DaMN – Discriminative and Mutually Nearest: Exploiting Pairwise Category Proximity for Video Action Recognition

Rui Hou[1], Amir Roshan Zamir[1], Rahul Sukthankar[2], and Mubarak Shah[1]

[1] Center for Research in Computer Vision at UCF, Orlando, USA
[2] Google Research, Mountain View, USA
http://crcv.ucf.edu/projects/DaMN/

Abstract. We propose a method for learning discriminative category-level features and demonstrate state-of-the-art results on large-scale action recognition in video. The key observation is that one-vs-rest classifiers, which are ubiquitously employed for this task, face challenges in separating very similar categories (such as running vs. jogging). Our proposed method automatically identifies such pairs of categories using a criterion of mutual pairwise proximity in the (kernelized) feature space, using a category-level similarity matrix where each entry corresponds to the one-vs-one SVM margin for pairs of categories. We then exploit the observation that while splitting such "Siamese Twin" categories may be difficult, separating them from the remaining categories in a two-vs-rest framework is not. This enables us to augment one-vs-rest classifiers with a judicious selection of "two-vs-rest" classifier outputs, formed from such discriminative and mutually nearest (DaMN) pairs. By combining one-vs-rest and two-vs-rest features in a principled probabilistic manner, we achieve state-of-the-art results on the UCF101 and HMDB51 datasets. More importantly, the same DaMN features, when treated as a mid-level representation also outperform existing methods in knowledge transfer experiments, both cross-dataset from UCF101 to HMDB51 and to new categories with limited training data (one-shot and few-shot learning). Finally, we study the generality of the proposed approach by applying DaMN to other classification tasks; our experiments show that DaMN outperforms related approaches in direct comparisons, not only on video action recognition but also on their original image dataset tasks.

1 Introduction

Attributes are mid-level visual concepts, such as "smiling", "brittle", or "quick" that are typically employed to characterize categories at a semantic level. In recent years, attributes have been successfully applied to a variety of computer vision problems including face verification [11], image retrieval [30], action recognition [15], image-to-text generation [1]. Category-level attributes are popular not only because they can represent the shared semantic properties of visual classes but because they can leverage information from known categories to enable existing classifiers to generalize to novel categories for which there exists limited training data.

D. Fleet et al. (Eds.): ECCV 2014, Part III, LNCS 8691, pp. 721–736, 2014.

Ideally, attributes should capture human-interpretable semantic characteristics that can be reliably recognized by machines from visual data. However, the focus on human-interpretation means that developing attribute classifiers typically demands a labor-intensive process involving manual selection of attribute labels and collection of suitable training data by domain experts (e.g., [12]).

Our stance is that while human interpretability of attributes is obviously desirable, it should be treated as a secondary goal. Thus, we seek fully automated methods that learn discriminative category-level features to serve as useful mid-level representations, directly from data.

We propose DaMN, a method for automatically constructing category-level features for multi-class problems based on combining one-vs-one, one-vs-rest and two-vs-rest classifiers in a principled manner. The key intuition behind DaMN is that similar activities (e.g., jogging vs. running) are often poorly represented by one-vs-rest classifiers. Rather than requiring methods to accurately split such "Siamese Twin" categories, we choose to augment one-vs-rest classifiers with a *judicious selection* of two-vs-rest classifiers that keep the strongly related categories together. The challenge is how best to identify such categories and then how to combine information from the different classifiers in a principled manner. Figure 1 (left) illustrates examples of category pairs identified as closely related by DaMN; the complete grouping of DaMN pairs extracted for UCF101 is shown in Figure 1 (right). It is important to note that the DaMN category pairs are (by construction) similar in kernel space and generally well separated from the remaining categories. By contrast, the manually constructed category-attribute matrix for UCF101 is not as amenable to machine classification despite having human-interpretable names [14]. Such experiences drive us to explore the idea of data-driven category features as an alternative to human selected attributes, with the hope that such features can still offer the benefits (such as cross-dataset generalization and one-shot learning) of traditional attributes.

We are not the first to propose such a data-driven approach. For instance, Farhadi et al. suggested searching for attributes by examining random splits of the data [5] and Bergamo & Torresani recently proposed Meta-class [2], an approach for identifying related image categories based on misclassification errors on a validation set. Although we are the first to apply such approaches for video action recognition, we do not claim this as a significant contribution of our work.

More importantly, our experiments on UCF101 and HMDB51 confirm that these automatically learned features significantly improve classification performance and are also effective vehicles for knowledge transfer to novel categories.

Our paper's contributions can be summarized as follows.

1. We propose a novel, general-purpose, fully automated algorithm that generates discriminative category-level features directly from data. Unlike earlier work, DaMN trains from all of the available data (no random test/train splits nor validation sets).
2. We evaluate DaMN on large-scale video action recognition tasks and show that: a) the proposed category-level features outperform the manually generated category-level attributes provided in the UCF101 dataset; b) DaMN is a strong choice for a mid-level action representation, enabling us to obtain the highest-reported results on the UCF101 dataset.
3. We show that DaMN outperforms existing methods on knowledge transfer, both across dataset and to novel categories with limited training data.

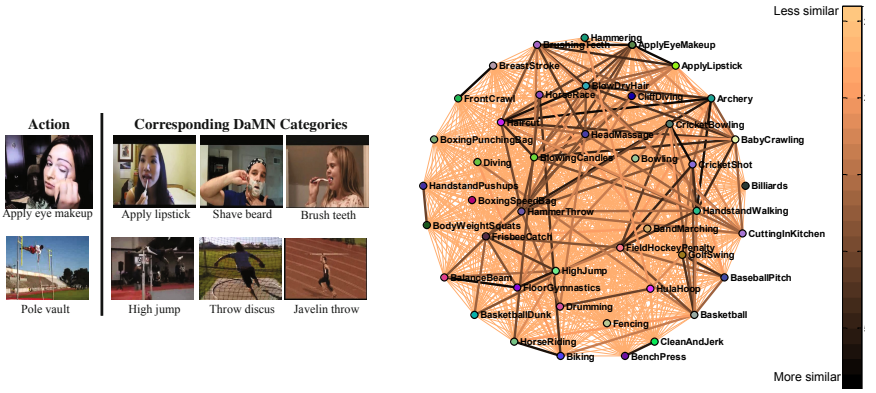

Fig. 1. Left: Examples of DaMN category pairs automatically discovered in UCF101. For the actions shown on the left, three sample mutually nearest categories are shown on the right. Explicitly representing such category-level information enables improved action recognition accuracy. **Right:** Visualization of the extracted pairwise category similarity (Section 3.1) for UCF101. To manage visual clutter, only the first 43 categories are shown, with edges shaded according to the pairwise similarity between the respective classes. We see that DaMN identifies meaningful semantic groups, such as *weightlifting*, *bat-and-ball sports*, and *face-centered actions*. Also note the presence of singleton categories.

4. While our focus is on video action recognition, the proposed method can also be applied to other problems and we show DaMN's superiority on the Animals-with-Attributes [12] dataset.

The remainder of the paper is structured as follows. Section 2 presents an overview of related work. Section 3 details the proposed method of generating DaMN category-level features. Section 4 describes video action recognition experiments on UCF101, knowledge transfer to HMDB51, one-shot learning and a study on semantic attributes vs. data driven category-level features. We also present results on image datasets, consider extensions beyond second-order relationships and provide a brief analysis of DaMN's computational complexity. Section 5 concludes and outlines promising directions for future work.

2 Related Work

This section reviews a representative subset of the related work in this area and details the key differences between the proposed approach and current efforts to directly learn discriminative attributes or category-level features from data.

2.1 Semantic Attributes

The majority of research on attributes focuses on how semantic attributes can better solve a diverse set of computer vision problems [1, 4, 8, 11, 15, 30] or enable new applications [12, 19]. Generally, specifying these semantic attributes and generating suitable

datasets from which to learn attribute classifiers is a difficult task that requires considerable effort and domain expertise. For instance, the popular animals-with-attributes dataset provided by Lampert et al. [12] relied upon a category/attribute matrix that was originally created by domain experts in the cognitive science [18] and AI [9] communities.

Traditionally, semantic attributes are constructed by manually selecting a set of terms that characterize the concept [15, 28]. An important line of work has explored ways to alleviate this human burden. Berg et al. [1] propose to automatically discover attributes by mining text and images on the web. Ferrari and Zisserman [7] learn attributes such as "striped" and "spotted" in a weakly supervised scheme from unsegmented images. Parikh and Grauman [20] propose an interactive scheme that efficiently uses annotator feedback.

While all these methods generate human-interpretable semantic attributes, our experiments motivate us to question whether semantic attributes are necessarily superior to those learned directly from low-level features. The recently released large-scale action recognition dataset, UCF101 [22] has been augmented by a set of category-level attributes, generated using human rater judgments, for the ICCV'13 THUMOS contest on Action Recognition [14]. Section 4 discusses that even though the THUMOS semantic attributes encode an additional source of information from human raters, they are significantly outperformed by the automatically learned DaMN features.

2.2 Data-Driven Category-Level Features

Category-level features (sometimes termed *data-driven attributes*) are mid-level representations that are typically learned from low-level features without manual supervision. Their main drawbacks are that: 1) the distinctions found in the feature space may not correspond in any obvious way to a semantic difference that is visible to humans; 2) unlike the attributes described in Section 2.1, which incorporate additional domain knowledge from human raters, it is unclear whether data-driven features should be expected to glean anything from the low-level features that a state-of-the-art classifier employing the same features would fail to extract.

Farhadi et al. [5] discover promising attributes by considering large numbers of random splits of the data. Wang & Mori [26] propose a discriminatively trained joint model for categories and their visual attributes in images. Liu et al. [16] combine a set of manually specified attributes with data-driven attributes for action recognition. Yang and Shah [27] propose the use of data-driven concepts for event detection in video. Mensink et al. [17] propose an efficient method for generalizing to new categories in image classification by extending the nearest class mean (NCM) classifier that is reminiscent of category-level features. Our work is related to that of Yu et al. [29], which independently proposes a principled approach to learning category-level attributes from data by formulating the similarity between two categories based on one-vs-one SVM margins. However, DaMN differs from [29] in several key respects. First, our features are real-valued while theirs (like most attributes in previous work) are binary. Second, our method centers on identifying pairs of strongly-connected categories (mutual near neighbors) while theirs is set up as a Rayleigh quotient optimization problem and solved using a greedy algorithm. Third, their application is in image classification while we are primarily interested in action recognition with a large number of classes. Nonetheless, we show in direct comparisons that DaMN outperforms [29] using their experimental methodology on image datasets.

At a high level, the work most closely related to DaMN is Bergamo & Torresani's Meta-Class features, which also seek to group categories in a data-driven manner in order to improve upon one-vs-rest classifiers. The fundamental difference is in how the two algorithms determine which categories to group: meta-class trains an auxiliary SVM classifier and treats the classification error (computed over a validation set) as a proxy for the similarity between two categories. By contrast, DaMN directly formulates principled measures for this category-level distance, such as the pairwise (one-vs-one) SVM margin. The latter is an improvement both in terms of theory and is experimentally validated in direct comparisons on several datasets.

2.3 Knowledge Transfer

Category-level features and attributes are well suited to vision tasks that require trained classifiers to generalize to novel categories for which there exists limited training data. This is because an classifier trained to recognize an attribute such as "furry" from cats and dogs is likely to generalize to recognizing other furry mammals. Attribute-driven knowledge transfer has been successfully demonstrated both in the one-shot [12] and zero-shot [19, 21] context. In the action recognition domain, Liu et al. [15] explore attribute-driven generalization for novel actions using manually specified action attributes.

Our approach for generalizing to novel categories is purely data-driven. At a high level, our approach to one-shot learning is related to the use of one-vs-rest classifiers or classemes [23] by which novel categories can be described; DaMN features can be viewed as augmenting one-vs-rest with the most useful subset of potential two-categories-vs-rest classifiers. Our experiments (Section 4) confirm that our DaMN category-level features are significantly superior to existing approaches in both cross-dataset and novel category generalization scenarios.

3 Proposed Approach: DaMN

Figure 2 (right) provides a visual overview of the DaMN process, detailed further in this section. Action recognition in video is typically formulated as a classification problem where the goal is to assign a category label $y \in \mathcal{Y}$ to each video clip $v \in \mathcal{V}$. Although our proposed method also works with more complex feature families, let us simplify the following discussion by assuming that the given video v is represented by some feature $\mathbf{x}_v \in \mathbb{R}^d$.

Let $n = |\mathcal{Y}|$ denote the number of categories and $\mathcal{F}$ the set of category-level features that describe how categories are related. The size of this set, $m = |\mathcal{F}|$; $m = n$ in the case of one-vs-rest classifiers, and $m > n$ for DaMN — with the number of two-vs-rest classifiers given by $m - n$. In general, the category-level relationships can be expressed by a binary-valued matrix $\mathbf{B} \in \mathbb{B}^{n \times m}$. The ith row of $\mathbf{B}$ contains the features for category y_i while the jth column of the matrix denotes the categories that share a given category-level feature. Each column is associated with a classifier, $f_j(\mathbf{x}_v)$ (such as a kernel SVM), trained either one-vs-rest or two-vs-rest, that operates on a given instance (video clip) v.

Our goal is to automatically identify the set of suitable features $\mathcal{F}$, train their associated classifiers $f(.)$ using the available training data and use the instance-level predictions to classify novel videos. In the additional task of one-shot learning, we use the same category-level features with an expanded set of categories, $\mathcal{Y}'$, for each of which we are given only a single exemplar.

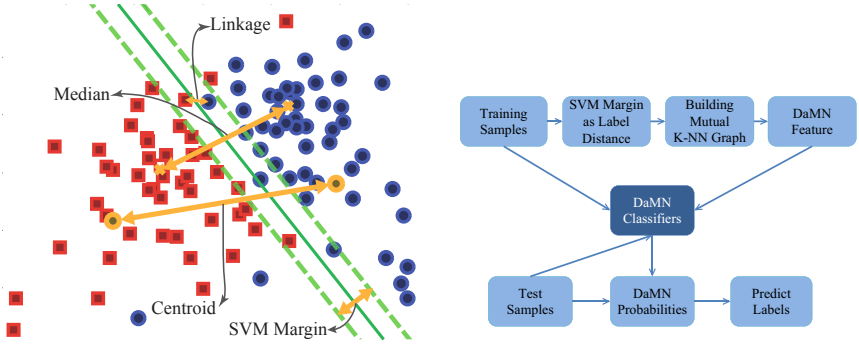

Fig. 2. Left: Illustration of different category-level distance metrics. The illustration uses a linear kernel only to simplify visualization; all of these, except for centroid, are actually performed in kernel space. **Right:** Overview of the DaMN Learning/Prediction Process.

3.1 Pairwise Category Similarity

We train one-vs-one SVM classifiers for every pair of categories to compute the category-level similarity. We propose a natural generalization from the instance-level similarity, typically expressed through some kernel function $K(.,.)$ that compares their respective low-level feature representation (e.g., using χ^2 for bag-of-words features) typically used in action recognition to category-level similarity as the margin of a one-vs-one SVM trained to discriminate the two categories, expressed in the dual form as

$$d_{y_i,y_j} = \sum_{\forall (p,q) \in y_i \cup y_j} \alpha_p \alpha_q (-1)^{I[c(p) \neq c(q)]} K(\mathbf{x}_p, \mathbf{x}_q) \tag{1}$$

where p and q are all pairs of instances from the union of the two categories, α their non-negative weights (> 0 only for support vectors), $c(.)$ is a function that returns the category of the given instance and $I[.]$ denotes the indicator function whose value is 1 when its argument is true and 0 otherwise. Figure 1 (right) visualizes the similarity values computed using this metric for UCF101; to manage visual clutter, we show only the first 43 categories, with edges shaded according to the pairwise similarity between respective categories. Note that even though there are $\binom{n}{2}$ such classifiers, they are quick to train since each classifier uses only a tiny subset of the training data — a fact exploited by one-vs-one multi-class SVMs.

Given similarities between categories (at the feature level), we seek to identify pairs of categories that are close. For this, we construct a mutual k-nearest neighbor (kNN) graph over categories. A mutual kNN graph is similar to the popular kNN graph except that nodes p and q are connected if and only if p lists q as among its k closest neighbors and q also lists p among its k closest neighbors. Let $G = (V, E)$ be a graph with n nodes, V, corresponding to the category labels y and weights given by:

$$w_{pq} = \begin{cases} 0 & y_p \text{ and } y_q \text{ are not mutual } k\text{NN}, \\ d_{y_p,y_q} & \text{otherwise.} \end{cases} \tag{2}$$

Unlike a k-nearest neighbor graph, where every node has degree k, a mutual kNN graph exhibits much sparser connectivity.

3.2 Constructing the DaMN Category-Level Feature Matrix

The DaMN category-level feature matrix is composed of two parts: 1) features to separate each category individually (identical to one-vs-rest); 2) pair classifiers designed to separate mutually proximate pairs of categories from the remaining classes. Among the $\binom{n}{2}$ possible category pairs, DaMN selects only those that are mutually proximate ($w_{pq} > 0$) as additional features.

Thus, $\mathbf{B}$ is an $n \times m$ matrix composed of two portions: an $n \times n$ identity matrix corresponding to the one-vs-rest classifiers augmented by additional columns for the selected category pairs. Each of the additional columns c in the second block is associated with a category pair (p, q) and the entries of the new column are given by:

$$B_{ic} = \begin{cases} 1 & \text{if } i = p \text{ or } i = q \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

3.3 Training Category-Level Feature Classifiers

Each column in $\mathbf{B}$ defines a partition over labels that is used to split the training set into positives (those instances whose labels possess the given feature) and negatives (instances lacking the feature). We learn a χ^2 kernel SVM using this data split to predict the new feature.

Since the first n columns of $\mathbf{B}$ have a single non-zero entry, they correspond to training regular one-vs-rest SVMs. The remaining $m - n$ columns have two non-zero entries and require training a set of somewhat unusual two-category-vs-rest SVMs to separate similar category pairs from the other classes.

3.4 Predicting Categories

Given a video v from the test set, we extract its low-level features x_v and pass it through the trained bank of DaMN classifiers, both one-vs-rest and two-vs-rest. We also have a set of one-vs-one classifiers to distinguish between the two categories in one DaMN pair. For the given video v, let $P(y_i)$ denote the probabilistic score returned by the one-vs-rest classifier of category y_i, and $P(y_i \oplus y_j)$ represent the score computed by the two-vs-rest classifier of the DaMN pair (y_i, y_j). Also, let $P(y_i|y_i \oplus y_j)$ denote the score returned by the one-vs-one classifier which distinguishes between the categories y_i and y_j. All of the SVM classifier scores are Platt-scaled to provide probabilistic values.

We now compute a score that quantifies the match between v and each candidate category y_i by combining the traditional one-vs-rest score with those two-vs-rest scores that involve y_i, weighted by a one-vs-one classifier between y_i and the other class in the DaMN pair. All of these individual probabilities are then combined to obtain the final score for category y_i:

$$T_{y_i} = \frac{P(y_i) + \sum_{\{j:(y_i,y_j)\in \text{DaMN}\}} P(y_i \oplus y_j) \times P(y_i|y_i \oplus y_j)}{\sum_{c=1}^{m} B_{ic}}. \tag{4}$$

The argument of the summation in the numerator is equivalent to the following probabilistic rule for finding the probability of event a: $P(a) = P(a \cup b) \times P(a|a \cup b)$. The set $\{j : (y_i, y_j) \in \text{DaMN}\}$ represents the DaMN pairs that involve the category y_i. Since there are several of such pairs, the score T_{y_i} is defined as the mean of the scores acquired

from all of those pairs as well as the score of the one-vs-rest classifier (i.e., $P(y_i)$). Therefore, the denominator of Eq. 4 is equal to the total number of DaMN pairs that involve category y_i plus one (to count for its one-vs-rest classifier).

Since, as seen in Figure 1 (right), some categories participate in many more DaMN pairs than others, the final score for such categories involves more terms. At the other extreme are categories that do not show up in any DaMN pair, for which only the one-vs-rest score is used; this situation corresponds to a highly distinctive category, which is easily discriminated from others (as evidenced by a high margin in kernel space). An intuitive illustration is that when computing a score for v with respect to the category *running*, we should focus on DaMN features generated by pairs like *jogging-running* or *walking-running* rather than those seeded by *makeup-shaving*.

Finally, we assign the test video to the category with the highest match score:

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} (T_y) \,. \tag{5}$$

3.5 Knowledge Transfer to Novel Categories

This section details how DaMN features enable generalization to novel categories through knowledge transfer. Consider the scenario where m features have been generated using abundant training data from n categories, resulting in an $m \times n$ binary matrix and an $n \times n$ adjacency matrix describing the weighted mutual kNN graph. We are now presented with n' novel categories, each containing only a few instances. The goal is to classify instances in the test set generated from all $n + n'$ categories (both existing and novel).

We proceed as follows. First, for each new label y', we determine its similarity to the known categories $\mathcal{Y}$ using the small amount of new data and Equation 1. Note that while the data from novel categories may be insufficient to enable accurate training of one-vs-rest classifiers, it is sufficient to provide a rough estimate of similarity between the new category and existing categories. From such rough estimate, we determine the k categories that are most similar to y' and synthesize a new row for the matrix $\mathbf{B}$ for the novel category from the rows of similar categories:

$$B_{y'j} = \sum_{y \in k\mathrm{NN}(y')} B_{yj}, \tag{6}$$

where $\sum$ is treated as the OR operator since $\mathbf{B}$ is a binary-valued matrix and kNN(.) returns the k categories most similar to the novel category.

At test time, we obtain scores from the bank of existing DaMN feature classifiers and determine the most likely category using Equations 4 and 5. Note that for the novel categories, Equation 4 employs the synthesized values in the matrix.

4 Experiments

Our experimental methodology focuses on clarity, thoroughness and reproducibility rather than tuning our system to squeeze the best possible results. Even under these conditions, as detailed below, DaMN generates the best classification results to date on both the UCF101 and HMDB51 datasets.

All of our experiments are conducted on the two latest publicly available action recognition datasets, UCF101 [22] and HMDB51 [10], which contain YouTube videos

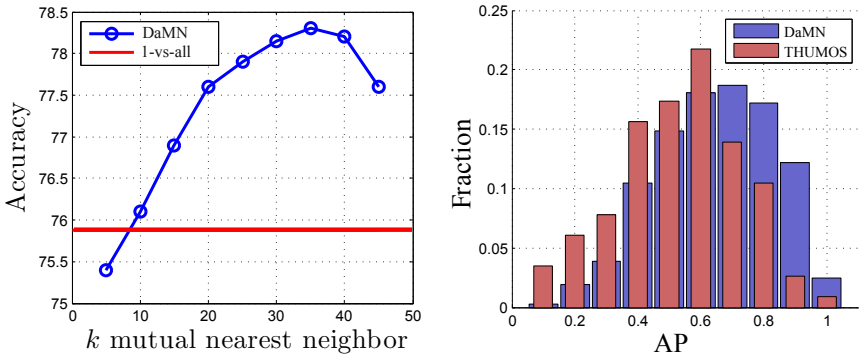

Fig. 3. Left: DaMN is tuned using a single hyper-parameter, k. Accuracy on UCF101 with DTF (late fusion) as k is varied. DaMN outperforms the strongest baseline for k>15 and even at its worst, DaMN outperforms all remaining methods. **Right:** Histogram showing the fraction of classifiers that achieve a given AP on UCF101. DaMN features can be detected more reliably than the manually prescribed attributes provided by THUMOS.

from 101 and 51 categories, respectively. For UCF101 experiments, we closely follow the protocol specified in the ICCV'13 THUMOS action recognition challenge [14], for which test/train splits, manually generated category-level semantic attributes and baseline results are provided.

We restrict ourselves to low-level features for which open source code is available, and select the Dense Trajectory Features (DTF) [25] and Improved Dense Trajectory Features (I-DTF) [24] based on their state-of-the-art results reported in recent competitions. DTF consists of four descriptors: histogram-of-gradients (HOG), histogram-of-flow (HOF), motion-based histograms (MBH) and trajectories. The descriptors are traditionally combined using early fusion (concatenation) but we also present results using the individual component descriptors and late fusion (using equally weighted combination of SVMs). For each descriptor, we generate a 4000-word codebook and build a standard bag-of-words representation for each video by aggregating bin counts over the entire clip.

For classification, we employ the popular LIBSVM [3] implementation of support vector machines with C=1 and the χ^2 kernel, since our features are histograms. Because DaMN already employs one-vs-one SVMs for determining category proximity, results from multi-class one-vs-one SVMs serve as a natural baseline. Following Yu et al. [29], we also train 101 one-vs-rest SVM classifiers to serve as a stronger baseline[1].

Implementing DaMN is straightforward, but we provide open-source code to enable the research community to easily duplicate our experiments and to employ DaMN in other domains.

[1] Others (e.g., the majority of the submissions to ICCV'13 THUMOS Challenge [14]) report that one-vs-rest SVMs consistently outperform multi-class one-vs-one SVMs for reasons that are not clearly understood; an observation that merits further study.

4.1 Action Recognition on UCF101 and HMDB51

All DaMN category-level features for UCF101 were generated with the single hyper-parameter k=35 (see below for experiments showing sensitivity to choice of k). Following the ICCV'13 THUMOS [14] protocol, we present results averaged over the 3 provided folds. The small number next to the mean accuracy in the tables is the standard deviation of the accuracy over the 3 folds. Table 1 summarizes the action recognition results. We present direct comparisons against a variety of baselines as well as DaMN's competitors, one-vs-rest and meta-class. Meta-class requires a validation set and cannot train on all of the available data; to eliminate the possibility that DaMN performs better solely due to the additional data, we provide additional rows (denoted DaMN$^-$ and one-vs-rest$^-$) where these algorithms were trained on a reduced dataset (instances in the validation set used by Meta-class are simply discarded). We also provide two baselines from the THUMOS contest: 1) the THUMOS contest baseline using STIP [13] + bag-of-words + χ^2 kernel SVM, and 2) a baseline computed using the manually generated THUMOS semantic attributes for UCF101. For the latter baseline, we employ the same methodology as DaMN to ensure a fair direct comparison. We make the following observations.

First, we note that switching from STIP to DTF features alone generates a large improvement over the THUMOS baseline and combining DTF components using late fusion (LF) is generally better than with the early fusion (EF) originally employed by the DTF authors. These are consistent with the results reported by many groups at the ICCV THUMOS workshop and are not claimed as a contribution. We simply note that STIP features are no longer a strong baseline for future work in this area.

Second, we observe that the manually generated THUMOS semantic features are outperformed by all of the methods. This drives a more detailed investigation (see Section 4.2).

Third, we note that the DaMN features in conjunction with *any* of the component features (either individual of fused) provides a consistent boost over both one-vs-rest or meta-class, regardless of experimental condition. In particular, DaMN outperforms meta-class even after discarding a portion of the data (which meta-class employs for estimating category-level similarity). Interestingly, meta-class outperforms one-vs-rest only when one-vs-rest is not given access to the full data (one-vs-rest$^-$); this demonstrates that, unlike DaMN, meta-class makes inefficient use of the available data and is not a recommended technique unless there is an abundance of training data. This additional training data boosts DaMN's accuracy by a further 6% in the late-fused DTF condition (DaMN vs. DaMN$^-$), which convincingly shows the benefits of the proposed approach over previous methods.

As discussed in Section 3, DaMN only employs a single hyper-parameter. Figure 3 (left) shows how UCF101 classification accuracy varies with k using the DTF features (late fusion). We observe that DaMN is better than the strongest baseline after k=15 and peaks on UCF101 around k=35. Even the worst instance of DaMN (k=5) is better than all of the remaining methods.

Table 2 shows action recognition results on HMDB51 using its standard splits [10]. The selected parameters for DaMN and the baselines for this dataset were the same as for UCF101. DaMN achieves state-of-the-art results on HMDB51, outperforming the recent results [24] that employ one-vs-rest SVM on Improved DTF features.

Table 1. UCF101 Results. DaMN consistently boosts results across all features. DaMN achieves the best reported results on UCF101 using I-DTF.

	I-DTF [24]	DTF (LF)	DTF (EF)	MBH	HOG	HOF	Traj.
DaMN	**87.00**±1.1	**78.33**±1.7	**75.93**±1.8	**73.25**±1.3	**57.60**±0.6	**57.42**±2.1	**55.18**±1.6
1-vs-rest	85.90±1.2	75.88±2.4	74.60±2.5	71.32±2.9	56.94±2.3	56.08±3.1	53.72±1.6
1-vs-1	79.12±1.9	69.25±3.3	68.32±3.6	66.00±2.4	51.40±3.2	51.80±2.3	48.63±1.1
DaMN$^-$	**80.03**±0.4	**71.82**±1.4	**70.04**±1.5	**66.73**±1.1	**51.63**±0.7	**49.83**±1.9	**49.74**±1.5
Meta-class	78.65±0.6	69.71±1.8	68.32±1.4	60.07±2.3	44.15±2.6	44.98±0.8	43.91±1.0
1-vs-rest$^-$	78.54±0.6	67.84±1.9	66.91±2.1	62.34±2.2	43.71±1.6	44.93±1.4	43.71±1.3
Semantic	58.99	50.19	49.73	51.56	32.68	43.77	33.85
THUMOS [14] baseline (STIP + BOW + χ^2 SVM): 43.9%							

Table 2. HMDB51 Results. DaMN achieves the best reported results on this dataset.

	DaMN	1-vs-rest	Meta-class
I-DTF	**57.88**±0.46	57.01±1.44	57.36±0.24

4.2 DaMN vs. THUMOS Semantic Attributes

To be of practical use, attributes need to be semantic as well as machine-detectable. The THUMOS attributes clearly capture the first criterion, since they were developed by human raters. However, since they are category-level there is a danger that the attributes envisioned by the raters may not actually be visible (or reliably detectable) in a given instance from that category.

The accuracy of an attribute classifier captures the reliability with which the given attribute can be recognized in data. Figure 3 (right) plots histograms of accuracy for THUMOS and DaMN classifiers and Table 3 (left) summarizes some key statistics.

Table 4 examines a natural question: how do different choices for the category-level distances illustrated in Figure 2 (left) impact DaMN's performance? We briefly describe the choices. Given the set of distances $\{d_{ij} : i \in I, j \in J\}$ for kernel distances between instances in categories I and J, *linkage* is defined as the minimum distance; *median* is the median distance between pairs; *average* is the mean over the distances in this set; and *centroid* is the distance between the centroids of each category (in feature space). *SVM Margin* is the margin for an one-vs-one SVM trained on instances from each category. We see that the SVM Margin is better for almost all features, is robust to outliers and is thus the default choice for DaMN in this paper.

We observe that the DaMN classifiers are more reliable than the THUMOS ones. The DaMN features were designed to identify pairs of categories that are mutually close (i.e., share an attribute) and to separate them from the remaining categories. As confirmed by these statistics, such a data-driven strategy enables very accurate recognition of attributes. Fortuitously, as shown in Figures 1 (left) and 1 (right), the selected pairs are also usually (but not always) meaningful to humans. Our results confirm that any price that we may pay in terms of human interpretability is more than offset by the gains we observe in recognition accuracy, both at the attribute and the category level.

In the remaining experiments, we evaluate how well DaMN enables knowledge transfer, both within and across datasets.

Table 3. Left: Performance vs. THUMOS semantic attributes. **Right**: Marginal benefits obtained by adding three-vs-rest classifiers to DaMN.

	mAP	StD	Min	Max
THUMOS	0.53	0.19	0.07	99.08
DaMN	**0.64**	0.18	0.12	95.74

	DTF (LF)	DTF (EF)	MBH	HOG	HOF	Traj.
pairs	**78.33**	75.93	**73.25**	**57.60**	**57.42**	55.18
triples	77.88	**76.30**	73.04	57.51	57.20	**55.28**

Table 4. Empirical evaluation of different category-level distance metrics (in the feature space projected using χ^2 kernel)

	DTF (LF)	MBH	HOG	HOF	Traj.
SVM Margin	**78.33**±1.7	**73.25**±1.3	**57.60**±0.6	57.42±2.1	**55.18**±1.6
Average	77.59±1.6	72.84±1.7	57.41±0.8	57.20±1.6	53.82±0.7
Linkage (Min)	77.38±1.8	72.66±1.1	57.12±0.3	57.05±1.8	55.03±1.4
Median	77.79±1.7	72.97±1.5	57.49±0.8	**57.50**±1.7	54.81±1.7
Centroid	77.22±1.9	72.64±1.5	57.49±0.8	57.00±2.0	54.60±1.6

4.3 Cross-Dataset Generalization to HMDB51

The focus of this experiment is to see how well the DaMN category-level features learned on UCF101 enable us to perform action recognition on a subset of 12 HMDB51 categories with no additional training. Since the category names are different and UCF101 categories more fine-grained, we roughly align them as shown in Table 5.

We follow the same experimental settings as in Section 4.1 but use just the MBH feature rather than late fusion for simplicity. We perform experiments using 3-fold cross-validation, with each run training on two folds of UCF101 and testing on the third fold of HMDB51. Table 5 shows results averaged over these three folds. We see that DaMN achieves an overall higher accuracy than both one-vs-rest and meta-class on cross-dataset generalization.

4.4 Generalization Performance with Limited Training Data

A popular use for attributes and category-level features is that they enable generalization to novel classes for which we have small amounts of training data; an extreme case of this is one-shot learning, where only a single exemplar is provided for each novel category [6].

For this experiment, we randomly select 10 categories to serve as "novel" and treat the remaining 91 as "known". Results are averaged on the three folds specified by THUMOS. We learn DaMN features (k=30) and train semantic and classeme (one-vs-rest) classifiers using the data from two folds of the known classes; we test on the entire third fold of the novel categories. As in cross-dataset scenario, all classifiers use the MBH feature rather than late fusion for simplicity.

We vary the number of training instances per novel category from 1 to 18, while ensuring that all three methods are given identical data. Figure 4 (left) summarizes the results (averaged over three folds). As in the UCF101 action recognition experiments (Section 4.1), the semantic attributes perform very poorly and DaMN does the best, outperforming both meta-class and one-vs-rest in every trial.

The difference between DaMN and the next best is much greater in this experiment, often more than 20%. This demonstrates that the information captured in DaMN generalizes much better across classes.

Table 5. Cross-Dataset Generalization: UCF101 → HMDB51

HMDB51 label	UCF101 IDs	DaMN	1-vs-rest	Meta-class
Brush hair	13	**57.78**	40.00	54.44
Climb	74 & 75	74.44	**83.33**	75.56
Dive	26	**77.78**	57.78	70.00
Golf	33	**66.67**	**66.67**	68.89
Handstand	30	43.33	**45.56**	41.11
Pullup	70	58.89	**68.89**	56.67
Punch	17 & 18	**81.11**	80.00	72.22
Pushup	72	**62.22**	56.67	54.44
Ride bike	11	72.22	**73.33**	67.78
Shoot ball	8	22.22	25.56	**30.00**
Shoot bow	3	**38.89**	36.67	**43.33**
Throw	7	**57.78**	**57.78**	54.44
Average		**59.44**±0.7	57.69±0.8	54.44±1.1

4.5 Extending DaMN to Image Datasets

Although DaMN was primarily developed for applications in action recognition, we recognize that the algorithm can be applied to any classification task. In order to demonstrate the generality of our method, we present a direct comparison against Yu et al.'s recently published method [29] for eliciting category-level attributes on the Animals with Attributes images dataset [12].

Figure 4 (right) shows the accuracy of DaMN, one-vs-rest and Yu et al. [29] using the experimental methodology and data splits prescribed in [29]. We see that one-vs-rest and Yu et al. perform similarly, but both are consistently outperformed by DaMN.

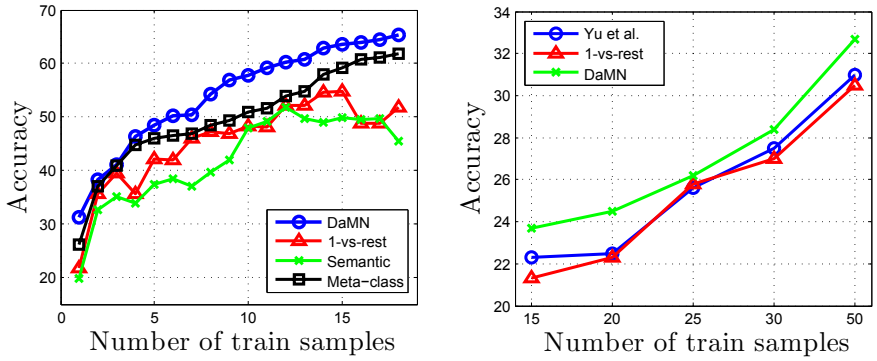

Fig. 4. Left: Knowledge transfer to novel categories with limited data (1 to 18 instances per category). DaMN consistently outperforms one-vs-rest, meta-class and THUMOS semantic attributes. **Right:** Generalization of DaMN to image datasets. DaMN outperforms one-vs-rest and Yu et al. [29] on the Animals with Attributes dataset.

4.6 Extending DaMN Beyond Pairs

Just as DaMN improves over one-vs-rest by judiciously mining suitable two-vs-rest classifiers, it is natural to explore whether one can obtain additional benefits by considering higher-level information, such as cliques consisting of 3 (or more) mutually near categories. Table 3 (right) shows that the improvements are marginal. We believe that this is due to several reasons: 1) there are relatively few such higher-order category groupings; 2) many of the relationships are already captured by the DaMN features since a triplet of similar categories also generates three two-vs-rest pairs; 3) the additional complexity of discriminating within the higher-order grouping of categories may not be merited, whereas the one-vs-one classifiers used to differentiate within the pair were already trained during the DaMN pair selection process. For these reasons, we do not extend DaMN beyond pair categories.

4.7 Computational Complexity

While DaMN may seem to be prohibitively more expensive in terms of computation than the popular one-vs-rest classifiers employed in the action recognition community, we show that training these additional classifiers is both computationally manageable and worthwhile since they generate such consistent (if small) improvements in mAP and classification accuracy. In the interests of space, we briefly summarize wall-clock times for the UCF-101 experiments. Total time for one-vs-one SVM training: 10.14s; one-vs-one SVM testing: 1682.72s; total one-vs-rest and two-vs-rest SVM training: 2752.74s; one-vs-rest and two-vs-rest testing: 2546.51s. For each descriptor type in DTF, DaMN employs ∼800 one-vs-one SVMs, ∼800 two-vs-rest SVMs and 101 one-vs-rest SVMs. Training time for the two-vs-rest classifiers is similar to the one-vs-rest since they have the same number of training instances, but with different labels.

5 Conclusion

We present a novel method for learning mid-level features in a discriminative, data-driven manner and evaluate on large-scale action recognition datasets. The DaMN features are selected on the basis of category-level mutual pairwise similarity and are shown to convincingly outperform existing approaches, both semantic as well as data-driven on a broad set of tasks.

The natural direction for future work is exploring how our category-level features can apply to problems outside action recognition (and possibly beyond computer vision). We believe that the image classification experiments where DaMN outperform recent approaches are a promising sign in this direction. We hope that our open source implementation encourages the community to use the DaMN approach on new tasks.

References

1. Berg, T.L., Berg, A.C., Shih, J.: Automatic attribute discovery and characterization from noisy web data. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 663–676. Springer, Heidelberg (2010)
2. Bergamo, A., Torresani, L.: Meta-class features for large-scale object categorization on a budget. In: Computer Vision and Pattern Recognition (2012)

3. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2(3) (2011), software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm
4. Duan, K., Parikh, D., Crandall, D., Grauman, K.: Discovering localized attributes for fine-grained recognition. In: Computer Vision and Pattern Recognition (2012)
5. Farhadi, A., Endres, I., Hoeim, D., Forsyth, D.: Describing objects by their attributes. In: Computer Vision and Pattern Recognition, pp. 1778–1785 (2009)
6. Fei-Fei, L., Fergus, R., Perona, P.: One-shot learning of object categories. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(4), 594–611 (2006)
7. Ferrari, V., Zisserman, A.: Learning visual attributes. In: NIPS (2007)
8. Jain, A., Gupta, A., Rodriguez, M., Davis, L.: Representing videos using mid-level discriminative patches. In: Computer Vision and Pattern Recognition (2013)
9. Kemp, C., Tenenbaum, J.B., Griffiths, T.L., Yamada, T., Ueda, N.: Learning systems of concepts with an infinite relational model. In: AAAI (2006)
10. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: HMDB: A large video database for human motion recognition. In: International Conference on Computer Vision (2011)
11. Kumar, N., Berg, A.C., Belhumeur, P.N., Nayar, S.K.: Attribute and simile classifiers for face verification. In: International Conference on Computer Vision (2009)
12. Lampert, C., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: Computer Vision and Pattern Recognition (2009)
13. Laptev, I., Marszałek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: Computer Vision and Pattern Recognition (2008)
14. Laptev, I., Piccardi, M., Shah, M., Sukthankar, R., Jiang, Y.G., Liu, J., Roshan Zamir, A.: THUMOS: ICCV 2013 workshop on action recognition with a large number of classes (2013)
15. Liu, J., Kuipers, B., Savarese, S.: Recognizing human actions by attributes. In: Computer Vision and Pattern Recognition, pp. 3337–3344 (2011)
16. Liu, J., Luo, J., Shah, M.: Recognizing realistic actions from videos "in the wild". In: Computer Vision and Pattern Recognition (2009)
17. Mensink, T., Verbeek, J., Perronin, F., Csurka, G.: Distance-based image classification: Generalizing to new classes at near-zero cost. IEEE Transactions on Pattern Analysis and Machine Intelligence 35(11) (2013)
18. Osherson, D.N., Stern, J., Wilkie, O., Stob, M., Smith, E.E.: Default probability. Cognitive Science 15(2) (1991)
19. Palatucci, M., Pomerleau, D., Hinton, G., Mitchell, T.: Zero-shot learning with semantic output codes. In: NIPS (2009)
20. Parikh, D., Grauman, K.: Interactively building a discriminative vocabulary of nameable attributes. In: Computer Vision and Pattern Recognition, pp. 1681–1688 (2011)
21. Rohrbach, M., Stark, M., Schiele, B.: Zero-shot learning in a large-scale setting. In: Computer Vision and Pattern Recognition (2011)
22. Soomro, K., Zamir, A.R., Shah, M.: UCF101: A dataset of 101 human action classes from videos in the wild. Tech. Rep. CRCV-TR-12-01, UCF Center for Research in Computer Vision (November 2012)
23. Torresani, L., Szummer, M., Fitzgibbon, A.: Efficient object category recognition using classemes. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 776–789. Springer, Heidelberg (2010)
24. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: International Conference on Computer Vision (2013)

25. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Action Recognition by Dense Trajectories. In: Computer Vision and Pattern Recognition, pp. 3169–3176 (2011), http://hal.inria.fr/inria-00583818/en
26. Wang, Y., Mori, G.: A discriminative latent model of object classes and attributes. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 155–168. Springer, Heidelberg (2010)
27. Yang, Y., Shah, M.: Complex events detection using data-driven concepts. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part III. LNCS, vol. 7574, pp. 722–735. Springer, Heidelberg (2012)
28. Yao, B., Jiang, X., Khosla, A., Lin, A.L., Guibas, L.J., Fei-Fei, L.: Action recognition by learning bases of action attributes and parts. In: International Conference on Computer Vision (2011)
29. Yu, F., Cao, L., Feris, R., Smith, J., Chang, S.F.: Designing category-level attributes for discriminative visual recognition. In: Computer Vision and Pattern Recognition (2013)
30. Yu, F.X.: Weak attributes for large-scale image retrieval. In: Computer Vision and Pattern Recognition, pp. 2949–2956 (2012)