The International Journal Of Engineering And Science (IJES)
|| Volume || 3 || Issue || 4 || Pages || 25-32 || 2014 ||
ISSN (e): 2319 – 1813 ISSN (p): 2319 – 1805

Action Recognition from Web Data

[1,] Prof. Ms. R.R. Tuteja, [2,] Ms. Shakeeba S. Khan
[1, 2,] *Department of Computer Science & Engg, PRMIT&R Amravati*

----------ABSTRACT----------
This paper proposes a generic approach of understanding human actions in uncontrolled video. The idea is to use images collected from the Web to learn representations of actions and use this knowledge to automatically annotate actions in videos. We use LDA to obtain a more compact and discriminative feature representation and binary SVMs for classification. Our approach is unsupervised in the sense that it requires no human intervention other than the text querying. We present experimental evidence that using action images collected from the Web. To our best knowledge, this is one of the first studies that try to recognize actions from web images.

Keywords - *Annotate actions, Event Recognition, Generic database, Histogram of Oriented Gradients, tagging videos.*

Date of Submission: 14 April 2014 Date of Publication: 25 April 2014

I. INTRODUCTION

Human actions are Interaction with environment on specific purpose. Most research in human action recognition to date has focused on videos taken in controlled environments working with limited action vocabularies. Motion is a very important cue for recognizing actions. However, real world videos rarely exhibit such consistent and relatively simple settings. Instead, there is a wide range of environments where the actions can possibly take place, together with a large variety of possible actions that can be observed. Towards a more generic action recognition system, we propose to "learn" action representations from the Web and while doing this, improve the precision of the retrieved action images. Recent experiments show that action recognition based on key poses from single video frames is possible. But if the system is recognizing actions from real world videos this method require training with very large amounts of videos. Finding enough labeled video data that covers a diverse set of poses is quite challenging. Where else Web is a rich source of information, with many action images taken under various conditions and these are roughly annotated; i.e., the surrounding text is a clue used by search engines about the content of these images. Our apprehension is that one can use such a collection of images to learn certain pose instances of an action. Thus our work tries to join two lines of research "Internet vision" and "action recognition" together and makes it possible for one to benefit from the other. For our aim we need shape descriptors that are able to model the variations caused by high articulations. Our approach starts with employing a pose extractor, and then representing the pose via distribution of its rectangular regions. By using classification and feature reduction techniques, we test our representation via supervised and unsupervised settings.

II. WORKING OF THE SYSTEM

The system first gathers images by simply querying the name of the action on a web image search engine like Google or Gigablast. Based on the assumption that the set of retrieved images contains relevant images of the queried action, we construct a dataset of action images in an incremental manner. This yields a large image set, which includes images of actions taken from multiple viewpoints in a range of environments, performed by people who have varying body proportions and different clothing. The images mostly present the "key poses" since these images try to convey the action with a single pose. Following figure 1 describe our system.

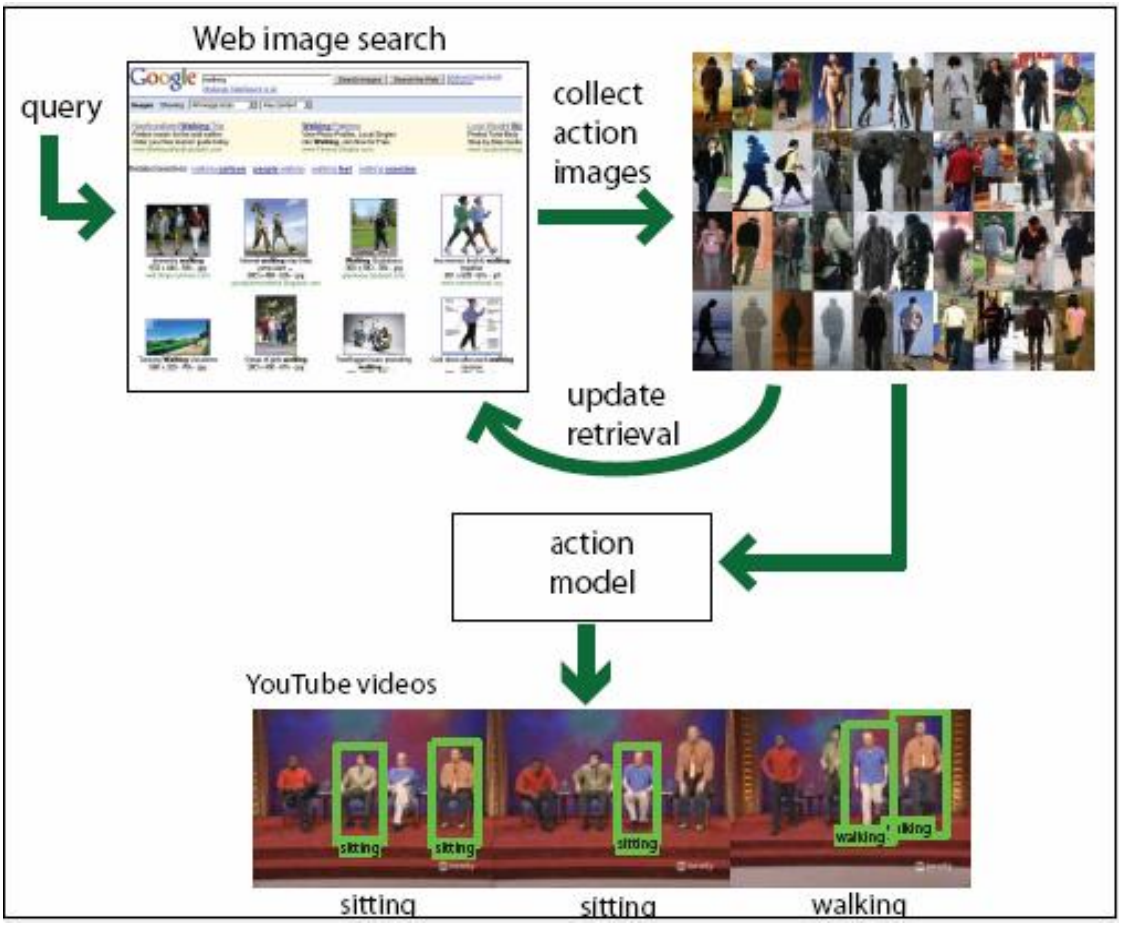

Fig 1. The overall system

We run an action query (such as "walking") through a web image search like Google or Gigablast. Then we incrementally build an action model (e.g., walking) and collect more images based on that model. We can then use the final image set for updating the retrieval result and for acquiring the action model for annotating poses in generic videos like those found on the YouTube web site. There are number of challenges that come across this representative data. First, the retrieved images are very noisy, since the Web is very diverse. For example, for a "walking" query, a search engine is likely to retrieve images of walking people along with images of walking shoes, dogs, signs, etc. Second, detecting and estimating the pose of humans in still images is more difficult than in videos, partly due to the background clutter and the lack of a foreground mask. In videos, foreground segmentation can exploit motion cues to great benefit. In still images, the only cue at hand is the appearance information and therefore, our model must address various challenges associated with different forms of appearance. One of the important strengths of our approach is that we can easily extend the vocabulary of actions, by simply making additional image search engine queries.

III. IMAGE REPRESENTATION

To begin, we are given the initial results of a keyword action query to an image search engine. First we extract the location of the human(s), for each of the retrieved images. If no humans are detected in an image, then that image is discarded. We use the implementation of Felzenswalb et al.'shuman detector which has been shown to be effective in detecting people in different poses.

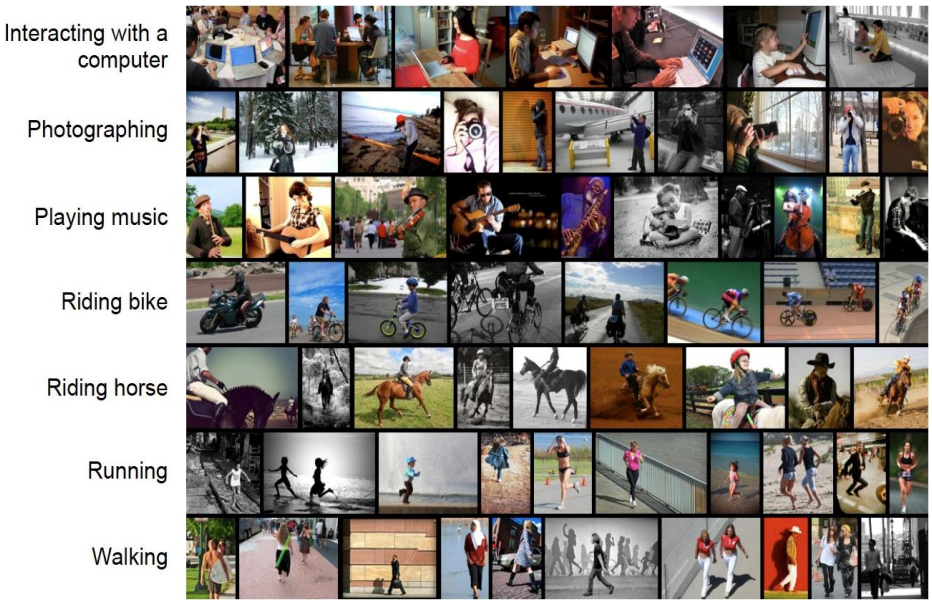

Fig 2. Some examples of collected images as the initial set for detecting actions

These show the output of the person detector. All are aligned by the head region (specified by the person detector). The rows correspond to actions "interacting with a computer, "photographing, "playing music," "riding bike," "riding horse," "running," "walking," respectively.

1.1 Head Alignment Step

The detected humans are not always centralized within their corresponding image. We solve this issue via an alignment step based on the head area response. Head detections are the most reliable parts of the detector, since there is high variance in the limb areas. So, for each image we take the detector's output for the head and update the image of the person so that the head area is positioned in the upper center of the image. Using this step, we achieve a rough alignment of the poses.

3.2 Feature Extraction Step

Once the humans are centralized, we extract an image descriptor for each detected area. The images collected from the web span a huge range of variability. In many cases, the background clutter impedes good pose estimation using state-of-the-art algorithms. Therefore, we need a descriptor which provides a good representation of the poses, and is able to cope with the background clutter. There are a number of algorithms for estimating human pose from a single image; however, we choose to avoid pose estimation altogether, mainly because: 1) pose estimation can be quite complex and can take a lot of processing time, and 2) most of the existing pose estimation algorithms require that the whole body must be visible.

Recently, researchers proposed new methods to address the more challenging event recognition task on video data sets captured under much less uncontrolled conditions, including movies and broadcast news videos Laptev et al. integrated local space-time features (i.e., Histograms of Oriented Gradient (HOG)), and SVM for action classification in movies, in order to locate the actions from movies. In most cases, a simple gradient filtering based HOG descriptor is affected significantly by noisy responses. Therefore, as an edge detector we use the probability of boundary (Pb) operator, which has been shown to perform well in delineating the object boundaries and then extract HOG features based on Pb responses. Although the outputs are by no means perfect, Pb tends to suppress small noises that can accumulate and dominate in HOG cells.

IV. BUILDING ACTION MODELS

After completion of the action query, person detection, and feature extraction steps, we have a set of images that depict instances of the queried action plus a large number of irrelevant images, which includes images of other actions and noise images. The next task is building our action model by removing non-relevant images from dataset.

1.2 Removing Non-Relevant Images from Dataset

An incremental learning procedure is used to detect and remove non-relevant images from the action image set. We start with the basic assumption that the first set of retrieved images is more likely to contain relevant ones. We take the first k = 20 images returned by each web source and combine these to form our initial training set. This set is still very noisy; a preliminary evaluation shows that only ≈ 35% − 50% of the images are of relevant actions. Also, these images contain people in various poses, and hence the dataset exhibits a multi-modal structure.

For incremental learning, we need a method that can give posterior probabilities, which we can then use to discriminate between action images with consistent poses of different viewpoints and noise images. One might consider using a density estimation based approach; however, such an approach has two major problems. First, it is hard to fit a density model because of non-relevant images and high variance of 2D features due to viewpoint changes. Second, action images may have similar contexts, such as people walking in a street or dancing in an indoor area. Consequently, it is likely that a density estimation procedure or a classifier without background information will generalize on the background features, such as the horizon line. We therefore follow a discriminative approach that provides estimates of posterior probabilities, but avoids these pitfalls. We force the classifier to generalize on features based on human pose rather than background (contextual) features.

We need a simple-enough classifier that learns just the common foreground features for a single action among different viewpoints and that is robust to the outliers in the foreground set. For this purpose, we use L1-regularized logistic regression, with the following probability model:

$$P(y = \pm 1|\mathbf{x}, \mathbf{w}) = \frac{1}{1 + exp(-y(\mathbf{w}^T\mathbf{x}))} \qquad (1)$$

Where x is the feature vector concatenated with 1 for the bias term, w is the weight vector and bias, and y is the class label. We train using L2-regularized logistic regression with the foreground set Fnoisy with labels yi = +1 and the background set B with labels yi = −1 at each iteration, by minimizing negative log using:

$$\min \sum_{i=1}^{N} log(1 + e^{(-y_i w^T x_i)}) + \frac{1}{2} w^T w, \qquad (2)$$

Where N is the number of training samples. Here L1-regularization provides implicit feature selection, L2-regularization is more suitable for our data at this step because of the multi-modality. Therefore, we use L2-regularization, where the noise is tolerated and the feature weights are preserved.

1.3 Incremental Model Update

This step starts with the initial classifier from the previous step; we iteratively go over the remaining set of retrieved images to build a larger dataset of action images. This is done by updating the dataset via selecting images with high posterior probability of being foreground, and retraining the logistic regression classifier. Since we will use the resulting set as the training set of the action model, the cost of introducing a false positive is much higher than leaving out some true positives. At each iteration step, the images with low posterior probability in the previous set are also removed in order to achieve high precision in the final dataset. We process the data by taking 10 pages of retrieved images at an iteration (typically ≈ 300 images) and terminate at around 100 pages (in total for each web query), resulting in around 10 iterations.

Using the above incremental procedure, we produce a cleaner image dataset for each action class. Given these datasets, we will train classifiers that discriminate between one particular action class and other action classes.

V. RECOGNIZING ACTIONS IN VIDEOS

After forming a dataset of action images and learning classifiers, we want to annotate actions in videos. This can be done by running the person detector in each video frame. Once the humans have been detected, then recognition involves: perturbing the image to account for errors in localizing the humans, tracking of detections across frames, and temporal smoothing of action labels.

1.4 Perturbation

In order to achieve better alignment of the test detections with the model, we extend the set of detections by applying small perturbations. For each human detection d_k, we apply two basic perturbations shift and scale. We shift the detection image to left and right, extend and shrink the size to get the candidate set of perturbations D, where

$$D = \{d_k, d_k^+, d_k^-, d_k^{left}, d_k^{right} \forall k \in K\}$$

We also take the mirror of these perturbations. Then, we apply our classifier on D. For each frame, we compute the posterior class probability P(at = c) at time t by marginalizing over the set of perturbations Dt. i.e.

$$P(a_t = c) = \sum_{d_k \in D_t} p(a_t = c, d_k) \qquad (3)$$

1.5 Tracking

Each frame can depict multiple people. We adopt a simple image based scheme for obtaining tracks of each person. We do this by initializing the tracker to the detections in the first frame. In consecutive frames, each new detection is added to the previous person track that has the closest spatial position. If detection cannot be associated with one of the existing tracks, a new track is initialized.

1.6 Smoothing

Due to noise in some frames and ambiguous intermediate poses, we expect to get several misclassifications. Hence to smooth these out, we use a dynamic programming approach and find the path with maximum classification probability in the person track based on action posteriors at each person's detection image. We assume a first-order Markov model and define the optimum path c = (c1, ..., cT) as

$$\arg\max_{\mathbf{c}} P(a_1 = c_1, ..., a_T = c_T | \mathbf{\Lambda}) =$$

$$\arg\max_{\mathbf{c}} p(a_1 = c_1) \prod_{t=2}^{T} (\mathbf{\Lambda}_{c_t, c_{t-1}} p(a_t = c_t)) \qquad (4)$$

where P(at = i) is the posterior probability for action i at time t ∈ 1, ..., T. $\mathbf{\Lambda}$ is the predefined transition probability matrix defined as follows

$$\mathbf{\Lambda}_{i,j} = \left\{ \begin{array}{l} 1/z \ , \ if \ i = j \\ \sigma/z \ , \ if \ i \neq j \end{array} \right\} \qquad (5)$$

where z is the normalization factor so that

$$\sum_j \mathbf{\Lambda}_{i,j} = 1.$$

We set σ = 0.25 to reduce rapid fluctuations between actions. This definition corresponds to building a graph with a node for each action at each frame in the track. We add an edge between all pairs of nodes between each consecutive frame in the track. Each edge is denoted by;

$$\mathbf{\Lambda}_{c_t, c_{t-1}} p(a_t = c_t)$$

Each edge represents the probability of selecting the action ct given the previous action c_{t-1}. We obtain the optimum path by using the Viterbi algorithm.

VI. EXPERIMENTAL RESULTS

6.1 Dataset

For recognition of actions we collected a dataset. For this purpose we utilize several query words related to each action on web search engines like Google and Yahoo Image Search. For querying each action, we combine the action word (e.g. walking) with pronouns like "person" and "people", in order to retrieve more relevant images. We collected images for seven different actions: interacting with a computer, photographing, playing music, riding bike, riding horse, running, walking.

6.2 Retrieval of Action Image

As our first experiment, we test if the incremental update procedure is helpful in increasing the precision rate of the retrieved images. Since our aim is to use the collected set of images as a training set for videos, we require high precision in the collected image set; therefore, we sacrifice some of the recall by setting the thresholds high in incremental model update. In the end of data collection step, the final set contains 401 interacting with a computer, 390 photographing, 561 playing music, 162 riding bike, 250 riding horse, 384 running and 307 walking images. The precision of all collected images are at recall level 15%. Since we want to evaluate our system independent of the choice of the person detector, initial queried images are filtered by the person detector. The precision rates improve up to 15% for the actions interacting with a computer, photographing, running, walking, riding horse and near 10% for playing music. The improvement is minor (3%) for the riding bike action. This is due to the high level of noise in the initial set of retrieved images. We observed that if the amount of no relevant images dominates the initial set, it becomes very difficult for our model to differentiate noise from relevant images and therefore, the resulting set includes a significant number of non-relevant images.

VII. ANNOTATION OF VIDEO

Our second experiment involves labeling the actions in videos by using the action models we form over web images. Besides our approach, for labeling actions in videos, we also tried two different classifiers: one-vs-all SVMs and multi-class SVMs. Both use RBF kernels and are trained using bootstrapping in order to handle the noise. We apply annotation on following set of frames from YouTube videos.

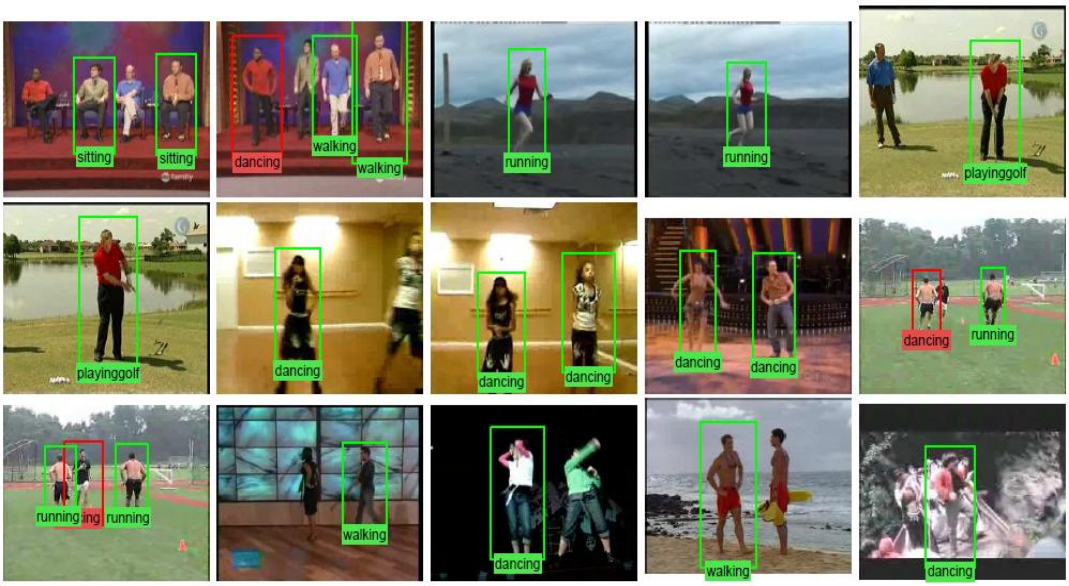

Fig 3. Example annotated frames from YouTube videos

We run the person detector on these frames and create separate tracks for each person. Then, by applying our action models learnt from web images and using temporal smoothing over each track, we get the final annotations. Our method inherently handles multiple people and multiple actions. Correct classifications are shown in green and misclassifications are in red.

We present the comparison of different classifiers and effects of smoothing procedure on YouTube action annotations in following table 1.

	No Smoothing	Smoothing
ovaSVM	55.03	57.79
multiSVM	59.35	68.60
multiLR_NMF	63.61	75.87

Table 1. Comparison of different classifiers and effects of smoothing on YouTube action annotations.

Chance level for each action is 20%. Our proposed method multiLR NMF outperforms SVMs both with or without smoothing. By the results, we observe that learning multiple local classifiers on poses is better than a single classifier for each action. Also, we see that temporal smoothing helps a lot and without smoothing, minor differences amongst posteriors affects the total labeling seriously. Figure 7 shows the confusion matrix for our method on this dataset.

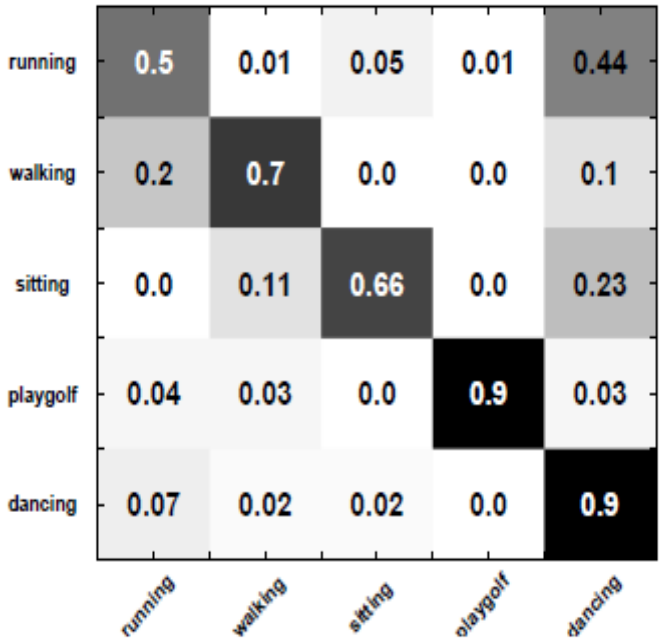

Fig. 4. Per frame confusion matrix for action annotation on YouTube videos.

Most of the confusion occurs between running and dancing actions. This is not surprising, since some of the dancing poses involve a running pose for the legs (e.g. in the "twist" dance, the legs are bend like running), therefore some confusion is inevitable. Moreover, when the arms are bent, it is quite easy for walking to be mixed up with dancing. This is the problem of composition of actions and should be handled as a separate problem.

VIII. CONCLUSION

In this paper, our aim is not to compete with action recognition algorithms that work purely on videos, but show with experimental evidence that web images can be used to annotate the videos taken in uncontrolled environments. We also address the problem of retrieving action images from the web and using them to annotate generic and challenging videos. The results are quite interesting; neither of the domains is controlled, yet, we can transfer the knowledge from the web images to annotate YouTube videos. Moreover, the approach we present here has some important features. The only supervision it has is from text queries. No more human intervention is needed. It handles multiple people and multiple actions inherently. What is more appealing is that it is easily extensible; run a new query for action 'x', clean the images and build the model, and you have a new action model. There is room for improvement. Action image retrieval brings a set of challenges: First, the data retrieved is quite noisy and consists of multiple modes due to the variance in poses and in people. This makes the problem more challenging and requires special attention. Second, from the retrieval point of view, the regular cues (like color, static templates) used in content-based retrieval of objects are likely to fail in this domain. We therefore use HOG features operating on Pb responses for describing action images. Additional pose cues will likely improve the performance. On the other hand, the retrieved data is quite diverse, and using this data effectively can be very beneficial. We have seen that the action images are also composite in nature (running and waving, for example), like the actions in video. Future work includes the exploration of this composition and improving methods for dealing with noise and multi-modality.

REFERENCES

[1] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In CVPR, 2008.
[2] N. Ikizler, R. G. Cinbis, and P. Duygulu. Recognizing actions in still images. In ICPR, 2008.
[3] N. Ikizler and D. Forsyth. Searching for complex human activities with no visual examples. IJCV, 80(3), 2008.
[4] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In ICCV, 2007.
[5] Y. Ke, R. Sukthankar, andM. Hebert. Event detection in crowded videos. In Int. Conf. on Computer Vision, 2007.
[6] T. K. Kim, S. F. Wong, and R. Cipolla. Tensor canonical correlation analysis for action classification. CVPR, 2007.
[7] I.Laptev,M.Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In CVPR, 2008.
[8] D. Lee and H. Seung. Algorithms for non-negative matrix factorization. In NIPS, 2001.
[9] F. Schroff, A. Criminisi, and A. Zisserman. Harvesting image databases from the web. In ICCV, 2007.
[10] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In ICPR, 2004.
[11] C. Thurau and V. Hlavac. Pose primitive based human action recognition in videos or still images. In CVPR, 2008.
[12] D. Tran and A. Sorokin. Human activity recognition with metric learning. In ECCV, 2008.
[13] S. Vijayanarasimhan and K. Grauman. Keywords to visual categories: Multiple-instance learning for weakly supervised object categorization. In CVPR, 2008.
[14] G. Wang and D. Forsyth. Object image retrieval by exploiting online knowledge resources. In CVPR, 2008.
[15] Y.Wang, H. Jiang, M. S. Drew, Z.-N. Li, and G. Mori. Unsupervised discovery of action classes. In CVPR, 2006.
[16] D. Weinland and E. Boyer. Action recognition using exemplarbased embedding. In CVPR, 2008.
[17] S. Zanetti, L. Zelnik-Manor, and P. Perona. A walk through the web's video clips. In First IEEE Workshop on Internet Vision, in CVPR, 2008.