Randomized Spatio-Temporal Pyramids for Egocentric Activity Recognition

Tomas McCandless, 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 activity recognition using the UC Irvine ADL (Activities of Daily Living) dataset [10]. The ADL dataset consists of hundreds of egocentric video clips captured by dozens of people while performing various everyday activities in their own homes, often related to hygeine or food preparation. Frames in the dataset are 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. We partition video clips into 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 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 outperforming that of 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. Strong classifiers trained using pools of biased partitions tend to outperform those which are trained using unbiased partitions.

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. [10]. 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 [11]. 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 [6, 1, 4]. 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.

1.1. Related Work

In [7], 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.

In [9], Marszalek *et al*. released a novel dataset based on Hollywood movies that contains twelve types of activities

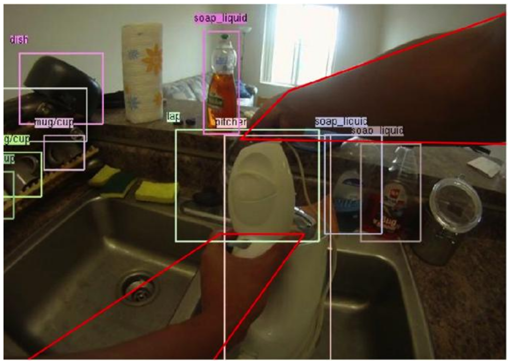

Figure 1. An example frame with annotations TODO: find a better image

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 settting 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 [10]. In this work the ADL dataset is introduced as well as detailed analysis of performance of several different classifiers. The ADL dataset consists of roughly 10 hours of video in total, collected from 20 people performing 18 types of unscripted actions in their own homes. There are 26 different types of detected objects, including 5 active and 21 passive objects. Ramanan compares the well known bag-of-words approach with a strict hard coded 2-level temporal pyramid. The temporal pyramid is simple and outperforms a classifier trained on bag-of-words histograms. The crucial contribution of [10] is that egocentric activity recognition is "all about the objects", particularly the objects being iteracted with, as recognition accuracy increases dramatically when ground truth object locations rather than detected locations are used to train the classifier.

Our algorithm is inspired by the work of [5], which uses a similar boosting approach with randomized spatial pyramids for 2D images, leading to increased robustness to intra-class variation. The proposed method 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 [8] includes a method for temporally segmenting egocentric video into events.

2. Approach

Our algorithm was inspired by the work of [5]. We use a version of the SAMME Ada-boost algorithm [12] to learn the most discriminative candidate partitions.

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_θ on Φ
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 θ, compute its classification error err_θ as the dot product product of w with the indicator vector of incorrect classifications using f_θ.

- Choose the pattern θ_j with minimum error err_j
- Compute the weight for θ_j as: $\alpha_j = \log\frac{1-err_j}{err_j} + \log(C-1)$
- Update the weight vector: $\mathrm{w}_i = w_i * \exp(\alpha_j * \mathbf{I}(f_{\theta_j}(V_i) \neq c_i))$.
- 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 develop two different representations of partition schemes: the first is simply a list of 4-tuples representing planar equations in video space. Each plane is independently rotated according to a parameterized probability. The second is an implementation of a kd-tree, where each level in the tree represents a set of cuts along a certain dimension. 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.

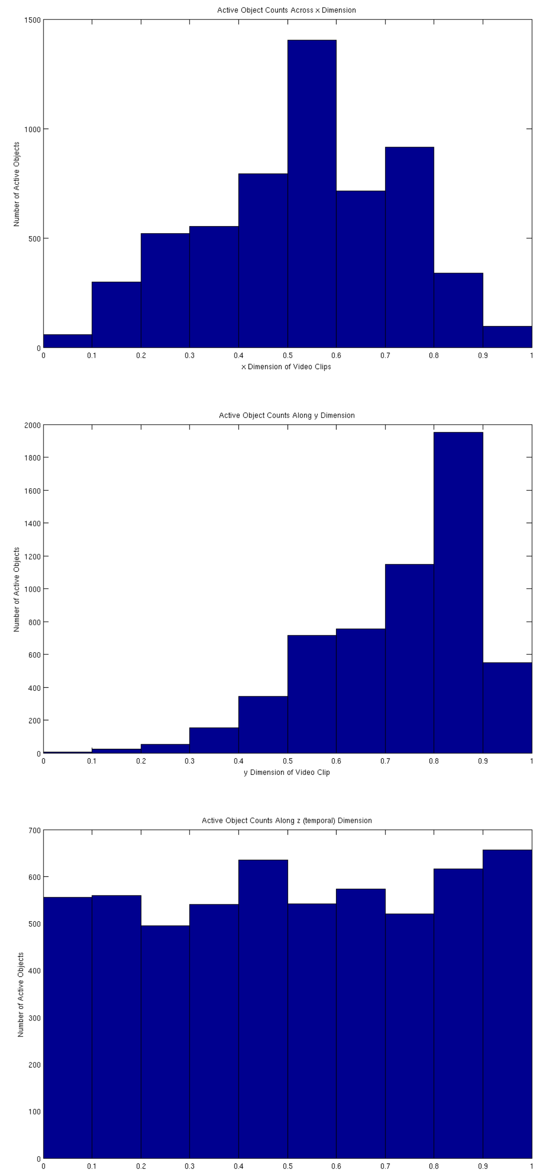

Figure 2. Histograms of counts of active objects across all 3 dimensions

3. Results

Mostly TODO at this point, but I want to discuss the following:

- graphic showing the effect on training error as the number of boosting rounds increases
- fix the number of boosting rounds, then show the effect of pool size
- graphic showing a few particularly discriminative partitions

The ADL dataset has been modified since the publication of [10]; because of this, running the published code gives slightly lower accuracy than the originally published numbers.

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

Object Type	bag	pyramid
O (published number)	24.7	32.7
AO (published number)	36.0	40.6
O (after modification)	26.6	29.0
AO (after modification)	34.9	36.9

Table 1. Overall classification accuracy on pre-segmented video clips before and after dataset modification

We create a heterogenous pool containing partitions of differing number of levels. We found that 3-level pyramids are often preferred to 2-level pyramids, and 4-level pyramids are often preferred to 3-level pyramids.

4. Conclusion and Future Work

We have presented an application of the well-known boosting framework with results outperforming 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 are "linear" in the sense that each cut is a linear equation of 3 variables, 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] 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

[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

[6] 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

[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

[8] 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

[9] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. *Computer Vision and Pattern Recognition, 2009.*, pages 2929–2936. 1

[10] 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

[11] 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

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