Randomized Spatio-Temporal Pyramids for Egocentric Activity Recognition

Tomas McCandless and Kristen Grauman
University of Texas at Austin
{tomas, grauman}@cs.utexas.edu

Abstract

Egocentric video and wearable computing have become increasingly prevalent in the past decade, resulting in a huge explosion in the amount of video content. In this paper, we present a novel approach for egocentric activity recognition using the UC Irvine ADL (Activities of Daily Living) dataset [12]. Existing work in activity recognition uses predefined binning schemes, which may fail to capture important temporal relationships between features. The method we present partitions video clips into 3-dimensional cuboids, based on many different multi-level randomized partitioning schemes, then concatenates object histograms over multiple levels to form feature vectors which we use to train a pool of weak SVM classifiers. Finally, we use a boosting algorithm to learn the most discriminative partitions 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 dimension of the dataset. 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. The problem of human activity recognition is in some ways less well defined than, say, object recognition for 2D images, in part due to the relative lack of datasets for activity recognition, and also because it is somewhat problematic to define a canonical representation for each type of action. In other words, it seems as though there can be higher intra-class variation for activity recognition than for, say, object recognition. Datasets geared towards activity recognition in the past have often consisted of actors performing scripted activities in a static and at times artificial environment, yet in order to develop robust and effective methods, we need datasets that are more organic in the sense that they depict unscripted activities in a natural environment such as a home or apartment. [12]. However, activity recognition and object recognition do share some similar properties. For instance, occlusion and background clutter are problems that arise in both problems.

A robust and accurate method for egocentric activity recognition would have many practical applications. For instance, a recent trend in wearable computing is so-called life logging which can assist patients suffering from memory loss [13]. However, with such large amounts of video, it becomes necessary to have a system for efficiently browsing video. A robust egocentric activity recognition system could automatically tag video clips with types of activities (this could be done either online or offline), thus allowing the user to, for instance, quickly find all clips in the past that depict making tea.

There are many clinical benchmarks used to evaluate patients everyday functional abilities [7, 1, 5]. These benchmarks are currently conducted in a hospital setting, but a robust system for egocentric activity recognition could greatly impact the workflow for patient evaluation, as such a system would allow for passive long term observation of patients in their own homes. This could lead to more accurate evaluations since it would be possible to collect far more data about individual patients. Such a system would also eliminate the need for patients to commute to a hospital to have evaluations done, thus reducing cognitive and physical burden on patients.

Previous work in activity recognition has employed a single strict hand-coded partition scheme [12], which may not be particularly robust to inter and intra-class variation. The work of [9] uses multiple candidate spatio-temporal grids for the task of activity recognition (but not in an egocentric setting), however each grid is hand-coded and only 24 candidate grids are considered. The work presented in [8] describes an effective method for learning the shapes of spatio-temporal regions on a per-class basis, but makes use of lower-level features and is not applied in an egocentric setting.

Spatial pooling of features in a learned way has been thoroughly explored [14], but to our knowledge there has been little work on learning the best way to pool spatio-temporal features.

Our method, however, builds on existing work by creating a larger number of candidate partitioning schemes in a randomized way. Our main novel contribution is the ability to bias this randomization step so that partitions in the resulting pool have a high probability of cutting through or around spatio-temporal regions which tend to contain active objects. We then pool spatio-temporal features in a learned way, selecting those partitioning schemes which are most discriminative.

1.1. Related Work

In [9], Laptev *et al*. investigate aligning movie scripts with video for the purpose of annotating human actions, and achieve 91.8% accuracy on the KTH dataset. The method presented in this paper uses a relatively small number of hard-coded schemes for spatio-temporal binning, which may fail to capture important spatio-temporal relationships between features.

In [11], 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 particular, Laptev *et al*. show how to learn relevant scene classes along with any correlations they may have with human activity.

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 to predict both activity and gaze. The activities in this published dataset are primarily related to food preparation.

The main work related to our own is that carried out in [12]. In this work the ADL dataset is introduced as well as detailed analysis of performance of several different classifiers.

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 unscripted actions are often related to hygiene or food preparation and are more varied than actions presented in previous datasets such as that presented in [4]. There are 26 different types of detected objects, including 5 active and 21 passive objects. 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. A comparison of the well known bag-of-words approach with a strict hard-coded 2-level temporal pyramid is presented. The temporal pyramid makes no cuts along the spatial dimensions, but is easy to implement, simple, and outperforms a classifier trained on bag-of-words histograms. The crucial contribution of [12] is that egocentric activity recognition is "all about the objects", particularly the objects being interacted with, as recognition accuracy increases dramatically when ground truth object locations rather than detected locations are used to train the classifier.

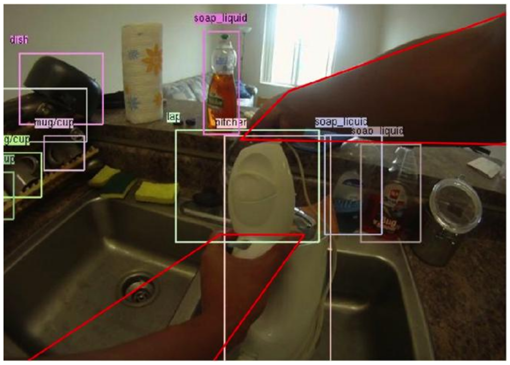

Figure 1. An example frame with annotations TODO: find a better image?

Our algorithm is inspired by the work of [6], which uses a a version of the SAMME Ada-boost algorithm [15] with randomized spatial pyramids for 2D images, leading to increased robustness to intra-class variation. However, the randomized pyramids are not biased in any way. The method introduced by [6] is benchmarked on three public datasets.

The video collected for the ADL dataset is available in a temporally presegmented format; the shots have been segmented into clips depicting activities. The work presented in [10] includes a method for temporally segmenting egocentric video into events.

2. Approach

Our boosting algorithm takes as input a collection of labeled training videos and a pool of candidate partition patterns. We train a separate weak SVM (using LIBSVM [2]) classifier on the feature vectors resulting from representing the training data using each candidate partition pattern. We set a weight for each training point p_i that is inversely proportional to the number of points with the same class as p_i. During each round of boosting we select the candidate partition θ_j that is most discriminative (has minimum training error), compute a weight for θ_j, and compute accuracy for the current version of the final strong classifier. We set the number of boosting rounds to 30, noting that additional boosting rounds only give a marginal boost to performance. Additionally, with a larger number of boosting rounds, overfitting becomes a possibility.

Algorithm 1 Training RSTP Classifier via Boosting

```
INPUT:
  • $N$ labeled training videos $\Phi = \{(V_i, c_i)\}_{i=1}^{N}$
  • A pool of 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 $\theta \in \Theta$
       – Train a multi-class classifier (SVM) $f_\theta$ on $\Phi$
    2. Initialize:
       – weight $w_i = \frac{1}{CN_{c_i}}$ for each video clip,
         where $N_{c_i}$ is the number of videos with label $c_i$.
       – current iteration number $j = 0$.
       – current accuracy $\sigma_j = 0$.
    3. For each round of boosting:
       – increment $j$.
       – Re-normalize the weight vector: $w_i = \frac{w_i}{\Sigma_i^N w_i}$.
       – For each pattern $\theta$, compute its classification
         error $err_\theta$ as the dot product product of $w$
         with the indicator vector of incorrect classifications using $f_\theta$.
       – Choose the pattern $\theta_j$ with minimum error $err_j$
       – Compute the weight for $\theta_j$ as:
         $\alpha_j = \log\frac{1-err_j}{err_j} + \log(C-1)$
       – Update the weight vector:
         $w_i = w_i * \exp(\alpha_j * \mathbf{I}(f_{\theta_j}(V_i) \neq c_i))$.
       – Generate the strong classifier:
         $F(V) = \arg\max_c \Sigma_{m=1}^{j} \alpha_m * \mathbf{I}(f_{\theta_m}(V) = c)$
```

The original version of the SAMME algorithm has each weak classifier f_θ trained on a randomly selected subset of the training dataset, but we train each of our weak classifiers on the full training dataset in order to reduce the number of randomized portions of our method, making it easier to reason about.

2.1. Partitions

We use k-d trees to represent partition schemes, where each level in the tree represents a set of cuts along a certain dimension, and we generate cuts in a round robin manner over dimensions (x, y, t) across levels in the tree. Cuts for child nodes are generated independently, and each cut is axis-aligned (we incorporate random shifts, but not random rotations). Initially, all randomized partitions were computed according to a uniform distribution. However, in an attempt to avoid generating partition schemes that are not sufficiently discriminative, we bias the partition generation step according to computed distributions of active object locations across training data.

From figure 2 we see that active objects tend to occur in the lower center of the field of view, and that active objects are nearly uniformly distributed across the temporal dimension. This is as expected, because the active objects are close to the hands which are in the lower field of view from an egocentric perspective. When generating a biased partition, we can choose to prefer cutting around regions that tend to contain active objects (denoted as bias type 2), or we can choose to prefer cutting through regions that tend to contain active objects (denoted as bias type 3). We denote by bias type 1 the method of using completely uniform distributions to generate partitions. For biased partitions, we generate the first cut along each dimension according to a weighted distribution corresponding to the observed active object regions in the training data, and we generate all subsequent child cuts using 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 [12], which gives bounding boxes and object labels. We compute histograms 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 simply concatenate the histograms computed for each level to form a single feature vector.

3. Results

The ADL dataset has been modified since the publication of [12]; because of this, running the published code gives slightly lower accuracy than the originally published numbers. We use the dataset available from the authors webpage at the time of writing to benchmark our method.

Table 1 shows a comparison of overall classification accuracy between our method and two methods presented in [12]. The temporal pyramid has two levels, formed by making a single cut along the temporal dimension. Row 1 shows results obtained using only passive detected objects, while row 2 shows results obtained using both active (being inter-

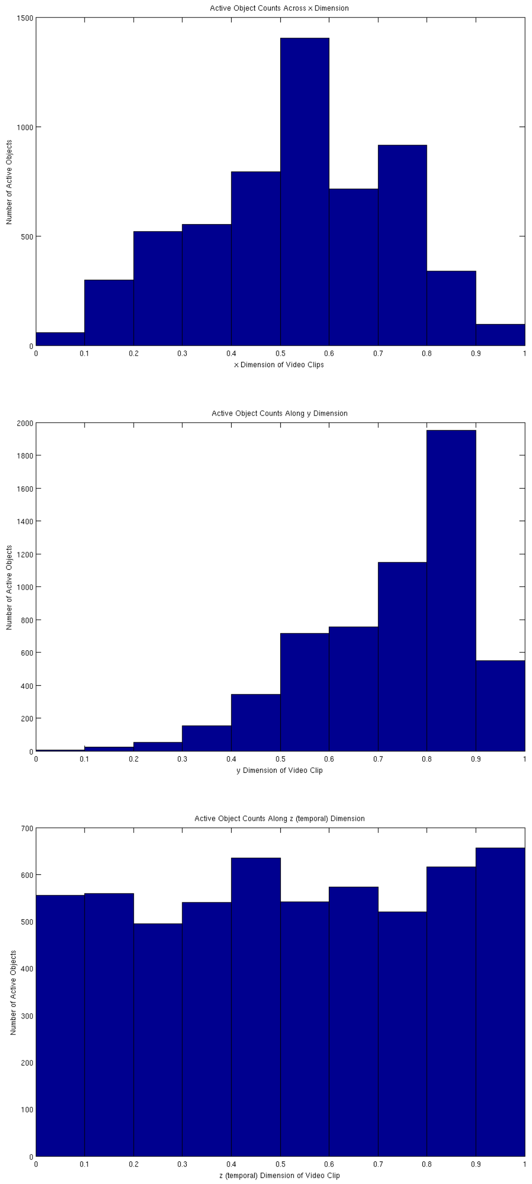

Figure 2. Histograms of counts of active objects across all 3 dimensions

Feature Type	BoW	Temporal Pyramid	RSTP
O	26.6	29.0	32.7
AO	34.9	36.9	37.9

Table 1. Overall classification accuracy on pre-segmented video clips.

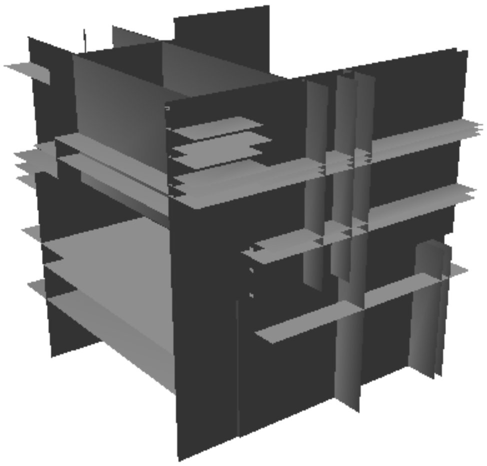

Figure 3. An example biased (type 3) partitioning scheme corresponding to a 3-level pyramid

acted with) and passive detected objects. The consideration of active objects when constructing feature vectors gives a significant improvement over just considering passive objects, and in both cases our method improves on the current state of the art.

The results shown in Table 1 are computed using a form of cross validation (use the video clips from person i as a held out validation set, and train on the video clips from the remaining people).

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 [12].

For this experiment we used a pool consisting of 100 4-level partitioning schemes. 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 [6], 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 oversegmentation of images. However, we found that in the 3D case 4-level pyramids give better overall accuracy than coarser-grained representations.

As seen in the confusion matrix, our method has particularly good performance for activity types 1 and 6 (“combing hair” and “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. Since the two activity types are similar it is not unexpected that a recognition system would confuse them often.

TODO: plot accuracy vs pool size for each bias type

TODO: accuracy on GA tech egocentric dataset

To support our claim that 4-level biased pyramids tend

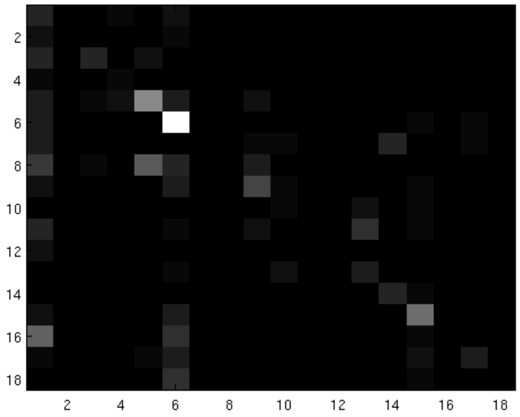

Figure 4. Confusion matrix for our method using active objects

to be most discriminative, we created a heterogeneous pool containing partitions of different types. Specifically, the heterogeneous pool contains 30 2-level partitions of each bias type, 30 3-level partitions of 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 partition were most often selected. We found that 3-level pyramids are often preferred to 2-level pyramids, and 4-level pyramids are often preferred to 3-level pyramids. 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 that improve upon the current state of the art. Our main novel contribution is a method for generating biased partition schemes. Future work could 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 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. Additionally, it may be worthwhile to investigate the performance of other variants of the boosting algorithm. The partitions we focus on contain cuts that are planar and axis-aligned, but it is also possible to carve up the video volume in 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. Catz, M. Itzkovich, E. Agranov, H. Ring, A. Tamir, et al. Scim–spinal cord independence measure: a new disability scale for patients with spinal cord lesions. *Spinal Cord*, 35(12):850, 1997. 1

[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. 2

[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. 2

[5] M. Itzkovich, I. Gelernter, F. Biering-Sorensen, C. Weeks, M. Laramee, B. Craven, M. Tonack, S. Hitzig, E. Glaser, G. Zeilig, et al. The spinal cord independence measure (scim) version iii: reliability and validity in a multi-center international study. *Disability & Rehabilitation*, 29(24):1926–1933, 2007. 1

[6] 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, 4

[7] B. Kopp, A. Kunkel, H. Flor, T. Platz, U. Rose, K.-H. Mauritz, K. Gresser, K. L. McCulloch, E. Taub, et al. The arm motor ability test: reliability, validity, and sensitivity to change of an instrument for assessing disabilities in activities of daily living. *Archives of physical medicine and rehabilitation*, 78(6):615, 1997. 1

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

[9] 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

[10] 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

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

[12] 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

[13] 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

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

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