Recognizing Human Actions by Attributes

Jingen Liu, Benjamin Kuipers, Silvio Savarese
Dept. of Electrical Engineering and Computer Science
University of Michigan
{liujg,kuipers,silvio}@umich.edu

Abstract

In this paper we explore the idea of using high-level semantic concepts, also called attributes*, to represent human actions from videos and argue that attributes enable the construction of more descriptive models for human action recognition. We propose a unified framework wherein manually specified attributes are: i) selected in a discriminative fashion so as to account for intra-class variability; ii) coherently integrated with data-driven attributes to make the attribute set more descriptive. Data-driven attributes are automatically inferred from the training data using an information theoretic approach. Our framework is built upon a latent SVM formulation where latent variables capture the degree of importance of each attribute for each action class. We also demonstrate that our attribute-based action representation can be effectively used to design a recognition procedure for classifying novel action classes for which no training samples are available. We test our approach on several publicly available datasets and obtain promising results that quantitatively demonstrate our theoretical claims.*

1. Introduction

In most of the traditional approaches for human action recognition, action models are typically constructed from patterns of low-level features and directly associated with class labels (say, *walking* or *golf-swinging* in Fig.1). It is clear, however, that this process is fundamentally reductive: rich visual temporal-spatial structures (such as those associated with the *golf-swinging*) can be hardly characterized by one single class label and would be better represented by considering multiple high-level semantic concepts describing the action. Inspired by recent formulations on object categorization [3, 8, 14, 32, 28], we call these high-level concepts "action attributes". Fig. 1 shows examples illustrating this intuition. For instance, the action *golf-swinging* may be effectively represented by introducing a number of attributes that can be directly associated with either the visual characteristics describing the spatial-temporal evolution of the actor (e.g., *single leg motion*, *arm over shoulder motion*, *torso up-down motion*) or with the contextual description of the scene wherein the action takes place (e.g., *outdoor*, *sport*). In this work we focus on the former types of attributes (extensions including the latter ones are straightforward) and argue that an action attribute-based representation enables a classification framework for human action recognition that is more descriptive and discriminative than traditional methods.

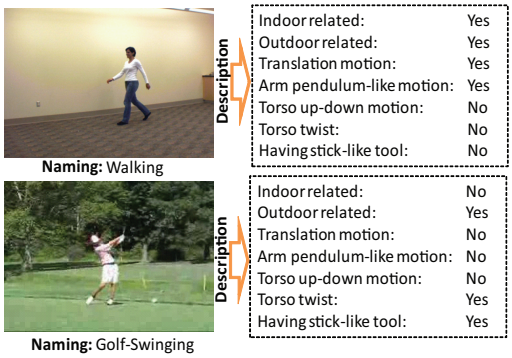


Figure 1. We propose to represent human actions by a set of attributes which can be directly associated with the visual characteristics describing the spatial-temporal evolution of the action in a video (e.g., *single leg motion*, *arm over shoulder motion*, *torso up-down motion*). We argue that an action attribute-based representation is more descriptive and discriminative for action recognition than traditional methods.

As also discussed in [8, 14] for object categorization, the ability to characterize actions by attributes is not only helpful for recognizing familiar actions, but it is also a powerful tool for recognizing action categories that have never been seen before (e.g., for which no training samples are available). This problem is also referred as *zero-shot* learning [17, 25] and it is based on the idea of transferring knowledge from known classes to unknown classes via attributes as a bridge. Clearly, for these methods the success of a zero-shot learning process heavily depends on the possibility of sharing attributes across classes.

While promising for the reasons discussed above, it is clear that an attribute-based representation has the drawback of being sensitive to the process of selecting attributes and associating them with relevant action classes. A conventional way for doing so is to: i) manually identify a list

of possible high-level concepts that can be reasonably used to characterize the set of action classes one wishes to classify (see Fig.1); ii) select which of these attributes occur in each class of interest. There are, however, two major open problems related to this selection/assignment process.

The first problem concerns the issue of selecting a set of attributes that are capable of representing the complete pool of action classes. If one selects attributes manually, it is clear that this process is subjective and arbitrary, and it does not guarantee that all of the critical visual spatial-temporal patterns characterizing an action class are successfully associated with attribute labels. To address this issue, we propose to integrate manually specified attributes with attributes that are automatically discovered from the data themselves using information theoretic methods. We call these attributes *data-driven* attributes. This procedure is inspired by previous methods that seek to discover semantic visual words [20] or intermediate concepts [19, 31] for action classification.

The second problem concerns the issue of selecting action attributes that are discriminative and yet able to capture the inherent intra-class variably of each action class. For instance, consider the action class *walking* and the attribute *two arms perform pendulum-like motion*. Some examples of *walking* may contain the attribute *two arms perform pendulum-like motion*; some others may not. So how does one associate the action class *walking* to the attribute *two arms perform pendulum-like motion*? To address this problem, we treat attributes as latent variables, and formulate the classification problem using a latent linear SVM framework akin to [11] for selecting the most discriminative and representative attributes for each action class. Our method has the key advantage of integrating in one unique optimization problem the process of forming the data-driven attributes and selecting the most discriminative attributes among the pool of (data-driven and manually selected) attributes.

1.1. Related work

The problem of action recognition has been widely explored in the computer vision community. Early action recognition frameworks focused on tracking, motion capture and the analysis of tracks [2]. More recently, great progress has been made by introducing more descriptive action representations such as space-time pattern templates[1, 21], 2D shape matching [33, 7], optical flow patterns [6, 33], trajectory-based representation [27], and bag-of-video-words [24, 19, 20]. Some approaches have attempted to integrate contextual information, such as space-time neighborhood features[13], object-scene-action mixture models[12], and spatial and temporal relations[22].

To our knowledge, few attempts have been made to utilize high-level concepts for the recognition of human actions. Some researchers have proposed to represent action by intermediate semantic features [19, 20, 9], which are conceptually similar to our data-driven attributes. These intermediate features are learned from a training dataset by (soft or hard) clustering low-level features based on their co-occurrence in training videos. The assumption is that frequently co-occurring low-level features are correlated at some conceptual level giving raising to intermediate representations (topics). Similarly, Wang *et al.* [31] use the hidden Conditional Random Fields for action recognition. The authors model an action class as a root template and a constellation of several hidden "parts", where the hidden "part" is a group of local patches that are implicitly correlated with some intermediate representation. In both methods the association between the intermediate features (or hidden "parts") and action class labels is not immediately known - given a set of discovered intermediate semantic features, one is unable to assign an action class to the observation unless the class has previously been learned from a labelled training dataset. This prevents these methods from being used when no training examples are available (zero-shot learning). Ramanan *et al.* [26] proposed to annotate videos by manually label the training videos with some movements, which seems similar to the attributes in our work.

Rather than discovering attributes from videos, works by [28, 3] have proposed to mine semantic object attributes from either web data [3, 28] or WordNet [28]. We believe, however, this strategy may not work well for human actions, since mining the semantic relationships of verbs (actions) from WordNet or web data is much more difficult than discovering the relationships between nouns (objects). This is because verbs do not have the same well-built ontological relationships found with nouns.

The idea of using an attribute-based representation as a guiding tool for object recognition is explored in [32, 14, 28, 3, 8]. These methods follow the intuition that manually specified attributes can be used to explicitly represent a visual class, thus helping the recognition [32, 8] or enabling the recognition of novel classes even when no prior training examples are available [14, 28]. To the best of our knowledge, however, our work is the first to apply *zero-shot learning* for action recognition using attributes. Moreover, the integration of manually specified and data-driven attributes makes our work unique among the existing work.

1.2. Our Contributions

Our goal is to investigate how action attributes can be used to improve human action recognition. The contributions of this work are three-fold. First, manually-specified attributes enable our approach to recognize novel action classes when no training examples are available. Second, we address intra-class variability by considering the attributes to be latent variables. We use latent SVM to search for the best configuration of attributes for each action. Finally, our approach integrates manually-specified and data-

driven attributes, making the attributes imply much more complete high-level human knowledge on actions. We test our approach on various publicly available action datasets. Experiments are conducted on the Olympic Sports dataset [23], the UIUC dataset[30], and the combination of three datasets: the UIUC dataset, the KTH dataset [29] , and the Weizmman dataset [21] (for a total of 21 action classes). Experimental results show that our attribute-based action representation is useful for recognizing novel action classes without training examples and can also significantly boost traditional action classification.

2. Attribute-Based Action Representation

Most previous works represent actions with low-level features $\mathbf{x} \in \mathbb{X}^d$, and solve the classification problem by defining a classifier $f : \mathbb{X}^d \rightarrow \mathbb{Y}$, that maps the feature vector to a class label $\mathbb{Y}$. But we believe that human actions are better described by action attributes. In this section, we briefly describe how we represent actions with a set of action attributes.

Follow the description of [25], we define an *action attribute space* $\mathbb{A}^m$ as an m dimensional semantic metric space in which each dimension encodes the value of a semantic property. This semantic space is spanned by a basis consisting of m attributes, $\{a_i\}_1^m$. In this space, each action class, as well as each action instance, is represented by one point as shown in Fig.2. As an example, suppose we have five attributes forming the basis: "translation of torso", "up-down torso motion", "arm motion", "arm over shoulder motion", "leg motion". Then the action class "walking" might be represented by a binary vector $\{1, 0, 1, 0, 1\} \in \mathbb{A}^5$, with each dimension indicating the presence or absence of the corresponding attribute. We use binary values for clarity, but continuous values are equally valid. Ideally, the positions of action instances in the space $\mathbb{A}^m$ should be superimposed to the position of each action class. In practice, however, if the absence or presence of each attribute is approximated by a confidence value (from 0 to 1), action instances from one action class will form a point cloud around the class. This is illustrated in Fig.2.

By introducing the attribute layer between the low-level features and action class labels, the classifier f which maps $\mathbf{x}$ to a class label, is decomposed into:

$$\mathcal{H} = \mathcal{L}(\mathcal{S}(\mathbf{x})) \qquad (1)$$
$$\mathcal{S} : \mathbb{X}^d \rightarrow \mathbb{A}^m \text{ and } \mathcal{L} : \mathbb{A}^m \rightarrow \mathbb{Y}$$

where $\mathcal{S}$ consists of m individual attribute classifiers $\{f_{a_i}(\mathbf{x})\}_{i=1}^m$, and each classifier maps $\mathbf{x}$ to the corresponding i-th axis (attribute) of $\mathbb{A}^m$, $\mathcal{L}$ maps a point $\mathbf{a} \in \mathbb{A}^m$ to a class label $y \in \mathbb{Y}$. The attribute classifiers are learned from a training dataset. Specifically, classifier $f_{a_i}(\mathbf{x})$ is trained by labeling the examples of all action classes whose attribute value $a_i = 1$ as positive examples and the rest as negative.

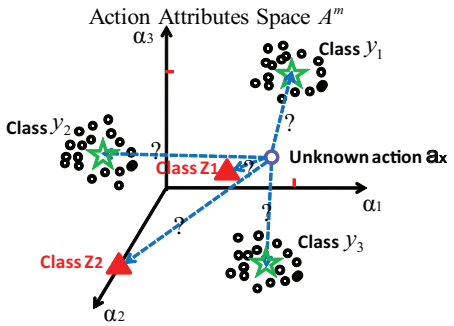


Figure 2. A set of action attributes form a semantic space $\mathbb{A}^m$, in which every action class or action instance is represented by a point. A "circle" represents an action instance; a "star" represents a known class y having training examples (the surrounding points), and "triangle" represents a novel class z for which no training examples are available but it is characterized by a manually specified attribute vector. In this space, an unknown action $\mathbf{a_x}$ that belongs to one of the unknown classes (say z_1) can be recognized by associating its closest class to it (see Sec 5 for details).

The mapping $\mathcal{L}$ can be defined manually or learned from a training dataset. The fact that action classes and action instances share the same semantic space and the capability to manually define $\mathcal{L}$ make it possible to recognize a novel action class (e.g., z_1 and z_2 in Fig.2) with no training samples available, which is addressed in Sec 5.

3. Attributes as Latent Variables

Given a training set, $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, $y_i \in \{-1, 1\}$, we want to learn a classification model for recognizing an unknown action $\mathbf{x}$. Although the attribute values of each training example are already known (they are inherited from their corresponding classes), intra-class variability may cause the attribute values to be inaccurate for specific members of the training class. Consequently, some action instances may be associated with subtly different sets of attributes even if they belong to the same action class. For example, the "jumping forward" actions from the UIUC action dataset have the "big pendulum-like arm motion" attribute, while the instances from the Weizmman dataset do not. This is a consequence of the inherent intra-class variability and the fact that associating attribute labels is a subjective process. We address this difficulty by treating attributes as latent variables.

Inspired by Felzenszwalb's *deformable part model* [11], which can handle the variability in part positions by treating them as latent variables, as well as the work by Wang *et al.* for object recognition [32], we consider each attribute as an abstract "part" of an action. In this way the location of an attribute in the space $\mathcal{A}^m$ is interpreted as a latent variable, $a_i \in [0, 1]$. Larger values of a_i indicate a higher probability that a video possesses this attribute.

Our goal is to learn a classifier $f_{\mathbf{w}} : \mathbb{X}^d \times \mathbb{Y} \rightarrow \mathbb{R}$ where $\mathbf{w}$ is the parameter vector. In testing, $f_{\mathbf{w}}$ is used to predict a new video $\mathbf{x}$, namely $y* = arg\max_{y \in \mathbb{Y}} f_{\mathbf{w}}(\mathbf{x}, y)$. This prediction is not completely characterized by the pair $(\mathbf{x}, y)$ alone, but also depends on its associated attribute values $\mathbf{a} \in$

$\mathbb{A}^m$. Specifically, $f_{\mathbf{w}}(\mathbf{x}, y) = \max_{\mathbf{a}} \mathbf{w}^T \Phi(\mathbf{x}, y, \mathbf{a})$, where $\Phi(\mathbf{x}, y, \mathbf{a})$ is a feature vector depending on raw feature $\mathbf{x}$, its class label y and its associated attributes $\mathbf{a}$, and $\mathbf{w}$ is a parameter vector providing a weight for each feature. The linear model is defined as,

$$\mathbf{w}^T \Phi(\mathbf{x}, y, \mathbf{a}) = \mathbf{w}_{\mathbf{x}} \varphi_1(\mathbf{x}) + \sum_{j \in \mathcal{A}} \mathbf{w}_{a_j}^T \varphi_2(\mathbf{x}, a_j) + \sum_{j,k \in \mathcal{A}} \mathbf{w}_{a_j, a_k}^T \varphi_3(a_i, a_j), \quad (2)$$

where $\mathbf{w} = \{\mathbf{w}_{\mathbf{x}}; \mathbf{w}_{a_j}; \mathbf{w}_{a_j, a_k}\}$, and $\mathcal{A}$ is an attribute set.

The potential function $\mathbf{w}_{\mathbf{x}} \varphi_1(\mathbf{x})$ provides the score measuring how well the raw feature $\varphi_1(\mathbf{x})$ of a video matches the action class template $\mathbf{w}_{\mathbf{x}}$ which is a set of coefficients learned from the raw features $\mathbf{x}$. If other potential functions in Eq. (2) are ignored, we can learn $\mathbf{w}_{\mathbf{x}}$ using a binary linear SVM. In our implementation, we use this observation to make the computation more efficient. Specifically, instead of keeping $\varphi(\mathbf{x})$ as a high-dimensional feature vector, we represent it as the score output of the pre-trained linear SVM. So $\mathbf{w}_{\mathbf{x}}$ is a scalar value used to weigh the SVM score. This strategy was used in [5, 32].

The potential function $\mathbf{w}_{a_j}^T \varphi_2(\mathbf{x}, a_j)$ provides the score of an individual attribute, and is used to indicate the presence of an attribute in the video $\mathbf{x}$. The initial value of a_j is inherited from its class label in the training phase, and is given by a pre-trained attribute classifier when testing (see Sec 2). The edge potential $\mathbf{w}_{a_j, a_k}^T \varphi_3(a_j, a_k)$ captures the co-occurrence of pair of attributes a_j and a_k. If each attribute a_i has A statuses (e.g., {0,1}), then each edge has $A \times A$ configurations. As a result, the feature vector $\varphi_3(a_j, a_k)$ of an edge is a $A \times A$ dimensional indicator for edge configurations. The associated $\mathbf{w}_{a_j, a_k}^T$ contains the weights for all configurations.

The parameter vector $\mathbf{w}$ is learned from a training dataset $\mathcal{D}$ by solving the following objective function:

$$\min_{\mathbf{w}} \lambda \parallel \mathbf{w} \parallel^2 + \sum_{i=1}^{n} \max(0, 1 - y_i \cdot f_{\mathbf{w}}(x_i)), \quad (3)$$

where the second term implements a soft-margin. The objective function 3 is semi-convex due to the inner max in $f_{\mathbf{w}}$. A local optimum can be obtained by the coordinate descent [11], as follows:

- Holding $\mathbf{w}$ fixed, find the best attribute configuration $\mathbf{a}'$ such that $\mathbf{w} \cdot \Phi(\mathbf{x}, y, \mathbf{a})$ is maximized.
- Holding $\mathbf{a}'$ fixed, search the best parameters $\mathbf{w}$ such that the objective Eq. 3 is minimized.

In our current implementation, the attribute graph is learned from the training data. For computational efficiency, each attribute has two statuses ($\{0\}$ and $\{1\}$). To find the best attribute configuration $\mathbf{a}$ for $f_{\mathbf{w}}(\mathbf{x}, y) = \max_{\mathbf{a}} \mathbf{w}^T \Phi(\mathbf{x}, y, \mathbf{a})$, we use belief propagation [10].

4. Learning Data-Driven Attributes

As aforementioned, we argue that manually-specified attributes can assist recognition because they provide high-level semantic information that can be used to improve the characterization of actions. However, the manual specification of attributes is subjective, and potentially useful (discriminative) attributes may be ignored. This may significantly affect the performance of classifiers. One way to overcome this weakness is to automatically learn attributes. We call these data-driven attributes, and argue that they have a complementary role in providing a more complete characterization of human actions. We propose to discover data-driven attributes by clustering low-level features while maximizing the system information gain. The intuition is that attributes may be characterized by a collection of low-level features that tend to co-occur in the training data.

Given two random variables $X \in \mathcal{X} = \{x_1, x_2, ..., x_n\}$ and $Y \in \mathcal{Y} = \{y_1, y_2, ..., y_m\}$, where $\mathcal{X}$ represents a set of *visual-words*, and $\mathcal{Y}$ is a set of action videos. The Mutual Information (MI) [4] $MI(X;Y)$ between X and Y expresses how much information from variable *visual-words* is contained in *action videos*, which provides a good measurement to evaluate the quality of low-level features grouping. It is clear that if two features x_i and x_j are semantically similar, then merging them will not cause significant loss of the information that X and Y share. Given a set of features $X = \{x_i\}_1^n$, we wish to obtain a set of clusters $\hat{X} = \{\hat{x}_t\}_1^T$. The quality of clustering is measured by the loss of MI,

$$\mathcal{L}_{inf}(\mathcal{D}, \Pi) = MI(X;Y) - MI(\hat{X};Y), \quad (4)$$

where $\mathcal{D}$ is the training data set and $\Pi = \{p(\hat{x_t}|x_i)\}$ is the partition of X to $\hat{X}$. After some mathematical derivation, we have [18],

$$\mathcal{L}_{inf}(\mathcal{D}, \Pi) = \sum_{t=1}^{T} \sum_{x_i \in \hat{x_t}} p(x_i) \cdot KL(p(Y|x_i), P(Y|\hat{x_t})), \quad (5)$$

where $KL(a, b)$ is the KL-divergence between two distributions. If we treat the distribution $p(Y|\hat{x}_t)$ as the cluster prototype (e.g., the centroid), and the prior $p(x_i)$ is uniformly distributed, then the loss of MI is the distance from the distribution $p(Y|x_i)$ to the cluster prototype.

We integrate the discovery of data-driven attributes into the framework of latent SVM. Suppose $\mathbf{h} \in \mathbb{H}^l$ (where $\mathbb{H}^l$ is the data-driven attribute space, with the basis $\hat{X}, l = |\hat{X}|$) is the data-driven attribute vector associated with $\mathbf{x}$, then our model is extended as follows,

$$\mathbf{w}^T \Phi(\mathbf{x}, y, \mathbf{a}, \mathbf{h}) = \mathbf{w}_{\mathbf{x}} \varphi_1(\mathbf{x}) + \sum_{i \in \mathcal{A}} \mathbf{w}_{a_i}^T \varphi_2(\mathbf{x}, a_i) + \sum_{j,k \in \mathcal{A}} \mathbf{w}_{a_j, a_k}^T \varphi_3(a_j, a_k) + \sum_{s \in \mathcal{H}} \mathbf{w}_{h_s}^T \varphi_4(\mathbf{x}, h_s) + \sum_{s,t \in \mathcal{H}} \mathbf{w}_{h_s, h_t}^T \varphi_5(h_s, h_t), \quad (6)$$

where $\mathbf{w}_{h_s}^T \varphi_4(\mathbf{x}, h_s)$ provides prediction of a class label by a data-driven attribute h_s, and $\mathbf{w}_{h_s,h_t}^T \varphi_5(h_s, h_t)$ measure the dependency between pairwise data-driven attributes, $\mathcal{H}$ is a set of data-driven attributes. If we consider both human-specified attributes and data-driven attributes to be latent variables, then for each example $\mathbf{x}$, we search for the best configuration of $\mathbf{a}$ and $\mathbf{h}$ such that $f_{\mathbf{w}}(\mathbf{x}, y) = \max_{\mathbf{a},\mathbf{h}} (w)^T \Phi(\mathbf{x}, y, \mathbf{a}, \mathbf{h})$. The extended objective is,

$$arg \min_{\mathbf{w},\Pi} \lambda \parallel \mathbf{w} \parallel^2 + \eta \mathcal{L}_{inf} + \sum_{i=1}^{n} \max(0, 1 - y_i \cdot f_{\mathbf{w}}(\mathbf{x}_i)), \quad (7)$$

where λ and η are tradeoff parameters. The rationale behind this model is that by minimizing the integrated objective function, we can find a set of latent data-driven attributes and the classification model $\mathbf{w}$ which i) predicts the data correctly with a large margin and ii) minimizes the loss of mutual information caused by feature merging.

Discovering data-driven attributes while treating both human-specified and data-driven action attributes as latent variables makes the objective function intractable. To simplify this process, we use two separate steps. We first find the best partition Π of X by minimizing the loss of MI $\mathcal{L}_{inf}$. This process produces the data-driven attributes. Once we have the data-driven attributes we only need to solve the latent SVM problem. Gradient descent methods can be used to minimize the loss of MI [18].

5. Knowledge Transfer Across Classes

Representing actions with a set of human-specified action attributes makes it possible to recognize a novel action class even when training examples are not available. This is accomplished by transferring knowledge from known classes (with training examples) to a novel class (without training examples), and using this knowledge to recognize instances of the novel class. To formulate the problem, let $\mathcal{T} = \{(\mathbf{x}_i, l_i)\}_{i=1}^n \subset \mathbb{X}^d \times \mathbb{Y}$ be a training set where $\mathbb{Y} = \{y_k\}_{k=1}^K$ consists of K training action classes. Given a set of novel classes $\mathbb{Z} = \{z_j\}_{j=1}^L$ that is disjoint from $\mathbb{Y}$, we seek to obtain a classifier $f : \mathbb{X}^d \rightarrow \mathbb{Z}$. Traditional classification fails to solve this problem since there are no training examples for $\mathbb{Z}$.

As Fig. 2 demonstrates, with human-specified attributes any action class is an m dimensional vector, say $a^y = (a_1^y, ..., a_m^y) \in \mathbb{A}^m$ for a training class y and $a^z = (a_1^z, ..., a_m^z) \in \mathbb{A}^m$ for a novel class z. An action instance is also represented by a point in $\mathbb{A}^m$. Ideally, the positions of action instances will be close to the positions of their corresponding action classes. The attribute vector of an action class is specified manually, while the attribute vector of an action instance is provided by m attribute classifiers. These attribute classifiers, namely the mapping $\mathcal{S} : \mathbb{X}^d \rightarrow \mathbb{A}^m$, are learned from training dataset $\mathcal{T}$ (see section 2). Given an unknown action $\mathbf{x}$ belonging to one of the action classes $\mathbb{Z}$, we first encode it into the attribute space by $\mathcal{S}(\mathbf{x}) \in \mathbb{A}^m$. We can then measure its Euclidean distances to all novel classes $\mathbb{Z}$, and assign it to the nearest class (in these experiments, the K-Nearest Neighbors (KNN) technique was used for classification). Notice that this assignment is possible because we know the mappings $\mathbb{A}^m \rightarrow \mathbb{Z}$ (manually specified) and $\mathbb{X}^d \rightarrow \mathbb{A}^m$ (learned from the training data), even if no training samples are available for the novel classes $\mathbb{Z}$.

6. Experiments and Discussion

6.1. Low-level feature extraction

We adopt the 1D-Gabor detector proposed in [24] to detect 3D interest points from videos of action. In our experiments, we set two parameters σ and τ of the Gabor filter to 2 and 1.5 respectively. The ST volumes around the points are extracted and gradient-based descriptors are learned by PCA. All descriptors are quantized to d (e.g., d = 1000 in our experiments) visual-words using the k-means algorithm. With the quantized vocabulary, each action is represented by a histogram vector $\mathbf{x} \in \mathbb{X}^d$ of visual-words.

6.2. Datasets and Action Attributes

We tested our framework on three publicly available datasets. First, experiments were performed on the UIUC action Dataset [30], which contains about 532 videos of 14 actions, such as *walk*, *hand-clap*, *jump-forward*, and *jump-jack*. The action classes in this dataset are very diverse, which is useful for our study. We manually defined 22 action attributes such as "standing with arm motion", "torso translation with arm motion", "leg fold and unfold motion" [18]. Second, experiments were conducted on a new dataset obtained by combining existing datasets into a larger one. The sources include the KTH dataset [15] (six classes and about 2,300 videos), the Weizmann dataset [21] (10 classes and about 100 videos), and the UIUC dataset (14 classes and about 500 videos), resulting in a combined dataset with 21 actions and 2910 videos in total. The three datasets are combined in order to: 1) produce sufficient number of action classes, which implies that more attributes can be introduced to characterize such classes, 2) add more variability across video sources. We defined 34 action attributes for the combined 21 actions [18]. Finally, experiments were also conducted on the Olympic Sports dataset, which is newly published by Niebles *et al.* [23]. As it is collected from YouTube, it contains realistic human actions.

6.3. Experimental Results

A. Recognizing novel action classes

As aforementioned, human-specified attributes can help recognize novel action classes. We use the *leave-two-classes-out-cross-validation* strategy [25, 17] in experiments on the UIUC dataset. Specifically, for each run we leave two classes out as novel classes ($|\mathbb{Z}| = 2$). The remaining classes are used for training. All 91 possible configurations of training and testing classes are used. Fig 3

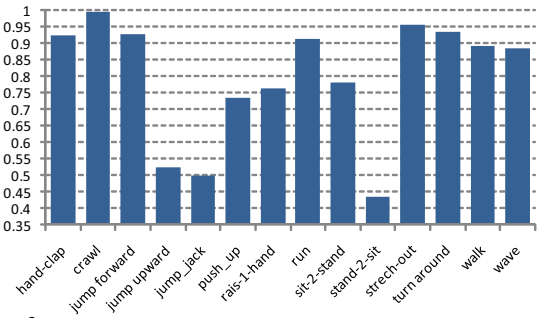
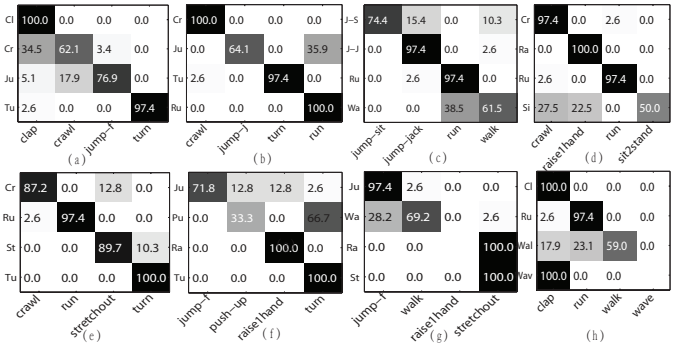


Figure 3. The average accuracy of leave-two-classes-out-cross-validation on the UIUC dataset for recognizing novel action classes.

shows the average accuracy of each action over all runs. We see that the majority of classes are recognized with a success rate of over 70%, and 8 of the classes approach 90%.

For the next experiment, we divide the UIUC dataset into two disjoint sets to make the problem more challenging. One set, $\mathbb{Y}$, contains 10 action classes, and is used for training. The other set, $\mathbb{Z}$, contains four classes and is used for testing. This strategy is similar to that of [14, 28]. The underlying rationale for knowledge transfer across action classes is that the testing and training classes share some common attributes. We follow this criteria to form our test cases. Fig. 4 (a)-(h) list the confusion tables for eight representative cases with varied performance. Some interesting observations can be made from the confusion tables. For example, in (a) some "crawl" actions are misclassified as "clap" since both classes share "alternate arm motion" while "crawl" does not have strong attributes to differentiate itself from "clap". Similarly, both "jump-forward" and "crawl" have "translation motion", so some "jump-forward" examples are misclassified into "crawl". Because no results under zero-shot learning are reported in the literature of the human action classification, we compare our results with one-shot learning, which occurs when each testing class has a single training example. We use KNN as a classifier, and tested on actions represented by raw features without attribute features. Fig.4 (I) compares average accuracies for zero-shot and one-shot learning. For 6 out of the 8 cases, our approach performs about 7% to 22% better than one-shot learning. This supports our claim that knowledge transfer from known classes to novel classes by action attributes can improve recognition.



(%)	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)
zero-shot learning	84.11	90.38	82.69	86.22	93.59	76.28	66.67	62.18
one-shot learning	74.00	77.69	75.58	64.99	80.06	78.04	59.08	67.56

(I)

Figure 4. Confusion table for four novel test classes. (a)-(h) correspond to eight different cases. The average accuracies for (a)-(h) are listed in the first row of table (I), and for comparison the second of (I) lists the average accuracies of the eight cases for "one-shot learning" (best viewed in PDF).

	0-shot	1-shot	10-shots	20-shots	Schuldet	Liu	Niebles
Box	56.7	61.5	80.9	83.4	97.9	96.0	99.2
Clap	84.3	36.9	46.7	55.2	59.7	92.9	96.5
Jog	9.0	13.4	39.3	50.1	60.4	87.0	78.2
Run	64.1	12.0	48.0	59.2	54.9	82.0	79.5
Walk	75.6	13.1	43.8	54.7	83.8	98.0	94.4
Wave	82.9	53.7	83.7	87.4	73.6	92.0	99.9
Ave. Acc.	62.1	31.8	57.1	65.0	71.7	91.3	91.3

Figure 5. The performance comparison on KTH dataset between our approach (column "0-shot") with "x-shots" learning (i.e., using x examples from each class as training) and other state-of-the-art training based approaches: Schuldet *et al.*[29], Liu *et al.* [19], and Niebles *et al.* [23].

Using our combined datasest, we conduct a more challenging experiment, using all six action classes from the KTH dataset as testing classes $\mathbb{Z}$, and action classes that are from the UIUC and Weizmman datasets but not included in $\mathbb{Z}$ as training classes $\mathbb{Y}$. The classification results are shown in Fig.5. For comparison, experiments are conducted with 1, 10 and 20 training examples from each action classes. Again, a KNN classifier is used for classification and actions are represented without using attributes. The results are shown in Fig.5. We see that "0-shot" generally performs much better than "1-shot" and "10-shot", and is competitive with "20-shot". The poorest performance is from the "jog" class. Confusion tables indicate that the majority of "jog" examples are misclassified as "run", which is understandable, as there are no attributes that are capable of distinguishing between the two classes in this experiment. Fig.5 also shows some results with the state-of-the-art bag-of-words approaches on KTH. They typically use very large training sets (e.g., [19] used more than 90% data for training), more discriminative classifiers like SVM, or temporal information [23]. Our results compare well to these approaches, and although our overall performance is slightly weaker, it is important to remember that our results come without any training examples in six of the action classes.

B. Attributes boosting traditional action recognition.

In this section, we present a series of experiments on the MIXED-Action dataset using our proposed framework in sections 3 and 4 to prove that action attributes do improve performance of traditional action recognition. Our results demonstrate that a significant improvement occurs with the use of manually-specified attributes.

We split the dataset into three parts: 40% for training attribute classifiers, 40% for training latent SVM, and 20% for testing. We treat the attributes associated with **x** as a

Ours	Dollar's	Niebles'	Liu's	Wang's	Laptev's	Raptis'
91.59	80.66	91.3	91.31	92.1	91.8	94.5

Figure 7. Performance of some state-of-the-art approaches (most of them are based on bag of words) on the KTH dataset. The first column is our result, the following columns correspond to Dollar *et al.* [24], Niebles *et al.*[23], Liu *et al.*[19] (without spatial structure information), Laptev *et al.*[16], Raptis *et al.*[27] (use tracklets features).

(%)	Average	bend	box	clap	crawl	jack	jog	jump-f	jump-s	pjump	push up	raise-h	run	side	sit-2-s	skip	stand-2-s	Stretch	turn	walk	wave1	wave2
raw-feature	53.1	0.0	98.7	93.1	77.8	18.2	67.5	54.5	66.7	0.0	100.0	11.1	80.0	0.0	75.0	0.0	100.0	11.1	66.7	97.8	0.0	97.6
specified attributes	72.1	33.3	94.9	94.3	100.0	63.6	93.8	81.8	100.0	50.0	83.3	100.0	66.7	0.0	75.0	0.0	100.0	22.2	100.0	92.3	70.0	92.7
raw-feature + specified attributes	78.7	100.0	100.0	95.4	100.0	81.8	87.5	81.8	100.0	0.0	83.3	77.8	77.8	0.0	87.5	0.0	100.0	100.0	100.0	94.5	90.0	95.1
data-driven attributes	47.4	0.0	94.9	93.1	22.2	0.0	61.3	45.5	66.7	0.0	100.0	0.0	78.9	0.0	62.5	0.0	100.0	0.0	77.8	97.8	0.0	95.1
raw-featrue + all attributes	83.1	100.0	92.4	94.3	100.0	90.9	70.0	81.8	100.0	50.0	66.7	100.0	82.2	50.0	87.5	0.0	100.0	100.0	100.0	83.5	100.0	96.3

Figure 6. The performance comparison among raw features human-specified attributes, data-driven attributes and various combination.

feature vector. We trained a binary classifier for each action class, and used a simple voting strategy to obtain multi-class recognition results (the binary classifier giving highest confidence value wins the vote). Fig. 6 shows the performance comparison between varied combinations of different types of features. Each row corresponds to a diagonal entry of a 21 by 21 confusion table. It can be seen that using raw-features alone, our system obtained 53.1% average recognition accuracy, while with human-specified attributes alone, the average accuracy is increased to 72.1%. Clearly, correctly specified attributes help traditional recognition significantly. This can be seen especially well for the action classes that do not have enough training examples (e.g., "bend", "jack", and "wave1"). This is because the attributes transfer knowledge from other classes to compensate for fewer available training examples. Combining both raw features and human-specified attributes, the performance is improved to 78.1%. By adding data-driven attributes with the human-specified attributes, the performance can be further improved by about 4.5%. We conclude that the data-driven attributes provide cues that are complementary to the human-specified attributes. It is worth noting that actions with few training examples (e.g., "side", "pjump" and "skip" those actions from the Weizmann action dataset) generally perform poorly even with the introducing attribute features. For example, "skip" is misclassified to other jump-like actions.

In order to relate our results with those of other methods, we conducted an experiment where we trained classifier on the MIXED dataset (both attribute classifiers and binary action classifiers), and tested on videos from the KTH dataset. Note that a classifier trained on 21 classes may be weaker than the one trained on 6 classes only, and we can not train

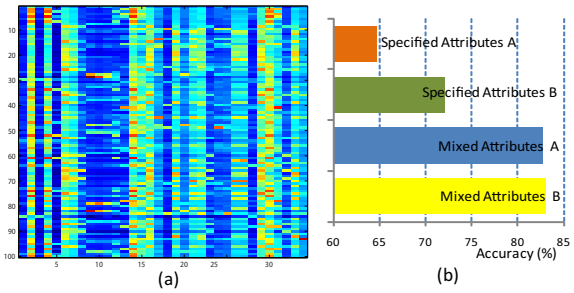


Figure 8. (a) *Dissimilarity* between 100 data-driven attributes (rows) and 34 manually-specified attributes (columns). Colder color has lower value. (b) The effect of removing a set of human-specified attributes. "Specified attributes" means only using this type of attributes for recognition. "B" indicates the performance before attributes removal, while "A" indicates the performance after removing the attributes. "Mixed Attributes" means using both manually-specified and data-driven attributes for recognition (best viewed in PDF).

on the KTH dataset as it does not have sufficient classes to learn robust attribute classifiers. We obtain average accuracies of $\{96.0\%, 95.0\%, 83.8\%, 85.4\%, 90.7\%, 98.7\%\}$ for "box", "clap", "jog", "run", "walk" and "wave" respectively. Fig.7 lists the performance of some state-of-the-art approaches based on bag of features. Notice that even though our binary classifiers are trained on the more complicated MIXED dataset, our result is still competitive.

To further demonstrate the correlation between manually-specified attributes and data-driven attributes, we show a *dissimilarity* map in Fig.8, where colder colors indicate less dissimilarity, namely stronger correlation. This map is constructed from the training data (i.e., the action-to-data-driven-attribute matrix and the action-to-specified-attribute matrix). The dissimilarity between two attributes is computed as the Euclidean distance between their corresponding column vectors. From this map, we see that some specified attributes (e.g., the human-specified attribute set $\bar{a} = \{1, 8, 9, 10, 11\}$, columns of Fig.8 (a)) are more correlated with data-driven attributes. The effect of this correlation can be seen in the following experiments. As Fig. 8 (b) shows, for recognition using manually-specified attributes only, removing $\bar{a}$ decreases the performance from 72% (i.e., "specified attribute B" in (b)) to 64%. However, for recognition using both manually-specified and data-driven attributes, removing $\bar{a}$ doesn't cause an obvious performance decrease (i.e., "Mixed Attributes B" vs. "Mixed Attributes A" in (b)). This shows that data-driven attributes can make up the information loss caused by removing some human-specified attributes.

6.4. Experiments on Olympic Sports Dataset

We validated our approach using the Olympic Sports dataset, which contains 16 action classes and about 781 videos, for recognizing novel action classes and traditional training based recognition. This dataset is more challenging because some of the videos contains camera motion and are taken from varied views. Moreover, many action classes have similar sub-actions, such as "Discus-throw", "Hammer-throw " and "Shot-put ". We defined 39 attributes on this dataset [18] and created a codebook with size of 2,000 for all experiments on this dataset.

A set of experiments for testing the ability to recog-

Case 1		Case 2		Case 3		Case 4		Case 5	
Clean-jerk	75.8	Diving-10m	52.6	Diving-3m	63.0	Clean-jerk	84.8	bowl	0.0
Diving-10m	61.4	Hammer-th	87.0	Javelin-th	32.0	Diving-3m	37.0	Hammer-th	91.3
High-jump	73.8	snatch	83.7	Pole-vault	72.5	Hammer-th	91.3	Javelin-th	88.0
Shot-put	68.3	Long-jump	87.0	Shot-put	71.4	Triple-jump	57.1	snatch	85.7
Ave. Acc.	69.8	Ave. Acc.	77.6	Ave. Acc.	59.7	Ave. Acc.	67.6	Ave. Acc.	66.3

Figure 9. The performance of recognizing novel testing classes. Five cases are listed. For each case, four classes are used for testing and the other 12 classes used for training.

raw-features	51.83
specified attributes	60.48
raw-features + specified attributes	63.60
Data-driven attributes	45.31
raw-feature+ all attributes	65.09

(a)

Our approach	74.38
Raw features	66.93
Niebles et al.	72.10
Laptev et al.	62.00

(b)

Figure 10. (a) Recognition performance on the Olympic Sports dataset using different features and attributes. (b) Performance comparison with some state-of-the-art approaches in terms of mean average precision. Both results corresponding to "Niebles et al." and "Laptev et al." are from [23].

nize novel action classes using manually specified attributes were performed. Four classes are selected as novel testing classes and the rest are used as training classes. Fig.9 shows results for five representative cases. Overall, the results are reasonable, although a few classes perform poorly. With 10-shot learning, the average accuracies for each of five cases is 47.7%, 60.1%, 52.8%, 58.3%, and 60.0% respectively, which are significantly inferior to the average accuracies of our approach (see Fig.9).

In addition, we follow the experimental setup in Sec 6.3 B to see if both manually-specified and data-driven attributes can improve recognition. The *average accuracy* (for multi-class classification) is shown in Fig.10 (a). Again, the manually-specified attributes boost the performance. Due to the noise dataset, the attribute classifiers are not well trained. It causes the improvement of performance for this dataset to be somewhat less significant than that of the combined dataset. For comparison with state-of-the-art approaches, we also compute the *average precision* (which differs from the *average accuracy*) for each class. Fig.10 (b) shows a comparison of mean average precision for different approaches.

7. Conclusion

In this paper, we have proposed to represent human actions by a set of intermediate concepts called action attributes which are either manually specified or learnt from the training data. We have introduced a unified framework wherein the action attributes can be effectively selected in a discriminative fashion. Extensive experiments have been carried to validate our claims and have confirmed our intuition that an attribute-based representation is a critical building block for modeling complex activities from videos.

8. Acknowledgements

Thanks for helpful comments from colleagues and the reviewers. This work is supported by the National Science Foundation (Grant CPS-0931474).

References

[1] A. Yilmaz and M. Shah. Actions sketch: A novel action representation. In *CVPR*, 2005.

[2] J. Aggarwal and Q. Cai. Human motion analysis: A review. In *Nonrigid and Articulated Motion Workshop*, 1997.

[3] T. Berg, A. Berg, and J. Shih. Automatic attribute discovery and characterization from noisy web data. In *ECCV*, 2010.

[4] T. Cover, J. Thomas, J. Wiley, et al. *Elements of information theory*. Wiley Online Library, 1991.

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

[6] A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *CVPR*, 2003.

[7] F. Lv and R. Nevatia. Single view human action recognition using key pose matching and viterbi path searching. In *ICCV*, 2007.

[8] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009.

[9] A. Fathi and G. Mori. Action recognition by learning mid-level motion features. In *CVPR*, 2008.

[10] P. Felzenszwalb and D. Huttenlocher. Efficient belief propagation for early vision. *IJCV*, 70(1):41–54, 2006.

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

[12] N. Ikizler-Cinbis and S. Sclaroff. Object, scene and actions: combining multiple features for human action recognition. In *ECCV*, 2010.

[13] A. Kovashka and K. Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In *CVPR*, 2010.

[14] C. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009.

[15] I. Laptev. On space-time interest points. *IJCV*, 64(2):107–123, 2005.

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

[17] H. Larochelle, D. Erhan, and Y. Bengio. Zero-data learning of new tasks. In *AAAI*, 2008.

[18] J. Liu, B. Kuipers, and S. Savarese. Recognizing human actions by attributes. In *Technical Report, the University of Michigan*, 2011.

[19] J. Liu and M. Shah. Learning human actions via information maximization. In *CVPR*, 2008.

[20] J. Liu, Y. Yang, and M. Shah. Learning semantic visual vocabularies using diffusion distance. In *CVPR*, 2009.

[21] M. Blank, M. Irani and R. Basri. Actions as space-time shapes. In *ICCV*, 2005.

[22] P. Matikainen, M. Hebert, and R. Sukthankar. Representing pairwise spatial and temporal relations for action recognition. In *ECCV*, 2010.

[23] J. Niebles, C. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, 2010.

[24] P. Dollar, V. Rabaud, G. Cottrell and S. Belongie. Hierarchical motion history images for recognizing human motion. In *ICCV workshop: VS-PETS*, 2005.

[25] M. Palatucci, D. Pomerleau, G. Hinton, and T. Mitchell. Zero-shot learning with semantic output codes. In *NIPS*, 2009.

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

[27] M. Raptis and S. Soatto. Tracklet Descriptors for Action Modeling and Video Analysis. In *ECCV*, 2010.

[28] M. Rohrbach, M. Stark, G. Szarvas, I. Gurevych, and B. Schiele. What helps where–and why? Semantic relatedness for knowledge transfer. In *CVPR*, 2010.

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

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

[31] Y. Wang and G. Mori. Max-margin hidden conditional random fields for human action recognition. In *CVPR*, 2009.

[32] Y. Wang and G. Mori. A discriminative latent model of object classes and attributes. In *ECCV*, 2010.

[33] Z. Lin, Z. Jiang and L.S. Davis,. Recognizing actions by shape-motion prototype trees. In *CVPR*, 2009.