Describing Objects by their Attributes

Ali Farhadi, Ian Endres, Derek Hoiem, David Forsyth
Computer Science Department
University of Illinois at Urbana-Champaign
{afarhad2,iendres2,dhoiem,daf}@uiuc.edu

Abstract

We propose to shift the goal of recognition from naming to describing. Doing so allows us not only to name familiar objects, but also: to report unusual aspects of a familiar object ("spotty dog", not just "dog"); to say something about unfamiliar objects ("hairy and four-legged", not just "unknown"); and to learn how to recognize new objects with few or no visual examples. Rather than focusing on identity assignment, we make inferring attributes the core problem of recognition. These attributes can be semantic ("spotty") or discriminative ("dogs have it but sheep do not"). Learning attributes presents a major new challenge: generalization across object categories, not just across instances within a category. In this paper, we also introduce a novel feature selection method for learning attributes that generalize well across categories. We support our claims by thorough evaluation that provides insights into the limitations of the standard recognition paradigm of naming and demonstrates the new abilities provided by our attribute-based framework.

1. Introduction

We want to develop computer vision algorithms that go beyond naming and infer the properties or attributes of objects. The capacity to infer attributes allows us to describe, compare, and more easily categorize objects. Importantly, when faced with a new kind of object, we can still say something about it (e.g., "furry with four legs") even though we cannot identify it. We can also say what is unusual about a particular object (e.g, "dog with spots") and learn to recognize objects from description alone.

In this paper, we show that our attribute-centric approach to object recognition allows us to do a better job in the traditional naming task and provides many new abilities. We focus on learning object attributes, which can be semantic or not. Semantic attributes describe parts ("has nose"), shape ("cylindrical"), and materials ("furry"). They can be learned from annotations and allow us to describe objects and to identify them based on textual descriptions. But they are not always sufficient for differentiating between object categories. For instance, it is difficult to describe the difference between cats and dogs, even though there are many visual dissimilarities. Therefore, we also learn non-semantic attributes that correspond to splits in the visual feature space. These can be learned by defining auxiliary tasks, such as to differentiate between cars and motorbikes using texture.

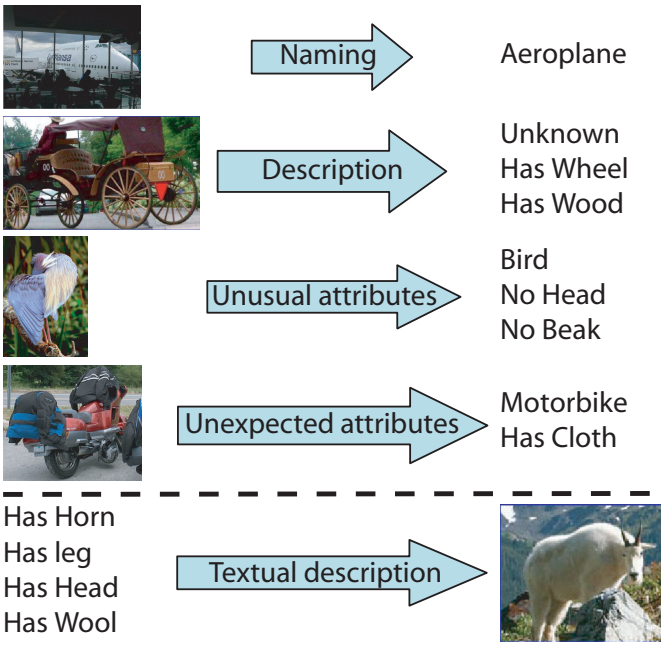

Figure 1: Our attribute based approach allows us not only to effectively recognize object categories, but also to describe unknown object categories, report atypical attributes of known classes, and even learn models of new object categories from pure textual description.

When learning the attributes, we want to be able to generalize to new types of objects. Generalizing both within categories and *across categories* is extremely challenging, and we believe that studying this problem will lead to new insights that are broadly applicable in computer vision. Training attribute classifiers in the traditional way (use all features to classify whether an object has an attribute) leads to poor generalization for some attributes across categories. This is because irrelevant features (such as color when learning shape) are often correlated with attributes for some sets of objects but not others. Instead, we propose to first select features that can predict attributes within an object class and use only those to train the attribute classifier.

For instance, to learn a "spots" detector, we would select features that can distinguish between dogs with and without spots, cats with and without spots, horses with and without spots, and so on. We then use only these selected features to train a single spot detector for all objects.

A key goal is to describe objects and to learn from descriptions. Two objects with the same name (e.g., "car") may have differences in materials or shapes, and we would like to be able to recognize and comment on those differences. Further, we may encounter new types of objects. Even though we can't name them, we would like to be able to say something about them. Finally, we would like to learn about new objects quickly, sometimes purely from a textual description. These are important tools for humans, and we are the first to develop them in computer vision at the object category level.

We have developed new annotations and datasets to test our ability to describe, compare, and categorize objects. In particular, using Amazon's Mechanical Turk [21], we obtained 64 attribute labels for each of the twenty objects in the PASCAL VOC 2008 [4] trainval set of roughly 12,000 instances. We also downloaded images using Yahoo! image search for twelve new types of objects and labeled them with attributes in a similar manner. To better focus on description, we perform experiments on objects that have been localized (with a bounding box) but not identified. Thus, we deal with the question "What is this?", but not "Where is this?" We want to show that our attribute-based approach allows us to effectively categorize objects, describe known and new objects, and learn to categorize new types of objects. We are particularly interested in the question of how well we can generalize to new types of objects, something that has not been extensively studied in past work.

Our experiments demonstrate that our attribute-based approach to recognition has several benefits. First, we can effectively categories objects. The advantage is particularly strong when few training examples are available, likely because attributes can be shared across categories and provide a compact but discriminative representation. Our tests also indicate that selecting features provide large gains in learning from textual description and reporting unusual attributes of objects. Surprisingly, we found that we can classify objects from a purely textual description as accurately as if we trained from several examples. These experimental results are extremely encouraging and indicate that attribute-based recognition is an important area for further study.

2. Background

Our notion of attributes comes from the literature on concepts and categories (reviewed in [15]). While research on "basic level" categories [19] indicates that people tend to use the same name to refer to objects (e.g., "look at that cat" instead of "look at that Persian longhair" or "look at that mammal"), there is much evidence [13] that category formation and assignment depends on what attributes we know and on our current goal. A cat in different contexts could be a "pet", "pest", or "predator." The fluid nature of object categorization makes attribute learning essential. For this reason, we make attribute learning the center of our framework, allowing us to go beyond basic level naming. We do not, however, attempt to resolve the long-standing debate between exemplar and prototype models; instead we experiment with a variety of classifiers. In this, we differ from Malisiewicz and Efros [14] who eschew categorization altogether, treating recognition as a problem of finding the most similar exemplar object (but without trying to say how that object is similar). Our model is also different from approaches like [24] because our attributes are more general than just textures.

Space does not allow a comprehensive review of current work on object recognition. The main contrast is that our work involves a form of generalization that is novel to the literature — we want our system to make useful statements about objects whose name it does not happen to know. This means that we must use an intermediate representation with known semantics (our attributes). It also means that we must ensure that we can predict attributes correctly for categories that were not used in training (section 4).

Ferrari and Zisserman [9] learn to localize simple color and texture attributes from loose annotations provided by image search. By contrast, we learn a broad set of complex attributes (shape, materials, parts) in a fully supervised manner and are concerned with generalization to new types of objects. We do not explicitly learn to localize attributes, but in some cases our feature selection method provides good localization as a side effect. Extensive work has been done in parts models for object recognition, but the emphasis is on localizing objects, usually with latent parts (e.g., [8, 20, 7]) learned for individual object categories. We differ from these approaches because of the explicit semantics of our attributes. We define explicit parts that can be shared across categories.Several researchers [2, 18, 12, 1, 22, 17] have shown that sharing features across multiple tasks or categories can lead to increased performance, especially when training data is limited. Our semantic attributes have a further advantage: they can be used to verbally describe new types of objects and to learn from textual description (without any visual examples).

3. Attributes and Features

We believe inferring attributes of objects is the key problem in recognition. These attributes can be semantic attributes like parts, shapes, and materials. Semantic attributes may not be always enough to distinguish all the categories of objects. For this reason we use discriminative attributes as well. These discriminative attributes take the form of comparisons borrowed from [5, 6],*"cats and dogs have it but sheep and horses don't"*.

Objects share attributes. Thus, by using predicted attributes as features, one can get a more compact and more discriminative feature space. Learning both semantic and discriminative attributes open doors to some new visual functions. We can not only recognize objects using predicted attributes as features, but also describe unfamiliar objects. Furthermore, these attribute classifiers can report

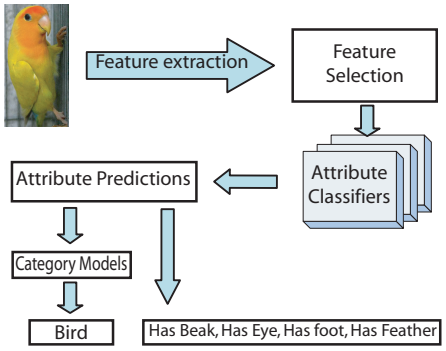

Figure 2: This figure summarizes our approach. We first extract base features. We then select features that are beneficial in learning attribute classifiers. We learn attribute classifiers using selected features. To learn object categories we use predicted attributes as features. Using attribute classifiers, we can do more than simple recognition. For instance, we can describe unknown classes, report atypical attributes, and learn new categories from very few examples.

the absence of typical attributes for objects, as well as presence of atypical attributes. Finally, we can learn models for new object classes using few examples. We can even learn new categories with *no* visual examples, using textual descriptions instead.

3.1. Base Features

The broad variety of attributes requires a feature representation to describe several visual aspects. We use color and texture, which are good for materials; visual words, which are useful for parts; and edges which are useful for shapes. We call these *base features*.

We use a bag of words style feature for each of these four feature types. Texture descriptors [23] are computed for each pixel, and quantized to the nearest 256 kmeans centers. The texture descriptor is extracted with a texton filterbank. Visual words are constructed with an HOG spatial pyramid, using 8x8 blocks, a 4 pixel step size, and 2 scales per octave. HOG descriptors are quantized to 1000 kmeans centers. Edges are found using a standard canny edge detector and their orientations are quantized into 8 unsigned bins. Finally, color descriptors are densely sampled for each pixel, and quantized to the nearest 128 kmeans centers. The color descriptor consists of the LAB values.

Having quantized these values, local texture, HOG, edge, and color descriptors inside the bounding box are binned into individual histograms. To represent shapes and locations, we also generate histograms for each feature type for each cell in a grid of three vertical and two horizontal blocks. This allows for coarse localization of attributes such as wheels which tend to appear at the bottom of the object. These seven histograms are stacked together resulting in a 9751 dimensional feature, which we refer to as the *base features*.

3.2. Semantic Attributes

We use three main types of semantic attribute. **Shape** attributes refer to 2D and 3D properties such as “is 2D boxy”, “is 3D boxy”,“is cylindrical“, etc. **Part** attributes identify parts that are *visible*, such as “has head”, “has leg”, “has arm”, “has wheel”, “has wing”, “has window”. **Material** attributes describe what an object is made of, including “has wood”, “is furry”, “has glass”, “is shiny”.

3.3. Discriminative Attributes

We do not yet have a comprehensive set of visual attributes. This means that, for example, instances of both cats and dogs can share all semantic attributes in our list. In fact, a Naive Bayes classifier trained on our ground truth attributes in Pascal can distinguish classes with only 74% accuracy. To solve this problem, we introduce auxiliary discriminative attributes. These new attributes take the form of random comparisons introduced in [6]. Each comparison splits a portion of the data into two partitions. We form these splits by randomly selecting one to five classes or attributes for each side. Instances not belonging to the selected classes or attributes are not considered. For example, a split would assign “cat” to one side and “dog” to the other side, while we don’t care where “motorbike” falls. Each split is further defined by a subset of base features, such as texture or color, to use for learning. For example, we might use texture to distinguish between “cats” and “dogs”. We then use a linear SVM to learn tens of thousands of these splits and pick those that can be well predicted using the validation data. In our implementation we used 1000 discriminative attributes.

4. Learning to Recognize Semantic Attributes

We want to accurately classify attributes for new types of objects. We also want our attribute classifiers to reflect the correct semantics of attributes. Simply learning classifiers by fitting them to all base features often fails to generalize the semantics of the attributes correctly (section 6.3).

4.1. Across Category Generalization by Within Category Prediction

Learning a “wheel” classifier on a dataset of cars, motorbikes, buses, and trains is difficult because all examples of wheels in this dataset are surrounded by “metallic” surfaces. The wheel classifier might learn “metallic” instead of “wheel”. If so, when we test it on a new dataset that happens to have wooden “carriage” examples, it will fail miserably, because there are not that many metallic surfaces around the wheel. What is happening is that the classifier is learning to predict a *correlated* attribute rather than the one we wish it to learn. This problem is made worse by using bounding boxes,instead of accurate segmentations. This is because some properties of nearby objects are likely to co-occur with object attributes. This behavior is not necessarily undesirable, but can cause significant problems if we must rely on the semantics of the attribute predictions. This is a major issue, because it results from training and testing on datasets with different correlation statistics, something we will always have to do because datasets will always be small compared to the complexity of the world.

Feature Selection: The standard strategy for dealing with generalization issues is to control variance by selecting a subset of features that can generalize well. However, conventional feature selection criteria will not apply to our problem because they are still confused by semantically irrelevant correlations — our "wheel" classifier does generalize well to cars, etc. (but not to carriages).

We use a novel feature selection criterion that decorrelates attribute predictions. Our criterion focuses on within category prediction ability. For example, if we want to learn a "wheel" classifier, we select features that perform well at distinguishing examples of cars with "wheels" and cars without "wheels". By doing so, we help the classifier avoid being confused about "metallic", as both types of example for this "wheel" classifier have "metallic" surfaces. We select the features using an L1-regularized logistic regression (because it assigns non-zero weights to a small subset of features [16]) trained for each attribute within each class, then pool examples over all classes and train using the selected features. For example, we first select features that are good at distinguishing cars with and without "wheel" by fitting an L1-regularized logistic regression to those examples. We then use the same procedure to select features that are good at separating motorbikes with and without wheels, buses with and without wheels, and trains with and without wheels. We then pool all those selected features and learn the "wheel" classifier over all classes using those selected features.

To test whether our feature selection decorrelate predicted attributes, we can look at changes in correlation across datasets. Throughout the paper we refer to features that we select by the procedure explained above as *selected features* and working with all features as *whole features*. For example, the correlation between ground-truth "wheel" and "metallic" in the a-Pascal dataset (section 5) is 0.71, and in the a-Yahoo dataset is 0.17. We train on the a-Pascal dataset with whole features and with selected features. In testing on the a-Yahoo dataset (section 5), the correlation between *predictions* by the "wheel" and "metallic" classifiers trained on whole features is 0.56 (i.e. predictions are biased to be correlated). When we do feature selection this correlation falls to 0.28, this shows that classifiers trained on selected features are less susceptible to biases in the dataset.

5. Datasets

We have built new datasets for exploring the object description problem. Our method for learning semantic attributes requires a ground truth labeling for each training example, but we must create our own, since no dataset exists with annotations for a wide variety of attributes which describe many object types. We collect our attribute annotations for each of twenty object classes in a standard object recognition dataset, PASCAL VOC 2008. We also collect the same annotations for a new set of images, called a-Yahoo. Labeling objects with their attributes can often be an ambiguous task. This can be demonstrated by imperfect inter-annotator agreement among "experts" (authors) and Amazon Turk annotators. The agreement among experts

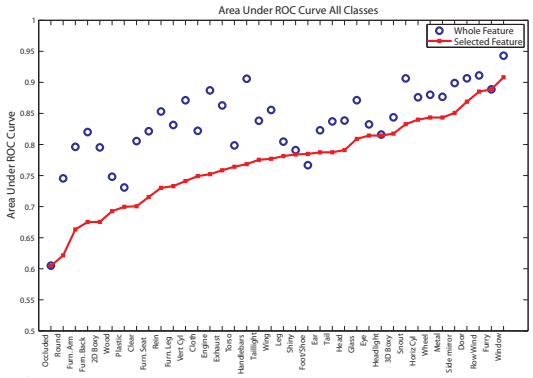

Figure 3: Attribute prediction for attribute classifiers trained on a-Pascal and tested on a-Pascal, comparing whole with selected features. We don't expect the feature selection to help in this case because we observe same classes during training and testing. This means that the correlation statistics are not changing during training and testing.

is 84.3%, between experts and Amazon Turk annotators is 81.4%, and among Amazon Turk annotators is 84.1%. Using Amazon Turk annotations, we are not biasing the attribute labels toward our own idea of attributes.

a-Pascal: The Pascal VOC 2008 dataset was created for classification and detection of visual object classes in variety of natural poses, viewpoints, and orientations. These objects classes cluster nicely, "animals", "vehicles", and "things". The object classes are: people, bird, cat, cow, dog, horse, sheep aeroplane, bicycle, boat, bus, car, motorbike, train, bottle, chair, dining table, potted plant, sofa, and tv/monitor. The number of objects from each category ranges from 150 to 1000, along with over 5000 instances of people. We collect annotations for semantic attributes for each object using Amazon's Mechanical Turk. We made a list of 64 attributes to describe Pascal objects. We do not claim to have attributes that exhaustively describe each class.

a-Yahoo: To supplement the *a-Pascal* dataset, we collect images for twelve additional object classes from the Yahoo image search, which we call the *a-Yahoo* set; these images are also labelled with attributes. The classes in a-Yahoo set are selected to have objects similar to a-Pascal, while having different correlations between the attributes selected on a-Pascal. For example, compare a-Yahoo's "wolf" category to a-Pascal's "dog"; a-Yahoo's "centaur" to a-Pascal's "people" and "horses". This allows us to evaluate the attribute predictors' generalization abilities. Objects in this set are: wolf, zebra, goat, donkey, monkey, statue of people, centaur, bag, building, jet ski, carriage, and mug.

These datasets are available at http://vision.cs.uiuc.edu/attributes/.

6. Experiments and Results

First we show how well we can assign attributes and use them to describe objects. We then examine the performance of using the attribute based representation in the traditional naming task and demonstrate new capabilities offered by this representation: learning from very few visual examples and learning from pure textual description. Finally we show

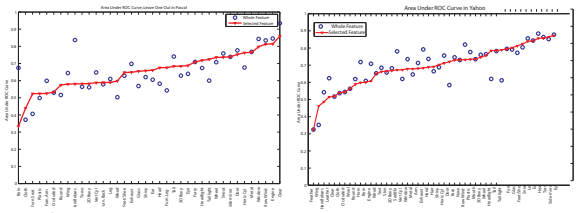

Figure 4: Attribute prediction for across category protocols. On the left is Leave-one-class-out case for Pascal and on the right is attribute prediction for Yahoo set. Only attributes relevant to these tasks are displayed. Classes are different during training and testing, thus we have across category generalization issues. Some attributes on the left, like "engine", "snout", and "furry", generalize well, some do not. Feature selection helps considerably for those attributes, like "taillight", "cloth", and "rein" that have problem generalizing across classes. Similar to leave one class out case, learning attributes on Pascal08 train set and testing them on Yahoo set involves across category generalization, right plot. We can, in fact, predict attributes for new classes fairly reliably. Some attributes, like "wing", "door", "headlight", and "taillight", do not generalize well. Feature selection improves generalization on those attributes. Toward the high end of this curve, where good classifiers sit, feature selection improves prediction of attribute with generalization issues and produce similar results for attributes without generalization issues. For better visualization purposes we sorted the plots based on selected features' area under ROC curve values.

benefits of our novel feature selection method compared to using whole features.

6.1. Describing Objects

Assigning attributes: There are two main protocols for attribute prediction: "within category" predictions, where train and test instances are drawn from the same set of classes, and "across category" predictions where train and test instances are drawn from different sets of classes. We do across category experiments using a leave-one-class-out approach, or a new set of classes on a new dataset. We train attributes in a-Pascal and test them in a-Yahoo. We measure our performance in attribute predictions by the area under the ROC curve, mainly because it is invariant to class priors. We can predict attributes for the *within category* protocol with the area under the curve of 0.834 (Figure 3).

Figure 4 shows that we can predict attributes fairly reliably for *across category* protocols. The plot on the left shows the leave-one-class-out case on a-Pascal and the plot on the right shows the same curve for a-Yahoo set.

Figure 5 depicts 12 typical images from a-Yahoo set with a subset of positively predicted attributes. These attribute classifiers are learned on a-Pascal train set and tested on a-Yahoo images. Attributes written in red, with red crosses, are wrong predictions.

Unusual attributes: People tend to make statements about unexpected aspects of known objects ([11], p101). An advantage of an attribute based representation is we can easily reproduce this behavior. The ground truth attributes specify which attributes are typical for each class. If a reliable attribute classifier predicts one of these typical attributes is absent, we report that it is not visible in the image. Figure 6 shows some of these typical attributes which are not visible in the image. For example, it is worth reporting when we do not see the "wing" an aeroplane is expected to have. To qualitatively evaluate this task we reported 752 expected attributes over the whole dataset which are not visible in the images. 68.2% of these reports are correct when compared to our manual labeling of those reports (Figure 6). On the other hand, if a reliable attribute classifier predicts an attribute which is not expected to be in the predicted class, we can report that, too (Figure 7). For example, birds don't have a "leaf", and if we see one we should report it. To quantitatively evaluate this prediction we evaluate 951 of those predictions by hand; 47.3% are correct. There are two important consequences. First, because birds never have leaves, we may be able to exploit knowledge of object semantics to reason that, in this case, the bird is in a tree. Second, because we can localize features used to predict attributes, we can show what caused the unexpected attribute to be predicted (Figure 8). For example, we can sometimes tell where the "metal" is in a pic-

Figure 5: This figure shows randomly selected positively predicted attributes for 12 typical images from 12 categories in Yahoo set. Attribute classifiers are learned on Pascal train set and tested on Yahoo set. We randomly select 5 predicted attributes from the list of 64 attributes available in the dataset. Bounding boxes around the objects are provided by the dataset and we are only looking inside the bounding boxes to predict attributes. Wrong predictions are written in red and marked with red crosses.

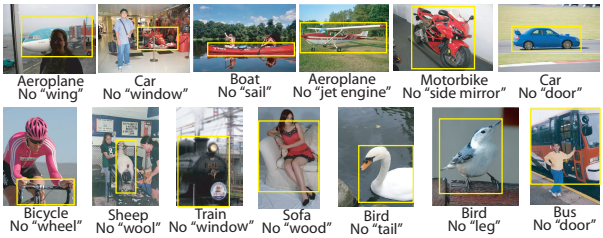

Figure 6: Reporting the absence of typical attributes. For example, we expect to see "Wing"in an aeroplane. It is worth reporting if we see a picture of an aeroplane for which the wing is not visible or a picture of a bird for which the tail is not visible.

Figure 7: Reporting the presence of atypical attributes. For example, we don't expect to observe "skin" on a dining table. Notice that, if we have access to information about object semantics, observing "leaf" in an image of a bird might eventually yield "The bird is in a tree". Sometimes our attribute classifiers are confused by some misleading visual similarities, like predicting "Horn" from the visually similar handle bar of a road bike.

Figure 8: This figure shows localization of atypical attributes for given classes. Not only do we report unexpected attributes, but we also can sometimes localize atypical attributes in images. For example, we don't expect to see "skin" in a motorbike, but when we do, we can localize the skin reasonably well. Red colored points correspond to selected features. This figure should be viewed in color.

PASCAL 08	Base Features	Whole Features		Selected Features	
		Sem. Attr.	All Attr.	Sem. Attr.	All Attr.
SVM	58.5 (35.5)	56.1 (34.3)	58.3 (38.1)	54.6 (28.4)	**59.4** (37.7)
Logistic Regression	54.6 (36.9)	51.2 (31.4)	53.4 (33.5)	51.8 (32.3)	53.5 (35.1)

Table 1: This table compares our accuracies in traditional naming task with two simple baselines. Since the Pascal08 dataset is heavily biased toward "people" category, we report both overall and mean per class accuracies. Mean per class accuracies appear in the parentheses. This table also compares using attributes trained on *selected features* with those trained on *whole features*. Columns marked as "All Attr." refer to the cases when classifiers use both predicted semantic and non-semantic attributes as features. Note that the attribute based representation does not help significantly in the traditional naming task but it offers new capabilities,Figure 9.

ture that has people and "metal."

6.2. Naming

Naming familiar objects: So far there is little evidence that our attribute based framework helps the traditional naming task. However, this framework allows us to learn new categories from very few visual examples or even with pure textual description. We compare our performance in naming task with two baselines, linear SVM and logistic regression applied to base features to directly recognize objects. We use the Pascal training set as our train/val set and use the Pascal validation set as our test set. Table 1 shows details of this experiment. A one vs. all linear SVM can recognize objects with the overall accuracy of 59.4% using our predicted attributes as features, comparing to the accuracy of 58.5% of base features. Because we assume that bounding boxes are provided, we can not directly compare our results with other methods in the literature. It is also worth noting differences between class confusions using our attribute based features and standard recognition methods. The biggest increase in confusions using our attribute based representation is between "chair" and "sofa". The biggest decrease is between "bike" and "people". The shifts in the confusions may be due to our encoding of semantics.

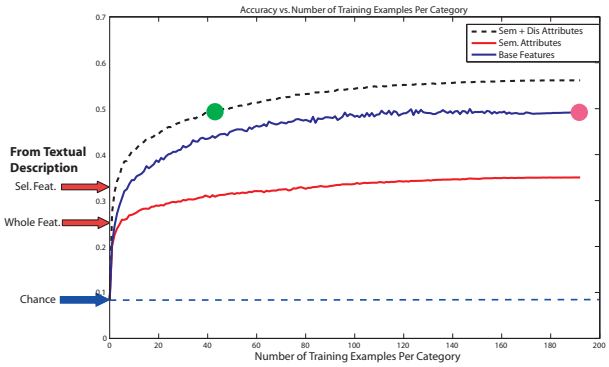

Figure 9: Accuracy vs. number of training examples per category. We can learn categories with considerably fewer examples. Using 4 examples per class with 1NN classifier, which is our only choice for so few examples, we can predict as well as with 20 examples per class using base features. If we use almost 200 examples per category (purple circles) on original features we are as good as using 40 examples (green circles) using our attributes. Note that the semantic attributes are not designed to maximize discrimination. Discriminative attributes provide similar performance when used with or without semantic attributes. Semantic attributes help us to achieve enhanced visual capabilities without any loss in discrimination. Another interesting point about our attribute base description is that we can recognize objects from pure textual description and NO visual examples. As depicted above, recognizing objects from textual description (red arrows) is as good as having almost 100 visual examples in semantic attribute space, 8 visual examples in base features and 3 in semantic and discriminative attribute space. Red arrows indicate the accuracy of learning new classes by textual description using whole and selected features.

Learning to Identify New Objects: The first test is to examine standard object recognition in new categories. We use predicted attributes as features and one-vs-all linear SVM as classifier. If we recognize classes in a-Yahoo set using attribute classifiers trained on a-Pascal, we get an overall accuracy of 69.8%. If we train attributes on a-Yahoo as well, we get an overall accuracy of 74.7%, comparing to 72.7% using base features.

We can also recognize new classes with notably fewer training examples than classifiers trained on base features (Figure 9). We choose a 1NN classifier for this task, mainly because we need to learn from very few examples per category. As plotted in this figure we can learn new categories using under 20% of the examples required by base features. This means that the overall accuracy of training on almost 40 images per category (green circles) using our attributes is equal to that of training on almost 200 images per category (purple circles) on base features.

Learning New Categories from Textual Description: A novel aspect of our approach is to learn new categories from pure textual descriptions. For example, we can learn new categories by describing new classes to our algorithm as this new class is "furry", "four legged", "has snout",and "has head". The object description is specified by a list of positive attributes, providing a binary attribute vector. We classify a test image by finding the nearest description to its predicted attributes. Figure 9 shows that by learning new categories from textual description we could get an accuracy of 32.5%, which is equal to having almost 100 visual examples in semantic attribute space, 8 visual examples in base feature space, and 3 examples in semantic and discrim-

Figure 10: Feature selection is necessary to localize attributes. For example, if we want to learn a "Hair" classifier we might end up learning a skin detector instead. This figure compares localization of attributes based on classifiers learned on selected features with those trained on whole features. Colored points are features with high positive response for attribute classifiers. This implies that by using whole features we may not obtain classifiers with the semantics we expect. For more results on localization of attributes using selected features see Figure 11

inative attribute space.

Rejection: When presented with an object from a new category, we want our model to recognize that it is doesn't belong to any known category. For example, object models trained on a-Pascal should all reject a category like "book" as unknown. The a-Yahoo set is an extremely challenging dataset in rejection tasks for object models trained on a-Pascal (one has "wolf", the other "dog", and so on). If we reject using confidences of one-vs.-all SVM's used to learn a-Pascal object models, we get chance performance (the area under the ROC curve for this rejection task using base features is 0.5). However, by using attributes we reject significantly better, with an AUC of 0.6.

6.3. Across Category generalization

Three tasks demand excellent across-category generalization: learning from very few examples; learning from textual descriptions; and reporting unusual attributes. For example, in learning a car category from very few example, the "wheel" classifier has to mean "wheel" when it fires, rather than predicting "wheel" and meaning "metallic". Experiments below show that selected features have significantly improved performance on these tasks compared to whole features. Furthermore, they allow us to localize attributes, and predict correlations accurately.

Semantics of Learned Attributes: Any task that relies strongly on the semantics of learned attributes seems to benefit from using selected features. For example, selecting features improves the results in learning from textual description from 25.2% to 32.5%, in reporting the absence of typical attributes from 54.8% to 68.2%, and in reporting the presence of atypical attributes from 24.5% to 47.3%.

Localization: Selected features can produce better reports of location for the relevant attribute, something whole features cannot do. Figure 10 compares localizations of three different attributes using selected and whole features.

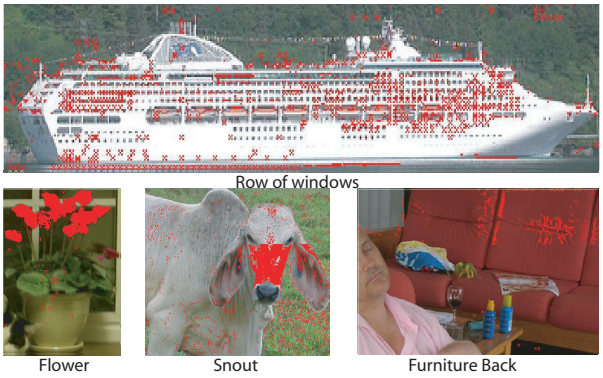

Figure 11: Attribute localization using selected features.

As expected, classifiers trained on whole features tend to pick correlated features rather than features related directly to the attributes. Instead, selected features can roughly localize attributes in some cases. Figure 11 shows more examples of localizations of attributes using selected features. These figures suggest that learning classifiers on whole features may result in classifiers that are confused about semantics.

Correlation: Attribute classifiers learned using whole features are biased to be correlated in the way the training set is correlated. This means that, when applied to a test set that has different statistics, the predictions of correlation do not agree with ground truth. Figure 12 shows histograms of differences between the correlation in predicted attributes and the correlation in ground truth attributes using both whole and selected features on a-Pascal and a-Yahoo images.

7. Discussion

Suppose we train a system to recognizes horses and people by mapping directly from image features to categories. If our system is then faced with a centaur, it will be completely clueless. To make a sensible report under these circumstances, the object representation must be in terms that are useful to describe many objects, even ones that do not appear in the training set. Attributes are the natural candidate. If we make attributes the central representation for object recognition, we are able to say more about an object than just its name. We can say how it is different from the usual member of its category (for example, noticing that a bicycle looks like it has horns, figure 7). Even if we don't happen to have a model of an object, we can make useful statements about it when we see it. We can build models using descriptions. For instance, we can recognize a goat based on the description "four-legged, has face, has horns, has fur". This means that we could learn by reading. To the best of our knowledge, we are the first in computer vision to provide these abilities to describe objects and learn from description. We expect further investigation of attribute-based models in object recognition to be very fruitful. For example, [10], which appears in the same proceedings, proposes an interesting application of attribute based representations for recognizing new categories of animals.

Cross-category generalization is essential to these visual functions, because they rely on the semantics of the attribute report being correct. The area has received little attention, but an improved understanding of cross-category generalization is essential to sustained progress in object recognition. To deal with novel objects, we must be confident we have semantically accurate reports of object properties in an image — e.g., we must know that "wheel" means "wheel", not some correlated property like "metal.

We have proposed one method to achieve cross-category generalization: select features that can predict the attribute within each class. This helps to decorrelate the attributes and leads to much improved learning by reading and attribute localization.

Another reason to understand cross-category generalization better is that correlation between target and other concepts causes widespread problems in the object recognition community. For instance, it is still difficult to tell whether pedestrian detectors perform well because pedestrian data sets are special, or because we are now excellent at detecting people. Evidence that we are excellent at detecting people would be a person detector trained on the INRIA dataset [3] that works well on the PASCAL-08 [4]. So far, such a detector is conspicuously absent; most current object detectors work well only when the training and test sets are very similar. Our work hints such detectors are likely learning as much about dataset biases as about the objects themselves. To distinguish between these phenomena, we should most likely devise tasks that, while not explicitly trained, can be accomplished if the target concept is well-learned. We have provided two examples – cross-dataset evaluation on new objects and looking at localization without training to localize – but there are likely many other such tasks.

8. Acknowledgments

This work was supported in part by the National Science Foundation under *IIS* – 0534837 and 0803603 and in part by the Office of Naval Research under *N*00014 – 01 – 1 – 0890 as part of the MURI program. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the NSF or the ONR. Derek Hoiem was supported by a Beckman fellowship. Ian Endres was supported by the Richard T.Cheng fellowship. The authors would like to thank Alexander Sorokin for insightful discussions on Mechanical Turk.

References

[1] Rie Kubota Ando and Tong Zhang. A high-performance semi-supervised learning method for text chunking. In *ACL05*, 2005. 2

[2] Rich Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997. 2

[3] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 8

[4] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. 2, 8

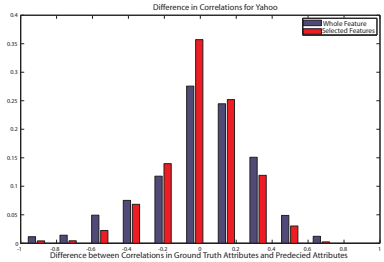

Figure 12: This histogram compares the differences between correlations in ground truth annotation and predicted attributes using selected and whole features in a-Yahoo. Using Whole features introduces dataset dependent correlations. Feature selection helps to reduce this correlation.

[5] A. Farhadi and Kamali. Learning to recognize activities from the wrong view point. In *ECCV*, 2008. 2

[6] Ali Farhadi, David A. Forsyth, and Ryan White. Transfer learning in sign language. In *CVPR*, 2007. 2, 3

[7] P. Felzenszwalb, D. Mcallester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008. 2

[8] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR03*, pages II: 264–271, 2003. 2

[9] V. Ferrari and A. Zisserman. Learning visual attributes. In *NIPS*, 2007. 2

[10] Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009. 7

[11] S.C. Levinson. *Pragmatics*. CUP, 1983. 5

[12] Nicolas Loeff, Ali Farhadi, and David Forsyth. Scene discovery by matrix factorization. Technical Report No. UIUCDCS-R-2008-2928, 2008. 2

[13] S.M. Kosslyn (Eds.) M.A. Gluck, J.R. Anderson. *Category Learning: Learning to Access and Use Relevant Knowledge, Memory and Mind, A Festschrift for Gordon H. Bower (Chapter 15 - pp. 229-246)*. MIT Press, 2008. 2

[14] Tomasz Malisiewicz and Alexei A. Efros. Recognition by association via learning per-exemplar distances. In *CVPR*, 2008. 2

[15] G.L. Murphy. *The big book of concepts*. MIT Press, 2004. 2

[16] Andrew Y. Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *ICML*, 2004. 4

[17] Andreas Opelt, Axel Pinz, and Andrew Zisserman. Learning an alphabet of shape and appearance for multi-class object detection. *International Journal of Computer Vision*, 80:16–44, 2008. 2

[18] Ariadna Quattoni, Micheal Collins, and trevor darrell. Learning visual representations using images with captions. In *Proc. CVPR 2007*, 2007. 2

[19] Eleanor Rosch, C. B. Mervis, W. D. Gray, D. M. Johnson, and Boyes P. Braem. Basic objects in natural categories. *Cognitive Psychology*, 8:382–439, 1976. 2

[20] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, volume 2, pages 1470–1477, 2003. 2

[21] A. Sorokin and D.A. Forsyth. Utility data annotation with amazon mechanical turk. In *CVPR Workshop on Internet Vision*, 2008. 2

[22] Antonio Torralba, Kevin P. Murphy, and William T. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In *CVPR*. 2004. 2

[23] Manik Varma and Andrew Zisserman. A statistical approach to texture classification from single images. *Int. J. Comput. Vision*, 62:61–81, 2005. 3

[24] Julia Vogel and Bernt Schiele. Natural scene retrieval based on a semantic modeling step. In *CIVR*, pages 207–215, 2004. 2