

# Learning To Detect Unseen Object Classes by Between-Class Attribute Transfer

Christoph H. Lampert   Hannes Nickisch   Stefan Harmeling  
Max Planck Institute for Biological Cybernetics, Tübingen, Germany

{firstname.lastname}@tuebingen.mpg.de

## Abstract

We study the problem of object classification when training and test classes are disjoint, i.e. no training examples of the target classes are available. This setup has hardly been studied in computer vision research, but it is the rule rather than the exception, because the world contains tens of thousands of different object classes and for only a very few of them image collections have been formed and annotated with suitable class labels.

In this paper, we tackle the problem by introducing attribute-based classification. It performs object detection based on a human-specified high-level description of the target objects instead of training images. The description consists of arbitrary semantic attributes, like shape, color or even geographic information. Because such properties transcend the specific learning task at hand, they can be pre-learned, e.g. from image datasets unrelated to the current task. Afterwards, new classes can be detected based on their attribute representation, without the need for a new training phase. In order to evaluate our method and to facilitate research in this area, we have assembled a new large-scale dataset, “Animals with Attributes”, of over 30,000 animal images that match the 50 classes in Osherson’s classic table of how strongly humans associate 85 semantic attributes with animal classes. Our experiments show that by using an attribute layer it is indeed possible to build a learning object detection system that does not require any training images of the target classes.

## 1. Introduction

Learning-based methods for recognizing objects in natural images have made large progress over the last years. For specific object classes, in particular faces and vehicles, reliable and efficient detectors are available, based on the combination of powerful low-level features, e.g. SIFT or HoG, with modern machine learning techniques, e.g. boosting or support vector machines. However, in order to achieve good classification accuracy, these systems require a lot of manually labeled training data, typically hundreds or thousands of example images for each class to be learned.

It has been estimated that humans distinguish between at least 30,000 relevant object classes [3]. Training conventional object detectors for all these would require mil-

### otter

black:    yes  
white:   no  
brown:   yes  
stripes:  no  
water:   yes  
eats fish: yes



### polar bear

black:    no  
white:    yes  
brown:   no  
stripes:  no  
water:   yes  
eats fish: yes



### zebra

black:    yes  
white:    yes  
brown:   no  
stripes:  yes  
water:   no  
eats fish: no



Figure 1. A description by *high-level attributes* allows the transfer of knowledge between object categories: after learning the visual appearance of attributes from any classes with training examples, we can detect also object classes that do not have any training images, based on which attribute description a test image fits best.

lions of well-labeled training images and is likely out of reach for years to come. 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 is different: although humans can learn and abstract well from examples, they are also capable of detecting completely unseen classes when provided with a high-level description. *E.g.*, 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*. We build on this paradigm and propose a system that is able to detect objects from a list of high-level attributes. The attributes serve as an intermediate layer in a classifier cascade and they enable the system to detect object classes, for which it had not seen a single training example.

Clearly, a large number of possible attributes exist and collecting separate training material to learn an ordinary classifier for each of them would be as tedious as for all object classes. But, 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 therefore make use of existing training data by merging images of several object classes. To learn, *e.g.*, 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.

## 2. Information Transfer by Attribute Sharing

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

### Learning with Disjoint Training and Test Classes:

Let  $(x_1, l_1), \dots, (x_n, l_n) \subset \mathcal{X} \times \mathcal{Y}$  be training samples where  $\mathcal{X}$  is an arbitrary feature space and  $\mathcal{Y} = \{y_1, \dots, y_K\}$  consists of  $K$  discrete classes. The task is to learn a classifier  $f : \mathcal{X} \rightarrow \mathcal{Z}$  for a label set  $\mathcal{Z} = \{z_1, \dots, z_L\}$  that is disjoint from  $\mathcal{Y}$ <sup>1</sup>.

Clearly, this task cannot be solved by an ordinary multi-class classifier. Figure 2(a) provides a graphical illustration of the problem: typical classifiers learn one parameter vector (or other representation)  $\alpha_k$  for each training class  $y_1, \dots, y_K$ . Because the classes  $z_1, \dots, z_L$  were 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.

In order to make predictions about classes, for which no training data is available, we need to introduce a coupling between classes in  $\mathcal{Y}$  and  $\mathcal{Z}$ . Since no training data for the unobserved classes is available, this coupling cannot be learned from samples, but has to be inserted into the system by human effort. This introduces two severe constraints on what kind of coupling mechanisms are feasible: 1) the amount of human effort to specify new classes should be minimal, because otherwise collecting and labeling training samples would be a simpler solution; 2) coupling data that requires only *common knowledge* is preferable over specialized expert knowledge, because the latter is often difficult and expensive to obtain.

### 2.1. Attribute-Based Classification:

We achieve both goals by introducing a small set of high-level semantic per-class attributes. These can be *e.g.* color

<sup>1</sup>The conditions that  $\mathcal{Y}$  and  $\mathcal{Z}$  are disjoint is included only to clarify the later presentation. The problem described also occurs if just  $\mathcal{Z} \not\subseteq \mathcal{Y}$ .

and shape for arbitrary objects, or the natural habitat for animals. Humans are typically able to provide good prior knowledge about such attributes, and it is therefore possible to collect the necessary information without a lot of overhead. Because the attributes are assigned on a per-class basis instead of a per-image basis, the manual effort to add a new object class is kept minimal.

For the situation where attribute data of this kind of available, we introduce *attribute-based classification*:

### Attribute-Based Classification:

Given 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*  $a \in \mathcal{A}$  is available, then we can learn a non-trivial 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* is indeed 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 multi-class classification:

*Direct attribute prediction (DAP)*, illustrated by Figure 2(b), 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  $\beta_m$ . At test time, these allow the prediction of attribute values for each test sample, from which the test class label 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 Figure 2(c), 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 is ordinary multi-class 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 can be inferred.

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 are 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 a test time as an intermediate

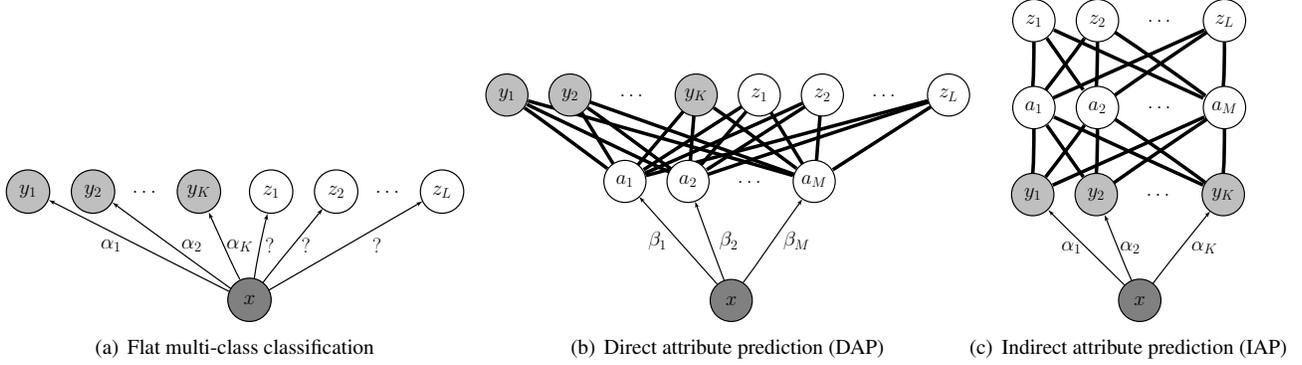


Figure 2. Graphical representation of the proposed across-class learning task: dark gray nodes are always observed, light gray nodes are observed only during training. White nodes are never observed but must be inferred. An ordinary, flat, multi-class classifier (left) learns one parameter  $\alpha_k$  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  $\beta_m$  are learned. At test time, attribute values can directly be inferred, and these imply output class label even for previously unseen classes. A multi-class based attribute classifier (right) combined both ideas: multi-class parameters  $\alpha_k$  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.

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 regularization step that creates only *sensible* attribute combinations and therefore makes the system more robust. In the following, we will develop implementations for both methods and benchmark their performance.

## 2.2. Implementation

Both cascaded classification methods, DAP and IAP, can in principle be implemented by combining 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 of Figures 2(b) and 2(c). For simplicity, we assume that all attributes have binary values such that the attribute representation  $a^y = (a_1^y, \dots, a_m^y)$  for any training class  $y$  are fixed-length binary vectors. Continuous attributes can in principle be handled in the same way by using regression instead of classification.

For DAP, we start by learning probabilistic classifiers for each attribute  $a_m$ . We use all images from all training classes as training samples with their label determined by the entry of the attribute vector corresponding to the sample’s label, *i.e.* a sample of class  $y$  is assigned the binary label  $a_m^y$ . The trained classifiers provide us with estimates of  $p(a_m|x)$ , from which we form a model for the complete *image–attribute* layer as  $p(a|x) = \prod_{m=1}^M p(a_m|x)$ . At test time, we assume that every class  $z$  induces its attribute vector  $a^z$  in a deterministic way, *i.e.*  $p(a|z) = \llbracket a = a^z \rrbracket$ , making use of Iverson’s bracket notation:  $\llbracket P \rrbracket = 1$  if the con-

dition  $P$  is true and it is 0 otherwise [19]. Applying Bayes’ rule we obtain  $p(z|a) = \frac{p(z)}{p(a^z)} \llbracket a = a^z \rrbracket$  as representation of the *attribute–class* layer. Combining both layers, we can calculate the posterior of a test class given an image:

$$p(z|x) = \sum_{a \in \{0,1\}^M} p(z|a)p(a|x) = \frac{p(z)}{p(a^z)} \prod_{m=1}^M p(a_m^z|x). \quad (1)$$

In the absence of more specific knowledge, we assume identical 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  $p(a_m) = \frac{1}{K} \sum_{k=1}^K a_m^{y_k}$  over the training classes as attribute priors.<sup>2</sup> As decision rule  $f : \mathcal{X} \rightarrow \mathcal{Z}$  that assigns the best output class from all test classes  $z_1, \dots, z_L$  to a test sample  $x$ , we use MAP prediction:

$$f(x) = \operatorname{argmax}_{l=1,\dots,L} \prod_{m=1}^M \frac{p(a_m^{z_l}|x)}{p(a_m^{z_l})}. \quad (2)$$

In order to implement IAP, we only modify the image–attribute stage: as first step, we learn a probabilistic multi-class classifier estimating  $p(y_k|x)$  for all training classes  $y_1, \dots, y_K$ . Again assuming a deterministic dependence between attributes and classes, we set  $p(a_m|y) = \llbracket a_m = a_m^y \rrbracket$ . The combination of both steps yields

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

so inferring the attribute posterior probabilities  $p(a_m|x)$  requires only a matrix–vector multiplication. Afterwards, we

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

continue in the same way as in for DAP, classifying test samples using Equation (2).

### 3. Connections to Previous Work

*Multi-layer* or *cascaded classifiers* have a long tradition in pattern recognition and computer vision: *multi-layer perceptrons* [29], *decision trees* [5], *mixtures of experts* [17] and *boosting* [14] are prominent examples of classification systems built as feed-forward architectures with several stages. Multi-class classifiers are also often constructed as layers of binary decisions, from which the final output is inferred, e.g. [7, 28]. These methods differ in their training methodologies, but they share the goal of decomposing a difficult classification problem into a collection of simpler ones. Because their emphasis lies on the classification performance in a fully supervised scenario, the methods are not capable of generalizing across class boundaries.

Especially in the area of computer vision, multi-layered 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 like edge and corner detectors [27]. Popular local feature descriptors, such as SIFT [21] or HoG [6], 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 non-informative image variations. Similarly, image segmentation has been proposed as an unsupervised method to extract contours that are discriminative for object classes [37]. Such pre-processing 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, *pictorial structures* [13], the *constellation model* [10] and recent *discriminatively trained deformable part models* [9] 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*, e.g. *arms* or *wheels*. However, it is a difficult problem to build a system that learns to detect exactly the parts described. Instead, the identification of parts is integrated into the training of the model, which often reduces the parts to co-occurrence patterns of local feature points, not to units with a semantic meaning. In general, parts learned this way do generalize across class boundaries.

#### 3.1. Sharing Information between Classes

The aspect of sharing information between classes has also been recognized as an interesting field before. A com-

mon idea is to construct multi-class 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 performance [26]. Similarly, one can reduce the number of feature calculations by actively selecting low-level features that help discrimination for many classes simultaneously [33]. Combinations of both approaches are also possible [39].

In contrast, *inter-class 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 intra-class variance in terms of distortions [22] or shapes and appearances [20]. Alternatively, features that are known to be discriminative for some classes can be reused and adapted to support the detection of new classes [1]. To our knowledge, no previous approach allows the direct incorporation of human prior knowledge. Also, all methods require at least some training examples and cannot handle completely new object classes.

A noticeable exception is [8] that uses high-level attributes to learn *descriptions* of object. Like our approach, this opens the possibility to generalize between categories.

#### 3.2. Learning Semantic Attributes

A different line of relevant research occurring as one building block for attribute-based classification is the *learning of high-level semantic attributes* from images. Prior work in the area of computer vision has mainly studied elementary properties like colors and geometric patterns [11, 36, 38], achieving high accuracy by developing task-specific features and representations. In the field of multimedia retrieval, the annual TRECVID contest [32] contains a subtask of *high-level feature extraction*. It has stimulated a lot of research in the detection of *semantic concepts*, including the categorization of scene types, e.g. *outdoor*, *urban*, and high-level actions, e.g. *sports*. Typical systems in this area combine many feature representations and, because they were designed for retrieval scenarios, they aim at high precision for low recall levels [34, 40].

Our own task of attribute learning targets a similar problem, but our final goal is not the prediction of few individual attributes. Instead, we want to infer class labels by combining the predictions of many attributes. Therefore, we are relatively robust to prediction errors on the level of individual attributes, and we will rely on generic classifiers and standard image features instead of specialized setups.

In contrast to computer science, a lot of work in *cognitive science* has been dedicated to studying the relations between object recognition and attributes. Typical questions in the field are how human judgements are influenced by characteristic object attributes [23, 31]. A related line of research studies how the human performance in object

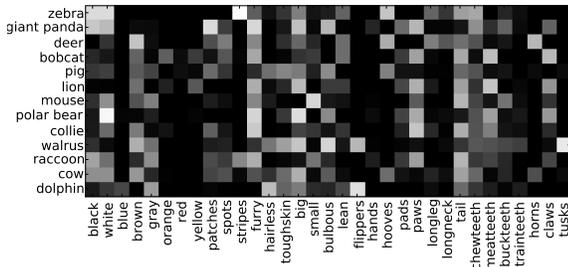


Figure 3. *Class–attribute* matrices from [24, 18]. The responses of 10 test persons were averaged to determine the real-valued association strength between attributes and classes. The darker the boxes, the less is the attribute associated with the class. Binary attributes are obtained by thresholding at the overall matrix mean.

detection tasks depends on the presence or absence of object properties and contextual cues [16]. 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 new dataset of animal images that allows us to make use of existing class-attribute association data, which was collected from cognitive science research.

#### 4. The *Animals with Attributes* Dataset

For their studies on attribute-based object similarity, Osherson and Wilkie [24] collected judgements from human subjects on the “*relative strength of association*” between 85 attributes and 48 animal classes. Kemp et al. [18] made use of the same data in a machine learning context and added 2 more animals classes. Figure 3 illustrates an excerpt of the resulting  $50 \times 85$  class-attribute matrix. However, so far this data was not usable in a computer vision context, because the animals and attributes are only specified by their abstract names, not by example images. To overcome this problem, we have collected the *Animals with Attributes* data.<sup>3</sup>

##### 4.1. Image Collection

We have collected example images for all 50 Osherson/Kemp animal classes by querying four large internet search engines, *Google*, *Microsoft*, *Yahoo* and *Flickr*, using the animal names as keywords. The resulting over 180,000 images were manually processed to remove outliers and duplicates, and to ensure that the target animal is in a prominent view in all cases. The remaining collection consists of 30475 images with at minimum of 92 images for any class. Figure 1 shows examples of some classes with the values of exemplary attributes assigned to this class. Altogether, animals are uniquely characterized by their attribute vector. Consequently, the *Animals with Attributes* dataset, formed

<sup>3</sup>Available at <http://attributes.kyb.tuebingen.mpg.de>

by combining the collected images with the semantic attribute table, can serve as a testbed for the task of incorporating human knowledge into an object detection system.

#### 4.2. Feature Representations

Feature extraction is known to have a big influence in computer vision tasks. For most image datasets, e.g. Caltech [15] and PASCAL VOC<sup>4</sup>, it has become difficult to judge the true performance of newly proposed classification methods, because results based on very different feature sets need to be compared. We have therefore decided to include a reference set of pre-extracted features into the *Animals with Attributes* dataset.

We have selected six different feature types: RGB color histograms, SIFT [21], rgSIFT [35], PHOG [4], SURF [2] and local self-similarity histograms [30]. The color histograms and PHOG feature vectors are extracted separately for all 21 cells of a 3-level spatial pyramids ( $1 \times 1$ ,  $2 \times 2$ ,  $4 \times 4$ ). For each cell, 128-dimensional color histograms are extracted and concatenated to form a 2688-dimensional feature vector. For PHOG, the same construction is used, but with 12-dimensional base histograms. The other feature vectors each are 2000-bin *bag-of-visual words* histograms.

For the consistent evaluation of attribute-based object classification methods, we have selected 10 test classes: *chimpanzee*, *giant panda*, *hippopotamus*, *humpback whale*, *leopard*, *pig*, *raccoon*, *rat*, *seal*. The 6180 images of those classes act as test data, whereas the 24295 images of the remaining 40 classes can be used for training. Additionally, we also encourage the use of the dataset for regular large-scale multi-class or multi-label classification. For this we provide ordinary *training/test* splits with both parts containing images of all classes. In particular, we expect the *Animals with Attributes* dataset to be suitable to test hierarchical classification techniques, because the classes contain natural subgroups of similar appearance.

#### 5. Experimental Evaluation

In Section 2 we introduced DAP and IAP, two methods for attribute-based classification, that allow the learning of object classification systems for classes for which no training samples are available. In the following, we evaluate both methods by applying them to the *Animals with Attributes* dataset. For DAP, we train a non-linear support vector machine (SVM) to predict each binary attribute  $a_1, \dots, a_M$ . All attribute SVMs are based on the same kernel, the sum of individual  $\chi^2$ -kernels for each feature, where the bandwidth parameters are fixed to the five times inverse of the median of the  $\chi^2$ -distances over the training samples. The SVM’s parameter  $C$  is set to 10, which had been determined a priori by cross-validation on a subset of the training

<sup>4</sup><http://www.pascal-network.org/challenges/VOC/>

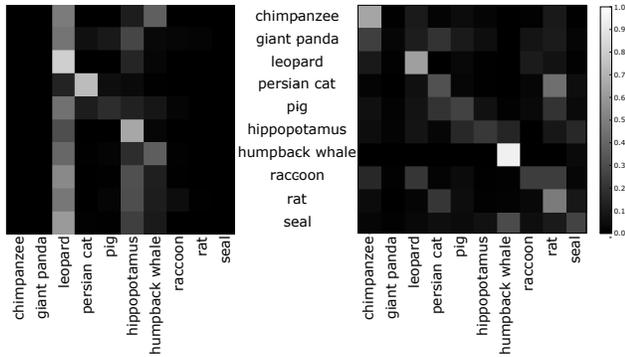


Figure 4. Confusion matrices between 10 test classes of the *Animals with Attributes* dataset (best viewed in color). Left: indirect attribute prediction (IAP), right: direct attributes prediction (DAP).

classes. In order to obtain probability estimates, we perform the SVM training using only on 90% of the training samples and use the remaining training data as validation set for *Platt scaling* [25]. For IAP, we use  $L^2$ -regularized multi-class logistic regression with the same feature representation as above. This directly gives us an estimate of the class posteriors  $p(y_k|x)$  that we turn into attribute posteriors by Equation (3). Because the linear training is much faster than the non-linear one, we can determine the regularization parameters  $C$  by five-fold cross-validation in this setup.

### 5.1. Results

Having trained the predictors for  $p(a_m|x)$  on the training part of the *Animals with Attributes* dataset, we use the test classes' attribute vectors and Equation (2) to perform multi-class classification for the test part of the dataset. This results in a multi-class accuracy of 40.5% for DAP, as measured by the mean of the diagonal of the confusion matrix, and a multi-class accuracy of 27.8% for IAP. Figure 4 shows the resulting confusion matrices for both methods. Clearly, this performance is significantly higher than the chance level of 10%, in particular for DAP, which proves our original statement on *attribute-based classification*: by sharing information via an attribute layer, it is possible to classify images of classes that were not part of the training data<sup>5</sup>.

Because of the better performance of DAP, in the following we will give more detailed results only for this method. Figure 6 show the quality of the individual attribute predictors on the test data. Note that for several attributes performance is not much better than random ( $AUC \approx 0.5$ ), whereas other features can be learned almost perfectly ( $AUC \approx 1$ ). This

<sup>5</sup>Note that, although the train/test split we use is somewhat *balanced* with regard to the nature of the classes, the performance for other splits are comparable: using classes 1–10, 11–20, etc. as test classes and the remaining ones as test classes, DAP achieves multiclass accuracies of 33.7%, 42.0%, 32.6%, 34.3%, 33.2% and IAP of 27.6%, 25.4%, 24.9%, 18.6%, 25.6%.

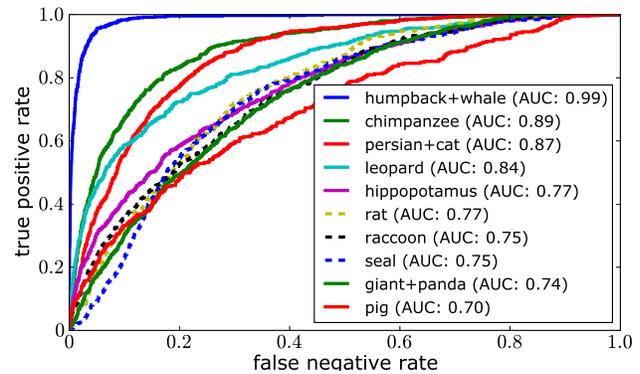


Figure 5. Detection performance of object classification with disjoint training and test classes (DAP method): ROC-curves and area under curve (AUC) for the 10 *Animals with Attributes* test classes.

shows the importance of having a large and diverse feature set. Note also, that even non-visual attributes like *smelly* can be learned better than chance, presumably because they are correlated with visual properties. Figure 5 depicts the the resulting *ROC curves* and their *area under curve* (AUC) values. One can see that for all classes, reasonable classifiers have been learned. With an AUC of 0.99, the performance for *humpback whale* is even on par with what fully supervised techniques typically achieve. Figure 7 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 Equation (1). One can see that the rankings for *humpback whales*, *chimpanzees*, *leopards*, *hippopotamuses* and *raccoons* are very reliable, whereas the other rankings contain several errors. Since all classifiers base their decisions on the same learned attributes, this suggests that either these classes are characterized by attributes that are more difficult to learn, or that the attribute characterization itself is less informative for these classes.

To our knowledge, no previous methods for object detection with disjoint training and test classes exist. Typical unsupervised techniques like clustering are not applicable in our setup, either, since we want to individually assign class labels to test images, not only identify groups of similar appearance within a large collection of test images. For comparison, we have therefore implemented a simple *one-shot learning* approach: it learns a diagonal covariance matrix of feature variance from the training classes and uses the resulting Mahalanobis-distance for nearest-neighbor classification. Each target class is represented by up to 10 images randomly taken from the test set. However, this setup achieves only accuracies between 14.3% for 1 training image and 18.9% for 10 training images, thereby not clearly improving over the chance level. This shows how difficult both the problem and the dataset are. To put our results into a larger perspective, nonetheless, we also compare our method to ordinary multi-class classification, using the standard setup of Figure 2(a) with a 50/50 split of test class im-

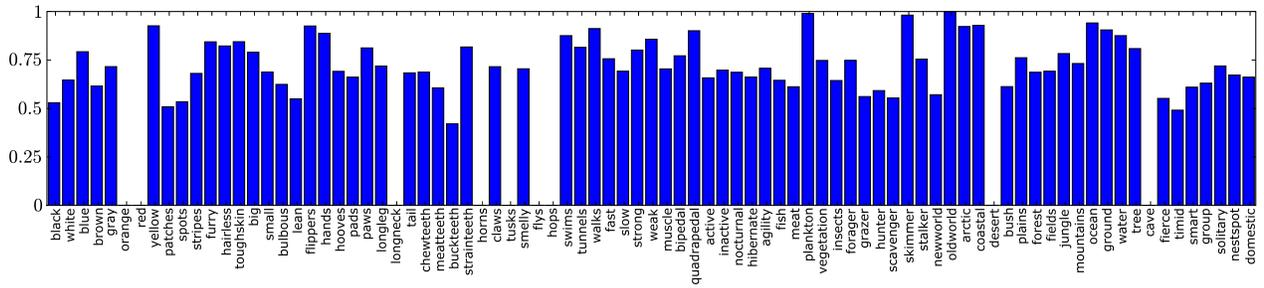


Figure 6. Quality of individual attribute predictors (trained on *train* classes, tested on *test* classes), as measured by by *area under ROC curve* (*AUC*). Attributes with zero entries have constant values for all test classes of the split chosen, so their *AUC* is undefined.

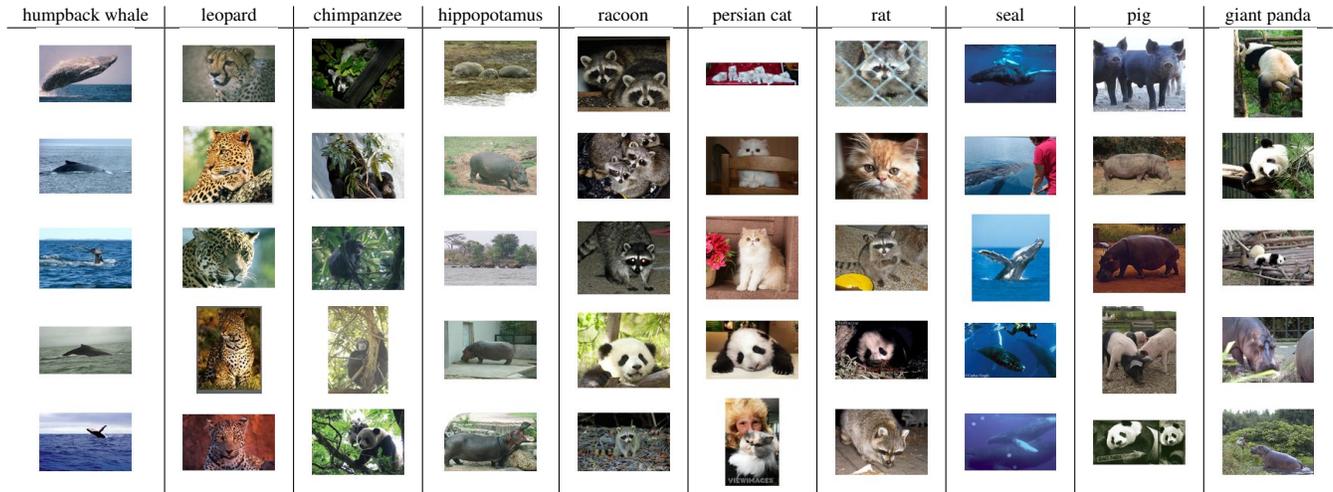


Figure 7. Highest ranking results for each test class in the *Animals with Attributes* dataset. Uniquely characterized classes are identified well, e.g. humpback whales and leopards. Confusions occur between visually similar categories, e.g. pigs and hippopotamuses.

ages for training and testing. Clearly, the availability of a large number of training samples from the same classes as the test data vastly simplifies the problem. With a resulting multi-class accuracy of 65.9%, supervised training does indeed perform better than the 40.5% achieved by *attribute-based learning*. However, given the different amount of training information included, we believe that the difference in accuracy is not discouragingly large, and that learning with attributes has the potential to complement supervised classification in areas where no or only few training examples are available.

## 6. Conclusion

In this paper, we have introduced *learning for 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 have proposed two methods for *attribute-based classification* that solve this problem by transferring information between classes. The transfer is achieved by an intermediate representation that consists of high level, semantic, per-class attributes, providing a fast and easy way to include human knowledge into the system.

Once trained, the system can detect any object category, for which a suitable characterization by attributes is available, and it does not require a re-training step.

Additionally, we have introduced the *Animals with Attributes* dataset: it consists over 30,000 images with pre-computed reference features for 50 animal classes, for which a semantic attribute annotation is available from studies in cognitive science. We hope that this dataset will facilitate research and serve as a testbed for *attribute-based classification*.

Starting from the proposed system, many improvements and extensions are possible. Clearly, better designed per-attribute and multi-class classifiers could improve the overall performance of the system, as could a per-attribute feature selection set, because clearly not all attributes relevant for a human can be determined from images. For an adaptive system that can grow to include new classes, it should be possible to increase the attribute set without retraining. It would also be very interesting to remove the amount of human effort, e.g. by letting a human define the attributes, but build an automatic system to label the image classes with the attribute values, possibly using textual information from

the internet. Ultimately, this could even lead to a fully automatic determination of both, the attributes and their values.

A different interesting direction for future work, is the question how attribute-based classification and supervised classification can be merged to improve the classification accuracy when training examples are available but scarce. This would make attribute-based classification applicable to existing transfer learning problems with many classes, but few examples per class, *e.g.* [12].

## Acknowledgments

This work was funded in part by the EC project CLASS, IST 027978, and the PASCAL2 network of excellence. We thank Charles Kemp for providing the Osherson/Wilkie *class-attribute* matrix and Jens Weidmann for several weeks of labeling work.

## References

- [1] E. Bart and S. Ullman. Cross-generalization: Learning novel classes from a single example by feature replacement. In *CVPR*, 2005.
- [2] H. Bay, A. Ess, T. Tuytelaars, and L. J. V. Gool. Speeded-up robust features (SURF). *CVIU*, 110(3), 2008.
- [3] I. Biederman. Recognition by components - a theory of human image understanding. *Psychological Review*, 94(2), 1987.
- [4] A. Bosch, A. Zisserman, and X. Muñoz. Representing shape with a spatial pyramid kernel. In *CIVR*, 2007.
- [5] L. Breiman, J. J. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth, 1984.
- [6] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In *CVPR*, 2005.
- [7] T. G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2, 1995.
- [8] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009.
- [9] P. F. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.
- [10] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, 2003.
- [11] V. Ferrari and A. Zisserman. Learning visual attributes. In *NIPS*, 2008.
- [12] M. Fink and S. Ullman. From Aardvark to Zorro: A benchmark for mammal image classification. *IJCV*, 77(1-3), 2008.
- [13] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Trans. Computer*, 22(1), Jan. 1973.
- [14] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1), Aug. 1997.
- [15] G. Griffin, A. Holub, and P. Perona. Caltech-256 Object Category Dataset. Technical Report 7694, California Institute of Technology, 2007.
- [16] T. Hansen, M. Olkkonen, S. Walter, and K. R. Gegenfurtner. Memory modulates color appearance. *Nature Neuroscience*, 9, 2006.
- [17] M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6(2), 1994.
- [18] C. Kemp, J. B. Tenenbaum, T. L. Griffiths, T. Yamada, and N. Ueda. Learning systems of concepts with an infinite relational model. In *AAAI*, 2006.
- [19] D. E. Knuth. Two notes on notation. *Amer. Math. Monthly*, 99(5), May 1992.
- [20] F. F. Li, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE PAMI*, 28(4), Apr. 2006.
- [21] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 60(2), 2004.
- [22] E. Miller, N. Matsakis, and P. Viola. Learning from one example through shared densities on transforms. In *CVPR*, 2000.
- [23] D. Osherson, E. E. Smith, T. S. Myers, E. Shafir, and M. Stob. Extrapolating human probability judgment. *Theory and Decision*, 2, March 1994.
- [24] D. N. Osherson, J. Stern, O. Wilkie, M. Stob, and E. E. Smith. Default probability. *Cognitive Science*, 15(2), 1991.
- [25] J. C. Platt. Probabilities for SV machines. In *Advances in Large Margin Classifiers*. MIT Press, 2000.
- [26] J. C. Platt, N. Cristianini, and J. Shawe-Taylor. Large margin DAGs for multiclass classification. In *NIPS*, 1999.
- [27] M. Ranzato, F. J. Huang, Y.-L. Boureau, and Y. LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *CVPR*, 2007.
- [28] R. Rifkin and A. Klautau. In defense of one-vs-all classification. *JMLR*, 5, 2004.
- [29] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In *Parallel Distributed Processing*. MIT Press, Cambridge, MA, 1986.
- [30] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *CVPR*, 2007.
- [31] S. A. Sloman. Feature-based induction. *Cognitive Psychology*, 25, 1993.
- [32] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and trecvid. In *ACM Multimedia Information Retrieval*, 2006.
- [33] A. Torralba and K. P. Murphy. Sharing visual features for multiclass and multiview object detection. *IEEE PAMI*, 29(5), 2007.
- [34] A. Ulges, C. Schulze, D. Keysers, and T. Breuel. Identifying relevant frames in weakly labeled videos for training concept detectors. In *CIVR*, 2008.
- [35] K. van de Sande, T. Gevers, and C. Snoek. Evaluation of color descriptors for object and scene recognition. In *CVPR*, 2008.
- [36] J. van de Weijer, C. Schmid, and J. Verbeek. Learning color names from real-world images. In *CVPR*, 2007.
- [37] J. Winn and N. Jojic. LOCUS: Learning object classes with unsupervised segmentation. In *ICCV*, volume I, 2005.
- [38] K. Yanai and K. Barnard. Image region entropy: a measure of "visualness" of web images associated with one concept. In *ACM Multimedia*, 2005.
- [39] P. Zehnder, E. K. Meier, and L. J. V. Gool. An efficient shared multi-class detection cascade. In *BMVC*, 2008.
- [40] Y.-T. Zheng, S.-Y. Neo, T.-S. Chua, and Q. Tian. Probabilistic optimized ranking for multimedia semantic concept detection via RVM. In *CIVR*, 2008.