Watching Unlabeled Video Helps Learn New Human Actions from Very Few Labeled Snapshots

Chao-Yeh Chen and Kristen Grauman
University of Texas at Austin
chaoyeh@cs.utexas.edu, grauman@cs.utexas.edu

Abstract

We propose an approach to learn action categories from static images that leverages prior observations of generic human motion to augment its training process. Using unlabeled video containing various human activities, the system first learns how body pose tends to change locally in time. Then, given a small number of labeled static images, it uses that model to extrapolate beyond the given exemplars and generate "synthetic" training examples—poses that could link the observed images and/or immediately precede or follow them in time. In this way, we expand the training set without requiring additional manually labeled examples. We explore both example-based and manifold-based methods to implement our idea. Applying our approach to recognize actions in both images and video, we show it enhances a state-of-the-art technique when very few labeled training examples are available.

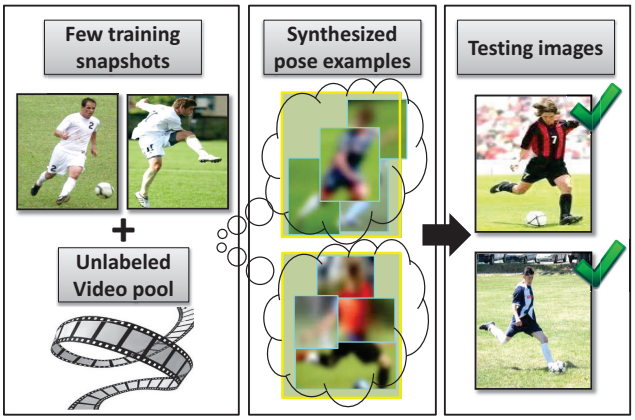

Figure 1. Our approach learns about human pose dynamics from unlabeled video, and then leverages that knowledge to train novel action categories from very few static snapshots. The snapshots and video (left) are used together to extrapolate "synthetic" poses relevant to that category (center), augmenting the training set. This leads to better generalization at test time (right), especially when test poses vary from the given snapshots.

1. Introduction

Action recognition is a challenging vision problem with applications in video search, surveillance, auto-tagging, and human-computer interfaces. While researchers have trained activity models from video data for many years, there is increasing interest in methods that can learn an action category from *static* image snapshots [17, 33, 34, 7, 36, 35]. Learning and predicting actions from an image (rather than a video) is appealing for several reasons. First, labeling training videos is more expensive; it requires more elaborate annotation interfaces, more work by annotators, and it can even be ambiguous in terms of when the action starts and ends. Second, collecting "staged" videos of activity can be tricky to do realistically; arguably, it is more manageable to capture realistic individual photos. Third, the ability to infer actions from static inputs has potential to aid object and scene recognition, since all three can serve as mutual context.

However, training an action recognition system with static snapshots presents its own challenges. How can we be assured to adequately cover the space of an action with an array of training photos? If benchmark datasets restrict train and test data to "canonical" instants of the action (e.g., the tennis player with the racquet outstretched), how will the methods generalize when faced with less exaggerated views in the wild? Can one photo sufficiently convey what is really a spatio-temporal event?

With these questions in mind, we contrast human capabilities with current methods. People can understand a human activity by looking at just a few static snapshots, yet today's systems typically require hundreds of such exemplars to learn an action category well. Human viewers have an important advantage, however: prior knowledge of how human poses tend to vary in time. This undoubtedly helps "fill the gaps" between a sparse set of snapshots, and thereby improves generalization. See Figure 1.

Building on this intuition, we propose an approach to

1063-6919/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPR.2013.80

learn action categories from a small number of static images by leveraging *prior observations of generic human motion* to augment the training process. Given unlabeled video, the system first learns how body pose changes over time. We assume this video has some human activity in it, and that humans are often detectable when present, but otherwise make no assumptions about *which* actions are present in the data. Then, given a small set of labeled images for an action category, the system uses the generic knowledge obtained from watching the video to extrapolate beyond those exemplars during training. In particular, it augments its labeled set with "synthetic" examples, which depict poses that could immediately precede or follow the given examples in time. In this way, we expand the training set without requiring additional manually labeled examples.

We explore two ways to implement our idea. The first uses an example-based representation of pose dynamics; we match the labeled training images to unlabeled video frames based on their pose similarity, and then augment the training set with the poses appearing before and after the matched frames. The second technique uses a manifold-based representation; we learn a nonlinear manifold over body poses, relying on the temporal nearness of the video frames to establish which should maintain proximity. Then, we map the static training instances to the manifold, and explore their neighborhoods on the manifold to augment the training set. In both cases, we adopt a part-based representation of pose, and use domain adaptation to account for the mismatch between the source images and the unlabeled video. We show that our synthetic expansions to the training set yield more accurate predictions, especially when labeled data is quite sparse. Notably, the gains come at no additional labeling cost, since we make no assumptions about which actions appear in the unlabeled video.

Limited prior work considers ways to synthetically pad training sets for recognition. This includes the now commonplace trick of inserting mirrored images for training object detectors (e.g., [20]), as well as the use of graphics software to generate images of people (often driven by mocap) with variable clothing and body sizes to train pose estimators [26, 12, 27]. We also share the goal of expanding the training pool for virtually no cost. However, whereas prior methods aim to introduce variation in (instantaneous) *appearance*, ours aims to imagine the variation *over time*. Furthermore, the source of our new examples is data-driven, not parameterized by hand.

We demonstrate the proposed approach to recognize actions in both static images and videos from multiple challenging datasets. The results show that by letting the system first "watch" generic video, it can successfully infer additional plausible poses that bolster training. For our target scenario where training examples are very few, our approach outperforms both a method limited to the original static exemplars, as well as alternative methods to pad the data by introducing appearance variation.

2. Related Work

Activity recognition and human motion analysis have a rich literature [1]. To learn activities from video, earlier work emphasized tracking and explicit body-part models (e.g., [19, 23, 22]). In parallel, many methods to estimate body pose have been developed, including techniques using nonlinear manifolds to represent the complex space of joint configurations [12, 32, 3, 16, 28, 29]; in contrast to our work, such methods assume silhouette (background-subtracted) inputs and/or derive models from mocap data, and are often intended for motion synthesis applications. More recently, researchers have considered how activity classes can be learned directly from lower-level spatio-temporal appearance and motion features—for example, based on bag-of-words models for video (e.g., [15, 31]). By sidestepping tracking and pose, this general strategy offers robustness and can leverage strong learning algorithms; on the other hand, the lack of top-down cues suggests more data is critical to learn the needed invariance.

More relevant to our problem, recent work considers action recognition in *static* images. During both training and testing, these algorithms use only static snapshots of the actions of interest. Most current methods rely on a combination of pose- and appearance-based descriptors [33, 17, 7, 36, 35]. In particular, "poselets" [4]—local part-based features mined for their consistency with fragments of body pose—have proven to be a promising representation [33, 17, 7], as well as high-level descriptors that also incorporate interactions with objects [8, 34, 7, 36]. We adopt the "poselet activation vector" representation [17] to describe pose in our implementation. Our focus is artificially expanding the training set for "free" via pose dynamics learned from unlabeled data, regardless of the specific descriptor. Thus, our contribution could potentially benefit any of these prior models.

We use domain adaptation to account for the possible mismatch in statistics between the video frames and images. Domain adaptation can improve event learning across two domains of videos, such as Web videos to consumer videos [10] or one benchmark dataset to another [5], and it can also help train an object detector from videos [21]. A novel technique to use multiple source domains and a mix of static and dynamic features is developed in [9].

As discussed above, our idea can also be viewed in the context of work that pads the training set with synthetic data. A standard way to expand object detection training data is by mirroring the images along the vertical axis (e.g., [20] and many others). This trick has even been employed to produce flipped versions of video sequences for activity recognition [31]. The availability of

humanoid models in graphics software, together with mocap data, make it possible to generate synthetic images useful for training action recognition [18] and pose estimation methods [26, 12, 27]. Web images noisily labeled by tags can also serve as a "free" source of data for action classification [13]. No prior work attempts to augment action images with unlabeled video samples, as we propose. Furthermore, whereas past methods aim to better populate the *appearance* space for a class (e.g., people in different clothes; the object facing the other direction), our idea is to better populate the *pose parameter space* as learned from unlabeled video.

3. Approach

Our approach augments a small set of static images labeled by their action category by introducing synthetic body pose examples. The synthetic examples extend the real ones locally in time, so that we can train action classifiers on a wider set of poses that are (likely) relevant for the actions of interest.

We first define the representation we use for pose (Sec. 3.1). Then, after describing our video data requirements (Sec. 3.2), we present two methods to infer synthetic pose examples; one is example-based (Sec. 3.3.1), the other is manifold-based (Sec. 3.3.2). Finally, we explain how we use a mix of real and synthetic data to train a classifier that can predict actions in novel static images (Sec. 3.4).

3.1. Representing Body Pose

We use a part-based representation of pose called a *poselet activation vector* (PAV), adopted from [17]. A poselet [4] is an SVM classifier trained to fire on image patches that look like some consistent fragment of human body pose. For example, one poselet might capture arms crossed against the chest, or a left leg bent at the knee, or even the whole body of a seated person. The PAV records the "activation strength" of all poselets appearing within a person bounding box. Specifically, after running a bank of P poselet classifiers on an image, we take those poselet detections that overlap with a person bounding box, and record a vector $\boldsymbol{p} = [p_1, \ldots, p_P]$ where p_i is the sum of the i-th classifier's probability outputs. Figure 2 shows this process, and the blurry images in Figure 3 depict example poselets in terms of the averaged image patches used to train them. We use the $P = 1200$ poselets provided by [17].

We use this descriptor because it captures human body pose at a high level, and it is robust to occlusion and cluttered backgrounds. While it is quite simple—essentially a histogram of local pose estimates—it is also powerful. The poselets themselves offer a rich encoding of diverse poses, and they are detectable in spite of differences in appearance (e.g., clothing, race). Further, since they are specific to body configurations, the PAV implicitly captures spatial layout. Since 2D HOG descriptors underly the poselet classifiers, they are naturally sensitive to substantial 3D viewpoint changes. This is fine for our data-driven approach, which will synthesize poses that expand exemplars as viewed from a similar viewpoint.

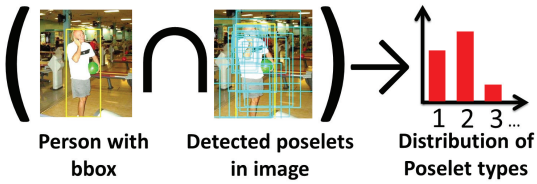

Figure 2. The PAV representation summarizes those detected poselets in the image that overlap with the person bounding box.

3.2. Unlabeled Video Data

Our method requires access to unlabeled videos containing human activity. The video has no action category labels associated with it, and the activity is not segmented in any way. In particular, we do *not* assume that the activities present belong to the same categories as we will observe in the static training images. The category-independence of the video data is crucial. We would like the system to build a model of human motion dynamics—typical changes of body pose over time—without knowing in advance what novel actions it will be asked to learn from snapshots. Intuitively, this suggests that a large and diverse set of clips would be ideal, as we cannot hope to extrapolate poses for inputs that are unlike anything the system has seen before. In our current implementation, we use video from the Hollywood dataset [15] to form the unlabeled pool.

We assume that the humans appearing in the video can often be detected and tracked, i.e., using state-of-the-art human detectors and tracking algorithms, so that we can extract pose descriptors from human bounding boxes. We also expect that the video and snapshots come from roughly similar sensor types, meaning that we would not attempt to use dynamics learned from overhead aerial video (where people are blobs of tens of pixels) to help recognition with snapshots taken on the ground (where people have substantially greater resolution and body parts are visible). This is a very mild requirement, since plenty of ground video is available to us via YouTube, Hollywood movies, and so on. In fact, our method explicitly builds in some flexibility to data source mismatches due to its use of domain adaptation, as we will discuss later.

To pre-process the unlabeled video, we 1) detect people and extract person tracks, 2) compute a PAV pose descriptor for each person window found, and 3) either simply index those examples for our exemplar-based method or else compute a pose manifold for our manifold-based method (both are defined in Sec. 3.3). Note that because this video is unlabeled, our method will enhance the training set with no additional manual effort.

3.3. Generating Synthetic Pose Examples

Our key idea is to expand limited training data by exploring unlabeled video, which implicitly provides rules governing how human pose changes over time for various activities. Thus, the heart of our method is to generate synthetic pose examples. We investigate two strategies: example-based and manifold-based.

Let $\mathcal{S} = \{(\boldsymbol{p}_1^i, y_1), \ldots, (\boldsymbol{p}_N^i, y_N)\}$ denote the N training snapshots our system receives as input, where the superscript i denotes *image*, and each $\boldsymbol{p}_j^i \in \mathbb{R}^P$ is a PAV descriptor with an associated action class label $y_j \in \{1, \ldots, C\}$ (e.g., running, answering phone, etc). Let $\{\boldsymbol{t}_1, \ldots, \boldsymbol{t}_K\}$ denote the K person tracks from the unlabeled video, and let each track $\boldsymbol{t}_k$ be represented by a sequence of PAV descriptors, $\boldsymbol{t}_k = (\boldsymbol{p}_{k_1}^v, \ldots, \boldsymbol{p}_{k_M}^v)$, where superscript v denotes *video*, and k_M is the number of frames in the k-th track.

3.3.1 Example-based strategy

Our example-based method treats the video as a non-parametric representation of pose dynamics. For each training snapshot pose $\boldsymbol{p}_j^i$, we find its nearest neighbor pose in any of the video tracks, according to Euclidean distance in PAV space. Denote that neighbor $\boldsymbol{p}_{j^*}^v$. Then, we simply sample temporally adjacent poses to $\boldsymbol{p}_{j^*}^v$ to form synthetic examples that will "pad" the training set for class y_j. Specifically, we take $\boldsymbol{p}_{j^*-T}^v$ and $\boldsymbol{p}_{j^*+T}^v$, the poses T frames before and T frames after the match (accounting for boundary cases if the neighbor starts or ends a track). See Figure 3.

We repeat this process for all training snapshots, yielding an expanded training set $\mathcal{S}^+$ with two new synthetic examples for each original snapshot: $\mathcal{S}^+ = \{\mathcal{S} \cup \{(\boldsymbol{p}_{j^*-T}^v, y_j), (\boldsymbol{p}_{j^*+T}^v, y_j)\}_{j=1}^N\}$. In our experiments, we set $T = 10$ in order to get frames showing poses that would occur just before or after the matched pose, without being too visually redundant. In preliminary tests, we found the method is not very sensitive to this parameter within the range $T = 5, \ldots, 20$, and simply fixed it at 10.

3.3.2 Manifold-based strategy

We also explore a method to extrapolate poses using a non-linear pose manifold. Whereas the example-based method extrapolates pose solely in the temporal dimension—and solely using one sequence at a time—the manifold variant unifies connections in both appearance and dynamics, and it effectively samples synthetic examples from a mix of sequences at once.

To construct the manifold, we use the locally linear embedding (LLE) algorithm [25]. LLE constructs a neighborhood-preserving embedding function that maps high-dimensional inputs in $\mathbb{R}^P$ to a low-dimensional non-linear manifold in $\mathbb{R}^d$. The manifold is represented as a set of globally consistent linear subspaces, and the solution to minimize its reconstruction error relies on an eigenvalue problem. The algorithm takes as input a set of data points and their respective k nearest neighbors, and returns as output all points' low-dimensional coordinates.

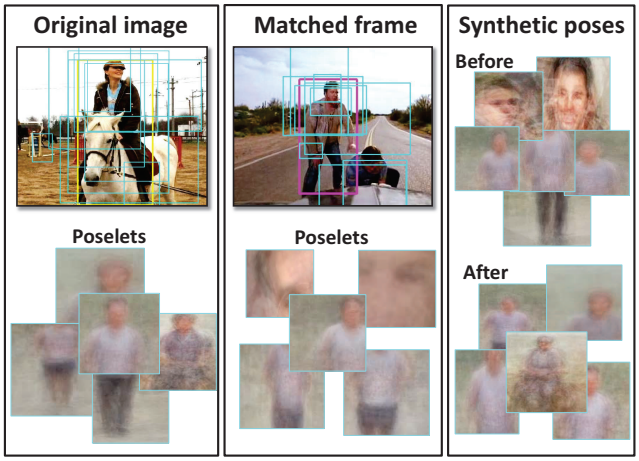

Figure 3. For each labeled training snapshot (top left), we use its pose description (depicted in bottom left) to find a neighbor in the unlabeled video (center panel). Then we synthesize additional training poses based on its temporal neighbors or nearby instances on a pose manifold (right panel). Best in color.

We use the PAVs from the unlabeled video to build the manifold. Recall that $\boldsymbol{p}_{k_q}$ denotes the PAV for the q-th frame within the k-th track in the unlabeled video (dropping the superscript v for clarity). We determine neighbors for LLE using a similarity function capturing both temporal nearness and pose similarity: $A(\boldsymbol{p}_{k_q}, \boldsymbol{p}_{j_r}) =$

$$\lambda \exp\left(-\left\|\boldsymbol{p}_{k_q} - \boldsymbol{p}_{j_r}\right\| / \sigma_p\right) + (1-\lambda) \exp\left(-\|q - r\| / \sigma_t\right),$$

where $\|q - r\| = \infty$ if $k \neq j$, that is, if the two inputs are from different tracks. Here σ_p and σ_t are scaling parameters, set to the average distance between all PAVs and frame numbers, respectively, and the weight λ controls the influence of the two terms. Note that an example's neighbors under A can span poses from both the same and different tracks. After applying the LLE embedding, each original PAV $\boldsymbol{p}^v \in \mathbb{R}^P$ has a low-dimensional counterpart $\hat{\boldsymbol{p}}^v \in \mathbb{R}^d$.

Next, for each training snapshot, we find nearby poses on the manifold to generate synthetic examples. Specifically, for snapshot $\boldsymbol{p}_j^i$ with nearest neighbor $\boldsymbol{p}_{j^*}^v$ in PAV space, we take the associated $\hat{\boldsymbol{p}}_{j^*}^v$ manifold coordinate, and compute its closest two embedded points from the video.[1] (We choose two simply to be consistent with the example-based method above.) Finally, we augment the training set similarly to above, putting the original PAVs for those two instances labeled with the snapshot's category into $\mathcal{S}^+$.

[1] One could alternatively use an out-of-sample extension to LLE [2] when collecting the manifold neighbors.

Discussion Whether example- or manifold-based, we stress that the synthetic examples exist in *pose* space—not raw image space. Thus, we are padding our training set with plausible poses that could immediately precede or follow the observed static snapshot poses, and ignoring surrounding context, objects, etc. Furthermore, it is entirely possible that the action the person in the video was performing when taking on that pose was *not* the action labeled in the static snapshot. Our idea is that the generic human motion dynamics gleaned from the unlabeled video allow us to extrapolate the poses observed in novel static images, at least to very near instants in time. This allows, for example, the system to infer that a kicking action could take on more diverse poses than the few available in the training set (compare left and right panels in Figure 1).

3.4. Training with a Mix of Real and Synthetic Poses

Finally, we use the augmented training set $\mathcal{S}^+$ to train SVM action classifiers to predict the labels of novel images. Rather than directly use the data as-is, we specifically account for the uncertainty in the synthetic examples in two ways. First, we employ domain adaptation to account for the potential mismatch in feature distributions between the labeled snapshots and unrelated video. Second, we use penalty terms in the SVM objective that put more emphasis on satisfying the label constraints for the real data examples compared to the synthetic ones.

Domain adaptation (DA) techniques are useful when there is a shift between the data distributions in a "source" and "target" domain. They typically transform the data in some way that accounts for this discrepancy—for example, by mapping to an intermediate space that shares characteristics of both domains. In our case, we can think of the static snapshots (whether training or testing) as the target domain, and the unlabeled video as the source domain.

We use the "frustratingly simple" DA approach of [6]. It maps original data in $\mathbb{R}^P$ to a new feature space of dimension $\mathbb{R}^{3P}$, as follows. Every synthetic (source) pose example $\boldsymbol{p}^v$ is mapped to $\boldsymbol{p}^{v'} = [\boldsymbol{p}^v, \boldsymbol{p}^v, \mathbf{0}]$, where $\mathbf{0} = [0, \ldots, 0] \in \mathbb{R}^P$. Every real (target) pose example is mapped to $\boldsymbol{p}^{i'} = [\boldsymbol{p}^i, \mathbf{0}, \boldsymbol{p}^i]$. This augmentation expands the feature space into a combination of three versions of it: a general version, a source-specific version, and a target-specific version. The classifier benefits from having access to all versions to find the most discriminative decision function.

Given the domain-adapted features, we train one-vs.-all SVM classifiers. During training, we want to reflect our lower confidence in the synthetic training examples, as well as account for the fact that they will outnumber the real examples. Thus, we use two separate constants for the slack penalty C in the standard SVM objective, in order to penalize violating label constraints on real data more heavily. Specifically, the cost for label errors on the real examples C_{real} is set to 1, while the cost for synthetic examples $C_{synth} \leq 1$ (set via cross-validation). This weighting, combined with the soft-margin SVM, will give some resilience to off-base synthetic pose examples wrongly hypothesized by our method. This can occur, for example, if the nearest PAV or manifold neighbor is quite distant and thus serves as a weak proxy for the training snapshot's pose.

4. Experimental Results

We demonstrate our approach on three datasets for recognizing activities in both images and videos.

4.1. Datasets

For the unlabeled video data, we use the training and testing clips from the Hollywood Human Actions dataset [15]. We stress that none of the activity labels are used from these clips. In fact, only one label in Hollywood overlaps with any of the data below (*phoning* is in both PASCAL and Hollywood). To get person tracks, we use the annotation tool provided by [30]. This allows us to focus our evaluation on the impact of our method, as opposed to the influence of a particular person tracking method.

For the recognition task with static test images, we test on both the 9 actions in the PASCAL VOC 2010 dataset [11] *(phoning, playing instrument, reading, riding bike, riding horse, running, taking photo, using computer, walking)* as well as 10 selected verbs from the Stanford 40 Actions dataset [36] *(climbing, fishing, jumping, playing guitar, riding a bike, riding a horse, rowing a boat, running, throwing frisbee, walking the dog)*. While the latter has 40 total verbs, we limit our experiments to those 10 where the baseline has reasonable precision using a body pose descriptor alone; many of the others are strongly characterized by the objects that appear in the scene. We call it Stanford 10. For PASCAL, we use (maximally) the 301 persons from the training set to train, and the 307 persons in the validation set to test. For Stanford 10, we randomly select (maximally) 250 and 1672 persons for training and testing, respectively, based on the train/test split suggested by the authors.

For the video recognition task, we compile a test set from multiple video sources, since no existing video dataset has both images and videos for a set of action labels. We gather 78 test videos from the HMDB51 [14], Action Similarity Labeling Challenge [14], and UCF Sports [24] datasets that contain activities also appearing in PASCAL: *phoning*, *riding bike*, *riding horse*, *running*, and *walking*. Note that the unlabeled video source remains the Hollywood data for this task; in all cases, the only labels our method gets are those on the static snapshots in PASCAL.

We fix the dimensionality for LLE $d = 10$, and the affinity weight $\lambda = 0.7$. We use χ^2-kernels for the SVMs, and set the SVM penalty $C_{synth} = 0.1$ for image recognition

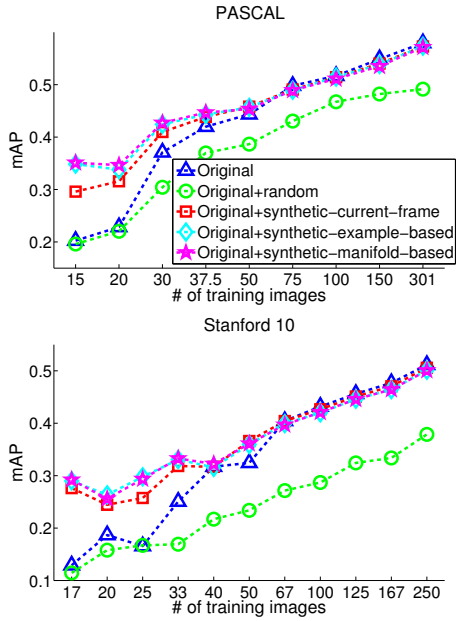

Figure 4. Accuracy on static action recognition datasets as a function of the number of training images. Our method shows dramatic gains with very few labeled snapshots, and maintains similar accuracy to the baseline when training exemplars are plentiful.

and $C_{synth} = 0.5$ for video recognition, based on validation with the PASCAL training data.

4.2. Recognizing Activity in Novel Images

The primary comparison of interest is to see whether recognition improves when adding our synthetic training data, versus a baseline that does everything else the same (i.e., PAV representation, SVM, etc.), but uses only the original training snapshots. This baseline corresponds to the state-of-the-art method of [17], and we denote it **Original** throughout. In addition, we provide two more baselines to help isolate the reason for our method's advantage. The first, **Original+random**, replaces our method's nearest neighbor selection with a randomly selected video pose. The second, **Original+synthetic-current-frame**, uses only the matched neighbor to synthesize an example (i.e., it lets $T = 0$). This baseline is useful to see the extent to which we need to extrapolate poses across *time* (dynamics), versus merely padding the data with variations in *appearance* (similar instances of the same pose).

Figure 4 shows the mean average precision (mAP) test accuracy as a function of the number of training images, for both static image datasets. To robustly estimate accuracy with few training samples, we run the experiment five times with different randomly sampled training images (when using less than all the data) and report the average. Our approach substantially boosts accuracy when few training snapshots are available. As expected, having only few exemplars accentuates our method's ability to "fill in" the

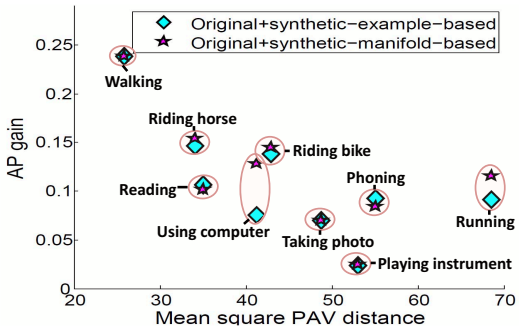

Figure 6. Per class accuracy gains by our methods as a function of the diversity of the original PASCAL data. See text.

related poses. On the other hand, when training examples are plentiful (hundreds), there is less to be gained, since more variation is already visible in the originals; in fact, our results are comparable to the baseline's in the rightmost part of the plots.[2] Adding poses from random frames degrades accuracy across the board, confirming that our method's gain is not due to having *more* pose examples; rather, it synthesizes *useful* ones relevant to the recognition task. Adding a pose from the neighbor frame itself ("current") increases the baseline's accuracy by synthesizing more varied appearances of the poses in the training set, but it is inferior to using the pose dynamics as proposed.

Figure 5 shows examples of images responsible for synthetic poses added to the original training set for PASCAL. We see how useful poses can be found *across* activity categories. For example, the bottom image of a man phoning has synthetic poses generated from a man who is not phoning—but who nonetheless takes on poses and facial expressions that could have been (were the objects in the scene different). In the special case that a familiar action actually appears in the unlabeled video, it too can help, as we see in the horse-riding and walking examples. In all examples, notice how the synthetic examples simulate slight variations over time. This is how our approach fleshes out the training set.

Note that our improvements are in spite of the fact that only one label overlaps between PASCAL and Hollywood, and zero overlap between Stanford 10 and Hollywood. We find that for the largest training set size on PASCAL ($N = 301$), 23 PASCAL images match to a Hollywood clip that shows the verb phoning. Among those 23, only two of them are themselves *phoning*. Hence, our results clearly show the category-independent nature of our approach. Poses from distinct actions are relevant to connect the dots between sparse exemplars.

Next we compare our example- and manifold-based strategies for gathering pose neighbors. The mAP averaged

[2] And our numbers roughly replicate those reported in [17] for PASCAL—we obtain 57.94 vs. 59.8 mAP when using all training data.

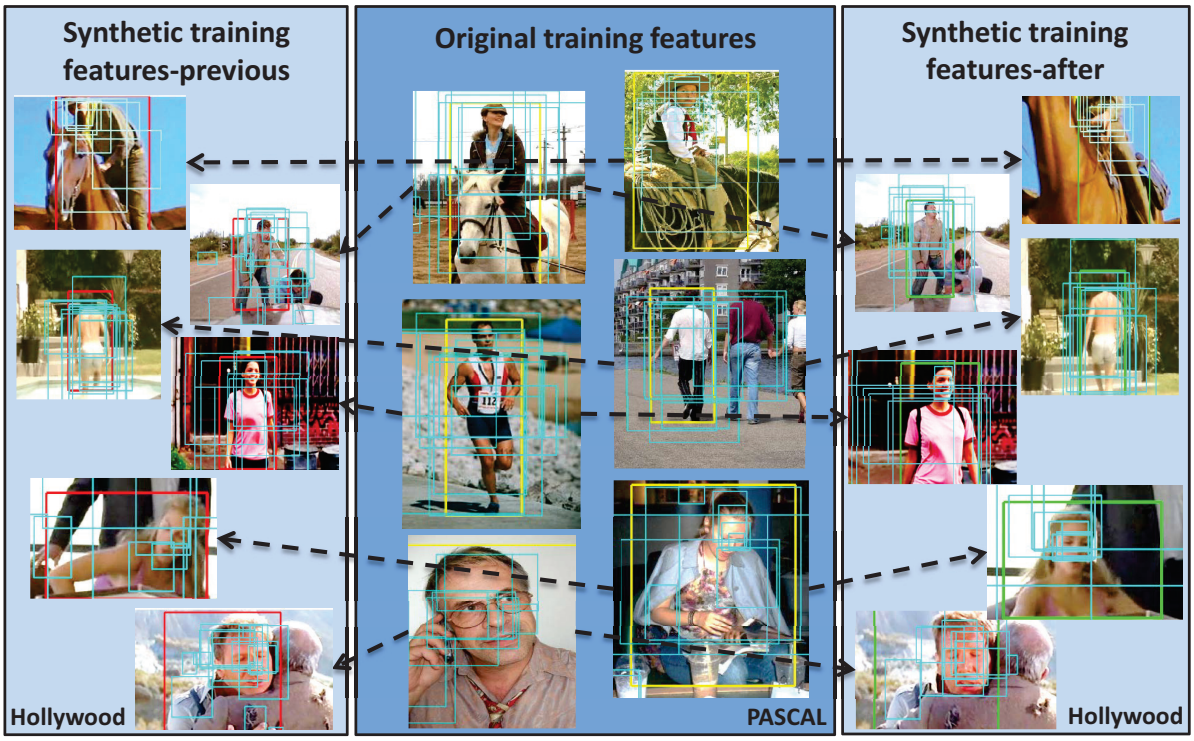

Figure 5. Six real examples showing the frames our method found in unlabeled video (left and right panels) and used to expand the original training poses in snapshots (center panel). Each pose in the center panel finds a neighbor in the unlabeled video $\boldsymbol{p}^v_{j*}$, which generates a synthetic example for what could come immediately before ($\boldsymbol{p}^v_{j*-T}$, left) and after ($\boldsymbol{p}^v_{j*+T}$, right) that pose. Red/yellow/green boxes denote person bounding boxes, and smaller cyan boxes denote poselet detections. Dotted arrows connect to corresponding synthetic frames.

over all classes (Fig. 4) is fairly similar for both. Figure 6 shows the AP gain of our two methods (compared to Original) for each individual class in PASCAL when training with $N = 20$ examples (ignore the x dimension for now). Indeed, for many classes their gains are similar. However, manifold-based has a noted advantage over example-based for the actions *running* and *using computer*. On Stanford 10, it is stronger for *running* and *climbing* (not shown). What these actions seem to have in common that they entail some repeated motion. We hypothesize the manifold does better in these cases since it captures both temporally nearby poses and appearance variations.

Figure 6 also shows that there is a correlation between those classes most benefited by our method and their lack of diversity. We measure diversity by the average inter-PAV distance among training examples. Low distance means low diversity. Just as a training set that is too small needs our method to fill in intermediate poses, so too a class whose examples are too tightly clustered in pose space (e.g., due to a dataset creator's unintentional bias towards "canonical poses") may benefit most from our method.

Table 1 isolates the impact of domain adaptation on our results, when the number of training examples $N = 30$. (The impact is very similar no matter the training set size.) We see that DA gives a modest but noticeable gain in accuracy for both variants of our method, showing it is worthwhile to model the potential data mismatch between the unlabeled video and training snapshots. We suspect the PAV pose descriptors are also playing a role in accounting for the domain shift, since they abstract away some nuisance factors that could differ between the two sources (e.g., lighting, scale).

Dataset	PASCAL		Stanford 10	
Domain-adaptation?	No	Yes	No	Yes
Example-based	0.4243	**0.4320**	0.3308	**0.3378**
Manifold-based	0.4271	**0.4327**	0.3328	**0.3404**

Table 1. Impact on mAP of domain adaptation on the static datasets.

4.3. Recognizing Activity in Novel Video

Next, we apply our method to predict activities in novel *video*, still using the same static image training set idea (see dataset details in Sec. 4.1). We use a simple voting approach to predict the label for the entire video. First, we classify each frame independently, generating a probability for each possible label $1, \ldots, C$. Then, we sum the probabilities across all frames to get the final prediction. Note that this test should allow our method to shine, since the novel videos will exhibit many intermediate poses that the original snapshots did not cover—but that our method will

	Original	Original+synthetic example-based	Original+synthetic manifold-based
Without DA	0.3846	**0.5128**	0.4872
With DA	N/A	**0.5382**	0.5128

Table 2. Accuracy of video activity recognition on 78 test videos from HMDB51+ASLAN+UCF data.

(ideally) synthesize. For this experiment, we transform the domain adapted features using $\boldsymbol{p}^{v'} = [\boldsymbol{p}^v, \mathbf{0}, \mathbf{0}]$, since the train, test, and synthetic data are all from different domains.

Table 2 shows the results. We compare our method to the Original baseline, and also show the impact of domain adaptation. Our method makes a substantial improvement in accuracy. Its synthetic padding of the data makes the training set less sparse, yielding more reliable predictions on the video frames. Domain adaptation again boosts the accuracy further.

5. Conclusions

We proposed a framework to augment training data for activity recognition without additional labeling cost. Our approach leverages knowledge of human pose patterns over time, as represented by an unlabeled video repository. To implement our idea, we explore simple but effective example- and manifold-based representations of pose dynamics, and combine them with a domain adaptation feature mapping that can connect the real and generated examples.

Our results classifying activities in three datasets show that the synthetic poses have significant impact when the labeled training examples are sparse. We demonstrate the benefits with a state-of-the-art local pose representation; however, our idea is not coupled specifically with that method, and it has potential to boost alternative descriptors in similar ways. In future work, we plan to investigate extensions that could account for not only poses, but also the objects and scenes with which the detected humans interact.

Acknowledgements This research is supported in part by DARPA CSSG N11AP20004.

References

[1] J. K. Aggarwal and Q. Cai. Human motion analysis: a review. *CVIU*, 73(3):428–440, 1999.

[2] Y. Bengio, J. Paiement, P. Vincent, O. Delalleau, N. L. Roux, and M. Ouiment. Out-of-sample extensions for LLE, Isomap, MDS, Eigenmaps, and spectral clustering. In *NIPS*, 2003.

[3] J. Blackburn and E. Ribeiro. Human motion recognition using isomap and dynamic time warping. In *Human Motion*, 2007.

[4] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*, 2009.

[5] L. Cao, Z. Liu, and T. Huang. Cross-dataset action detection. In *CVPR*, 2010.

[6] H. Daume III. Frustratingly easy domain adaptation. In *ACL*, 2007.

[7] V. Delaitre, J. Sivic, and I. Laptev. Learning person-object interactions for action recognition in still images. In *NIPS*, 2011.

[8] C. Desai, D. Ramanan, and C. Fowlkes. Discriminative models for static human-object interactions. In *Wrkshp on Structured models in Computer Vision*, 2010.

[9] L. Duan, D. Xu, and S.-F. Chang. Exploiting web images for event recognition in consumer videos: a multiple source domain adaptation approach. In *CVPR*, 2012.

[10] L. Duan, D. Xu, I. Tsang, and J. Luo. Visual event recognition oin videos by learning from web data. In *CVPR*, 2010.

[11] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. In *IJCV*, volume 88, pages 303–338, June 2010.

[12] K. Grauman, G. Shakhnarovich, and T. Darrell. Inferring 3d structure with a statistical image-based shape model. In *ICCV*, 2003.

[13] N. Ikizler-Cinbis, R. Cinbis, and S. Sclaroff. Learning actions from the web. In *ICCV*, 2009.

[14] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *ICCV*, 2011.

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

[16] C. Lee and A. Elgammal. Human motion synthesis by motion manifold learning and motion primitive segmentation. In *AMDO*, 2006.

[17] S. Maji, L. Bourdev, and J. Malik. Action recognition from a distributed representation of pose and appearance. In *ICCV*, 2011.

[18] P. Matikainen, R. Sukthankar, and M. Hebert. Feature seeding for action recognition. In *ICCV*, 2011.

[19] D. Moore, I. Essa, and M. Hayes. Exploiting human actions and object context for recognition tasks. In *CVPR*, 1999.

[20] C. Papageorgiou and T. Poggio. A trainable system for object detection. *IJCV*, 38(1):15–33, 2000.

[21] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, 2012.

[22] D. Ramanan and D. Forsyth. Automatic annotation of everyday movements. In *NIPS*, 2003.

[23] C. Rao and M. Shah. View-Invariance in Action Recognition. In *CVPR*, 2001.

[24] M. D. Rodriguez, J. Ahmed, and M. Shah. Action MACH a spatio-temporal maximum average correlation height filter for action recognition. In *CVPR*, 2008.

[25] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. In *Science*, volume 290, pages 2323–2326, December 2000.

[26] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter sensitive hashing. In *ICCV*, 2003.

[27] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from a single depth image. In *CVPR*, 2011.

[28] N. Tang, C. Hsu, T. Lin, and H. Liao. Example-based human motion extrapolation based on manifold learning. In *ICME*, 2011.

[29] G. Taylor, I. Spiro, C. Bregler, and R. Fergus. Learning invariance through imitation. In *CVPR*, 2011.

[30] C. Vondrick, D. Ramanan, and D. Patterson. Efficiently scaling up video annotation with crowdsourced marketplaces. In *ECCV*, 2010.

[31] H. Wang, M. Ullah, A. Klaser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *BMVC*, 2009.

[32] J. Wang, D. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *TPAMI*, pages 283–298, Feb 2008.

[33] W. Yang, Y. Wang, and G. Mori. Recognizing human actions from still images with latent poses. In *CVPR*, 2010.

[34] B. Yao and L. Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *CVPR*, 2010.

[35] B. Yao and L. Fei-Fei. Action recognition with exemplar based 2.5d graph matching. In *ECCV*, 2012.

[36] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. J. Guibas, and L. Fei-Fei. Action recognition by learning bases of action attributes and parts. In *ICCV*, 2011.

Please see `http://vision.cs.utexas.edu/projects/action-snapshot/` for dataset annotation and project page.