To appear, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012.

Discovering Localized Attributes for Fine-grained Recognition

Kun Duan
Indiana University
Bloomington, IN
kduan@indiana.edu

Devi Parikh
TTI-Chicago
Chicago, IL
dparikh@ttic.edu

David Crandall
Indiana University
Bloomington, IN
djcran@indiana.edu

Kristen Grauman
University of Texas
Austin, TX
grauman@cs.utexas.edu

Abstract

*Attributes are visual concepts that can be detected by machines, understood by humans, and shared across categories. They are particularly useful for fine-grained domains where categories are closely related to one other (*e.g. *bird species recognition). In such scenarios, relevant attributes are often local (*e.g. *"white belly"), but the question of how to choose these local attributes remains largely unexplored. In this paper, we propose an interactive approach that discovers local attributes that are both discriminative and semantically meaningful from image datasets annotated only with fine-grained category labels and object bounding boxes. Our approach uses a latent conditional random field model to discover candidate attributes that are detectable and discriminative, and then employs a recommender system that selects attributes likely to be semantically meaningful. Human interaction is used to provide semantic names for the discovered attributes. We demonstrate our method on two challenging datasets, Caltech-UCSD Birds-200-2011 and Leeds Butterflies, and find that our discovered attributes outperform those generated by traditional approaches.*

1. Introduction

Most image classification and object recognition approaches learn statistical models of low-level visual features like SIFT and HOG. While these approaches give state-of-the-art results in many settings, such low-level features and statistical classification models are meaningless to humans, thus limiting the ability of humans to understand object models or to easily contribute domain knowledge to recognition systems. Recent work has introduced *visual attributes* as intermediate-level features that are both machine-detectable and semantically meaningful (e.g. [2, 3, 6, 9, 11, 14, 27]). Attributes help to expose the details of an object model in a way that is accessible to humans: in bird species recognition, for example, they can explicitly model that a cardinal has a "red-orange beak," "red body," "sharp crown," "black face," etc. Attributes are particularly attractive for fine-grained domains like animal species where the categories are closely related, so that a common attribute vocabulary exists across categories. Attributes also enable innovative applications like zero-shot learning [15, 18] and image-to-text generation [6, 18].

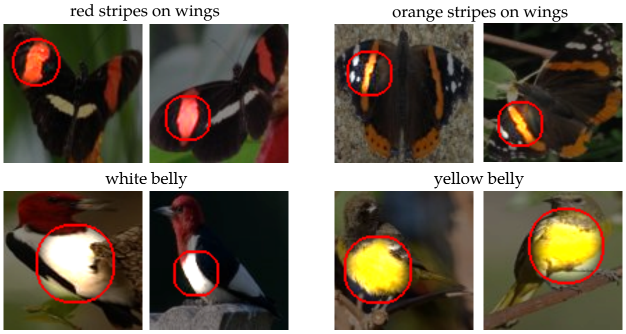

Figure 1. Sample local and semantically meaningful attributes automatically discovered by our approach. The names of the attributes are provided by the user-in-the-loop.

So where do these attributes come from? Most existing work uses hand-generated sets of attributes (*e.g.* [3, 14]), but creating these vocabularies is time-consuming and often requires a domain expert (*e.g.* an ornithologist familiar with the salient parts of a bird). Moreover, while these attributes are guaranteed to be human-understandable (which suffices for human-in-the-loop classification applications [3]), they may not be machine-detectable and hence may not work well in automatic systems. Some recent work has discovered image-level attributes (*e.g.* "outdoors" or "urban") automatically [17], but such global attributes are of limited use for fine-grained object classification in which subtle differences between object appearances are important.

Discovering *local* attributes (like those illustrated in Figure 1) is significantly harder because a local attribute might correspond to features at different unknown positions and scales across images. Automatic techniques to do this have generally either found attributes that are discriminative or that are meaningful to humans, but not both. Finding discriminative local regions (e.g. [27]) works well for attaining good image classification performance, but the regions may not be semantically meaningful and thus not useful for applications like zero-shot learning and automatic image description. On the other hand, mining text can produce attribute vocabularies that are meaningful (*e.g.* [2]) but not necessarily complete, discriminative, or detectable.

In this paper, we propose an interactive system that discovers discriminative local attributes that are both machine-detectable and human-understandable from an image dataset annotated with fine-grained category labels and object bounding boxes. At each iteration in the discovery process, we identify two categories that are most confusable given the attributes that have been discovered so far; we call these two categories an *active split*. We use a latent conditional random field model to automatically discover candidate local attributes that separate these two classes. For these candidates, we use a recommender system to identify those that are likely to be semantically meaningful to a human, and then present them to a human user to collect attribute names. Candidates for which the user can give a name are added to the pool of attributes, while unnamed ones are ignored. In either case, the recommender system's model of semantic meaningfulness is updated using the user's response. Once the discovery process has built a vocabulary of local attributes, these attributes are detected in new images and used for classification.

To the best of our knowledge, ours is the first system to discover vocabularies of local attributes that are both machine-detectable and human-understandable, and that yield good discriminative power on fine-grained recognition tasks. We demonstrate our approach through systematic experiments on two challenging datasets: Caltech-UCSD Birds-200-2011 [22] and Leeds Butterflies [24]. We find on these datasets that our discovered local attributes outperform those generated by human experts and by other strong baselines, on fine-grained image classification tasks.

2. Related work

Visual attributes for classification and recognition have received significant attention over the last few years. Much of this work assumes that the attribute vocabulary is defined ahead of time by a human expert [3, 14, 15, 25]. An exception is the work of Parikh and Grauman [17] which proposes a system that discovers the vocabulary of attributes. Their system iteratively selects discriminative hyperplanes between two sets of images (corresponding to two different subsets of image classes) using global image features (e.g. color, GIST); it would be difficult to apply this approach to find local attributes because of the exponential number of possible local regions in each image.

A few papers have studied how to discover local attributes. Berg et al. [2] identify attributes by mining text and images from the web. Their approach is able to localize attributes and rank them based on visual characteristics, but these attributes are not necessarily discriminative and thus may not perform well for image classification; they also require a corpus of text and images, while we just need images. Wang and Forsyth [23] present a multiple instance learning framework for both local attributes and object classes, but they assume attribute labels for each image are given. In contrast, our approach does not require attribute labels; we discover these attributes automatically.

Related to our work on local attribute selection is the extensive literature on learning part-based object models for recognition (e.g. [7, 8, 26]). These learning techniques usually look for highly distinctive parts – regions that are common within an object category but rare outside of it – and they make no attempt to ensure that the parts of the model actually correspond to meaningful semantic parts of an object. Local attribute discovery is similar in that we too seek distinctive image regions, but we would like these regions to be shared across categories and to have semantic meaning. Note that while most semantically meaningful local attributes are likely to correspond to semantic parts of objects, we view attributes as more general: an attribute is potentially any visual property that humans can precisely communicate or understand, even if it does not correspond to a traditionally-defined object part. For example "red-dot in center of wings" is a valid local attribute, even though there is not a single butterfly part that corresponds to it.

Finally, our work is also related to the literature on automatic object discovery and unsupervised learning of object models [4, 12, 16]. However, these methods aim to find objects that are *common* across images, while we are interested in finding *discriminative* local regions that will maximize classification performance.

3. Approach

We first consider the problem of finding discriminative and machine-detectable visual attributes in a set of training images. We then describe a recommender system that finds candidates that are likely to be human-understandable and presents them to users for human verification and naming.

3.1. Latent CRF model formulation

We assume that each image in the training set has been annotated with a class label (e.g. species of bird) and object bounding box similar to [24, 27],[1] but that the set of possible attributes and the attribute labels for each image are unknown. We run a hierarchical segmentation algorithm on the images to produce regions at different scales, and assume that any attribute of interest corresponds to *at most* one region in each image. This assumption reduces the

[1] [24] in fact requires the user to interactively segment the object out.

computational complexity and is reasonable because the hierarchical segmentation gives regions at many scales.

Formally, we are given a set of annotated training images $\mathcal{I} = (\mathcal{I}_1, \ldots, \mathcal{I}_M)$, with each exemplar $\mathcal{I}_i = (I_i, y_i)$ consisting of an image I_i and a corresponding class label y_i. For now we assume a binary class label, $y_i \in \{+1, -1\}$; we will generalize this to multiple classes in Section 3.4. Each image I_i consists of a set of overlapping multi-scale regions produced by the hierarchical segmentation. To find a discriminative local attribute for these images, we look for regions in positive images, one per image, that are similar to one another (in terms of appearance, scale and location) but not similar to regions in negative images. We formulate this task as an inference problem on a latent conditional random field (L-CRF) [19], the parameters of which we learn via a discriminative max-margin framework in the next section.

First consider finding a single attribute k for the training set $\mathcal{I}$. For each image we want to select a single region $l_i^k \in I_i$ such that the selected regions in the positive images have similar appearances to one another, but are different from those on the negative side. We denote the labeling for the entire training set as $L_k = (l_1^k, \ldots, l_M^k)$, and then formulate this task in terms of minimizing an energy function [5],

$$E(L_k|\mathcal{I}) = \sum_{i=1}^{M} \phi_k(l_i^k|\mathcal{I}_i) + \sum_{i=1}^{M}\sum_{j=1}^{M} \psi_k(l_i^k, l_j^k|\mathcal{I}_i, \mathcal{I}_j), \quad (1)$$

where $\phi_k(l_i^k|\mathcal{I}_i)$ measures the preference of a discriminative classifier trained on the selected regions to predict the category labels, while $\psi_k(l_i^k, l_j^k|\mathcal{I}_i, \mathcal{I}_j)$ measures pairwise similarities and differences between the selected regions. In particular, we define the unary term as,

$$\phi_k(l_i^k|\mathcal{I}_i) = -y_i w_k^T \cdot f(l_i^k) \quad (2)$$

where $f(l_i^k)$ denotes a vector of visual features for region l_i^k and w_k is a weight vector. We will use several different types of visual features, as discussed in Section 4; for now we just assume that there are d feature types that are concatenated together into a single n-dimensional vector. The weights are learned as an SVM on the latent regions from positive and negative images (discussed in Section 3.2).

The pairwise consistency term is given by,

$$\psi_k(l_i^k, l_j^k|\mathcal{I}_i, \mathcal{I}_j) = \begin{cases} \vec{\alpha}_k^+ \cdot D(f(l_i^k), f(l_j^k)) + \beta_k^+ & \text{if } y_i, y_j = +1 \\ \beta_k^0 & \text{if } y_i, y_j = -1 \\ \vec{\alpha}_k^- \cdot D(f(l_i^k), f(l_j^k)) + \beta_k^- & \text{otherwise,} \end{cases} \quad (3)$$

where $D(\cdot, \cdot)$ is a function $\mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^d$ that given two feature vectors computes a distance for each feature type, $\vec{\alpha}_k^-$ and $\vec{\alpha}_k^+$ are weight vectors, and $\vec{\beta}_k = (\beta_k^-, \beta_k^+, \beta_k^0)$ are constant bias terms (all learned in Section 3.2). This pairwise energy function encourages similarity among regions in positive images and dissimilarity between positive and negative regions. We allow negative regions to be different from one another since they serve only as negative exemplars; thus we use a constant β_k^0 as the edge potential between negative images in lieu of a similarity constraint.

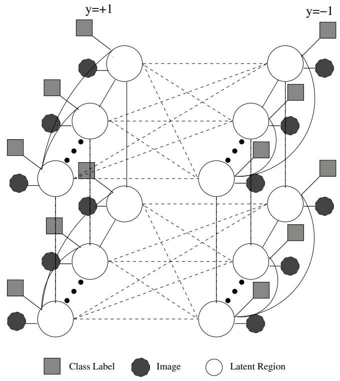

Figure 2. Our L-CRF model for one active split with $K = 2$ attributes, where white circles represent latent region variables (l_i^k), shaded circles represent observed image features (I_i), and squares represent observed image class labels (y_i).

The energy function presented in equation (1) defines a first-order Conditional Random Field (CRF) graphical model. Each vertex of the model corresponds to an image, and the inference problem involves choosing one of the regions of each image to be part of the attribute. Edges between nodes reflect pairwise constraints across images, where here we use a fully-connected graphical model such that there is a constraint between every image pair.

The single attribute candidate identified by the L-CRF may not necessarily be semantically meaningful, but there may be other candidates that can discriminate between the two categories that are semantically meaningful. To increase the chances of finding these, we wish to identify multiple candidate attributes. We generalize the above approach to select $K \geq 2$ attributes for a given split by introducing an energy function that sums equation (1) over all K attributes. We encourage the CRF to find a set of *diverse* attributes by adding an additional term that discourages spatial overlap among selected regions,

$$E(\mathcal{L}|\mathcal{I}) = \sum_{k=1}^{K} E(L_k|\mathcal{I}) + \sum_{i=1}^{M}\sum_{k,k'} \delta(l_i^k, l_i^{k'}|\mathcal{I}_i), \quad (4)$$

where $\mathcal{L} = (L_1, \ldots, L_K)$ denotes the latent region variables, δ measures spatial overlap between two regions,

$$\delta(l_i^k, l_i^{k'}|\mathcal{I}_i) = \sigma \cdot \frac{area(l_i^k \cap l_i^{k'})}{area(l_i^k \cup l_i^{k'})}, \quad (5)$$

and $\sigma \geq 0$ is a scalar which is also learned in the next section. This term is needed because we want a diverse set of candidates; without this constraint, the CRF may find a

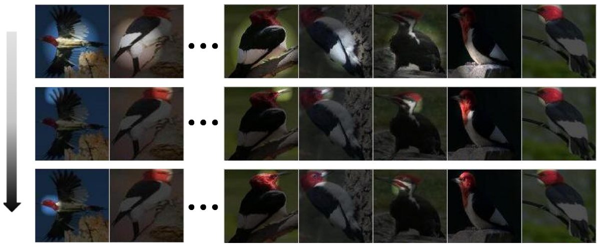

Figure 3. Sample latent region evolution on an active split, across three iterations (top to bottom). The latent region selected by the CRF on each positive image in each iteration is shown. These variables converged after three iterations to roughly correspond to the bird's red head. Best viewed on-screen and in color.

set of very similar candidates because those are most discriminative. Intuitively, $\delta(\cdot)$ penalizes the total amount of overlap between regions selected as attributes. Minimizing the energy in equation (4) also corresponds to an inference problem on a CRF; one can visualize the CRF as a three-dimensional graph with K layers, each corresponding to a single attribute, with the edges in each layer enforcing the pairwise consistency constraints ψ among training images and the edges between layers enforcing the anti-overlap constraints δ. The vertices within each layer form a fully-connected subgraph, as do the vertices across layers corresponding to the same image. Figure 2 illustrates the CRF for the case of two attributes.

3.2. Training

There are two sets of model parameters that must be learned: the weight vectors w_k in the unary potential of the CRF, and the parameters of the pairwise potentials ψ and δ which we can concatenate into a single vector $\vec{\alpha}$,

$$\vec{\alpha} = (\vec{\alpha}_1^-, \ldots, \vec{\alpha}_K^-, \vec{\alpha}_1^+, \ldots, \vec{\alpha}_K^+, \vec{\beta}_1, \ldots, \vec{\beta}_K, \sigma).$$

We could easily learn these parameters if we knew the correct values for the latent variables $\mathcal{L}$, and we could perform CRF inference to estimate the values of the latent variables if we knew the parameters. To solve for both, we take an iterative approach in the style of Expectation-Maximization. We initialize the latent variables $\mathcal{L}$ to random values. We then estimate w_k in equation (2) for each k by learning a linear SVM on the regions in L_k, using regions in positive images as positive exemplars and regions in negative images as negative exemplars. Holding w_k fixed, we then estimate the pairwise parameters $\vec{\alpha}$ via a standard latent structural SVM (LSSVM) framework,

$$\min_{\vec{\alpha}} \lambda\|\vec{\alpha}\|^2 + \xi, \text{ such that } \forall \tilde{l_i} \in I_i, \forall \tilde{y_i} \in \{+1,-1\}, \quad (6)$$

$$E(\{\tilde{l_i}\}|\{(I_i, \tilde{y_i})\}) - \min_{\mathcal{L}^*} E(\mathcal{L}^*|\mathcal{I}) \geq \Delta(\{\tilde{y_i}\}, \{y_i\}) - \xi$$

where $\xi \geq 0$ is a slack variable and the loss function is defined as the number of mislabeled images,

$$\Delta(\{\tilde{y_i}\}, \{y_i\}) = \sum_i \mathbb{1}_{\tilde{y_i} \neq y_i}.$$

We solve this quadratic programming problem using CVX [10]. Since there are an exponential number of constraints in equation (6), we follow existing work on structured SVMs [21] and find the most violated constraints, in this case using tree-reweighted message passing (TRW) [13] on the CRF. Once the CRF parameters have been learned, we hold them fixed and estimate new values for the latent variables $\mathcal{L}$ by performing inference using TRW. This process of alternating between estimating CRF parameters and latent variable values usually takes 3 to 5 iterations to converge (Figure 3). In our experiments we use $K = 5$. This takes about 3-5 minutes on a 3.0GHz server.

The above formulation was inspired by Multiple Instance CRFs [4, 5], but with some important differences (besides application domain). Our formulation is a standard latent structural SVM in which we minimize classification error, whereas the loss function in [5] is based on incorrect instance selections. Their unary potential is pre-trained instead of being updated iteratively. Finally, our model simultaneously discovers multiple discriminative candidate attributes (instances).

3.3. Attribute detection

To detect attributes in a new image I_t, we simply add I_t to the L-CRF as an additional node, fixing the values of the latent variables for the training image nodes. We perform CRF inference on this new graph to estimate both the class label $\hat{y}_t$ and its corresponding region label $\hat{l}_t \in I_t$. If $\hat{y}_t = 1$, then we report a successful detection and return $\hat{l}_t$, and otherwise report that I_t does not have this attribute. Note that this inference is exact and can be done in linear time.

3.4. Active attribute discovery

Having shown how to automatically discover attributes for images labeled with one of two classes (positive or negative), we now describe how to discover attributes in a dataset with multiple category labels, $y_i \in \{1, \ldots, N\}$. We would like to discover an attribute vocabulary that collectively discriminates well among all categories. It is intractable to consider all $O(N^2)$ possible binary splits of the labels, so we use an iterative approach with a greedy heuristic to try to actively prioritize the order in which splits are considered. At each iteration, we identify the two categories that are most similar in terms of the presence and absence of attributes discovered so far. We use these two categories to define an active split, and find a set of discriminative attributes for this split using the procedure above. We then add these to our attribute set, and repeat the process.

3.5. Identifying semantic attributes

The approach we described in previous sections is able to discover K candidate discriminative local attributes for each active split, but not all of these will be meaningful at a semantic level. We now describe how to introduce a minimal amount of human feedback at each iteration of the discovery process in order to identify candidates that are discriminative *and* meaningful. Of the K candidates, we first identify the candidate that is most discriminative – *i.e.* that increases the performance of a nearest neighbor classifier the most on held out validation data. We present this candidate to a human user by displaying a subset of the positive training images from the corresponding active split marked with the hypothesized attribute regions determined by the L-CRF (see Figure 8). If the user finds the candidate meaningful (and thus provides it with a name), it is added to our vocabulary of attributes. If not, that candidate is rejected, and we select the second most discriminative candidate in our pool of K candidates. If none of the candidates is judged to be meaningful, no attribute is added to our pool, and we identify the second most confusing pair of categories as our next active split.

In order to reduce user response time we propose an attribute recommender system that automatically prioritizes candidates before presenting them to a user. It uses past user feedback to predict whether the user is likely to find a new candidate attribute meaningful. Our recommender system is based on the hypothesis that users judge the meaningfulness of an attribute by whether it is located on consistent parts of the object across the positive instances (e.g. if the regions in the images correspond to the same nameable part of a bird).

We use a simple approach to measure the spatial consistency of an attribute with respect to the object (illustrated in Figure 4). At each active split, we train our attribute recommendation system using all attribute candidates that have been presented to human users so far, with accepted ones as positive exemplars and rejected ones as negative exemplars. Note that the L-CRF model (Section 3.1) can also encourage spatial consistency among training images (as we will see in Section 4); however those constraints are only pairwise, whereas the features here are higher-order statistics capturing the set of regions as a whole. Our recommender system is related to the nameability model of [17], but that model was restricted to global image-level attributes, whereas we model whether a group of local regions are likely to be deemed consistent and hence meaningful by a human.

4. Experiments

We now test our proposed approach to local attribute discovery. We use data from two recent datasets with fine-grained category labels: a subset of the Caltech-UCSD Birds-200-2011 (CUB) [22] dataset containing about 1,500 images of 25 categories of birds, and the Leeds Butterfly (LB) [24] dataset, which contains 832 images from 10 categories of butterflies. We apply a hierarchical segmentation algorithm [1] on each image to generate regions, and filter out background regions by applying GrabCut [20] using the ground truth bounding boxes provided by the datasets (for LB, using a bounding box around the GT segmentation mask in order to be consistent with CUB). Most images contain about $100 - 150$ such regions.

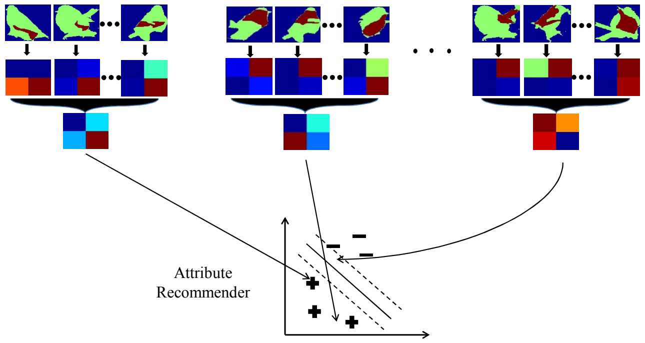

Figure 4. Illustration of the recommender system. A background mask is estimated for each image (top row, blue) using GrabCut [20]. The foreground mask is divided into a 2×2 grid. For each attribute region in the positive images (top row, dark red), we measure its spatial overlap with each grid cell shown in the second row, where degree of overlap is represented by colors ranging from dark blue (no overlap) to dark red (high overlap). Averaging these features across all positive images in the split (third row) gives a representation for the candidate attribute. We add two extra dimensions containing the mean and standard deviation of the areas of the selected regions, creating a 6-D feature vector to train a classifier. This is a positive exemplar if the candidate is deemed meaningful by the user, and negative otherwise.

For the region appearance features $f(\cdot)$ in equations (2) and (3), we combine a color feature (color histogram with 8 bins per RGB channel), a contour feature (gPb [1]), a size feature (region area and boundary length), a shape feature (an 8×8 subsampled binary mask), and spatial location (absolute pixel location of the centroid). For the distance function $D(\cdot, \cdot)$ in equation (3), we compute χ^2 distances for the color, contour, size, and shape features, and Euclidean distance for the spatial location feature. During learning, we constrain the weights of $\vec{\alpha}_k^+$ and $\vec{\alpha}_k^-$ corresponding to the spatial location feature to be positive to encourage candidates to appear at consistent locations. The weights in $\vec{\alpha}_k^+$ and $\vec{\alpha}_k^-$ corresponding to other feature types are constrained to be nonnegative and nonpositive, respectively, to encourage visual similarity among regions on the positive side of an active split and dissimilarity for regions on opposite sides. The bias terms $\vec{\beta}_k$ are not constrained.

Exhaustive data collection for all 200 categories in the

CUB birds dataset is not feasible because it would require about a million user responses. So we conduct systematic experiments on three subsets of CUB: ten randomly-selected categories, the ten hardest categories (defined as the 10 categories for which a linear SVM classifier using global color and gist features exhibits the worst classification performance), and five categories consisting of different species of warblers (to test performance on very fine-grained category differences). Each dataset is split into training, validation, and testing subsets. For CUB these subsets are one-half, one-quarter, and one-quarter of the images, respectively, while for LB each subset is one-third.

We use Amazon's Mechanical Turk to run our human interaction experiments. For each dataset, we generate an exhaustive list of all possible active splits, and use an "offline" collection approach [17] to conduct systematic experiments using data from real users without needing a live user-in-the-loop. We present attribute visualizations by superimposing on each latent region a "spotlight" consisting of a 2-D Gaussian whose mean is the region centroid and whose standard deviation is proportional to its size (and including a red outline for the butterfly images to enhance contrast). We do this to "blur" the precise boundaries of the selected regions, since they are an artifact of the choice of segmentation algorithm and are not important. We present each candidate attribute to 10 subjects, each of whom is asked to name the highlighted region (*e.g.* belly) and give a descriptive word (*e.g.* white). See Figure 8. We also ask the subjects to rate their confidence on a scale from 1 ("no idea") to 4 ("very confident"); candidates with mean score above 3 across users are declared to be semantically meaningful.

4.1. Attribute-based image classification

We now use our discovered attributes for image classification. We detect attributes in validation images and learn linear SVM and nearest-neighbor classifiers, and then detect attributes and measure performance on the test subset. We represent each image as a binary feature vector indicating which attributes were detected. Each category is represented as the average feature vector of its training images. The nearest-neighbor classifier works by assigning the test image to the category with the closest feature vector (similar to [15]). The SVM classifier is trained directly on the above binary features using cross-validation to choose parameters.

Figure 5 presents classification results on CUB birds and LB butterflies, comparing the attribute vocabularies produced by our **Proposed** technique with two baselines that are representative of existing approaches in the literature. These results do not include the recommender system; we evaluate that separately. **Hand-listed** uses the expert-generated attributes provided with the datasets. These are guaranteed to be semantically meaningful but may not be discriminative. **Discriminative only**, at the other extreme, greedily finds the most discriminative candidates and hopes for them to be semantic. At each iteration (i.e. active split) among K candidates, it picks the one that provides the biggest boost in classification performance on a held-out validation set. Candidates that are not semantic (and hence not attributes) are dropped in a post-process. As reference, we also show performance if *all* discriminative candidates are used (semantic or not). This **Upper bound** performance depicts the sacrifice in performance one makes in return for semantically meaningful attributes.

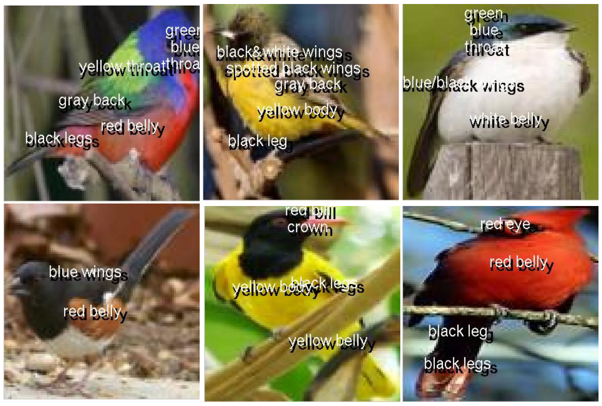

Figure 7. Examples of automatic text generation.

We see in Figure 5 that our proposed method performs significantly better than either the hand-listed attributes or discriminative only baselines. These conclusions are stable across all of the datasets and across both SVM and nearest neighbor classifiers. Hand-listed can be viewed as a semantic-only baseline (since the human experts likely ignored machine-detectability while selecting the attributes) and discriminative only can be thought of as focusing only on discriminative power and then addressing semantics after-the-fact. Our proposed approach that balances both these aspects performs better than either one.

We also evaluate our recommender system as shown in Figure 6. We see that using the recommender allows us to gather more attributes and achieve higher accuracy for a fixed number of user iterations. The recommender thus allows our system to reduce the human effort involved in the discovery process, without sacrificing discriminability.

4.2. Image-to-text Generation

Through the interaction with users, our process generates names for each of our discovered attributes; Figure 8 shows some examples. We can use these names to produce textual annotations for unseen images. We list the name of the attribute with maximum detection score among all candidates detected on the detected region. Sample annotation results are shown in Figure 7 using the system trained on the 10 random categories subset of the CUB birds dataset. Note that some of these images belong to categories that

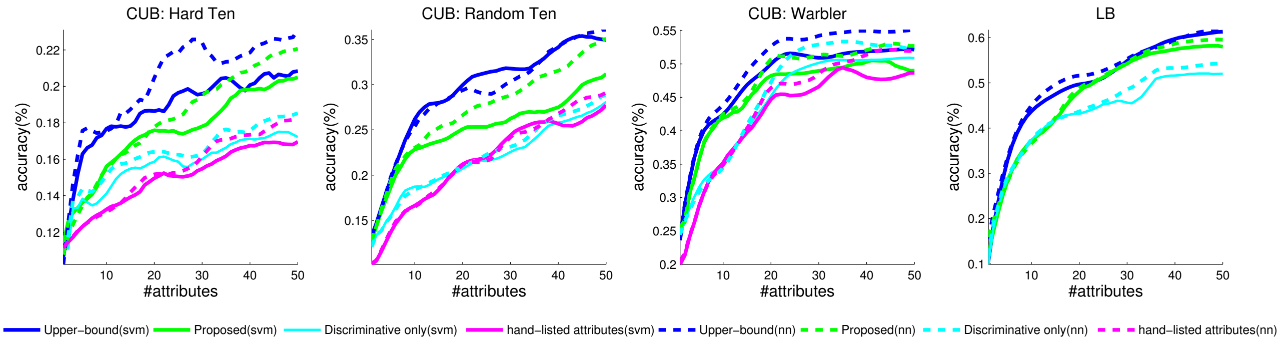

Figure 5. Image classification performance on four datasets with SVM and nearest neighbor (nn) classifiers, and using four different attribute discovery strategies: attributes selected by a purely discriminative criterion (**Upper bound**), a purely discriminative criterion from which non-meaningful candidates are removed by post-processing (**Discriminative only**), attributes produced by a human expert (**Hand-listed**), and our proposed approach which includes human interaction (**Proposed**). Classification statistics are averaged over 10 trials. The LB dataset does not include ground truth attributes so we do not evaluate hand-listed attributes on this dataset.

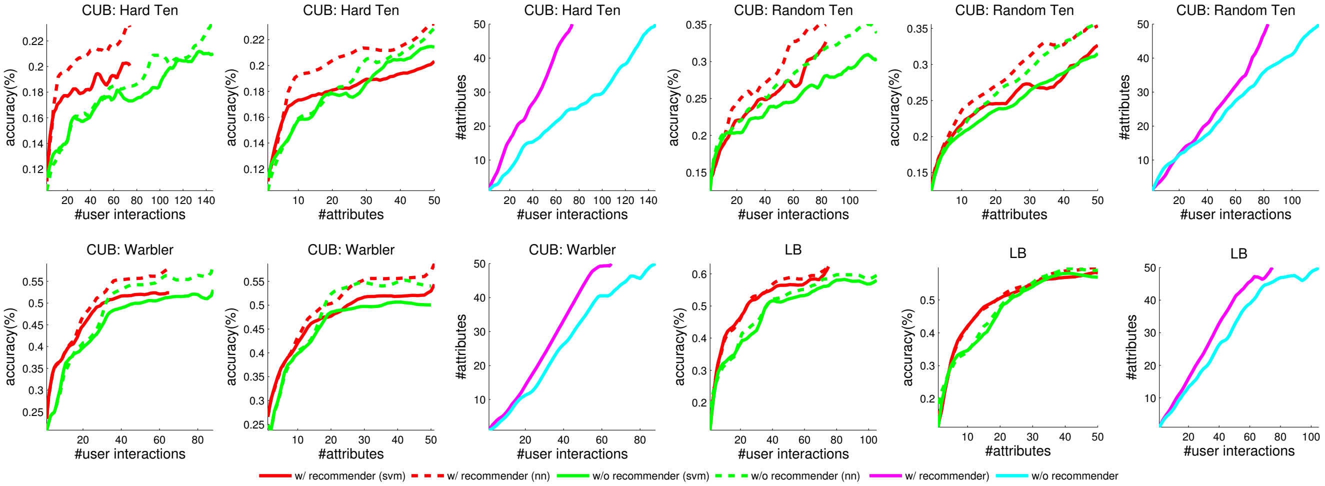

Figure 6. Classification performance of the Proposed system with and without using the recommender.

our system has never seen before and were not part of the discovery process at all. Being able to meaningfully annotate unseen images demonstrates the ability of our system to find human-understandable and machine-detectable attributes that can be shared across categories.

We can use the fact that several of the attribute names provided by our users match the hand-selected attribute names given in the CUB dataset to evaluate the detection accuracy of our attributes.[2] Some attributes that have high accuracy include blue wing (71.2%), red eye (83.3%), yellow belly (72.5%), red forehead (75.7%), and white nape (71.7%). Others are less accurate: spotted wing (67.2%), orange leg (60.3%), white crown (61.7%). In computing these accuracies, we use all positive examples that have the attribute, and randomly sample the same number of negative examples. We also observe that our approach is able to discover some interesting attributes that were not provided in the hand-selected annotations, including “sharp bill”, and “long/thin leg.”

5. Conclusion

We have presented a novel approach for discovering localized attributes for fine-grained recognition tasks. Our system generates local attributes that are both discriminative and human understandable, while keeping human effort to a minimum. Our approach intelligently selects active splits among training images, looking for the most discriminative local information. Involving a human in the loop, it identifies semantically meaningful attributes. We propose a recommender system that prioritizes likely to be meaningful candidate attributes, thus saving user time. Results on different datasets show the advantages of our novel local attribute discovery model as compared to existing approaches to determining an attribute vocabulary. In future work, we

[2] The hand-selected annotations are not used in our discovery process; we use them only as ground-truth for measuring detection accuracy.

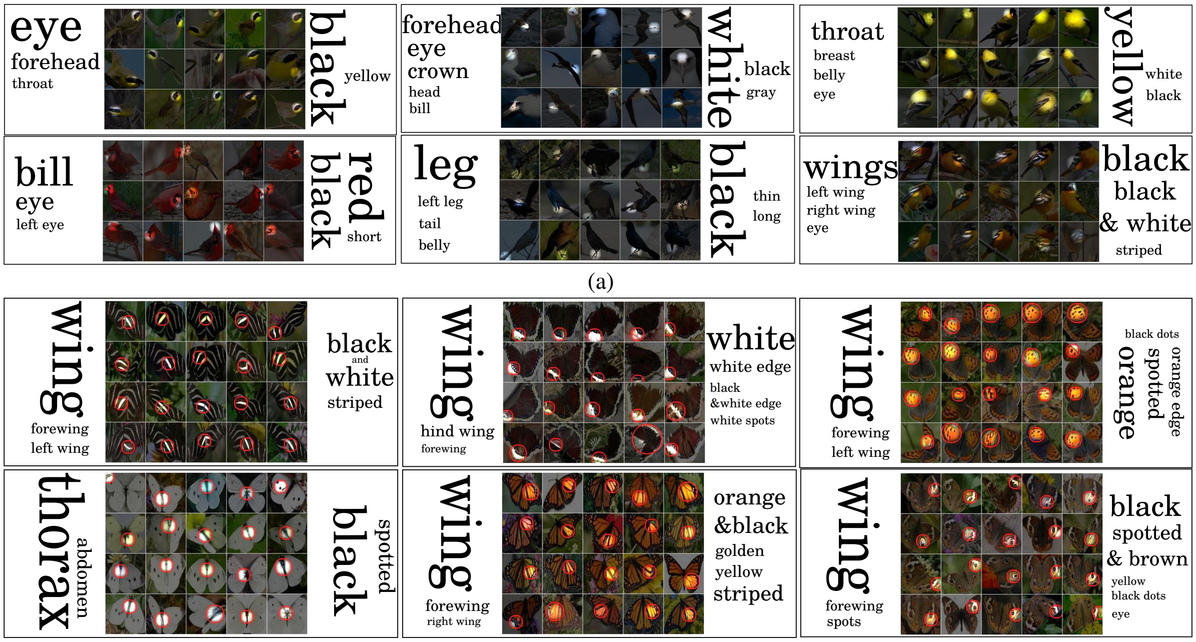

Figure 8. Some local attributes discovered by our approach, along with the semantic attribute names provided by users (where font size is proportional to number of users reporting that name), for *(a)* CUB birds, and *(b)* LB butterflies. Best viewed on-screen and in color.

would like to find links between local attributes and object models, in order to bring object detection into the loop of discovering localized attributes, such that both tasks benefit from each other. We would also like to study how to better incorporate human interactions into recognition techniques.

Acknowledgments: The authors thank Dhruv Batra for discussions on the L-CRF formulation, and acknowledge support from NSF IIS-1065390, the Luce Foundation, the Lilly Endowment, and the IU Data-to-Insight Center.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *PAMI*, 33:898–916, 2011.

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

[3] S. Branson, C. Wah, B. Babenko, F. Schroff, P. Welinder, P. Perona, and S. Belongie. Visual recognition with humans in the loop. In *ECCV*, 2010.

[4] T. Deselaers, B. Alexe, and V. Ferrari. Localizing objects while learning their appearance. In *ECCV*, 2010.

[5] T. Deselaers and V. Ferrari. A conditional random field for multiple-instance learning. In *ICML*, 2010.

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

[7] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 32(9):1627–1645, 2010.

[8] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, 2003.

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

[10] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, v1.21. http://cvxr.com/cvx.

[11] S. J. Hwang, F. Sha, and K. Grauman. Sharing features between objects and their attributes. In *CVPR*, 2011.

[12] G. Kim and A. Torralba. Unsupervised detection of regions of interest using iterative link analysis. In *NIPS*, 2009.

[13] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *PAMI*, 28(10):1568–1583, 2006.

[14] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and Simile Classifiers for Face Verification. In *ICCV*, 2009.

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

[16] Y. J. Lee and K. Grauman. Foreground focus: Unsupervised learning from partially matching images. *IJCV*, 85(2):143–166, 2009.

[17] D. Parikh and K. Grauman. Interactively building a discriminative vocabulary of nameable attributes. In *CVPR*, 2011.

[18] D. Parikh and K. Grauman. Relative attributes. In *ICCV*, 2011.

[19] A. Quattoni, S. Wang, L.-P. Morency, M. Collins, and T. Darrell. Hidden-state conditional random fields. *PAMI*, 29(10):1848–1852, 2007.

[20] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: interactive foreground extraction using iterated graph cuts. In *SIGGRAPH*, 2004.

[21] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 6:1453–1484, 2005.

[22] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 dataset. In *CVPR Workshop on Fine-Grained Visual Categorization*, 2011.

[23] G. Wang and D. Forsyth. Joint learning of visual attributes, object classes and visual saliency. In *ICCV*, 2009.

[24] J. Wang, K. Markert, and M. Everingham. Learning models for object recognition from natural language descriptions. In *BMVC*, 2009.

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

[26] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, 2011.

[27] B. Yao, A. Khosla, and L. Fei-Fei. Combining randomization and discrimination for fine-grained image categorization. In *CVPR*, 2011.