which partitioning schemes are most discriminative and form a final strong classifier with accuracy that improves upon the current state of the art. Our main novel contribution is a method for creating biased partition schemes based on observed distributions of active object locations across each spatial and temporal dimension of the video clips. We found that partitions which cut through spatio-temporal regions that tend to contain active objects are often more discriminative than unbiased partitions and partitions that cut around such active object regions.

1. Introduction

Activity recognition is becoming an increasingly canonical problem in computer vision as researchers are beginning to explore the domain more thoroughly and several relevant datasets have been released [13, 12, 4, 11]. Datasets geared towards activity recognition in the past have often consisted of actors performing scripted activities in a static and at times artificial environment [13]. However, in order to develop robust and effective recognition methods, we need datasets that are more organic in the sense that they depict unscripted activities occurring in a natural environment such as a home or apartment.

A robust and accurate method for egocentric activity recognition would have many useful practical applications. For instance, a recent trend in wearable computing is so-called life logging which can assist patients suffering from memory loss [14]. A robust egocentric activity recognition system could automatically tag video clips with types of activities, thus allowing the user to, for example, quickly find all clips recorded in the past that depict making tea.

Egocentric activity recognition in a daily living context differs from non-egocentric activity recognition because activities can have long-term temporal dependencies and actions can be interrupted by other actions. For example, a user might wash a few dishes while waiting for a cup of tea to brew. The familiar bag-of-words approach can be used to aggregate space-time features with reasonable performance, but ultimately falls short because it fails to capture temporal dependencies between features. The pyramid is a well-known extension of a pure bag-of-words model that encodes spatial relationships between features by recursively subdividing images or video and extracting features from each spatial bin [8], yielding impressive results across a range of applications.

Existing work in egocentric activity recognition has shown that activities are well-defined by the types of objects that are interacted with by users during particular actions [11]. Previous work in egocentric activity recognition has often employed a single strict hand-coded partition scheme [11], which may not be particularly robust to inter and intra-class variation. The work of [7] uses multiple candidate spatio-temporal grids for the task of activity recognition (but not in an egocentric setting), however each grid is predefined and only 24 possible candidate grids are considered. With a small pool of schemes for imposing spatial information, the most discriminative space-time relationships between features may not be captured. The work presented in [6] describes an effective state-of-the-art method for learning the shapes of spatio-temporal regions on a per-class basis, but makes use of lower-level features such as optical flow, rather than object locations, and is not applied in an egocentric setting.

Spatial pooling of features in a learned way for object recognition in 2D images has been thoroughly explored [15], but to our knowledge there has been little work on learning the best way to pool spatio-temporal features in video.

Our proposed method, however, builds on existing work by creating a large number of candidate partitioning schemes in a randomized way. We then pool spatio-temporal features in a learned way, using a boosting algorithm to select those partitioning schemes which are most discriminative. Our main novel contribution is the ability to bias the randomization step such that resulting partition schemes have a high probability of cutting through or around spatio-temporal regions which are known to tend to contain active objects. We call pyramids generated according to this method object-centric pyramids. We found that partitioning schemes that have a larger number of levels and are biased to cut through regions known to contain active objects are often the most discriminative. We evaluate the performance of our method using a cross-validation experiment and found that our method using object-centric pyramids improves upon the current state of the art.

1.1. Related Work

In [7], Laptev *et al*. investigate automatically aligning movie scripts with video for the purpose of annotating human actions, and achieve 91.8% accuracy on the KTH dataset. Previous work on spatial pyramids [1, 8] is extended by defining the spatio-temporal pyramid representation of video clips, but uses a relatively small number of predefined schemes for spatio-temporal binning, which may fail to capture important spatio-temporal relationships between features. There are 6 possible spatial grids and 4 temporal binning schemes, resulting in a total of 24 possible spatio-temporal partition schemes.

In [10], Marszalek *et al*. released a novel dataset based on Hollywood movies that contains twelve types of activities and ten different classes of scenes. The main contribution of this paper is based on the observation that the visual content of a human's environment can impose useful constraints on the type of activity occurring. For instance, food preparation activities frequently occur in a kitchen environment.

In [3], Fathi *et al*. focus on the relationship between gaze and activity recognition in an egocentric setting and develop methods to predict activity given gaze, gaze given activity, and both activity and gaze given neither. The activities in this published dataset are primarily related to food preparation.

The main work related to our own is that carried out in [11]. In this work the ADL dataset is introduced as well as detailed analysis of performance of several different classifiers. Composite object models are developed for the purpose of object detection. These models take advantage of the fact that objects can have vastly different appearances when they are being interacted with. For example, a refrigerator or microwave has a different appearance when it is open and being interacted with. A comparison of the well-known bag-of-words approach with a strict predefined 2-level temporal pyramid using both space-time interest points and the results of the part-based object detectors is presented. The temporal pyramid makes a single cut along the temporal dimension and no cuts along the spatial dimensions, but is simple, easy to implement, and outperforms a classifier trained on bag-of-words histograms. The use of detected objects as features offers impressive improvement over using space-time interest points (STIP). The crucial contribution of [11] is that egocentric activity recognition is "all about the objects", particularly the objects being interacted with, as recognition accuracy increases dramatically when locations of active objects in addition to passive objects are used as features.

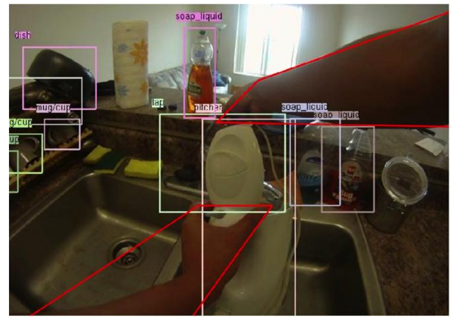


Figure 1. An example frame from the ADL dataset with annotations for hand position and detected objects

The video collected for the ADL dataset is available in a temporally presegmented format; each video has been segmented into clips depicting activities. Egocentric video is captured as a continuous stream, and thus a pre-processing step of temporal segmentation into discrete events is required. There is a large amount of literature on temporal segmentation of video. For instance, work presented in [9] includes a method for automatic temporal segmentation of egocentric video into events.

Our algorithm is inspired by the work of [5], which uses a a version of the SAMME Ada-boost algorithm [16] with randomized spatial pyramids for 2D images, leading to increased robustness to intra-class variation based on results from benchmarks on three publicly available datasets. However, in contrast to our own work, the randomized pyramids are not biased in any way.

2. Approach

Our boosting algorithm takes as input a collection of labeled training videos and a pool of candidate partition patterns. We use the output of the aforementioned object detectors trained on composite object models as our features to be pooled. We train a separate "weak" multi-class SVM (using LIBSVM [2]) classifier on the feature vectors resulting from representing the training data using each candidate partition pattern θ. We set a weight w_i for each training point p_i that is inversely proportional to the number of points with the same class as p_i. Giving larger weights to training examples of infrequently occurring actions helps to mitigate any bias resulting from imbalanced training data. During each round of boosting we select the candidate partition θ_j that is most discriminative (has minimum weighted training error, which is computed as the dot product between the weight vector w and an indicator of incorrect classifications using f_θ) Next, we compute a weight for θ_j, and compute accuracy for the current version of the final strong classifier, which maximizes a weighted sum of classifications produced by each weak classifier. We set the number of boosting rounds to 30.

Algorithm 1: Training RSTP Classifier via Boosting
INPUT:

- N labeled training videos $\Phi = \{(V_i, c_i)\}_{i=1}^N$
- A pool of M partition patterns $\Theta = \{\theta\}$

OUTPUT:

- A strong video classifier F. For an unlabeled video V, $c = F(V)$ is the predicted label for V.

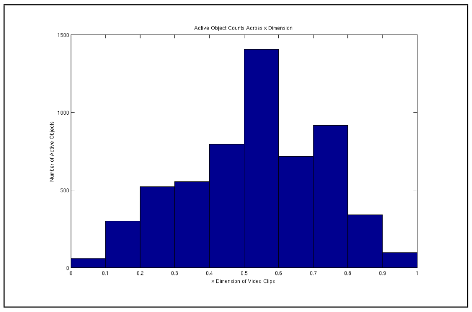
1. For each $\theta \in \Theta$:
    - Compute the representations of each $V_i \in \Phi$ using θ and train a multi-class classifier (SVM) f_θ on the resulting feature vectors.
2. Initialize:
    - A weight $w_i = \frac{1}{CN_{c_i}}$ for each video clip, where N_{c_i} is the number of videos with label c_i, and C is the number of distinct labels in the training data.
    - Current iteration number $j = 0$.
    - Current accuracy $\sigma_j = 0$.
3. For each round of boosting:
    - Increment j.
    - Re-normalize the weight vector:
    $$\forall i, w_i = \frac{w_i}{\Sigma_i^N w_i}.$$
    - For each pattern θ, compute its weighted classification error:
    $$e_\theta = w \cdot \mathbf{I}(f_\theta(V) \neq c)$$
    - Choose the pattern θ_j with minimum weighted classification error e_j.
    - Compute the weight for θ_j:
    $$\alpha_j = \log \frac{1-e_j}{e_j} + \log(C-1)$$
    - Update the weight vector:
    $$w_i = w_i \cdot \exp(\alpha_j \cdot \mathbf{I}(f_{\theta_j}(V_i) \neq c_i)).$$
    - Generate the current strong classifier:
    $$F(V) = \text{argmax}_c \Sigma_{m=1}^j \alpha_m \cdot \mathbf{I}(f_{\theta_m}(V) = c)$$

The original version of the SAMME algorithm has each weak classifier f_θ trained on a randomly sampled subset of the training examples, but we train each of our weak classifiers on the full training dataset.

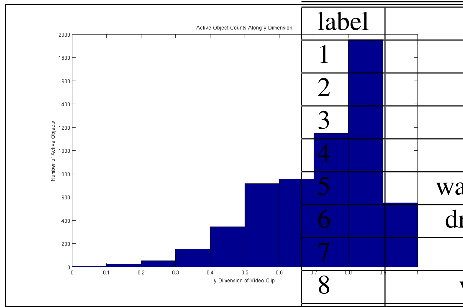
2.1. RSTP Implementation

We use k-d trees to represent randomized partition schemes. Each level in the tree represents a set of cuts along a certain dimension, and we generate cuts for subsequent levels in the tree. in a round-robin manner over dimensions (x, y, t). Cuts for child nodes are generated independently, and each cut is axis-aligned (we incorporate random shifts, but not random rotations). To construct a partition scheme that is easily applicable to videos of arbitrary size, we consider partitioning an "idealized" video clip that has all dimensions normalized to length 1. To generate a single cut we sample a random number from a distribution subject to any constraints imposed by "parent cuts" and use this as a randomized offset for an appropriate axis-aligned plane. To construct an unbiased partition scheme we sample from a uniform distribution.

To represent a video clip as a randomized spatio-temporal pyramid (RSTP) using a particular partition scheme we use the output of object detectors trained in [11], which gives bounding boxes and object labels for each extracted frame. We compute histograms of detected objects for each individual level in the pyramid, where level 0 is the entire video clip volume and level i is all the cells of depth i in the k-d tree. Note that level i has 8^i leaf cells. To form the final RSTP representation, we concatenate the histograms computed for each level to form a single feature vector. We can choose whether or not to include detected active objects when forming an RSTP representation of a video clip, however taking active objects into account gives a substantial improvement to overall classification accuracy.



(a)



(b)

Figure 2. Histograms of detected active objects across the x and y spatial dimensions of training data. Active objects tend to appear in the lower center field of view. There is a slight bias favoring the right side of the field of view because many users are right-handed.

label	activity type
1	combing hair
2	make up
3	brushing teeth
4	dental floss
5	washing hands/face
6	drying hands/face
7	laundry
8	washing dishes
9	moving dishes
10	making tea
11	making coffee
12	drinking water/bottle
13	drinking water/tap
14	preparing cold food/snack
15	vacuuming
16	watching tv
17	using computer
18	using cell phone

Table 1. Types of activities present in the ADL dataset.

2.2. Object-Centric Cuts (OCC)

The key novel contribution of our work is the ability to create meaningfully biased partition schemes that tend to be more discriminative than unbiased partition schemes. To accomplish this, we replace the uniform distribution with a discrete approximation of the distribution of active objects across each dimension (x, y, t) and otherwise proceed normally.

From figure 2 we see that active objects often tend to occur in the lower center of the field of view. This conforms to our expectations, because the active objects are close to the hands which are in the lower field of view from an egocentric perspective. Active objects tend to occur on the right side of the field of view slightly more often because a large percentage of users are right-handed. Since different clips can have varying lengths with respect to time, we normalize the length of each video clip to 1 and consider relative temporal locations of active objects. The distribution of active objects across the temporal dimension is nearly uniform. When generating a biased partition scheme, we can choose to prefer splits that cut around regions that tend to contain active objects (denoted as bias type 2), or we can choose to prefer splits that cut through regions that tend to contain active objects (denoted as bias type 3). We denote by bias type 1 the method of using uniform distributions to generate partitions. For biased partitions, we generate the first split along each dimension according to a weighted distribution corresponding to the histograms of observed active object regions in the training data, and we generate all subsequent child cuts using a uniform distribution. We do not consider locations of passive objects at all during the generation of biased partition schemes. Since active objects are located in close spatial proximity to hands, creating biased partition schemes can be interpreted as implicitly taking into account information about hand locations.

Figure 3 depicts an example 3-level partition scheme created using bias type 3. The salient feature to note is that visible splits along the y dimension correspond to the observed distribution of active objects along the y dimension.

3. Results

The ADL dataset consists of hundreds of egocentric video clips (roughly 10 hours of video in total) collected from 20 people performing 18 types of unscripted actions in their own homes. These naturally occurring actions are often related to hygiene or food preparation and are more varied than actions presented in previous datasets such as that of [4]. There are 26 different types of detected objects, including 5 active and 21 passive objects. Object detectors are trained on videos from the first 6 people and tested on the videos from the remaining 14 people.

Each frame in the dataset is annotated with activity labels and bounding boxes for detected objects and hand positions, Additionally, each object is tagged as active or passive depending on whether it is being interacted with. The ADL dataset has been modified since the publication of [11]; because of this, running the published code gives slightly lower accuracy than the originally published numbers. We use the modified version of the dataset available from the authors webpage at the time of writing to benchmark our method. One difficulty that can arise within egocentric activity recognition is that activities can be temporarily interrupted by other activities. For instance, while waiting for tea to brew a subject may watch TV. For cases of such interruptions, to avoid unnecessary complications resulting from frames being annotated with multiple activities, the ADL

Feature Type	BoW	Temporal Pyramid	RSTP	RSTP+OCC
STIP	16.5%	22.8%	-	-
O	26.6%	29.0%	-	32.7
AO	34.9%	36.9%	33.7%	38.7

Table 2. Overall classification accuracy on pre-segmented video clips, evaluated using a form of cross validation. Our boosted RSTP classifier improves on the current state of the art.

dataset simply uses the label of the interrupting action when a longer action is disrupted.

Table 2 shows a comparison of overall classification accuracy between our method and two methods presented in [11]. The temporal pyramid has two levels, formed by making a single cut along the temporal dimension and no cuts along the spatial dimensions.

Row 1 shows results originally published in [11] which we reproduce to illustrate the advantage of using higher level features such as detected objects over low level features such as space-time interest points. Row 2 shows results obtained using only passive detected objects, while row 3 shows results obtained using both active (being interacted with) and passive detected objects. The consideration of active objects when constructing feature vectors gives a significant improvement over just including passive objects, and in both cases our method improves on the current state of the art.

The results shown in Table 2 and Figure 4 are computed using a form of cross validation (the video clips from person i are used as a held out validation set, and training occurs using the video clips from the remaining people). Following [11], we exclude videos from the first 6 people (because they were used to train the object detectors) from our experiments. Feature vectors are computed using detections for both active and passive objects. The results for bag of words and temporal pyramids (2 level, with a single cut along the temporal dimension) are both presented in [11].

For this experiment we used pools of 4-level partitioning schemes of varying sizes with a varying number of boosting rounds. The numbers presented in Table 2 were obtained with 5 boosting rounds and a pool of size 70. Each partitioning scheme was of bias type 3, meaning the cuts for level 1 were biased such that they had a tendency to cut through regions containing active objects, and the cuts for levels 2 and 3 were drawn from a uniform distribution. The work of [5], which uses a similar pyramid-based boosting approach for 2D image recognition, found that using pyramids with more than 3 levels actually led to a decrease in overall accuracy due to over-segmentation of image space. However, we found that in the 3D case 4-level pyramids give better overall accuracy than coarser-grained representations.

As seen in Figure 3, our method has particularly good performance for activity types 5 and 6 ("combing hair" and

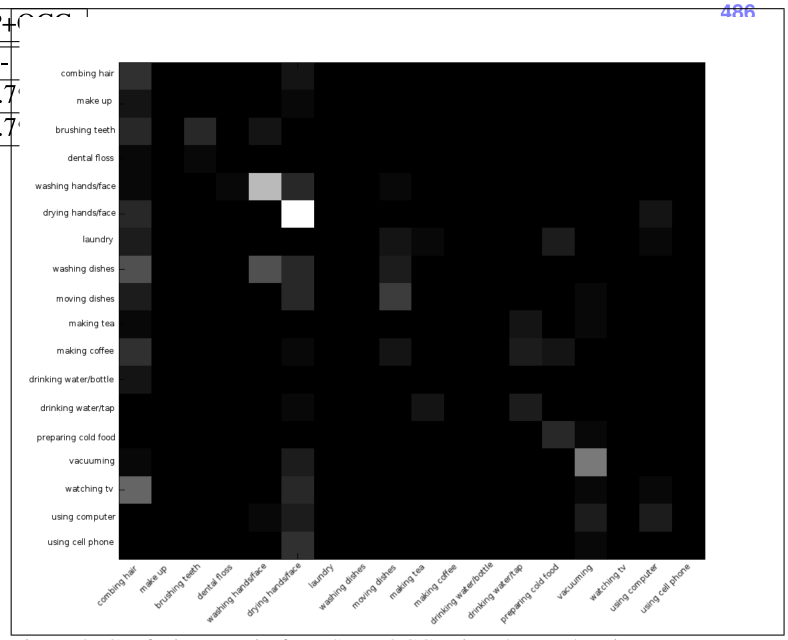


Figure 3. Confusion matrix for RSTP+OCC using detected active and passive objects

"drying hands/face", respectively). Some activity types on which our method does poorly are 10 and 11, which are "making tea" and "making coffee", respectively (see Table 1 for a full listing of activity types present in the ADL dataset). Since the two activity types are similar in the sense that they involve the same active objects, it is not unexpected that a recognition system would confuse them often.

To illustrate the improvement on accuracy obtained from using a pool of biased partitions, we created 3 separate pools containing 4-level partition schemes of each bias type and repeatedly ran the cross-validation experiment, adding additional partitions to each pool between runs. The pool containing partitions of bias type 3 consistently outperformed the pool of bias type 2, which consistently outperformed the unbiased pool. The results from this experiment are depicted in figure 4. We were initially surprised to find that increasing pool size can sometimes negatively impact overall classification accuracy, however we believe this is due to the inherent bias between the training and the test data. In other words, sometimes a partition scheme with small training error on the train data that gets selected during a round of boosting can have a larger training error on the test data. A similar behavior was discovered in the case of boosting spatial pyramids for 2D image recognition [5].

To further support our claim that 4-level pyramids of bias type 3 tend to be most discriminative, we created a heterogeneous pool containing partition schemes of several different types. Specifically, the heterogeneous pool contains 30 2-level partitions of each bias type, 30 3-level partitions of

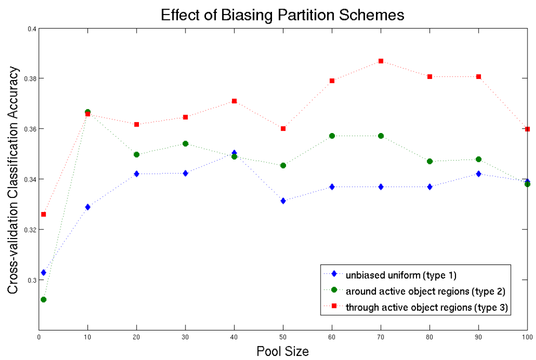


Figure 4. Effect of using biased partition schemes. The pool containing type 3 biased partition schemes consistently outperforms the other pools.

each bias type, and 30 4-level partitions of each bias type, for a total of 270 partitions. We generated 20 random 50/50 train/test splits, fixed the number of boosting rounds to 5, and observed which types of partitions were most often selected during boosting rounds. We found that partition schemes with more levels tend to get selected more often during boosting rounds. Specifically, we found that 2-level pyramids of bias type 3 were selected 21% of the time, 3-level pyramids of bias type 3 were selected 19% of the time, and 4-level pyramids of bias type 3 were selected 37% of the time. 2-level and 4-level unbiased pyramids were never selected. Thus, biased partition schemes that cut through regions that tend to contain active objects are clearly more discriminative than other types of partition schemes, especially those which are unbiased.

4. Conclusion and Future Work

We have presented an application of the well-known boosting framework with results for an egocentric activity recognition task that improve upon the current state of the art. Our main novel contribution is a method for generating biased partition schemes based on observations of active object locations throughout the training data (Object-Centric Cuts). Future work could potentially incorporate different types of biases when generating partitions. The ADL dataset also includes annotations for hand positions, which we have incorporated implicitly through our generation of biased partitions biased relative to regions which tend to contain active objects. However, it could be possible to incorporate explicit information given by hand positions to obtain better classification results. The partitions we focus on contain cuts that are planar and axis-aligned (we consider random shifts but not random rotations, and we do not consider non-planar splits), but it is possible to carve up the video volume in more advanced non-linear ways. Such a method would involve more sophisticated computational geometry, but may yield a more discriminative partitioning scheme that could lead to better classification accuracy.

References

[1] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. *Proceedings of the 6th ACM international conference on Image and video retrieval*, 2007. 2

[2] C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, pages 27:1–27:27, 2011. 3

[3] A. Fathi, Y. Li, and J. M. Rehg. Learning to recognize daily actions using gaze. *Proceedings of the 12th European conference on Computer Vision - Volume Part II*, 2012. 2

[4] A. Fathi, X. Ren, and J. Rehg. Learning to recognize objects in egocentric activities. *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3281–3288, 2011. 1, 4

[5] Y. Jiang, J. Yuan, and G. Yu. Randomized spatial partition for scene recognition. *Proceedings of the 12th European conference on Computer Vision - Volume Part II*, pages 730–743, 2012. 2, 5

[6] A. Kovashka and K. Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. 2010. 1

[7] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. *Computer Vision and Pattern Recognition, 2008.*, pages 1–8. 1, 2

[8] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, pages 2169–2178, 2006. 1, 2

[9] Y. J. Lee, J. Ghosh, and K. Grauman. Discovering important people and objects for egocentric video summarization. *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1346–1353, 2012. 2

[10] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. *Computer Vision and Pattern Recognition, 2009.*, pages 2929–2936. 2

[11] H. Pirsiavash and D. Ramanan. Detecting activities of daily living in first-person camera views. *Computer Vision and Pattern Recognition, 2012.*, pages 2847–2854. 1, 2, 3, 4, 5

[12] M. Rodriguez, J. Ahmed, and M. Shah. Action mach a spatio-temporal maximum average correlation height filter for action recognition. pages 1–8, 2008. 1

[13] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, 3:32–36 Vol.3, 2004. 1

[14] A. J. Sellen, A. Fogg, M. Aitken, S. Hodges, C. Rother, and K. Wood. Do life-logging technologies support memory for the past?: an experimental study using sensecam. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 81–90, 2007. 1

[15] G. Sharma and F. Jurie. Learning discriminative spatial representation for image classification. *British Machine Vision Conference (BMVC)*, 2011. 2

[16] J. Zhu, S. Rosset, H. Zou, and T. Hastie. Multi-class adaboost. *Ann Arbor*, 1001(48109):1612, 2006. 2