Attribute-Based Classification for Zero-Shot Visual Object Categorization

Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling

Abstract—We study the problem of object recognition for categories for which we have no training examples, a task also called zero-data or zero-shot learning. This situation has hardly been studied in computer vision research, even though it occurs frequently; the world contains tens of thousands of different object classes, and image collections have been formed and suitably annotated for only a few of them. To tackle the problem, we introduce attribute-based classification: Objects are identified based on a high-level description that is phrased in terms of semantic attributes, such as the object's color or shape. Because the identification of each such property transcends the specific learning task at hand, the attribute classifiers can be prelearned independently, for example, from existing image data sets unrelated to the current task. Afterward, new classes can be detected based on their attribute representation, without the need for a new training phase. In this paper, we also introduce a new data set, Animals with Attributes, of over 30,000 images of 50 animal classes, annotated with 85 semantic attributes. Extensive experiments on this and two more data sets show that attribute-based classification indeed is able to categorize images without access to any training images of the target classes.

Index Terms—Object recognition, vision and scene understanding

1 Introduction

The field of object recognition in natural images has made tremendous progress over the last decade. For specific object classes, in particular *faces*, *pedestrians*, and *vehicles*, reliable and efficient detectors are available, based on the combination of powerful low-level features, such as SIFT [1] or HoG [2], with modern machine learning techniques, such as *support vector machines* (SVMs) [3], [4] or *boosting* [5]. However, to achieve good classification accuracy, these systems require a lot of manually labeled training data, typically several thousand example images for each class to be learned.

While building recognition systems this way is feasible for categories of large common or commercial interest, one cannot expect it to solve object recognition for all natural categories. It has been estimated that humans distinguish between approximately 30,000 basic object categories [6], and many more subordinate ones, such as different breeds of dogs or different car models [7]. It has even been argued that there are infinitely many potentially relevant categorization tasks because humans can create new categories on the fly, for example, *"things to bring to a camping trip"* [8]. Training conventional object detectors for all these would require millions or billions of well-labeled training images and is likely out of reach for many years, if it is possible at all. Therefore, numerous techniques for reducing the number of necessary training images have been developed, some of which we will discuss in Section 3. However, all of these techniques still require at least some labeled training examples to detect future object instances.

Human learning works differently: Although humans can, of course, learn and generalize well from examples, they are also capable of identifying completely new classes when provided with a high-level description. For example, from the phrase *"eight-sided red traffic sign with white writing,"* we will be able to detect *stop signs*, and when looking for *"large gray animals with long trunks,"* we will reliably identify *elephants*. In this work, which extends our original publication [9], we build on this observation and propose a system that is able to classify objects from a list of high-level semantically meaningful properties that we call *attributes*. The attributes serve as an intermediate layer in a classifier cascade and they enable the system to recognize object classes for which it had not seen a single training example.

Clearly, a large number of potential attributes exist and collecting separate training material to learn an ordinary classifier for each of them would be as tedious as doing so for all object classes. Therefore, one of our main contributions in this work is to show how, instead of creating a separate training set for each attribute, we can exploit the fact that meaningful high-level concepts transcend class boundaries. To learn such attributes, we can make use of existing training data by merging images of several object classes. To learn, for example, the attribute *striped*, we can use images of zebras, bees, and tigers. For the attribute *yellow*, zebras would not be included, but bees and tigers would still prove useful, possibly together with canary birds. It is this possibility to obtain knowledge about attributes from different object classes and, vice versa, the fact that each attribute can be used for the detection of many object classes that makes our proposed learning method statistically efficient.

- *C.H. Lampert is with the Institute of Science and Technology Austria, Am Campus 1, Klosterneuburg 3400, Austria. E-mail: chl@ist.ac.at.*
- *H. Nickisch is with Philips Research, Röntgenstrasse 24-26, 22335 Hamburg, Germany. E-mail: hannes@nickisch.org.*
- *S. Harmeling is with the Max Planck Institute for Intelligent Systems, 72076 Tübingen, Germany. E-mail: stefan.harmeling@tuebingen.mpg.de.*

Manuscript received 5 Sept. 2012; revised 15 Mar. 2013; accepted 12 July 2013; published online 29 July 2013.
Recommended for acceptance by D. Forsyth.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2012-09-0701.
Digital Object Identifier no. 10.1109/TPAMI.2013.140.

2 Information Transfer by Attribute Sharing

We begin by formalizing the problem and our intuition from the previous section that the use of attributes allows us to transfer information between object classes. We first define the exact situation of our interest:

Learning with disjoint training and test classes. *Let $\mathcal{X}$ be an arbitrary feature space and let $\mathcal{Y}=\{y_1,\dots,y_K\}$ and $\mathcal{Z}=\{z_1,\dots,z_L\}$ be sets of object categories, also called classes. The task of learning with disjoint training and test classes is to construct a classifier $f:\mathcal{X}\rightarrow\mathcal{Z}$ by making use of training examples $(x_1,l_1),\dots,(x_n,l_n)\subset\mathcal{X}\times\mathcal{Y}$ even if $\mathcal{Y}\cap\mathcal{Z}=\emptyset$.*[1]

Fig. 2a illustrates graphically why this task cannot be solved by ordinary multiclass classification: Standard classifiers learn one parameter vector (or other representation) α_k for each training class $y_1,\dots,y_K$. Because the classes $z_1,\dots,z_L$ are not present during the training step, no parameter vector can be derived for them, and it is impossible to make predictions about these classes for future samples.

To make predictions about classes for which no training data are available, one needs to introduce a coupling between the classes in $\mathcal{Y}$ and $\mathcal{Z}$. Since no training data for the unobserved classes are available, this coupling cannot be learned from samples, but it has to be inserted into the system by human effort. Preferably, the amount of human effort to specify new classes should be small because otherwise collecting and labeling training samples might be a simpler solution.

2.1 Attribute-Based Classification

We propose a solution for learning with disjoint training and test classes by introducing a small set of high-level semantic attributes that can be specified either on a per-class or on a per-image level. While we currently have no formal definition of what should count as an attribute, in the rest of the manuscript, we rely on the following characterization.

Attributes. *We call a property of an object an* attribute, *if a human has the ability to decide whether the property is present or not for a certain object.*[2]

Attributes are typically nameable properties, for example, the color of an object, or the presence or absence of a certain body part. Note that the definition allows properties that are not directly visible but related to visual information, such as an animal's natural habitat. Fig. 1 shows examples of classes and attributes.

An important distinction between attributes and arbitrary features is the aspect of *semantics*: Humans associate a meaning with a given attribute name. This allows them to create annotation directly in form of attribute values, which can then be used by the computer. Ordinary image features, on the other hand, are typically computable, but they lack the human interpretability.

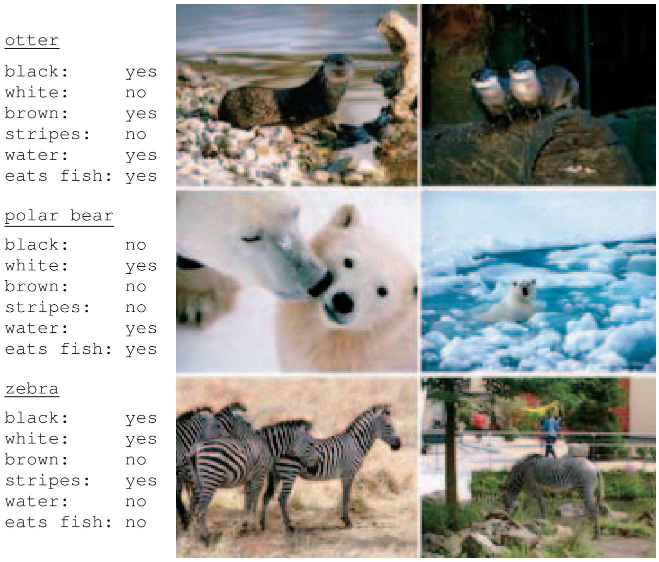


Fig. 1. Examples from the *Animals with Attributes*: object classes with per-class attribute annotation.

It is possible to assign attributes on a per-image basis, or on a per-class basis. The latter is particularly helpful, since it allows the creation of attribute annotation for a new classes with minimal effort. To make use of such attribute annotation, we propose *attribute-based classification*.

Attribute-based classification. *Assume the situation of* learning with disjoint training and test classes. *If for each class $z\in\mathcal{Z}$ and $y\in\mathcal{Y}$, an attribute representation $a^z,a^y\in\mathcal{A}$ is available, then we can learn a nontrivial classifier $\alpha:\mathcal{X}\rightarrow\mathcal{Z}$ by transferring information between $\mathcal{Y}$ and $\mathcal{Z}$ through $\mathcal{A}$.*

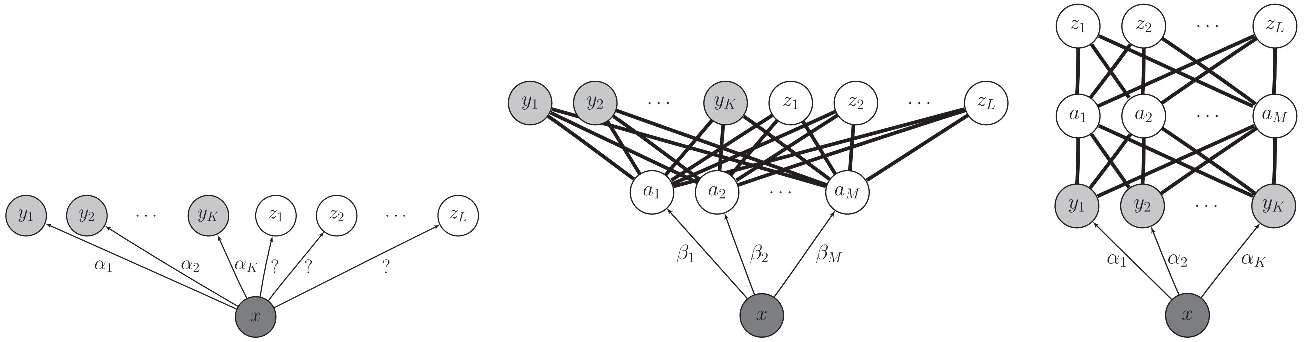
In the rest of this paper, we will demonstrate that *attribute-based classification* indeed offers a solution to the problem of *learning with disjoint training and test classes*, and how it can be practically used for object classification. For this, we introduce and compare two generic methods to integrate attributes into multiclass classification.

Direct attribute prediction (DAP), illustrated in Fig. 2b, uses an in-between layer of attribute variables to decouple the images from the layer of labels. During training, the output class label of each sample induces a deterministic labeling of the attribute layer. Consequently, any supervised learning method can be used to learn per-attribute parameters β_m. At test time, these allow the prediction of attribute values for each test sample, from which the test class labels are inferred. Note that the classes during testing can differ from the classes used for training, as long as the coupling attribute layer is determined in a way that does not require a training phase.

Indirect attribute prediction (IAP), depicted in Fig. 2c, also uses the attributes to transfer knowledge between classes, but the attributes form a connecting layer between two layers of labels: one for classes that are known at training time and one for classes that are not. The training phase of IAP consists of learning a classifier for each training class, as it would be the case in ordinary multiclass classification. At test time, the predictions for all training classes induce a labeling of the attribute layer, from which a labeling over the test classes is inferred.

[1] It is not necessary for $\mathcal{Y}$ and $\mathcal{Z}$ to be disjoint for the problems described to occur, $\mathcal{Z}\not\subseteq\mathcal{Y}$ is sufficient. However, for the sake of clarity, we only treat the case of disjoint class sets in this work.

[2] In this manuscript, we only consider *binary-valued* attributes. More general forms of attributes have already appeared in the literature; see Section 3.



(a) Flat multi-class classification (b) Direct attribute prediction (DAP) (c) Indirect attribute prediction (IAP)

Fig. 2. Graphical representation of the proposed across-class learning task: Dark gray nodes are always observed; light gray nodes are observed only during training. White nodes are never observed but must be inferred. An ordinary, flat, multiclass classifier (left) learns one parameter set α_k for each training class. It cannot generalize to classes $(z_l)_{l=1,\dots,L}$ that are not part of the training set. In an attribute-based classifier (middle) with fixed class-attribute relations (thick lines), training labels $(y_k)_{k=1,\dots,K}$ imply training values for the attributes $(a_m)_{m=1,\dots,M}$, from which parameters β_m are learned. At test time, attribute values can directly be inferred, and these imply output class label even for previously unseen classes. A multiclass-based attribute classifier (right) combines both ideas: Multiclass parameters α_k are learned for each training class. At test time, the posterior distribution of the training class labels induces a distribution over the labels of unseen classes by means of the class-attribute relationship.

The major difference between both approaches lies in the relationship between training classes and test classes. Directly learning the attributes results in a network where all classes are treated equally. When class labels are inferred at test time, the decision for all classes is based only on the attribute layer. We can expect it, therefore, to also handle the situation where training and test classes are not disjoint. In contrast, when predicting the attribute values indirectly, the training classes occur also at test time as an intermediate feature layer. On the one hand, this can introduce a bias, if training classes are also potential output classes during testing. On the other hand, one can argue that deriving the attribute layer from the label layer instead of from the samples will act as a regularization step that creates only *sensible* attribute combinations and, therefore, makes the system more robust. In the following, we will develop realizations for both methods and benchmark their performance.

2.2 A Probabilistic Realization

Both classification methods, DAP and IAP, are essentially metastrategies that can be realized by combining existing learning tools: a supervised classifier or regressor for the *image-attribute* or *image-class* prediction with a parameter-free inference method to channel the information through the *attribute* layer. In the following, we use a probabilistic model that reflects the graphical structures in Figs. 2b and 2c. For simplicity, we assume that all attributes have binary values such that the attribute representation $a = (a_1, \dots, a_M)$ for any class are fixed-length binary vectors. Continuous attributes can, in principle, be handled in the same way by using regression instead of classification. A generalization to relative attributes [10] or variable length descriptions should also be possible, but lies beyond the scope of this paper.

2.2.1 *Direct Attribute Prediction*

For DAP, we start by learning probabilistic classifiers for each attribute a_m. As training samples, we can use all images from all training classes, as labels, we use either per-image attribute annotations, if available, or we infer the labels from the entry of the attribute vector corresponding to the sample's label, i.e., all samples of class y have the binary label a^y_m. The trained classifiers provide us with estimates of $p(a_m \mid x)$, from which we form a model for the complete *image-attribute* layer as $p(a \mid x) = \prod_{m=1}^{M} p(a_m \mid x)$. At test time, we assume that every class z induces its attribute vector a^z in a deterministic way, i.e., $p(a \mid z) = [[a = a^z]]$, where we have made use of Iverson's bracket notation [11]: $[[P]] = 1$ if the condition P is true and it is 0 otherwise. By applying Bayes' rule, we obtain $p(z \mid a) = \frac{p(z)}{p(a^z)} [[a = a^z]]$ as the representation of the *attribute-class* layer. Combining both layers, we can calculate the posterior of a test class given an image:

$$p(z \mid x) = \sum_{a \in \{0,1\}^M} p(z \mid a) p(a \mid x) = \frac{p(z)}{p(a^z)} \prod_{m=1}^{M} p\left(a^z_m \mid x\right). \quad (1)$$

In the absence of more specific knowledge, we assume identical test class priors, which allows us to ignore the factor $p(z)$ in the following. For the factor $p(a)$, we assume a factorial distribution $p(a) = \prod_{m=1}^{M} p(a_m)$, using the empirical means $p(a_m) = \frac{1}{K} \sum_{k=1}^{K} a^{y_k}_m$ over the training classes as attribute priors.[3] As decision rule $f : \mathcal{X} \to \mathcal{Z}$ that assigns the best output class from all test classes $z_1, \dots, z_L$ to a test sample x, we then use MAP prediction:

$$f(x) = \operatorname*{argmax}_{l=1,\dots,L} p(z = l \mid x) = \operatorname*{argmax}_{l=1,\dots,L} \prod_{m=1}^{M} \frac{p\left(a^{z_l}_m \mid x\right)}{p\left(a^{z_l}_m\right)}. \quad (2)$$

2.2.2 *Indirect Attribute Prediction*

To realize IAP, we only modify the image-attribute stage: As a first step, we learn a probabilistic multiclass classifier estimating $p(y_k \mid x)$ for each training classes y_k, $k = 1, \dots, K$. As for DAP, we assume a deterministic dependence between

[3] In practice, the prior $p(a)$ is not crucial to the procedure and setting $p(a_m) = \frac{1}{2}$ yields comparable results.

attributes and classes, setting $p(a_m \mid y) = [[a_m = a_m^y]]$. The combination of both steps yields

$$p(a_m \mid x) = \sum_{k=1}^{K} p(a_m \mid y_k) p(y_k \mid x), \tag{3}$$

so in comparison to DAP, we only perform an additional matrix-vector multiplication after evaluating the classifiers. With the estimate of $p(a \mid x)$ obtained from (3) we proceed in the same way as in for DAP, i.e., we classify test samples using (2).

3 Related Work

Multilayer or *cascaded classifiers* have a long tradition in pattern recognition and computer vision: *multilayer perceptrons* [12], *decision trees* [13], *mixtures of experts* [14], and *boosting* [15] are prominent examples of classification systems built as feed-forward architectures with several stages. Multiclass classifiers are also often constructed as layers of binary decisions from which the final output is inferred, for example, [16], [17]. These methods differ in their training methodologies, but they share the goal of decomposing a difficult classification problem into a collection of simpler ones. However, their emphasis lies on the classification performance in a fully supervised scenario, so the methods are not capable of generalizing across class boundaries.

Especially in the area of computer vision, multilayered classification systems have been constructed, in which intermediate layers have interpretable properties: *Artificial neural networks* or *deep belief networks* have been shown to learn interpretable filters, but these are typically restricted to low-level properties, such as edge and corner detectors [18]. Popular local feature descriptors, such as SIFT [1] or HoG [2], can be seen as hand-crafted stages in a feed-forward architecture that transform an image from the pixel domain into a representation invariant to noninformative image variations. Similarly, image segmentation has been formulated as an unsupervised method to extract contours that are discriminative for object classes [19]. Such preprocessing steps are generic in the sense that they still allow the subsequent detection of arbitrary object classes. However, the basic elements, local image descriptors or segments shapes, alone are not reliable enough indicators of generic visual object classes, unless they are used as input to a subsequent statistical learning step.

On a higher level of abstraction, *pictorial structures* [20], the *constellation model* [21], and recent *discriminatively trained deformable part models* [22] are examples of the many methods that recognize objects in images by detecting *discriminative parts*. In principle, humans can give descriptions of object classes in terms of such *parts*, for example, *arms* or *wheels*. However, it is a difficult problem to build a system that learns to detect exactly the parts described. Instead, the above methods identify parts in an unsupervised way during training, which often reduces the parts to reproducible patterns of local feature points, not to units with a semantic meaning. In general, parts learned this way do not generalize across class boundaries.

3.1 Sharing Information between Classes

The aspect of sharing information between classes has attracted the attention of many researchers. A common idea is to construct multiclass classifiers in a cascaded way. By making similar classes share large parts of their decision paths, fewer classification functions need to be learned, thereby increasing the system's prediction speed [23]. Similarly, one can reduce the number of feature calculations by actively selecting low-level features that help discrimination for many classes simultaneously [24]. Combinations of both approaches are also possible [25].

In contrast, *interclass transfer* does not aim at higher speed, but at better generalization performance, typically for object classes with only few available training instances. From known object classes, one infers prior distributions over the expected intraclass variance in terms of distortions [26] or shapes and appearances [27]. Alternatively, features that are known to be discriminative for some classes can be reused and adapted to support the detection of new classes [28]. To our knowledge, no previous approach allows the direct incorporation of human prior knowledge. Also, all above methods require at least some training examples of the target classes and cannot handle completely new objects.

A notable exception is [29] that, like DAP and IAP, aims at classification with disjoint train and test set. It assumes that each class has a *description* vector, which can be used to transfer between classes. However, because these description vectors do not necessarily have a semantic meaning, they cannot be obtained from human prior knowledge. Instead, an additional data source is needed to create them, for example, data samples in a different representation.

3.2 Predicting Semantic Attributes

A second relevant line of related work is the *prediction of high-level semantic attributes* for images. Prior work in the area of computer vision has mainly studied elementary properties, such as colors and geometric patterns [30], [31], [32], achieving high accuracy by developing task-specific features and representations. In the field of multimedia retrieval, similar tasks occur. For example, the TRECVID contest [33] contains a task of *high-level feature extraction*, which consists of predicting *semantic concepts*, in particular scene types, for example, *outdoor*, *urban*, and high-level actions, for example, *sports*. It has been shown that by combining searches for several such attributes, one can build more powerful retrieval database mechanisms, for example, of faces [34], [35].

Instead of relying on manually defined attributes, it has recently been proposed to identify attributes automatically. Parikh and Grauman [36] introduced a semiautomatic technique for this that combines classifier outputs with human feedback. Sharmanska et al. [37] propose an unsupervised technique for augmenting existing attribute representations with additional nonsemantic binary features to make them more discriminative. It has also been shown that new attributes can be found by text mining [38], [39], [40], and that object classes themselves can act as attributes for other tasks [41]. Berg et al. [40] showed that instead of predicting only the presence or absence of an attribute, their occurrence can also be localized within the

TABLE 1
Animal Classes of the *Animals with Attributes* Data Set

skunk	polar bear	beaver	giraffe	*leopard*
lion	killer whale	bobcat	wolf	*pig*
fox	grizzly bear	collie	tiger	*hippopotamus*
ox	chihuahua	otter	cow	*seal*
mole	dalmatian	antelope	weasel	*persian cat*
sheep	spider monkey	hamster	mouse	*chimpanzee*
horse	blue whale	squirrel	buffalo	*rat*
bat	siamese cat	elephant	moose	*humpback whale*
zebra	rhinoceros	rabbit	walrus	*giant panda*
deer	german shepherd	dolphin	gorilla	*raccoon*

The 40 classes of the first four columns are used for training, the 10 classes of the last column (in italics) are the test classes.

image. Other alternative models for predicting attributes from images include conditional random fields [42], and probabilistic topic models [43]. Scheirer et al. [44] introduced an alternative technique for turning the output of attribute classifiers into probability estimates based on extremal value theory. The concept that attributes are properties of single images has also been generalized: Parikh and Grauman [10] introduced *relative attributes*, which encode a comparison between two images instead of specifying an absolute property, for example, *is larger than*, instead of *is large*.

3.3 Other Uses of Semantic Attributes

In parallel to our original work [9], Farhadi et al. [45] introduced the concept of predicting high-level semantic attributes of objects with the objective of being able to *describe* objects, even if their class membership is unknown.

Numerous follow-up papers have explored even more applications of attributes in computer vision tasks, for example, for scene classification [46], face verification [35], action recognition [47], and surveillance [48]. Rohrbach et al. [49] performed an in-depth analysis of attribute-based classification for transfer learning. Kulkarni et al. [50] used attribute predictions in combination with object detection and techniques from natural language processing to automatically create descriptions of images in natural language of images. Attributes have also been suggested as feedback mechanisms to improve image retrieval [51] and categorization [52].

3.4 Related Work Outside of Computer Science

In comparison to computer science, *cognitive science* research started much earlier to study the relations between object recognition and attributes. Typical questions in the field are how human judgements are influenced by characteristic object attributes [53], [54], and how the human performance in object detection tasks depends on the presence or absence of object properties and contextual cues [55]. Since one of our goals is to integrate human knowledge into a computer vision task, we would like to benefit from the prior work in this field, at least as a source of high-quality data that, so far, cannot be obtained by an automatic process. In the following section, we describe a data set of animal images that allows us to leverage established class-attribute association data from the cognitive science research community.

TABLE 2
Eighty-Five Semantic Attributes of the
Animals with Attributes Data Set in Short Form

black	toughskin	tail	bipedal	stalker	mountains
white	bulbous	horns	active	skimmer	water
blue	lean	claws	inactive	cave	newworld
brown	flippers	tusks	nocturnal	fierce	oldworld
gray	hands	smelly	hibernate	arctic	timid
orange	hooves	flies	agility	coastal	smart
red	longleg	hops	fish	desert	group
yellow	pads	swims	meat	bush	solitary
patches	paws	tunnels	plankton	plains	nestspot
spots	longneck	walks	vegetation	forest	domestic
stripes	chewteeth	fast	insects	fields	
furry	meatteeth	slow	forager	jungle	
hairless	buckteeth	strong	grazer	tree	
big	strainteeth	weak	hunter	ocean	
small	quadrapedal	muscle	scavenger	ground	

Longer forms given to human subject for annotation were complete phrases, such as has flippers, eats plankton, *or* lives in water.

4 The Animals with Attributes (AwA) Data Set

In the early 1990s, Osherson et al. [56] collected judgements from human subjects on the *"relative strength of association"* between 85 semantic attributes and 48 mammals. Kemp et al. [57] later added two more classes and their attributes for a total of 50×85 class-attribute associations.[4] The full list of classes and attributes can be found in Tables 1 and 2. Besides the original continuous-valued matrix, also a binary version was created by thresholding the original matrix at its overall mean value; see Fig. 3 for excerpts from both matrices. Note that because of the data collection process, the data are not completely error free. For example, contrary to what is specified in the binary matrix, panda bears do not have buck teeth, and walruses do have tails.

Our goal in creating the *Animals with Attributes* data set[5] was to make this attribute-matrix accessible for computer vision experiments. We collected images by querying the image search engines of *Google*, *Microsoft*, *Yahoo*, and *Flickr* for each of the 50 animals classes. We manually removed outliers and duplicates as well as images in which the target animals were not in prominent enough view to be recognizable. The remaining image set has 30,475 images, where the minimum number of images for any class is 92 (mole) and the maximum is 1,168 (collie). Fig. 1 shows exemplary images and their attribute annotation.

To facilitate use by researchers from outside of computer vision, and to increase the reproducibility of results, we provide precomputed feature vectors for all images of the data set. The representations were chosen to reflect different aspects of the images (color, texture, shape), and to allow easy use with off-the-shelf classifiers: HSV color histograms, SIFT [1], rgSIFT [58], PHOG [59], SURF [60], and local self-similarity histograms [61]. The color histograms and PHOG feature vectors are extracted separately for all 21 cells of a three-level spatial pyramids (1×1, 2×2, 4×4). For each cell, 128-dimensional color histograms are extracted and concatenated to form a 2,688-dimensional feature vector. For PHOG, the same pyramid is used, but with 12-dimensional

[4] http://www.psy.cmu.edu/~ckemp/code/irm.html.
[5] http://www.ist.ac.at/~chl/AwA/.

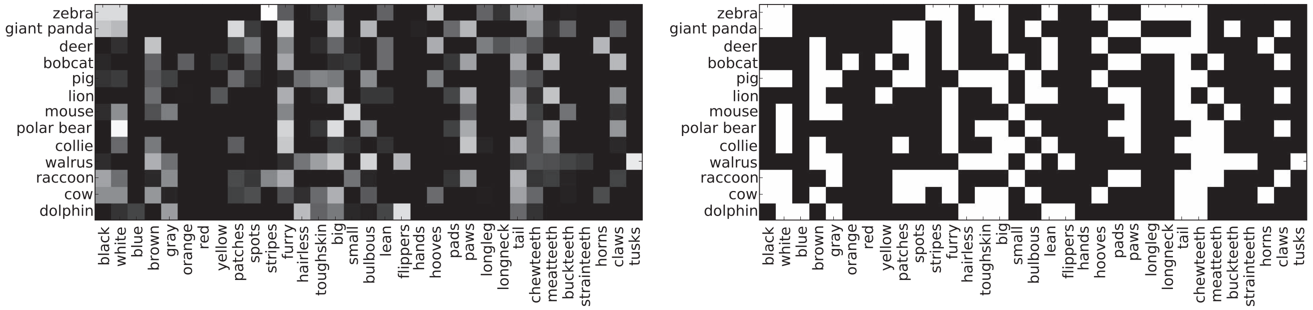


Fig. 3. Real-valued (left) and binary-valued (right) *class-attribute* matrices of the *Animals with Attributes* data set. Shown are 13×33 excerpts of the complete 50×85 matrices.

base histograms in each cell. The other feature vectors each are *bag-of-visual-words* histograms obtained from quantizing the original descriptors with 2,000-element codebooks that were obtained by k-means clustering on 250,000 element subsets of the descriptors.

We define a fixed split of the data set into 40 classes (24,295 images) to be used for training, and 10 classes (6,180 images) to be used for testing; see Table 1. This split was not done randomly, but much of the diversity of the animals in the data set (water/land-based, wild/domestic, etc.) is reflected in the training as well as in the test set of classes. The assignments were based only on the class names and before any experiments were performed, so in particular, the split was not designed for best zero-shot classification performance. Random train-test splits of similar characteristics can be created by fivefold cross validation (CV) over the classes.

5 Other Data Sets for Attribute-Based Classification

Besides the Animals with Attributes data set, we also perform experiments on two other data sets of natural images for which attribute annotations have been released. We briefly summarize their characteristics here. An overview is also provided in Table 3.

5.1 aPascal-aYahoo

The aPascal-aYahoo data set[6] was introduced by Farhadi et al. [45]. It consists of a 12,695-image subset of the PASCAL VOC 2008 data set[7] and 2,644 images that were collected using the Yahoo image search engine. The PASCAL part serves as training data, and the Yahoo part as test data. Both sets have disjoint classes (20 classes for PASCAL, 12 for Yahoo), so learning with disjoint training and test classes is unavoidable. Attribute annotation is available on the image level: Each image has been annotated with 64 binary attribute that characterize shape, material, and the presence of important parts of the visible object. As image representation, we rely on the precomputed color, texture, edge orientation, and HoG features that the authors of [45] extracted from the objects' bounding boxes (as provided by the PASCAL VOC annotation) and released as part of the data set.

5.2 SUN Attributes

The *SUN Attributes*[8] data set was introduced by Patterson and Hays [46]. It is a subset of the *SUN Database* [62] for fine-grained scene categorization and consists of 14,340 images from 717 classes (20 images per class). Each image is annotated with 102 binary attributes that describe the scenes' material and surface properties as well as lighting conditions, functions, affordances, and general image layout. For our experiments, we rely on the feature vectors that are provided by the authors of [46] as part of the data set. These consists of GIST, HOG, self-similarity, and geometric color histograms.

6 Experimental Evaluation

In this section, we perform an experimental evaluation of the DAP and the IAP model on the Animals with Attributes data set as well as the other data sets described above.

Since our goal is the categorization of classes for which no training samples are available, we always use training and test set with disjoint class structure.

For DAP, we train one nonlinear support vector machine for each binary attributes, $a_1, \dots, a_M$. In each case, we use 90 percent of the images of the training classes for training, with binary labels for the attribute, which are either obtained from the class-attribute matrix by assigning each image the attribute value of its class, or by per-image attribute annotation, where available. We use the remaining 10 percent of training images to estimate the parameters of a sigmoid curve for Platt scaling, to convert the SVM outputs into probability estimates [63].

At test time, we apply the trained SVMs with Platt scaling to each test image and make test class predictions using (2).

For IAP, we train one-versus-rest SVMs for each training class, again using a 90/10 percent split for training of the decision functions, and of the sigmoid coefficients for Platt scaling. At test time, we predict a vector of class probabilities for each test image. We L^1-normalize this vector such that we can interpret it as a posterior distribution over the training classes. We then use (3) to predict attribute values, from which we obtain test class predictions by (2) as above.

[6] http://vision.cs.uiuc.edu/attributes/.

[7] http://www.pascal-network.org/challenges/VOC/.

[8] http://cs.brown.edu/~gen/sunattributes.html.

TABLE 3
Characteristics of Data Sets with Attribute Annotation: Animals with Attributes [9], aPascal/aYahoo (aP/aY) [45], SUN Attributes (SUN) [46]

Dataset	AwA	aP/aY	SUN
# Images	30475	15339	14340
# Classes	50	32	717
# Attributes	85	64	102
Annotation Level	per class	per image	per image
Annotation Type (real-valued or binary)	both	binary	binary

6.1 SVM Kernels and Model Selection

To achieve optimal performance of the SVM classifiers, we use established kernel functions and perform thorough model selection. All SVMs are trained with linearly combined χ^2-kernels: For any D-dimensional feature vectors, $h(x) \in \mathbb{R}^D$ and $h(\bar{x}) \in \mathbb{R}^D$, of images x and $\bar{x}$, we set $k(x,\bar{x}) = \exp(-\gamma\chi^2(h(x),h(\bar{x})))$ with $\chi^2(h,\bar{h}) = \sum_{i=1}^{D}\frac{(h_i-\bar{h}_i)^2}{h_i+\bar{h}_i}$. For DAP, the bandwidth parameter γ is selected in the following way: For each attribute, we perform fivefold cross validation, computing the receiver operating characteristic (ROC) curve of each predictor and averaging the areas under the curves (AUCs) over the attributes. The result is a single *mean attrAUC* score for any value of the bandwidth. We perform this estimation for $\bar{\gamma} = c\gamma \in \{0.01, 0.03, 0.1, 0.3, 1, 3, 10\}$, where $c = \frac{1}{n^2}\sum_{i,j=1}^{n}\chi^2(h(x_i),h(x_j))$, i.e., we parameterize γ relative to the average χ^2-distance of all points in the training set. $\bar{\gamma} = 3$ was consistently found as best value.

Given L different feature functions, $h_1,\dots,h_K$, we obtain L kernel functions $k_1,\dots,k_L$, and we use their unnormalized sum as the final SVM kernel, $k(x,\bar{x}) = \sum_{l=1}^{L} k_l(x,\bar{x})$. Once we fixed the kernel, we identify the SVMs C parameter among the values $\{0.01, 0.03, 0.1, \dots, 30, 100, 3{,}000, 1{,}000\}$ in an analogous procedure. We perform fivefold cross validation for each attribute, and we pick C that achieves the highest mean attrAUC. Note that we use the same C values for all attribute classifiers. Technically, this would not be necessary, but we prefer it to avoid large scaling differences between the SVM outputs of different attribute predictors. Also, one can expect the optimal C values to not vary strongly between different attributes, because all classifiers use the same kernel matrix and differ only in their label annotation.

For IAP, we use the same kernel as for DAP and determine C using fivefold cross validation similar to the procedure one described above, except that we use the mean area under the ROC curve of class predictions (*mean classAUC*) as selection criterion.

6.2 Results

We use the above-described procedures to train DAP and IAP models for all data sets. For DAP, where applicable, we use both per-image or per-class annotation to find out whether the time-consuming per-image annotation is necessary. For the data set with per-image attribute annotation, we create class-attribute matrices by averaging all attribute vectors of each class and thresholding the resulting real-valued matrix at its global mean value. Besides experiments with fixed train/test splits of classes, we also perform experiments with random class split using fivefold cross validation for Animals with Attributes (i.e., 40 training classes, 10 test classes), and 10-fold cross validation for SUN Attributes (approximately 637 ± 1 classes for training and 70 ± 1 classes for testing). We measure the quality of the prediction steps in terms of normalized multiclass accuracy (MC acc.) on the test set (the mean of the diagonal of the confusion matrix). We also report areas under the ROC curve for each test class z and attribute a, when their posterior probabilities $p(z \mid x)$ and $p(a \mid x)$, respectively, are treated as ranking measures over all test images.

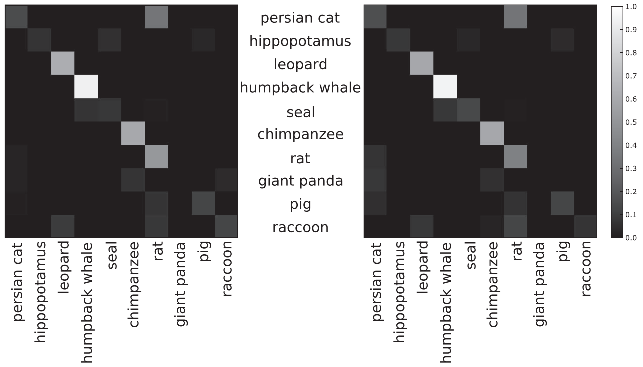


Fig. 4. Confusion matrices between 10 test classes of the *Animals with Attributes* data set. Left: Indirect attribute prediction. Right: Direct attributes prediction.

In the following, we show detailed results for *Animals with Attributes* and summaries of the results for the other data sets.

6.2.1 Results—Animals with Attributes

The *Animals with Attributes* data set comes only with per-class annotation, so there are two models to compare: per-class DAP and per-class IAP. Fig. 4 shows the resulting confusion matrices for both methods. The class-normalized multiclass accuracy can be read off from the mean value of the diagonal as 41.4 percent for DAP and 42.2 percent for IAP. While the results are not as high as a supervised method could achieve, it nevertheless clearly proves our original claim about attribute-based classification: *By sharing information via an attribute layer, it is possible to classify images of classes for which we had no training examples.* As a baseline, we compare against a zero-shot classifier, where for each test class, we identify the most similar training class and predict using a classifier for it trained on all training data. We use two different methods to define the similarity between the classes' attribute representations: Hamming distance or cross correlation. As it turns out, both variants make almost identical decisions, resulting in multiclass accuracies of 30.7 and 30.8 percent. This is clearly better than chance performance, but below the results of DAP and IAP.

Using random class splits instead of the predefined one, we obtain slightly lower multiclass accuracies of 34.8/44.8/34.7/35.1/36.3 percent (average 37.1 percent) for DAP, and 33.4/42.8/27.3/31.9/35.3 percent (average 34.1 percent) for IAP. Again, the baselines achieve clearly lower results: 32.4/31.9/28.1/25.3/20.9 percent (average 27.7 percent) for

TABLE 4
Numeric Results on the *Animals with Attributes* Data Set in Percent: Multiclass Accuracy for DAP, IAP, Class-Transfer Classifier Using Cross Correlation (CT-cc) or Hamming Distance (CT-H) of Class Attributes, and Chance Performance (rnd)

(a) default train/test class split

method	DAP	IAP	CT-cc	CT-H	rnd
MC acc.	41.4	42.2	30.7±0.2	30.8±0.2	10.0
classAUC	81.4	80.0	73.4	73.4	50.0
attrAUC	72.8	72.1	–	–	50.0

(b) five random splits (mean±std.dev.)

method	DAP	IAP	CT-cc	CT-H	rnd
MC acc.	37.1±3.9	34.1±5.1	27.7±4.3	27.3±4.0	10.0
classAUC	80.4±3.1	76.3±5.5	72.4±2.7	72.8±3.1	50.0
attrAUC	70.7±3.5	69.7±3.8	–	–	50.0

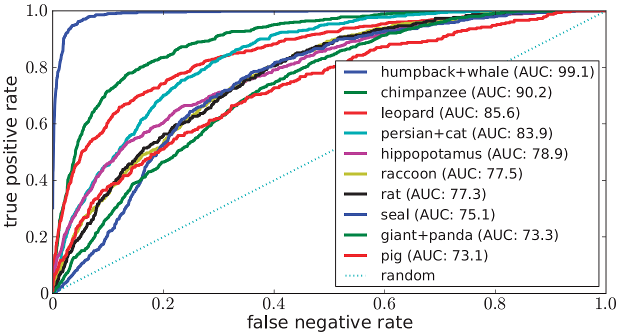


Fig. 5. Retrieval performance of attribute-based classification (DAP method): ROC curves and area under curve for the 10 *Animals with Attributes* test classes.

the cross-correlation version, and 33.0/29.0/28.4/25.3/20.9 percent (average 27.3 percent) for the version based on Hamming distance.

The quantitative results for all method are summarized in Table 4. One can see that the differences between the two approaches, DAP and IAP, are relatively small. One might see a slight overall advantage for DAP, but as the large variance between class splits is rather high, this could also be explained by random fluctuations. To avoid redundancy, we give detailed results only for the DAP model in the rest of this section.

Another measure of prediction performance besides multiclass accuracy is how well the predicted posterior probability of any of the test classes can be used to retrieve images of this class from the set of all test images. We evaluate this by plotting the corresponding *ROC curves* in Fig. 5 and report their AUC. One can see that for all classes, reasonable classifiers have been learned with AUCs clearly higher than the chance level 0.5. With an AUC of 0.99, the performance for *humpback whale* is even on par of what we can expect to achieve with fully supervised learning techniques. Fig. 6 shows the five images with highest posterior score for each test class, therefore allowing to judge the quality of a hypothetical image retrieval system based on (1). One can see that the rankings for *humpback whales, leopards,* and *hippopotamuses* are very reliable. Confusions that occur are typically between animals classes with similar characteristics, such as a whale mistaken for a seal, or a racoon mistaken for a rat.

Because all classifiers base their decisions on the same learned attribute classifiers, one can presume that the easier classes are characterized either by more distinctive attribute vectors or by attributes that are easier to learn from visual data. We believe that the first explanation is not correct, since the matrix of pairwise distances between attribute vectors does not resemble the confusion matrices in Table 4.

We, therefore, analyze the quality of the individual attribute predictors in more detail. Fig. 7 summarizes their quality in terms of the area under the ROC curve

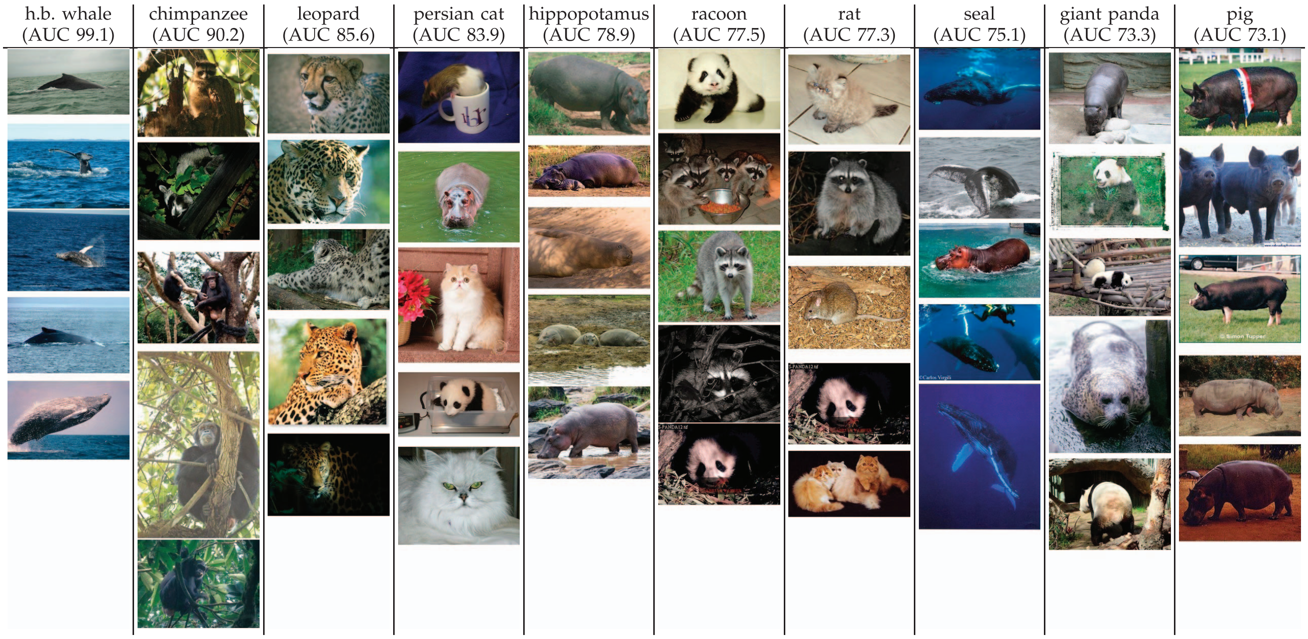


Fig. 6. Highest ranking results for each test class in the *Animals with Attributes* data set. Classes with unique characteristics are identified well, for example, humpback whales and leopards. Confusions occur between visually similar categories, for example, pigs and hippopotamuses.

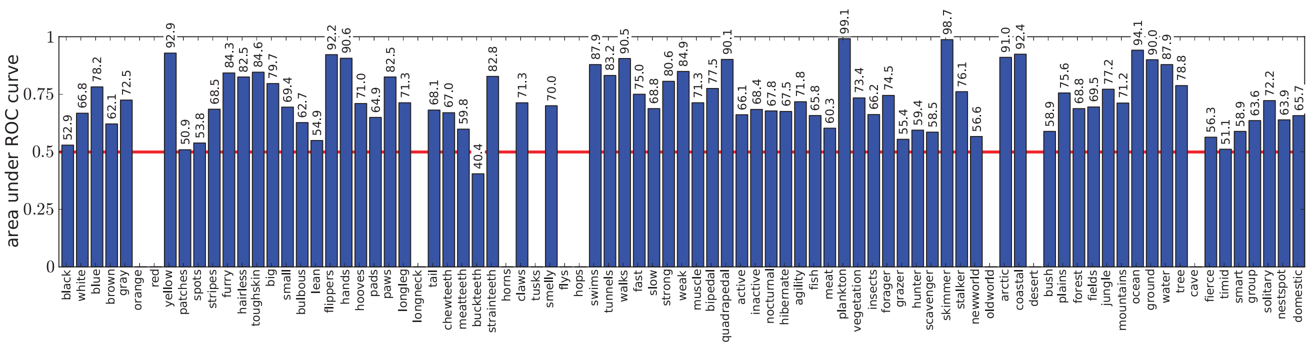


Fig. 7. Quality of individual attribute predictors (trained on *train* classes, tested on *test* classes), as measured by the area under the ROC curve. Attributes without entries have constant values for all test classes, so their ROC curve cannot be computed.

(attrAUC). Missing entries indicate that all images in the test set coincided in their value for this attribute, so no ROC curve can be computed. Fig. 8 shows, for a selection of attributes, the five images of highest posterior score within the test set.

On average, attributes can be predicted clearly better than random (the average AUC is 72.4 percent, whereas random prediction would have 50 percent). However, the variance within the predictions is large, ranging from near perfect prediction, for example, for *is yellow* and *eats plankton*, to essentially random performance, for example, on *has buckteeth* or *is timid*. Contrary to what one might expect, attributes that refer to visual properties are not automatically predicted more accurately than others. For example, *is blue* is identified reliably, but *is brown* is not. Overall good performance is also achieved on several attributes that describe body parts, such as *has paws*, or the natural habitat *lives in trees*, and even on nonvisual properties like, such as, *is smelly*. There are two explanations for this effect: On the one hand, attributes that are clearly visual, such as colors, can still be hard to predict from a global image representation because they typically reflect information that is localized within only the object region. On the other hand, nonvisual attributes can often still be predicted from image information because they occur correlated with visual properties, for example, characteristic texture. It is known that the integration of such contextual information can improve the accuracy of visual classifiers, for example, road regions helps the detection of cars. However, it remains to be seen if this effect will be sufficient for purely nonvisual attributes, or whether it would be better in the long run to replace nonvisual attributes by the visual counterparts they are correlated with.

Another interesting observation is that the system learned to correctly predict attributes such as *is big* and *is small*, which are ultimately defined only by context. While this is desirable in our setup, where the context is consistent, it also suggests that the learned attribute predictors themselves are context dependent and cannot be expected to generalize to object classes very different from the training classes.

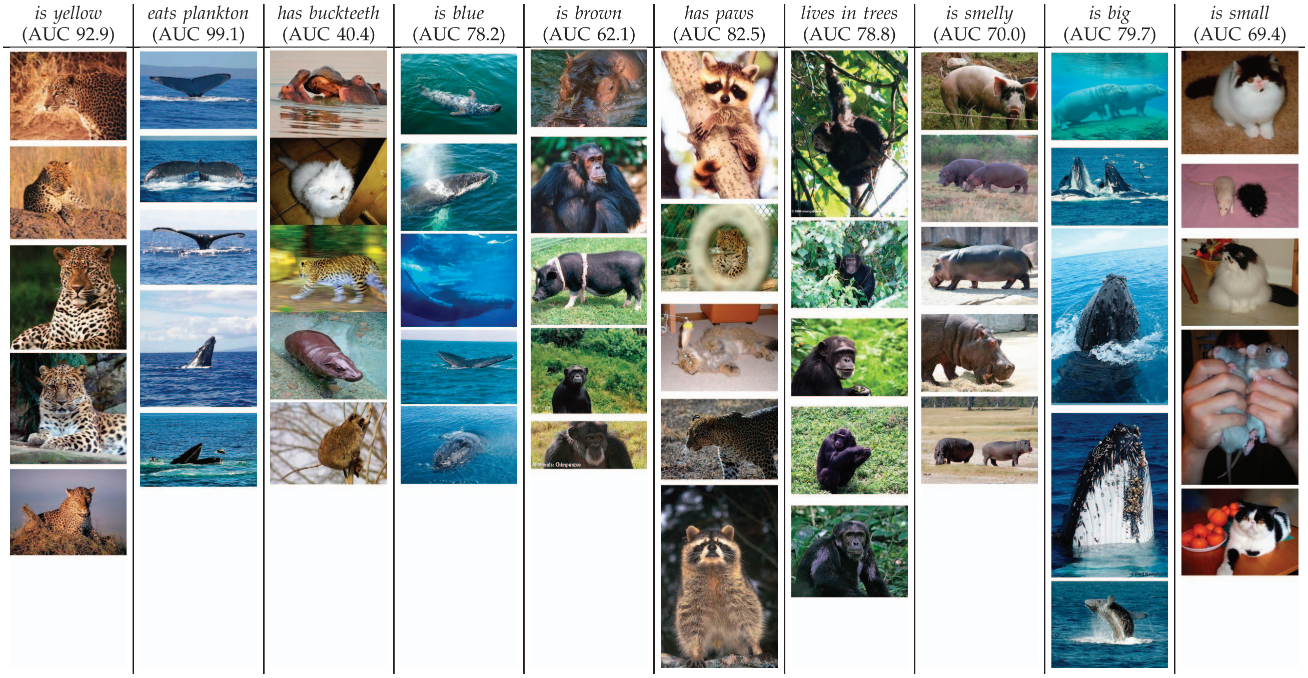

is yellow (AUC 92.9)
eats plankton (AUC 99.1)
has buckteeth (AUC 40.4)
is blue (AUC 78.2)
is brown (AUC 62.1)
has paws (AUC 82.5)
lives in trees (AUC 78.8)
is smelly (AUC 70.0)
is big (AUC 79.7)
is small (AUC 69.4)

Fig. 8. Highest ranking results for a selection of attribute predictors (see Section 6.2) learned by DAP on the *Animals with Attributes* data set.

TABLE 5
Numeric Results on the aPascal/aYahoo and the SUN Attributes Data Sets in Percent: DAP with Per-Image Annotation (DAP-I), DAP with Per-Class Annotation (DAP-C), IAP, Class-Transfer Classifier (CT-H), and Chance Performance (rnd)

(a) *aPascal-aYahoo*, default train/test split

method	DAP-I	DAP-C	IAP	CT-H	rnd
MC acc.	16.8	19.1	16.9	16.7±0.5	8.3
classAUC	76.9	76.5	75.4	64.2	50.0
attrAUC	70.6	73.7	73.1	–	50.0

(b) *SUN Attributes*, ten splits (mean±std.dev.)

method	DAP-I	DAP-C	IAP	CT-H	rnd
MC acc.	18.1±1.2	22.2±1.6	18.0±1.5	12.9±1.3	1.4
level2 acc.	40.2±2.1	46.6±1.7	41.1±2.1	32.6±2.0	6.2
level1 acc.	74.2±4.0	85.7±2.1	82.1±2.5	74.2±2.0	33.3
class mAUC	90.5±0.7	92.3±0.7	87.9±0.7	77.1±0.0	50.0
attrAUC	82.0±0.6	83.9±0.8	82.7±0.8	–	50.0

TABLE 6
Mean Mutual Information between Individual Attributes and Class Labels with Per-Class or Per-Image Annotation

Dataset	per class	per image
Animals with Attributes	0.736	—
aPascal-aYahoo	0.532	0.245
SUN Attributes	0.671	0.211

6.2.2 Results—Other Data Sets

We performed the same evaluation as for the *Animals with Attributes* data set also for the other data sets. Since these data sets have per-image attribute annotation in addition to per-class attribute annotation, we obtain results for two variants of DAP: trained with per-image labels and trained with per-class labels. In both cases, test time inference is done with per-class labels, since we still assume that no examples of the test classes are available. As additional baseline, we use the class transferred classifier as in Section 6.2.1. Since both variants perform almost identically, we report only results for the one based on Hamming distance. The results are summarized in Tables 5a and 5b.

For the SUN data set, we measure the classification performance based on the three-level SUN hierarchy suggested in [62]. At test time, the ground truth label and the predicted class label each corresponds to one path in the hierarchy (or multiple paths, since the hierarchy is not a tree, but a directed acyclic graph). A prediction is considered correct at a certain level, if both paths run through a common node in that level. At the third level, each class is a separate leaf, so level-3 accuracy is identical to the unnormalized multiclass accuracy, which coincides with the diagonal of the confusion matrix in this case, since all classes have the same number of images. However, at levels 1 and 2, semantically similar classes are mapped to the same node, and confusions between these classes are, therefore, disregarded. Note that the values obtained for the SUN data set are not directly comparable to earlier supervised work using these data. Because we split the data into disjoint train (90 percent) and test classes (10 percent), fewer classes of the data set are present at test time.

The results for both data sets confirm our observations from the *Animals with Attributes* data set. DAP (both variants) as well as IAP achieves far better than random performance in terms of multiclass accuracy, mean per-class AUC, and mean per-attribute AUC, and also better than the Hamming distance-based baseline classifier.

A more surprising observation is that per-image attribute annotation, as it is available for the aPascal and SUN Attributes data sets, does not improve the prediction accuracy compared to the per-class annotation, which is much easier to create. We currently do not have a definite explanation for this. However, two additional observations suggest that the reason might be a *bias-variance* effect: First, per-image attribute annotation does not follow class boundaries, so its mutual information of the ground truth attribute annotation with the class labels is lower than for per-class annotation (see Table 6). Second, the visual learning tasks defined by per-image annotation do not seem easier learnable than the per-class counterparts, as indicated by the reduced mean attribute AUC in Tables 4, 5a, and 5b. Likely this is because per-class annotation is correlated with many other visual properties in the images and therefore often easy to predict, whereas per-image annotation singles out the actual attribute in question.

In combination, we expect the per-image annotation to lead to less *bias* in the training problem, therefore having the potential for better attribute classifiers given enough data. However, because of the harder learning problem, the resulting classifiers have higher variance when trained on a fixed amount of data. We take the results as a sign that the second effect is currently the dominant source of errors. We plan to explore this hypothesis in future work by studying the learning curves of attribute learning with per-class and per-image annotation for varying amounts of training data.

There is also a second, more empirical, explanation: Per-class training of attribute classifiers resembles recent work on discriminatively learned image representations, such as *classemes* [64]. These have been found to work well for image categorization tasks, even for categories that are not part of the classemes set. A similar effect might hold for per-class trained attribute representations: Even if their interpretation as semantic image properties is not as straightforward as for classifiers trained with per-image annotation, they might simply lead to a good image representation.

6.2.3 Comparison to the Supervised Setup

Besides the relative comparison of the different methods to each other, we also try to highlight how DAP and IAP perform on an absolute scale. We, therefore, compare our method to ordinary multiclass classification with a small number of training examples. For each test class, we randomly pick a fixed number of training examples and use them to train a one-versus-rest multiclass SVM, which we evaluate using the remaining images of the test classes. The kernel function and parameters are the same as for the IAP model. Fig. 7 summarizes the results in form of the mean over 10 such splits. For an easier comparison, we also repeat the range of values that zero-shot with DAP or IAP achieved (see Tables 4, 5a, and 5b).

By comparing the last column to the others, one sees that on the Animals with Attributes data set, attribute-based classification achieves results on par with supervised

TABLE 7
Numeric Results of One-versus-Rest Multiclass SVMs Trained with $n \in \{1, 2, 3, 4, 5, 10, 15, 20\}$ Training Examples from Each Test Class in Comparison to the Results Achieved by Zero-Shot Learning with DAP and IAP (in Percent)

(a) mean class accuracy

	$n = 1$	2	3	4	5	10	15	20	*zero-shot*
AwA (def.)	23.6	26.2	29.9	32.7	34.0	39.9	42.9	45.4	*41.4–42.2*
AwA (5-CV)	20.1	23.4	26.3	27.7	29.7	35.7	39.8	40.6	*34.1–37.1*
aP/aY	22.6	29.6	33.9	38.1	40.0	48.5	50.7	57.1	*16.9–19.1*
SUN	13.1	18.6	22.7	25.8	28.5	36.8	–	–	*18.0–22.2*

(b) mean classAUC

	$n = 1$	2	3	4	5	10	15	20	*zero-shot*
AwA (def.)	67.4	72.4	74.0	75.5	76.6	81.2	82.6	84.1	*80.0–81.4*
AwA (5-CV)	63.8	67.0	69.9	71.3	72.7	77.6	80.5	82.1	*76.3–80.4*
aP/aY	68.7	74.2	76.2	79.3	80.4	85.3	86.0	89.2	*75.4–76.9*
SUN	76.9	82.2	84.9	86.9	88.1	91.3	–	–	*87.9–92.3*

training with 10-15 training examples per test class, i.e., 100-150 training images in total. On the aPascal, the attribute representations perform worse. Their results are comparable to supervised training with at most one example per class, if judged by multiclass accuracy, and two to three examples per class, if judged by mean classAUC. On the SUN data set, approximately two examples per class (142 total) are necessary for equal mean class accuracy, and 5-10 examples per class (355 to 710 total) for equal mean AUC. Note, however, that all the above comparisons may overestimate the power of the supervised classifiers: In a realistic setup with so few training examples, model selection is problematic, whereas to create Table 7, we just reused the parameters obtained by thorough model selection for the IAP model.

Interpreting the low performance on the aPascal-aYahoo data set, one has to take the background of this data set into account. Its attributes were selected to provide additional information about object classes, not to discriminate between them. While the resulting attribute set is comparably difficult to learn (see Table 5(a)), each attribute on average contains less information about the class labels (see Table 6), mainly because several of the attributes are meaningful only for a small subset of the categories. We conclude from this that attributes that are useful to describe objects from different categories are not automatically also useful to distinguish between the categories, a fact that should be taken into account in the future creation of attribute annotation for image data sets.

Overall, we do not think that the experiments we presented are sufficient to make a definite statement about the quality of attribute-based versus supervised classification. However, we believe that the results confirm the intuition that a larger ratio of attributes to classes improves the prediction performance. However, not only the number of attributes matters, but also how informative the chosen attributes are about the classes.

7 Conclusion

In this paper, we introduced *learning with disjoint training and test classes*. It formalizes the problem of learning an object classification systems for classes for which no training images are available. We proposed two methods for *attribute-based classification* that solve this problem by transferring information between classes. In both cases, the transfer is achieved by an intermediate representation that consists of high-level semantic attributes that provide a fast and simple way to include human knowledge into the system. To predict the attribute level, we either rely on classifiers trained directly on attribute annotation (DAP), or we infer the attribute layer from classifiers trained to identify other classes (indirect attribute prediction). Once trained, the system can detect new object categories, if a suitable characterization in terms of attributes is available for them, and it does not require retraining.

As a second contribution, we introduced the *Animals with Attributes* data set: It consists of over 30,000 images with precomputed reference features for 50 animal classes, for which a semantic attribute annotation is available that has been used in earlier cognitive science work. We hope that this data set will foster research and serve as a testbed for attribute-based classification.

7.1 Open Questions and Future Work

Despite the promising results of the proposed system, several questions remain open and require future work. For example, the assumption of disjoint training and test classes is clearly artificial. It has been observed, for example, in [65], that existing methods, including DAP and IAP, do not work well if this assumption is violated, since their decisions become biased toward the previously seen classes. In the supervised scenario, methods to overcome this limitation have been suggested, for example, [66], [67], but a unified framework that includes the possibility of zero-shot learning is still missing.

A related open problem is how zero-shot learning can be unified with supervised learning when a small number of labeled training examples are available. While some work in this direction exists (see our discussion in Section 3), we believe that it will also be able to extend DAP and IAP for this for purpose. For example, one could make use of their probabilistic formulation to define an attribute-based prior that is combined with a likelihood term derived from the training examples.

Beyond the specific task of multiclass classification, there are many other open questions that will need to be tackled if we want to make true progress in solving the grand tasks of computer vision: How do we handle the problem that many object categories are rare? How can we build object recognition systems that adapt and incorporate new categories that they encounter? How can we integrate human knowledge about the visual world besides specifying training examples? We believe that attribute-based classification will be able to help in answering at least some of these questions.

Acknowledgments

This work was in part funded by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013)/ERC grant agreement no 308036. The authors would like to thank Charles Kemp for providing the Osherson/Wilkie *class-attribute* matrix and Jens Weidmann for his help on creating the *Animals with Attributes* data set.

REFERENCES

[1] D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int'l J. Computer Vision,* vol. 60, no. 2, pp. 91-110, 2004.

[2] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2005.

[3] B. Schölkopf and A.J. Smola, *Learning with Kernels.* MIT Press, 2002.

[4] C.H. Lampert, "Kernel Methods in Computer Vision," *Foundations and Trends in Computer Graphics and Vision,* vol. 4, no. 3, pp. 193-285, 2009.

[5] R.E. Schapire and Y. Freund, *Boosting: Foundations and Algorithms.* MIT Press, 2012.

[6] I. Biederman, "Recognition by Components: A Theory of Human Image Understanding," *Psychological Rev.,* vol. 94, no. 2, pp. 115-147, 1987.

[7] B. Yao, A. Khosla, and L. Fei-Fei, "Combining Randomization and Discrimination for Fine-Grained Image Categorization," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2011.

[8] G.L. Murphy, *The Big Book of Concepts.* MIT Press, 2004.

[9] C.H. Lampert, H. Nickisch, and S. Harmeling, "Learning to Detect Unseen Object Classes by Between-Class Attribute Transfer," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2009.

[10] D. Parikh and K. Grauman, "Relative Attributes," *Proc. IEEE Int'l Conf. Computer Vision (ICCV),* 2011.

[11] D.E. Knuth, "Two Notes on Notation," *Am. Math. Monthly,* vol. 99, no. 5, pp. 403-422, 1992.

[12] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, "Learning Internal Representations by Error Propagation," *Parallel Distributed Processing,* MIT Press, 1986.

[13] L. Breiman, J.J. Friedman, R.A. Olshen, and C.J. Stone, *Classification and Regression Trees.* Wadsworth, 1984.

[14] M.I. Jordan and R.A. Jacobs, "Hierarchical Mixtures of Experts and the EM Algorithm," *Neural Computation,* vol. 6, no. 2, pp. 181-214, 1994.

[15] Y. Freund and R.E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *J. Computer and System Sciences,* vol. 55, no. 1, pp. 119-139, 1997.

[16] T.G. Dietterich and G. Bakiri, "Solving Multiclass Learning Problems via Error-Correcting Output Codes," *J. Artificial Intelligence Research,* vol. 2, pp. 263-286, 1995.

[17] R. Rifkin and A. Klautau, "In Defense of One-vs-All Classification," *J. Machine Learning Research,* vol. 5, pp. 101-141, 2004.

[18] M. Ranzato, F.J. Huang, Y.-L. Boureau, and Y. LeCun, "Unsupervised Learning of Invariant Feature Hierarchies with Applications to Object Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2007.

[19] J. Winn and N. Jojic, "LOCUS: Learning Object Classes with Unsupervised Segmentation," *Proc. IEEE Int'l Conf. Computer Vision (ICCV),* vol. 1, 2005.

[20] M.A. Fischler and R.A. Elschlager, "The Representation and Matching of Pictorial Structures," *IEEE Trans. Computers,* vol. 22, no. 1, pp. 67-92, Jan. 1973.

[21] R. Fergus, P. Perona, and A. Zisserman, "Object Class Recognition by Unsupervised Scale-Invariant Learning," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2003.

[22] P.F. Felzenszwalb, D. McAllester, and D. Ramanan, "A Discriminatively Trained, Multiscale, Deformable Part Model," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2008.

[23] J.C. Platt, N. Cristianini, and J. Shawe-Taylor, "Large Margin DAGs for Multiclass Classification," *Proc. Advances in Neural Information Processing Systems (NIPS),* 1999.

[24] A. Torralba and K.P. Murphy, "Sharing Visual Features for Multiclass and Multiview Object Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 5, pp. 854-869, May 2007.

[25] P. Zehnder, E.K. Meier, and L.J.V. Gool, "An Efficient Shared Multi-Class Detection Cascade," *Proc. British Machine Vision Conf. (BMVC),* 2008.

[26] E. Miller, N. Matsakis, and P. Viola, "Learning from One Example through Shared Densities on Transforms," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2000.

[27] F.F. Li, R. Fergus, and P. Perona, "One-Shot Learning of Object Categories," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 4, pp. 594-611, Apr. 2006.

[28] E. Bart and S. Ullman, "Cross-Generalization: Learning Novel Classes from a Single Example by Feature Replacement," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2005.

[29] H. Larochelle, D. Erhan, and Y. Bengio, "Zero-Data Learning of New Tasks," *Proc. 23rd Nat'l Conf. Artificial Intelligence,* vol. 1, no. 2, pp. 646-651, 2008.

[30] K. Yanai and K. Barnard, "Image Region Entropy: A Measure of Visualness of Web Images Associated with One Concept," *Proc. 13th Ann. ACM Int'l Conf. Multimedia,* pp. 419-422, 2005.

[31] J. Van De Weijer, C. Schmid, J. Verbeek, and D. Larlus, "Learning Color Names for Real-World Applications," *IEEE Trans. Image Processing,* vol. 18, no. 7, pp. 1512-1523, July 2009.

[32] V. Ferrari and A. Zisserman, "Learning Visual Attributes," *Proc. Advances in Neural Information Processing Systems (NIPS),* 2008.

[33] A.F. Smeaton, P. Over, and W. Kraaij, "Evaluation Campaigns and TRECVid," *Proc. Eighth ACM Int'l Workshop Multimedia Information Retrieval,* 2006.

[34] N. Kumar, P.N. Belhumeur, and S.K. Nayar, "Facetracer: A Search Engine for Large Collections of Images with Faces," *Proc. European Conf. Computer Vision (ECCV),* 2008.

[35] N. Kumar, A. Berg, P. Belhumeur, and S. Nayar, "Describable Visual Attributes for Face Verification and Image Search," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 33, no. 10, pp. 1962-1977, Oct. 2011.

[36] D. Parikh and K. Grauman, "Interactively Building a Discriminative Vocabulary of Nameable Attributes," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* pp. 1681-1688, 2011.

[37] V. Sharmanska, N. Quadrianto, and C.H. Lampert, "Augmented Attribute Representations" *Proc. European Conf. Computer Vision (ECCV),* 2012.

[38] K. Yanai and K. Barnard, "Image Region Entropy: A Measure of 'Visualness' of Web Images Associated with One Concept," *Proc. 13th Ann. ACM Int'l Conf. Multimedia,* 2005.

[39] J. Wang, K. Markert, and M. Everingham, "Learning Models for Object Recognition from Natural Language Descriptions," *Proc. British Machine Vision Conf. (BMVC),* 2009.

[40] T.L. Berg, A.C. Berg, and J. Shih, "Automatic Attribute Discovery and Characterization from Noisy Web Images," *Proc. European Conf. Computer Vision (ECCV),* 2010.

[41] L.J. Li, H. Su, Y. Lim, and L. Fei-Fei, "Objects as Attributes for Scene Classification," *Proc. First Int'l Workshop Parts and Attributes at European Conf. Computer Vision,* 2010.

[42] Y. Wang and G. Mori, "A Discriminative Latent Model of Object Classes and Attributes," *Proc. European Conf. Computer Vision (ECCV),* pp. 155-168, 2010.

[43] X. Yu and Y. Aloimonos, "Attribute-Based Transfer Learning for Object Categorization with Zero/One Training Example," *Proc. European Conf. Computer Vision (ECCV),* pp. 127-140, 2010.

[44] W.J. Scheirer, N. Kumar, P.N. Belhumeur, and T.E. Boult, "Multi-Attribute Spaces: Calibration for Attribute Fusion and Similarity Search," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2012.

[45] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing Objects by their Attributes," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2009.

[46] G. Patterson and J. Hays, "SUN Attribute Database: Discovering, Annotating, and Recognizing Scene Attributes," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2012.

[47] J. Liu, B. Kuipers, and S. Savarese, "Recognizing Human Actions by Attributes," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2011.

[48] R. Feris, B. Siddiquie, Y. Zhai, J. Petterson, L. Brown, and S. Pankanti, "Attribute-Based Vehicle Search in Crowded Surveillance Videos," *Proc. ACM Int'l Conf. Multimedia Retrieval (ICMR),* article 18, 2011.

[49] M. Rohrbach, M. Stark, G. Szarvas, I. Gurevych, and B. Schiele, "What Helps Where—and Why? Semantic Relatedness for Knowledge Transfer," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2010.

[50] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A.C. Berg, and T.L. Berg, "Baby Talk: Understanding and Generating Simple Image Descriptions," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* pp. 1601-1608, 2011.

[51] A. Kovashka, D. Parikh, and K. Grauman, "Whittlesearch: Image Search with Relative Attribute Feedback," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2012.

[52] A. Parkash and D. Parikh, "Attributes for Classifier Feedback," *Proc. European Conf. Computer Vision (ECCV)*, 2012.

[53] D. Osherson, E.E. Smith, T.S. Myers, E. Shafir, and M. Stob, "Extrapolating Human Probability Judgment," *Theory and Decision*, vol. 36, no. 2, pp. 103-129, 1994.

[54] S.A. Sloman, "Feature-Based Induction," *Cognitive Psychology*, vol. 25, pp. 231-280, 1993.

[55] T. Hansen, M. Olkkonen, S. Walter, and K.R. Gegenfurtner, "Memory Modulates Color Appearance," *Nature Neuroscience*, vol. 9, pp. 1367-1368, 2006.

[56] D.N. Osherson, J. Stern, O. Wilkie, M. Stob, and E.E. Smith, "Default Probability," *Cognitive Science*, vol. 15, no. 2, pp. 251-269, 1991.

[57] C. Kemp, J.B. Tenenbaum, T.L. Griffiths, T. Yamada, and N. Ueda, "Learning Systems of Concepts with an Infinite Relational Model," *Proc. Nat'l Conf. Artificial Intelligence (AAAI)*, 2006.

[58] K.E.A. van de Sande, T. Gevers, and C.G.M. Snoek, "Evaluation of Color Descriptors for Object and Scene Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2008.

[59] A. Bosch, A. Zisserman, and X. Muñoz, "Representing Shape with a Spatial Pyramid Kernel," *Proc. Int'l Conf. Content-Based Image and Video Retrieval (CIVR)*, 2007.

[60] H. Bay, A. Ess, T. Tuytelaars, and L.J.V. Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346-359, 2008.

[61] E. Shechtman and M. Irani, "Matching Local Self-Similarities across Images and Videos," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2007.

[62] J. Xiao, J. Hays, K.A. Ehinger, A. Oliva, and A. Torralba, "SUN Database: Large-Scale Scene Recognition from Abbey to Zoo," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 3485-3492, 2010.

[63] J.C. Platt, "Probabilities for SV Machines," *Advances in Large Margin Classifiers.* MIT Press, 2000.

[64] L. Torresani, M. Szummer, and A. Fitzgibbon, "Efficient Object Category Recognition Using Classemes," *Proc. European Conf. Computer Vision (ECCV)*, pp. 776-789, Sept. 2010.

[65] K.D. Tang, M.F. Tappen, R. Sukthankar, and C.H. Lampert, "Optimizing One-Shot Recognition with Micro-Set Learning," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2010.

[66] W.J. Scheirer, A. Rocha, A. Sapkota, and T.E. Boult, "Toward Open Set Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 35, no. 7, pp. 1757-1772, July 2013.

[67] T. Tommasi, N. Quadrianto, B. Caputo, and C.H. Lampert, "Beyond Data Set Bias: Multi-Task Unaligned Shared Knowledge Transfer," *Proc. Asian Conf. Computer Vision (ACCV)*, 2012.

Christoph H. Lampert received the PhD degree in mathematics from the University of Bonn in 2003. He was a senior researcher at the German Research Center for Artificial Intelligence in Kaiserslautern and a senior research scientist at the Max Planck Institute for Biological Cybernetics in Tübingen. He is currently an assistant professor at the Institute of Science and Technology Austria, where he heads a research group for computer vision and machine learning. He has received several international and national awards for his research, including the Best Paper Prize of CVPR 2008 and Best Student Paper Award of ECCV 2008. In 2012, he was awarded an ERC Starting Grant by the European Research Council. He is an associate editor of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* and an action editor for the *Journal of Machine Learning Research*.

Hannes Nickisch received degrees from the Université de Nantes, France, in 2004, and the Technical University Berlin, Germany, in 2006. During his PhD work at the Max Planck Institute for Biological Cybernetics, Tübingen, Germany, he worked on large-scale approximate Bayesian inference and magnetic resonance image reconstruction. Since 2011, he has been with Philips Research, Hamburg, Germany. His research interests include medical image processing, machine learning, and biophysical modeling.

Stefan Harmeling studied mathematics and logic at the University of Münster (Dipl Math 1998) and computer science with an emphasis on artificial intelligence at Stanford University (MSc 2000). During his doctoral studies, he was a member of Prof. Klaus-Robert Müller's research group at the Fraunhofer Institute FIRST (Dr rer nat 2004). Thereafter, he was a Marie Curie fellow at the University of Edinburgh from 2005 to 2007, before joining the Max Planck Institute for Biological Cybernetics/Intelligent Systems. He is currently a senior research scientist in Prof. Bernhard Schölkopf's Department of Empirical Inference at the Max Planck Institute for Intelligent Systems (formerly Biological Cybernetics). His research interests include machine learning, image processing, computational photography, and probabilistic inference. In 2011, he received the DAGM Paper Prize, and in 2012 the Günter Petzow Prize for outstanding work at the Max Planck Institute for Intelligent Systems.