Learning Actions From the Web

Nazli Ikizler-Cinbis, R. Gokberk Cinbis, Stan Sclaroff
Computer Science Department, Boston University, Boston, MA

{ncinbis,gcinbis,sclaroff}@cs.bu.edu

Abstract

This paper proposes a generic method for action recognition in uncontrolled videos. 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. Our approach is unsupervised in the sense that it requires no human intervention other than the text querying. Its benefits are two-fold: 1) we can improve retrieval of action images, and 2) we can collect a large generic database of action poses, which can then be used in tagging videos. We present experimental evidence that using action images collected from the Web, annotating actions is possible.

1. Introduction

Most research in human action recognition to date has focused on videos taken in controlled environments working with limited action vocabularies. Standard datasets, like KTH [21] and Weizmann [2], formed for this purpose are well-explored in various studies, e.g. [17, 8, 10, 19, 23] and many more. 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. The main observation behind our approach is as follows. Recent works [19, 22, 27] show that action recognition based on key poses from single video frames is possible. However, these methods require training with large amounts of video, especially if the system is to recognize actions in real world videos. Finding enough labeled video data that covers a diverse set of poses is quite challenging. The Web, on the other hand, 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 intuition is that one can use
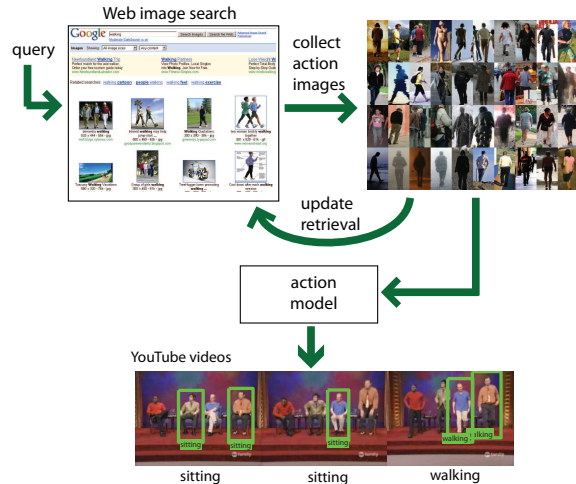

Figure 1. The overall system. We run an action query (such as "walking") through a web image search like Google or Yahoo. Then we incrementally build an action model (e.g., walking) and collect more images based on that model. We can then use the final image set for updating the retrieval result and for acquiring the action model for annotating poses in generic videos like those found on the YouTube web site.

such a collection of images to learn certain pose instances of an action. By doing this, 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.

Figure 1 illustrates our system. The system first gathers images by simply querying the name of the action on a web image search engine like Google or Yahoo. 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.

There are challenges that come at the expense of this broad and 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. Our method must perform well in the presence of such noise. 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.

We use the resulting dataset to annotate actions in videos of uncontrolled environments, like YouTube videos. 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.

Models trained with image data, of course, will be inferior to action models trained on videos solely; however, these models can serve as a basis for pruning the possible set of actions in a given video. Some actions like "sitting down" and "standing up" are impossible to distinguish just from 2D images – especially if the pose is a middle stage of the action. For this reason, in this work, we restrict our domain to actions which have characteristics that can be identifiable from a single monocular image, such as "running," "walking," "sitting," etc.

Our main contributions are:

- addressing the problem of action image retrieval and proposing a system which incrementally collects action images from the Web by simple text querying,
- building action models by using the noisy set of images in an unsupervised fashion, and
- using the models to annotate human actions in uncontrolled videos, such as YouTube videos.

Our method first collects an initial image set for each action by querying the web. For the initial set of images retrieved for each action we fit a logistic regression classifier to discriminate the foreground features of the related action from the background. Using this initial classifier, we then incrementally collect more action images and, at the same time, refine our model. This iterative process yields a more "cleaned" image set for that action where a good many of the non-relevant images are removed. We use non-negative matrix factorization on this set to find the different pose clusters for that action. We then train separate local action classifiers for each group of images and use these classifiers to annotate the poses in YouTube videos.

2. Related Work

Content-based image retrieval research has focused primarily on the retrieval of certain (and mostly unarticulated) objects; see [25, 13, 20, 24] for some recent work. These works mostly rely on the knowledge of object classes and generic feature extraction. To our knowledge, no prior work has dealt with the retrieval of action images. Moreover, while these works provide methods to collect datasets from the web, the final datasets are not mostly leveraged for further tasks. We accomplish this by making use of the collected dataset in a separate real-world domain.

Recognizing actions in still images is a widely ignored problem in computer vision. Wang, et al. [26] utilize deformable template matching for computing the distance between human poses, so that similar poses can be grouped together. Ikizler et al. apply a rectangle-based pose descriptor to static images [6]. Thurau and Hlavac [22] approach the problem by using non-negative matrix factorization on pose primitives. In essence, they learn the pose primitives from non-cluttered videos and apply these to images and find the closest pose. In this work, we consider the opposite case: we learn poses from images, fit an action model and use this to classify actions in the cluttered videos.

Person detection and pose estimation in still images, on the other hand, is widely studied. Some recent work includes part-based detectors [5, 4]. Bissacco, et al. [1] use an LDA approach for estimating the pose in images with shared backgrounds. Okada and Soatto [18] present a system to estimate the 3D pose of the persons from cluttered images using SVMs and piecewise linear regressors. These person detection and pose estimation methods can be used as first steps of action recognition in still images.

For recognizing actions in videos, there is a vast amount of work in the literature. Amongst these, some approaches require static or uniform backgrounds and easily extractable silhouettes [2, 23] and some approaches work with more realistic videos in the presence of background clutter [9, 15, 11, 7]. Our work is in this second category, but it does not require video data for training; and it is intended to build a bridge between Internet vision and action recognition.

Little work has been done with generic videos like YouTube videos, where the resolution is low and the recording environment is nonuniform. Zanetti, et al. [28] recently noted the challenges of working with web videos. Niebles, et al. [16] present a method to detect moving people in such videos. Tran, et al. [23] detect actions in YouTube Badminton videos with fairly static backgrounds. Our method is applicable to videos with a broader range of settings.

3. Image Representation

To begin, we are given the initial results of a keyword query to an image search engine. For each of the retrieved images, we first extract the location of the human(s). If no humans are detected in an image, then that image is discarded. We use the implementation of Felzenswalb et al.'s human detector [5], which has been shown to be effective in detecting people in different poses.

Figure 2. Some examples of collected images as the initial set. 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 "running," "walking," "sitting," "playing golf," "dancing" respectively. The irrelevant images are bordered with red. Notice the high amount of variability and noise images in this set.

Head Alignment. The detected humans are not always centralized within their corresponding bounding box. 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 bounding box of the person so that the head area is positioned in the upper center of the bounding box. Using this step, we achieve a rough alignment of the poses. **Feature Extraction.** 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 (see Fig 2). 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. This is tricky, since we neither have the silhouette, nor the perfect bounding box for the humans in the images. 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, which is not always the case in our problem. For some actions, the web search may yield images where only half or some of the body is visible (see Fig 2).

In human detection, the Histogram of Oriented Gradients (HOG) has been shown to be successful [3]; however, the clutter in web images makes it difficult to obtain a useful

pose description. 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 [14] and then extract HOG features based on Pb responses. Although the outputs are by no means perfect, Pb tend to suppress small noises that can accumulate and dominate in HOG cells. Figure 3 shows an example comparison between HOGs obtained using a $[-1 \ 0 \ +1]^\mathrm{T}$ filter vs the Pb operator. In our implementation, we resize each image to 128×64 and then extract PbHOGs in 8×8 cells. Our final feature vector is the 1152-dimensional normalized HOG cell vector plus the total magnitude of the responses.

4. Building Action Models

After completion of the 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 to obtain a less noisy dataset that can serve as a training set in building our action model.

4.1. Removing Non-Relevant Images from Dataset

We use an incremental learning procedure 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

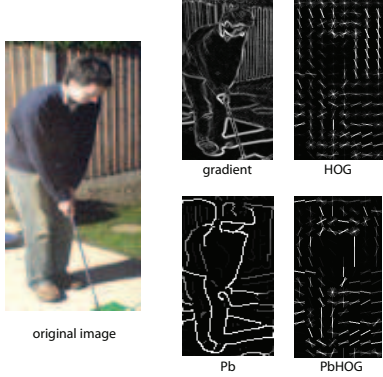gradient   HOG

original image

Pb   PbHOG

Figure 3. In order to decrease the effect of background clutter, we first apply the Pb operator to the image and extract the HOG features on the responses, forming our PbHOG descriptor. As seen in this figure, the straight background edges caused by the texture of the back wall are reduced and the HOGs are more concentrated towards the foreground person.

combine these to form our initial training set. This set is still very noisy; a preliminary evaluation shows that only $\approx 40\% - 55\%$ of the images are of relevant actions. Moreover, these images contain people in various poses; thus, 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. For this purpose, we use a background set, which is obtained by automatically selecting random bounding boxes where the human detector does not fire.

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 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 $\mathbf{x}$ is the feature vector concatenated with 1 for the bias term, $\mathbf{w}$ is the weight vector and bias, and y is the class label. We train using L2-regularized logistic regression with the foreground set F_{noisy} with labels $y_i = +1$ and the background set B with labels $y_i = -1$ at each iteration, by minimizing negative log-likelihood:

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

where N is the number of training samples. In our experiments, we have found that this logistic regression classifier effectively selects the foreground features without overfitting to the noisy training set.

Note that although L1-regularization provides implicit feature selection, L2-regularization is more suitable for our data at this step because of the multi-modality. L1-regularization tends to force sparsity in the feature domain and due to the high level of articulation, it tends to suppress important detail in high variance areas, and the weights tend to shift towards more stable regions of the body, like the head and shoulder area. Therefore, we use L2-regularization, where the noise is tolerated and the feature weights are preserved.

4.2. Incremental Model Update

Starting 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. We do this by updating the dataset via selecting images that have high posterior probability of being foreground, and retraining the logistic regression classifier. Note that, 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. So we set the inclusion threshold (τ_I) to be as high as 0.99. 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 adaptively lower (by steps of 0.01) this exclusion threshold (τ_E) so that the model is able to accommodate higher variations as the set extends. 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. This process is summarized in Alg. 1.

4.3. Learning Classifiers for Action Discrimination

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.

Since action images are taken from multiple viewpoints, the dataset for each action tends to be multi-modal. To deal with this multi-modality, we propose to first cluster the data for an action into multiple modes via non-negative matrix

Algorithm 1 Incremental collection of action images for a single action

1: Run query set Q in Google or Yahoo image search
2: Preprocess retrieved images:
3:    Run person detector
4:    Align images w.r.t. head positions
5:    Extract PbHOG features
6: Build action model using $S_{a_i} = \{$first n images$\}$
7: **while** more images remain **do**
8:    Compute $p(y|x)$ for next n images
9:    Include all new images with $p(y|x) > \tau_I$ to S_{a_i}
10:    Retrain logistic regression classifier on S_{a_i}
11:    Compute $p(y|x)$ for all $x \in S_{a_i}$
12:    Exclude images with $p(y|x) < \tau_E$ from S_{a_i}
13: **end while**

factorization (NMF) [12]. NMF decomposes the data into non-negative additive components on which we can then form our local classifiers. For each action set, we factor out the basis vectors $\mathbf{W}$ where $\mathbf{X} = \mathbf{WH}$. For finding $\mathbf{W}$ and $\mathbf{H}$, we minimize the divergence $D(\mathbf{X}||\mathbf{WH})$ with respect to $\mathbf{W}$ and $\mathbf{H}$ such that $\mathbf{W}, \mathbf{H} > 0$ using the multiplicative update rules defined as $\mathbf{H}_{a\mu} \leftarrow \mathbf{H}_{a\mu} \frac{\sum_i \mathbf{W}_{ia} \mathbf{X}_{i\mu}/(\mathbf{WH})_{i\mu}}{\sum_k \mathbf{W}_{ka}}$ and $\mathbf{W}_{ia} \leftarrow \mathbf{W}_{ia} \frac{\sum_\mu \mathbf{H}_{a\mu} \mathbf{X}_{i\mu}/(\mathbf{WH})_{i\mu}}{\sum_v \mathbf{H}_{av}}$ [12].

The basis vectors found by NMF are the components that depict the PbHOG pattern associated with each mode in the data. Fig. 4 shows example components.

We then cluster the images based on their $\mathbf{H}$, the encoding vectors. We do this by grouping the images with the same maximum response basis vector together. For number of components, we choose $k = 5$. This is based on the number of common viewpoints for each action in the dataset (2 for lateral views, 2 for $\mp 45°$ views and 1 for frontal/back view). By using a small the number of components (e.g. 5), we are able to discover poses within the data, rather than the parts of the human body. Figure 5 shows example clusters for the running action. Despite not being noise-free, each of these clusters roughly corresponds to different viewpoints/poses of actions.

We train separate local logistic regression classifiers on different clusters of each action in an one-vs-all manner. Since the viewpoint/pose variance within each cluster is lower, the local classifiers become more robust to outliers in the training set. For training, we use the background images and images of other actions as the negative class. To classify a new image, we choose the local classifier with highest posterior probability for an action, and then use the soft-max operator on the posteriors of all actions. The final posterior probability for each action a_i is defined as $p(a_i|\mathbf{x}) = \frac{\exp(\max_k w_{i,k}^T \mathbf{x})}{\sum_j \exp(\max_k w_{j,k}^T \mathbf{x})}$ where $w_{i,k}$ is the weight vector (and bias) for i^{th} action's k^{th} cluster.

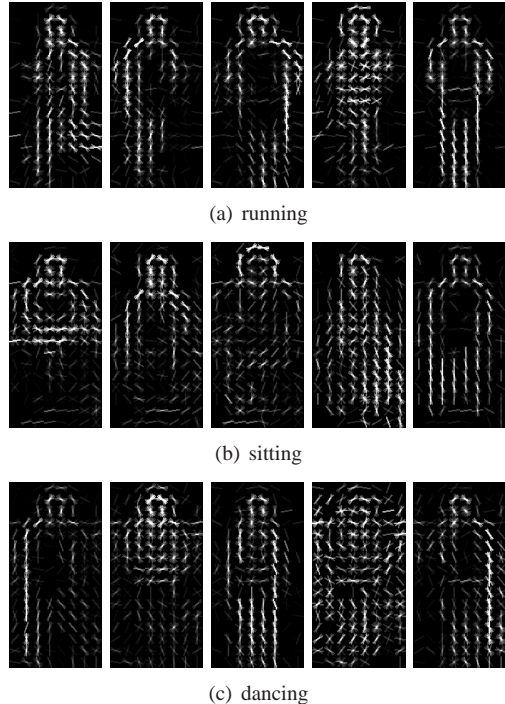
(a) running

(b) sitting

(c) dancing

Figure 4. Basis vectors of the actions found by NMF. Most of the time, NMF is successful in identifying certain poses of each action. However, some of the basis vectors are scrambled due to the high variance of the poses, as with the dancing action.

5. Recognizing Actions in Video

Having formed a dataset of action images and then learned classifiers, we want to annotate actions in videos. To do this, we first run the person detector [5] in each video frame. Once the humans have been detected, then recognition involves: perturbing the bounding box to account for errors in localizing the humans, tracking of detections across frames, and temporal smoothing of action labels.

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 bounding box to left and right, extend and shrink the size to get the candidate set of perturbations $D = \{d_k, d_k^+, d_k^-, d_k^{left}, d_k^{right} \forall k \in K\}$. We also take the mirror of these perturbations. Then, we apply our classifier on D. For each frame, we compute the posterior class probability $P(a_t = c)$ at time t by marginalizing over the set of perturbations D_t. That is,

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

Tracking. Each frame can depict multiple people. We adopt a simple bounding box 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,

Figure 5. Example images from clusters of the "running" action formed after NMF step. Each of these clusters corresponds to NMF basis vectors shown in Fig 4(a). Although some noise images remain, the clusters mostly include certain poses/viewpoints of the action.

each new detection is added to the previous person track that has the closest spatial position. If a detection cannot be associated with one of the existing tracks, a new track is initialized. This simple scheme has several, well-known weaknesses. For example, it may easily fail if the camera is moving extremely fast. However, it turns out that this method provides sufficient accuracy (with a few extra tracks) for our purpose.

Smoothing. Due to noise in some frames and ambiguous intermediate poses, we expect to get several misclassifications. 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 detection bounding box. We assume a first-order Markov model and define the optimum path $\mathbf{c} = (c_1, ..., c_T)$ as

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

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

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

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

where z is the normalization factor so that $\sum_j \mathbf{\Lambda}_{i,j} = 1$. We set $\sigma = 0.25$ 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 $\mathbf{\Lambda}_{c_t, c_{t-1}} p(a_t = c_t)$ represents the probability of selecting the action c_t given the previous action c_{t-1}. We obtain the optimum path by using the Viterbi algorithm.

6. Experimental Evaluation

We evaluate our method in two applications: improving the precision of action images retrieved from the Web and annotation of actions in YouTube videos.

6.1. Dataset

To collect the image dataset, 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. running) with pronouns like "person" and "people", in order to retrieve more relevant images. We collected images for five different actions: running, walking, sitting, playing golf and dancing.

We use the video dataset provided by Niebles et al. [16] for testing our action models. This dataset consists of YouTube videos that have considerably low resolution and moving cameras. This dataset has been used for person detection purposes and does not include action annotations. We annotated 11 videos from this dataset, 775 frames in total, which includes the five actions in combination. Note that each video may contain more than one action, and since we will do frame by frame annotation, our method does not require action segmentation prior to application.

6.2. Action Image Retrieval

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 (Section 4.2). In the end of data collection step, the final set contains 384 running, 307 walking, 313 sitting, 162 playing golf and 561 dancing images. The precision rates for each action at 15% recall (following [20, 24]) are shown in Fig. 6. Since we want to evaluate our system independent of the choice of the person detec-
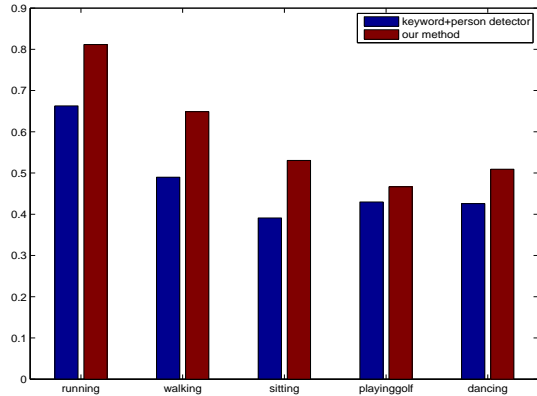
Figure 6. The precision of collected images at recall level 15%. Our method improves upon the precision of web image search in all actions. The lowest improvement occurs in playing golf action due to the high rate of noise images in the initial retrieved set.

tor, initial queried images are filtered by the person detector. The precision rates improve up to 15% for the actions running, walking, sitting and near 10% for dancing. The improvement is minor (3%) for the playing golf action. This is due to the high level of noise in the initial set of retrieved images (see Fig. 2). We observed that if the amount of non-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.

6.3. Video Annotation

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 present the comparison of these techniques and the effect of our smoothing procedure in Table 1. 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. 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 (see Figure 8 image on row 1 column 2). This is the problem of composition of actions [7] and should be handled as a seperate problem.

	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. The percentages shown are the average accuracies per frame.
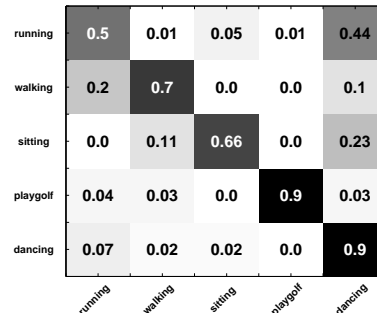

Figure 7. Per frame confusion matrix for action annotation on YouTube videos. Most of the confusion occurs between dancing and running actions. This is not surprising, because some of the dancing poses look very similar to running.

7. Discussion and Conclusion

In this work, we address the problem of retrieving action images from the web and using them to annotate generic and challenging videos. 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. 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.

In addition, 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 na-
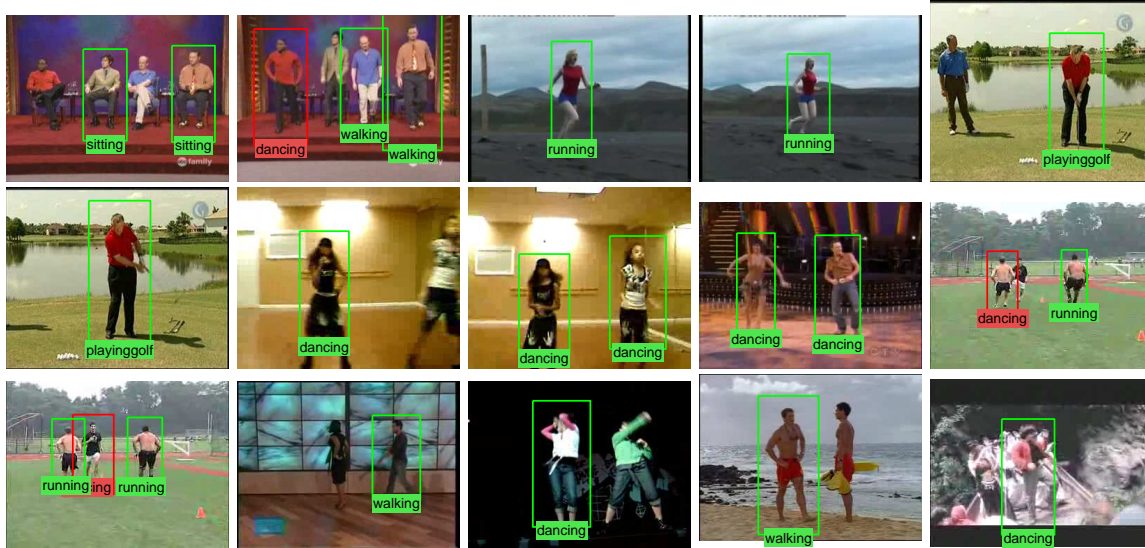
Figure 8. Example annotated frames from YouTube videos of Niebles et al. [16]. We run the person detector [5] on these frames and create separate tracks for each person. Then, by applying our action models learnt from web images and using temporal smoothing over each track, we get the final annotations. Note that, our method inherently handles multiple people and multiple actions. Correct classifications are shown in green and misclassifications are in red.

ture (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] A. Bissacco, M.-H. Yang, and S. Soatto. Detecting humans via their pose. In *NIPS*, 2006. 2

[2] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, 2005. 1, 2

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages I: 886–893, 2005. 3

[4] P. Dollár, B. Babenko, S. Belongie, P. Perona, and Z. Tu. Multiple component learning for object detection. In *ECCV*, 2008. 2

[5] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008. 2, 5, 8

[6] N. Ikizler, R. G. Cinbis, and P. Duygulu. Recognizing actions in still images. In *ICPR*, 2008. 2

[7] N. Ikizler and D. Forsyth. Searching for complex human activities with no visual examples. *IJCV*, 80(3), 2008. 2, 7

[8] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In *ICCV*, 2007. 1

[9] Y. Ke, R. Sukthankar, and M. Hebert. Event detection in crowded videos. In *Int. Conf. on Computer Vision*, 2007. 2

[10] T. K. Kim, S. F. Wong, and R. Cipolla. Tensor canonical correlation analysis for action classification. *CVPR*, 2007. 1

[11] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008. 2

[12] D. Lee and H. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, 2001. 5

[13] L.-J. Li, G. Wang, and L. Fei-Fei. Optimol: automatic object picture collection via incremental model learning. In *CVPR*, 2007. 2

[14] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color and texture cues. *PAMI*, 26, 2004. 3

[15] K. Mikolajczyk and H. Uemura. Action recognition with motion-appearance vocabulary forest. In *CVPR*, 2008. 2

[16] J. C. Niebles, B. Han, A. Ferencz, and L. Fei-Fei. Extracting moving people from internet videos. In *ECCV*, 2008. 2, 6, 8

[17] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. In *BMVC*, 2006. 1

[18] R. Okada and S. Soatto. Relevant feature selection for human pose estimation and localization in cluttered images. In *ECCV*, 2008. 2

[19] K. Schindler and L. van Gool. Action snippets: How many frames does human action recognition require? In *CVPR*, 2008. 1

[20] F. Schroff, A. Criminisi, and A. Zisserman. Harvesting image databases from the web. In *ICCV*, 2007. 2, 6

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

[22] C. Thurau and V. Hlavac. Pose primitive based human action recognition in videos or still images. In *CVPR*, 2008. 1, 2

[23] D. Tran and A. Sorokin. Human activity recognition with metric learning. In *ECCV*, 2008. 1, 2

[24] S. Vijayanarasimhan and K. Grauman. Keywords to visual categories: Multiple-instance learning for weakly supervised object categorization. In *CVPR*, 2008. 2, 6

[25] G. Wang and D. Forsyth. Object image retrieval by exploiting online knowledge resources. In *CVPR*, 2008. 2

[26] Y. Wang, H. Jiang, M. S. Drew, Z.-N. Li, and G. Mori. Unsupervised discovery of action classes. In *CVPR*, 2006. 2

[27] D. Weinland and E. Boyer. Action recognition using exemplar-based embedding. In *CVPR*, 2008. 1

[28] 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. 2