Animals on the Web

Tamara L. Berg
University of California, Berkeley
Computer Science Division
millert@cs.berkeley.edu

David A. Forsyth
University of Illinois, Urbana-Champaign
Department of Computer Science
daf@cs.uiuc.edu

Abstract

*We demonstrate a method for identifying images containing categories of animals. The images we classify depict animals in a wide range of aspects, configurations and appearances. In addition, the images typically portray multiple species that differ in appearance (*e.g. *ukari's, vervet monkeys, spider monkeys, rhesus monkeys, etc.). Our method is accurate despite this variation and relies on four simple cues: text, color, shape and texture.*

Visual cues are evaluated by a voting method that compares local image phenomena with a number of visual exemplars for the category. The visual exemplars are obtained using a clustering method applied to text on web pages. The only supervision required involves identifying which clusters of exemplars refer to which sense of a term (for example, "monkey" can refer to an animal or a bandmember).

Because our method is applied to web pages with free text, the word cue is extremely noisy. We show unequivocal evidence that visual information improves performance for our task. Our method allows us to produce large, accurate and challenging visual datasets mostly automatically.

1. Introduction

There are currently more than 8,168,684,336[1] web pages on the Internet. A search for the term "monkey" yields 36,800,000 results using Google text search. There must be a large quantity of images portraying "monkeys" within these pages, but retrieving them is not an easy task as demonstrated by the fact that a Google image search for "monkey" yields only 30 actual "monkey" pictures in the first 100 results. Animals in particular are quite difficult to identify because they pose difficulties that most vision systems are ill-equipped to handle, including large variations in aspect, appearance, depiction, and articulated limbs.

We build a classifier that uses word and image information to determine whether an image depicts an animal. This classifier uses a set of examples, harvested largely automatically, but incorporating some supervision to deal with polysemy-like phenomena. Four cues are combined to determine the final classification of each image: nearby words, color, shape, and texture. The resulting classifier is very accurate despite large variation in test images. In figure 1 we show that visual information makes a substantial contribution to the performance of our classifier.

We demonstrate one application by harvesting pictures of animals from the web. Since there is little point in looking for, say, "alligator" in web pages that don't have words like "alligator", "reptile" or "swamp", we use Google to focus the search. Using Google text search, we retrieve the top 1000 results for each category and use our classifier to re-rank the images on the returned pages. The resulting sets of animal images (fig 3) are quite compelling and demonstrate that we can handle a broad range of animals.

For one of our categories, "monkey", we show that the same algorithm can be used to label a much larger collection of images. The dataset that we produce from this set of images is startlingly accurate (81% precision for the first 500 images) and displays great visual variety (fig 5). This suggests that it should be possible to build enormous, rich sets of labeled animal images with our classifier.

1.1. Previous Work:

Object Recognition has been thoroughly researched, but is by no means a solved problem. There has been a recent explosion of work in appearance based object recognition using local features, in particular on the Caltech-101 Object Categories Dataset introduced in [8]. Some methods use constellation of parts based models trained using EM [10, 8]. Others employ probabilistic models like pLSA or LDA [20, 19]. The closest method to ours employs a nearest neighbor based deformable shape matching [4] to find correspondences between objects. Object recognition is unsolved, but we show that whole image classification can be successful using fairly simple methods.

There has been some preliminary work on voting based methods for image classification in the Caltech-101 Dataset using geometric blur features [3]. In an alternative forced choice recognition task this method produces quite rea-

[1] Google's last released number of indexed web pages

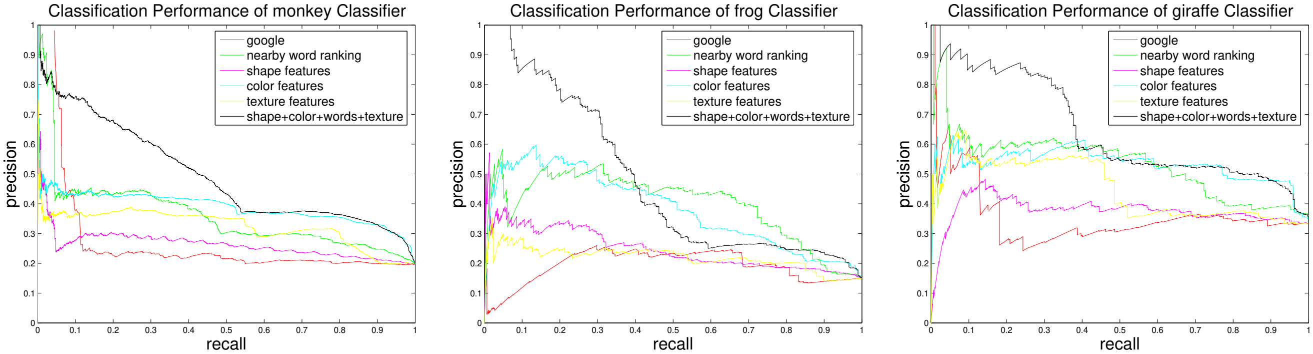

Figure 1. Classification performance on Test images (all images except visual exemplars) for the "monkey" (**left**), "frog" (**center**) and "giraffe" (**right**) categories. Recall is measured over images in our collection, not all images existing on the web. "monkey" results are on a set of 12567 images, 2456 of which are true "monkey" images. "frog" results are on a set of 1964 images, 290 of which are true "frog" images. "giraffe" results are on a set of 873 images, 287 of which are true "giraffe" images. Curves show the Google text search classification (**red**), word based classification (**green**), geometric blur shape feature based classification (**magenta**), color based classification (**cyan**), texture based classification (**yellow**) and the final classification using a combination of cues (**black**). Incorporating visual information increases classification performance enormously over using word based classification alone.

sonable results (recognition rate of 51%) as compared with the best previously reported result using deformable shape matching (45%) [4] [2].Our work uses a modified voting method for image retrieval that incorporates multiple sources of image and text based information.

Words + Pictures: Many collections of pictures come with associated text: collections of web pages, news articles, museum collections and collections of annotated images and video. There has been extensive work on fusing the information available from these two modalities to perform various tasks such as clustering art [2], labeling images [1, 15, 12] or identifying faces in news photographs [6]. However, in all of these papers the relationship between the words and pictures has been explicit; pictures annotated with key words or captioned photographs or video. On web pages where we focus our work, the link between words and pictures is less clear.

Our model of image re-ranking is related to much work done on relevance models for re-ranking data items by assigning to each a probability of relevance. Jeon *et al* [13] is the work most closely related to ours in this area. They use a text and image based relevance model to re-rank search results on a set of Corel images with associated keywords.

In addition, there has been some recent work on re-ranking Google search results using only images [11, 9] and on re-ranking search results using text plus images [21]. Our work proposes a similar task to the last paper, using the text and image information on web pages to re-rank the Google search results for a set of queries. However, by focusing on animal categories we are working with much richer, more difficult data, and can show unequivocal benefits from a visual representation.

Animals are demonstrably among the most difficult classes to recognize [4, 8]. This is because animals often take on a wide variety of appearances, depictions and aspects. Animals also come with the added challenges of articulated limbs and the fact that multiple species while looking quite different in appearance have the same semantic category label, *e.g.* "African leopards", "black leopards" and "clouded leopards".

There has been some work on recognizing animal categories using deformable models of shape [17, 18]. However, they concentrate on building a single model for appearance and would not be able to handle the large changes in aspect or multiple species that we find in our data.

2. Dataset

We have collected a set of 9,972 web pages using Google text search on 10 animal queries: "alligator","ant", "bear", "beaver", "dolphin", "frog", "giraffe", "leopard", "monkey" and "penguin". From these pages we extract 14,051 distinct images of sufficiently large size (at least 120x120 pixels).

Additionally, we have collected 9,320 web pages using Google text search on 13 queries related to monkey: "monkey", "monkey primate","monkey species","monkey monkeys", "monkey animal", "monkey science","monkey wild","monkey simian","monkey new world","monkey old world", "monkey banana", "monkey zoo","monkey Africa". From these pages we extract 12,866 images of sufficient size, 2,569 of which are actual monkey images.

Animals: In addition to the aforementioned difficulties of visual variance, animals have the added challenge of having evolved to be hard to spot. The tiger's stripes, the gi-

[2]At the time of publishing two new methods based on spatial pyramid matching [14] and k-NN SVMs [22] have since beat this performance with respectively 56% and 59% recognition rates for 15 training examples per class.

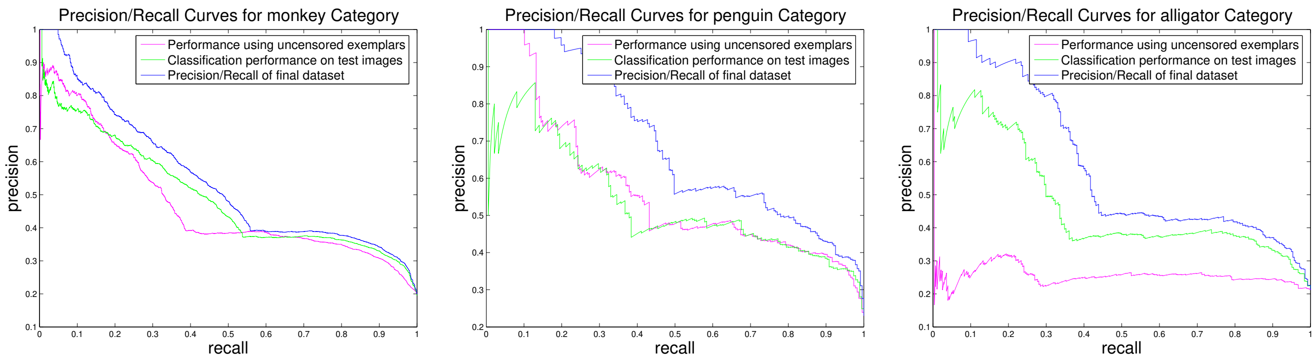

Figure 2. Our method uses an unusual form of (very light) supervisory input. Instead of labeling each training image, we simply identify which of a set of 10 clusters of example images are relevant. Furthermore, we have the option of removing erroneous images from clusters. For very large sets of images, this second process has little effect (compare the magenta and blue curves for "monkey" **left**), but for some smaller sets it can be helpful (*e.g.* "alligator" **right**). On a strict interpretation of a train/test split, we would report results only on images that do not appear in the clusters (**green**). However, for our application – building datasets – we also report a precision/recall curve for the accuracy of the final dataset produced (**blue**). For larger datasets the curves reported for the classification performance and dataset performance tend towards one another (**green** and **blue**). Recall is measured over images in our collection, not all images existing on the web. We show results for "monkey" (**left**) on a set of 12866 images containing 2569 "monkey" images, "penguin" (**center**) on a set of 985 images containing 193 "penguin" images, and "alligator" (**right**) on a set of 1311 containing 274 "alligator" images.

raffe's patches and the penguin's color all serve as camouflage, impeding segmentation from their surroundings.

Web Pages and Images: One important purpose of our activities is building huge reference collections of images. Images on the web are interesting, because they occur in immense numbers, and may co-occur with other forms of information. Thus, we focus on classifying images that appear on web pages using image and local text information.

Text is a natural source of information about the content of images, but the relationship between text and images on a web page is complex. In particular, there are no obvious indicators linking particular text items with image content (a problem that doesn't arise if one confines attention to captions, annotations or image names which is what has been concentrated on in previous work). All this makes text a noisy cue to image content if used alone (see the green curves in figure 1). However, this noisy cue can be helpful, if combined appropriately with good image descriptors and good examples. Furthermore, text helps us focus on web pages that may contain useful images.

3. Implementation

Our classifier consists of two stages, training and testing. The training stage selects a set of images to use as visual exemplars (exemplars for short) using only text based information (Secs 3.1-3.3). We then use visual and textual cues in the testing stage to extend this set of exemplars to images that are visually and semantically similar (Sec 3.4).

The training stage applies Latent Dirichlet Allocation (LDA) to the words contained in the web pages to discover a set of latent topics for each category. These latent topics give distributions over words and are used to select highly likely words for each topic. We rank images according to their nearby word likelihoods and select a set of 30 exemplars for each topic.

Words and images can be ambiguous (*e.g.* "alligator" could refer to "alligator boots" or "alligator clips" as well as the animal). Currently there is no known method for breaking this polysemy-like phenomenon automatically. Therefore, at this point we ask the user to identify which topics are relevant to the concept they are searching for. The user labels each topic as relevant or background, depending on whether the associated images and words illustrate the category well. Given this labeling we merge selected topics into a single relevant topic and unselected topics into a background topic (pooling their exemplars and likely words).

There is an optional second step to our training process, allowing the user to swap erroneously labeled exemplars between the relevant and background topics. This makes the results better, at little cost, but isn't compulsory (see figures 2 and 4). This amounts to clicking on incorrectly labeled exemplars to move them between topics. Typically the user has to click on a small number of images since text based labeling does a decent job of labeling at least the highest ranked images. For some of the 10 initial categories, the results are improved quite a bit by removing erroneous exemplars. Whereas, for the extended monkey category removal of erroneous exemplars is largely unnecessary (compare magenta and green in fig 2). This suggests that if we were to extend each of our categories as we did for the monkey class this step would become superfluous.

In the testing stage, we rank each image in the dataset according to a voting method using the knowledge base we have collected in the training stage. Voting uses image in-

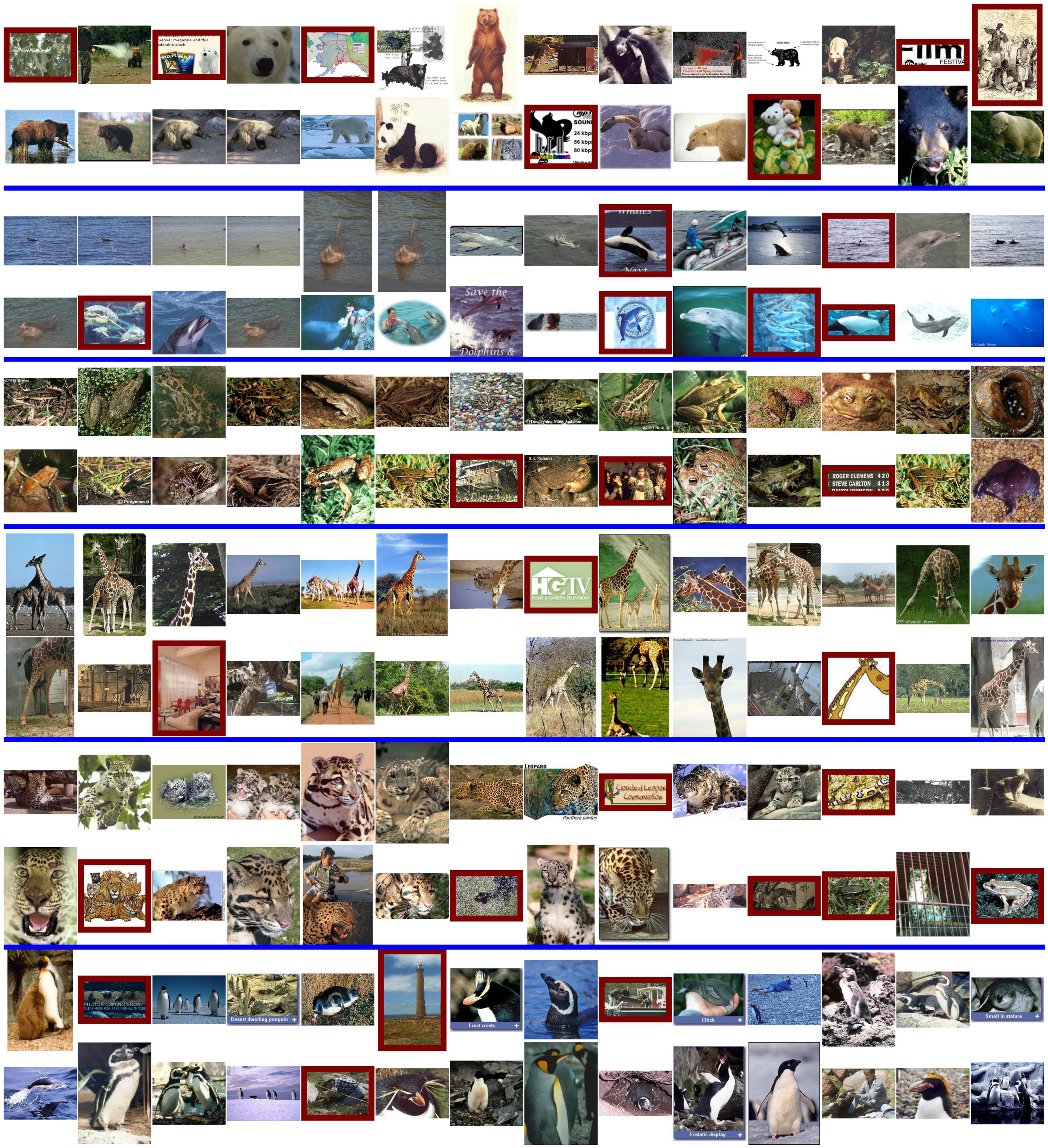

Figure 3. Top images returned by running our classifiers on a set of Test Images (the whole collection excluding visual exemplars) for the "bear", "dolphin", "frog", "giraffe", "leopard", and "penguin" categories. Most of the top classified images for each category are correct and display a wide variety of poses ("giraffe"), depictions ("leopard" – heads or whole bodies) and even multiple species ("penguins"). Returned "bear" results include "grizzly bears", "pandas" and "polar bears". Notice that the returned false positives (dark red) are quite reasonable; teddy bears for the "bear" class, whale images for the "dolphin" class and leopard frogs and leopard geckos for the "leopard" class. Drawings, even though they may depict the wanted category are also counted as false positives (*e.g*. dolphin and leopard drawings). Our image classification inherently takes advantage of the fact that objects are often correlated with their backgrounds (*e.g*. "dolphins" are usually in or near water, "giraffes" usually co-occur with grass or trees), to label images.

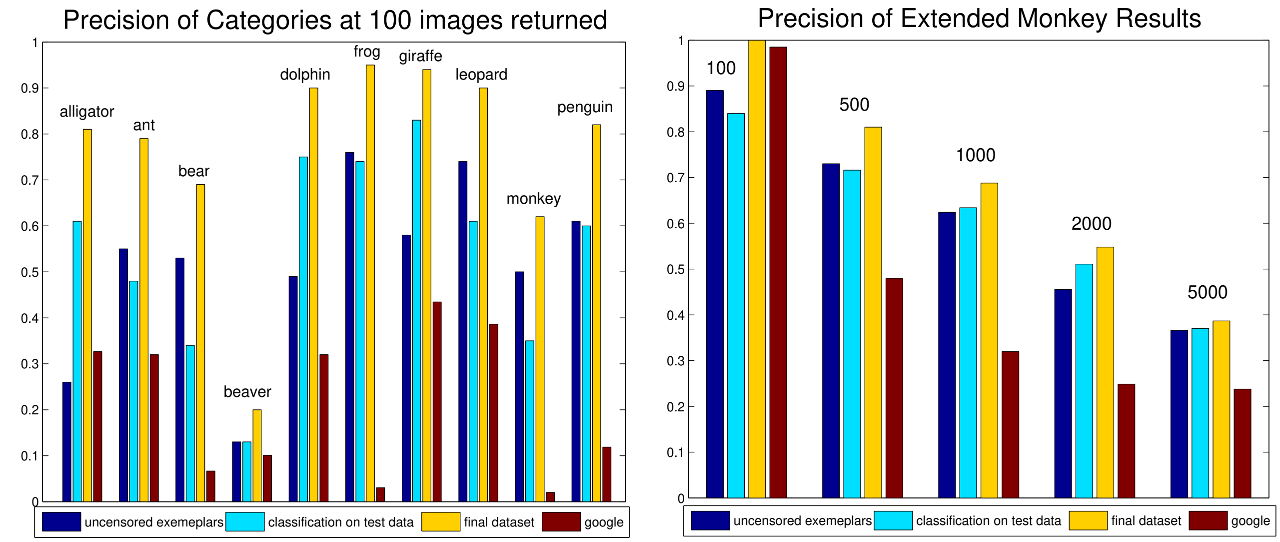

Figure 4. **Left:** Precision of the first 100 images for our 10 original categories: "alligator", "ant", "bear", "beaver", "dolphin", "frog", "giraffe", "leopard", "monkey", "penguin". Bar graphs show precision from the original Google text search ranking (**red**), for our classifier trained using uncensored exemplars (**blue**), and using corrected exemplars (**cyan**), described in section 3. One application of our system is the creation of rich animal datasets; precision of these datasets is shown in yellow. In all categories we outperform the Google text search ranking, sometimes by quite a bit ("giraffe", "penguin"). **Right:** Using multiple queries related to monkeys we are able to build an enormously rich and varied dataset of monkey images. Here we show the precision of our dataset (**yellow**) at various levels of recall (100, 500, 1000, 2000 and 5000 images). We also show the classification performance of the Google text search ranking (**red**) as well as two variations of our classifier, trained using uncensored (**blue**) and supervised exemplars (**cyan**) as described in section 3.

formation in the form of shape, texture and color features as well as word information based on words located near the image. By combining each of these modalities a better ranking is achieved than using any of the cues alone.

3.1. Text Representation

For each image, because nearby words are more likely to be relevant to the image than words elsewhere on the page, we restrict consideration to the 100 words surrounding the image link in its associated web page. The text is described using a bag of words model as a vector of word counts of these nearby words. To extract words from our collection of pages, we parse the HTML, compare to a dictionary to extract valid word strings and remove common English words.

LDA [7] is applied to all text on the collected web pages to discover a set of 10 latent topics for each category. LDA is a generative probabilistic model where documents are modeled as an infinite mixture over a set of latent topics and each topic is characterized by a distribution over words. Some of these topics will be relevant to our query while others will be irrelevant.

Using the word distributions learned by LDA, we extract a set of 50 highly likely words to represent each topic. We compute a likelihood for each image according to its associated word vector and the word likelihoods found by LDA.

3.2. Image Representation

We employ 3 types of image features, shape based geometric blur features, color features and texture features. We sample 50-400 local shape features (randomly at edge points), 9 semi-global color features and 1 global texture feature per image.

The geometric blur descriptor [5] first produces sparse channels from the gray scale image, in this case, half-wave rectified oriented edge filter responses at three orientations yielding six channels. Each channel is blurred by a spatially varying Gaussian with a standard deviation proportional to the distance to the feature center. The descriptors are then sub-sampled and normalized.

For our color representation we subdivide each image into 9 regions. In each of these regions we compute a normalized color histogram in RGB space with 8 divisions per color channel, 512 bins total. We also compute local color histograms with radius 30 pixels at each geometric blur feature point for use in gating of the geometric blur features as described in section 3.4.

Texture is represented globally across the image using histograms of filter outputs as in [16]. We use a filter bank consisting 24 bar and spot type filters: first and second derivatives of Gaussians at 6 orientations, 8 Laplacian of Gaussian filters and 4 Gaussians. We then create histograms of each filter output.

3.3. Exemplar Initialization

Using LDA we have computed a likelihood of each image under each topic as described in section 3.1. We tentatively assign each image to its most likely topic. For each topic, we select the top 30 images – or fewer if less than 30 images are assigned – as exemplars. These exemplars often have high precision, a fact that is not surprising given that most successful image search techniques currently use only textual information to index images.

3.4. Shape, Color, Texture and Word Based Voting

For each image, we compute features of 3 types: shape, color and texture. For each type of feature we create two pools; one containing positive features from the relevant exemplars and the other negative features from the background exemplars. For each feature of a particular type in a query image, we apply a 1-nearest neighbor classifier with similarity measured using normalized correlation to label the feature as the relevant topic or the background topic.

For each visual cue (color, shape, and texture), we compute the sum of the similarities of features matching positive exemplars. These 3 numbers are used as the cue scores for the image. For each image, we normalize each cue score to lie between 0 and 1 by dividing by the maximum color, shape or texture cue score computed over all images. In this way the cues are used to independently rank the images (by labeling each image with a score between 0 and 1).

Shape Feature Gating: We modify the voting strategy for the shape feature voting. Shape features tend to match at two extremes, either the best match is a good one and has a high score or the match is a poor one with a lower score. We prune out low score matches from the voting process allowing features to vote only if their match score is quite good (normalized correlation score above 0.95). We also apply a color based gating to the geometric blur matches. If the local color of the best match is significantly different from the query feature, we don't allow the feature to vote. Pruning and gating helps to disallow features from voting that are unsure about their votes and improves the shape feature voting performance significantly – especially in the higher recall range – as well as overall classification performance.

Words: We also compute a word score for each image by summing the likelihood under the relevant topic model found by LDA of words near the image on the associated page as described in section 3.1. We normalize the word score by dividing by the maximal word score over all images. This gives us a 4^{th} ranking of the images by labeling each image with a score between 0 and 1.

Cue Combination: We combine our 4 independent cue scores using a linear combination with convex weights. Currently the 4 cues are equally weighted. While equal weighting usually performs near to the optimal combination, some cues may perform better overall or for specific classes. For example, texture is a good cue for the "leopard" class while color is a good feature for the "frog" class. In the future we hope to learn this cue combination from evaluation on our training exemplars.

One **powerful** advantage of independent cue based voting is that it allows for the fact that each cue may work well for some images, but poorly for others. The errors made by each cue seem to be somewhat independent (see fig 1). Thus, by combining the different cues, we are able to achieve much better results than using any cue in isolation.

4. Results

We build quite effective classifiers for our initial 10 animal categories (see figure 4). For all categories our method (cyan) outperforms Google text search (red) in classification performance on the top 100 images returned. The giraffe and frog classifiers are especially accurate, returning 74 and 83 true positives respectively. Because exemplar based voting incorporates multiple templates per category we are able to retrieve images across different poses, aspects, and even species.

The top results returned by our classifiers are usually quite good. Figure 3 shows the top results returned by 6 of our classifiers: "bear", "dolphin", "frog", "giraffe", "leopard" and "penguin". Even the false negatives returned by our classifier are often reasonable, teddy bears for the "bear" class, whale images for the "dolphin" class and leopard frogs for the "leopard" class. Drawings and toy animals, even though they may depict the correct category are counted as false positives.

Visual Information makes a substantial contribution to search (fig 1). Our classifier uses a combination of visual cues: color (cyan), shape (magenta), texture (yellow), and textual cues: nearby words (green). Their combined classification performance (black) outperforms the classification power of any single cue by a substantial margin. This shows that current text based systems could be improved by the use of visual information. We also significantly exceed the original Google ranking (red) which we compute over images based on the order of the associated page in the Google text search results.

While shape is the cue favored by most recent object recognition systems, it is often less informative than color or texture for our dataset. This is due to the extreme variance in aspect and pose of animals. Color is often a good cue, especially for classes like "dolphin" and "frog". Texture performs well for the "giraffe" and "leopard" classes. Word ranking works well for some classes ("bear") and quite poorly for others ("penguin"). Because our cues were each used independently to rank images, we could easily incorporate a wider range of cues.

Censored vs Uncensored Exemplars: Our method uses an unusual form of (very light) supervisory input. Instead of labeling each training image, we simply identify which of a set of 10 clusters of example images are relevant. Furthermore, we have the option of removing erroneous images from clusters. For very large sets of images, this second process has little effect (compare the magenta and blue curves

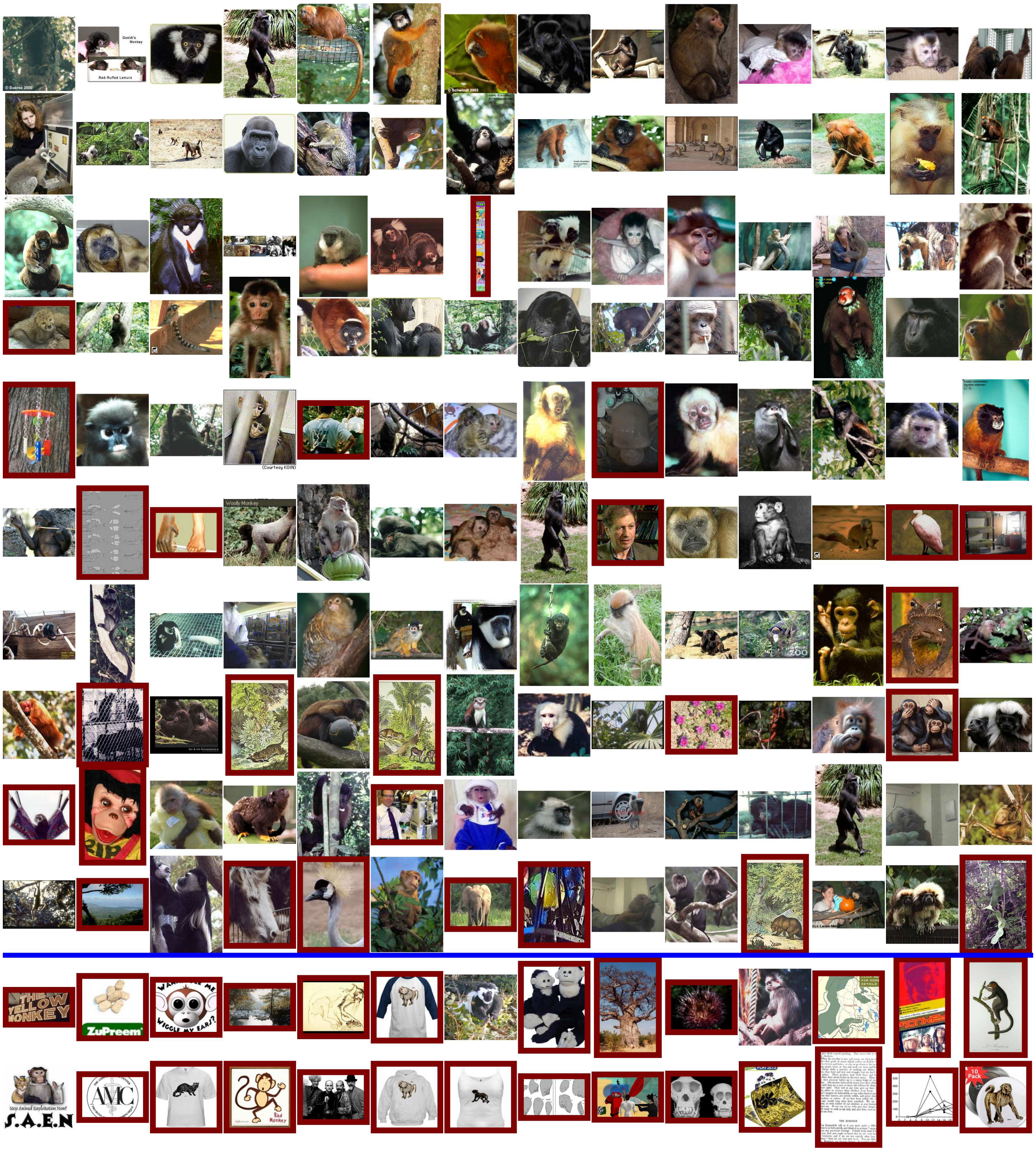

Figure 5. Images sampled from the dataset of monkey images that we produce. False positives are bordered in dark red. The first 10 rows are sampled every 4^{th} image from the top 560 results, while the last two rows are sampled every 250^{th} image from the last 5,000-12,866 results. Our monkey dataset is quite accurate, with a precision of 81% for the top 500 images, and a precision of 69% for the top 1000 images. Deciding which images are relevant to a query doesn't have a single interpretation. We chose to include primates like apes, lemurs, chimps and gibbons, though we didn't include things such as monkey figurines (row 8, col 13), people (row 6, col 9), monkey masks (row 9, col 2) and monkey drawings (row 4, col 1), (row 8, col 4). Our results include a huge range of aspects and poses as well as a depictions in different settings (*e.g.* trees, cages and indoor setings). Our animal image classifiers inherently take advantage of the fact that objects are often correlated with their backgrounds ("monkeys" are often in trees and other greenery).

in figure 2 for "monkey"), but for smaller sets it can be helpful (*e.g*. "alligator").

We believe that exemplars selected using LDA tend to be easier to classify using words because we selected them based on their high word likelihood. As a result, if we exclude them from testing, classification performance appears worse than it is. In figure 2, we show classifiers trained using both uncensored and censored exemplars. The uncensored case (magenta) is tested on the whole set of images, while the censored case is tested excluding the exemplars (green). The censored classifier should always perform better than the uncensored since it is provided with cleaner training data, but we see that in some cases the uncensored classifier has better accuracy. This is because by excluding the exemplar images, we bias our test set to be more difficult than the entire set of images returned by Google. This is not a phenomenon previously seen since exemplars are usually chosen at random from the set of images.

Beaver is the only class on which we perform poorly. Because the returned Google results contain only 67 "beavers" in 1087 images and most returned pages don't refer to "beavers", LDA didn't find a latent topic corresponding to the animal and the resulting classifier failed.

Dataset: We produce an extremely good dataset of 10 animal categories containing pictures of animals in a wide variety of aspects, depictions, appearances, poses and species. Figure 2 shows precision recall curves for datasets produced for the "monkey", "penguin" and "alligator" collections (blue), and figure 4 shows the precision of our dataset (yellow) for the top 100 images of each category.

Extended Monkey Category: For the "monkey" class we collected a much larger set of images using multiple related queries. Having this much data allowed us to build an extremely powerful classifier using the same procedure as for the initial 10 categories. Figure 2 shows that our "monkey" classifier is startlingly accurate using both supervised (green) and uncensored exemplars (magenta).

The "monkey" dataset that we produce is incredibly rich and varied. Figure 5 shows samples from the top 500 images in our monkey dataset (top 10 lines of images), and samples from the bottom 7000 images (bottom 2 lines of images). In the dataset we create 81% of the top 500 images are "monkey" pictures, and 69% of the top 1000 images are "monkeys". Our monkey dataset contains monkeys in a variety of poses, aspects, and depictions as well as a large number of monkey species and other related primates including lemurs, chimps and gibbons. Our results suggest that it should be possible to build enormous, clean sets of labeled animal images for many semantic categories.

References

[1] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. Blei, and M. Jordan. Matching words and pictures. *JMLR*, pages Vol 3, 1107–1135, 2003.

[2] K. Barnard, P. Duyguly, and D. Forsyth. Clustering art. In *CVPR*, June 2001.

[3] A. C. Berg. Phd thesis. To Appear.

[4] A. C. Berg, T. L. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondence. In *CVPR*, June 2005.

[5] A. C. Berg and J. Malik. Geometric blur for template matching. In *CVPR*, June 2001.

[6] T. L. Berg, A. C. Berg, J. Edwards, and D. Forsyth. Who's in the picture? In *NIPS*, Dec. 2004.

[7] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *JMLR*, 3:993–1022, Jan. 2003.

[8] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models for 101 object categories. *CVIU*, To appear.

[9] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search. In *ICCV*, Oct. 2005.

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

[11] R. Fergus, P. Perona, and A. Zisserman. A visual category filter for google images. In *ECCV*, May 2004.

[12] I. Giridharan, P. Duygulu, S. Feng, P. Ircing, S. Khudanpur, D. Klakow, M. Krause, R. Manmatha, H. Nock, D. Petkova, B. Pytlik, and P. Virga. Joint visual-text modeling for automatic retrieval of multimedia documents. In *ACM Multimedia Conference*, Nov. 2005.

[13] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *ACM SIGIR*, 2003.

[14] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching. In *CVPR*, June 2006.

[15] J. Li and J. Wang. Automatic linguistic indexing of pictures by a statistical modeling approach. *PAMI*, pages Vol 25, no. 9, 1075–1088, 2003.

[16] J. Puzicha, Y. Rubner, C. Tomasi, and J. M. Buhmann. Empirical evaluation of dissimilarity measures for color and texture. In *ICCV*, 1999.

[17] D. Ramanan, D. Forsyth, and K. Barnard. Detecting, localizing and recovering kinematics of textured animals. In *CVPR*, June 2005.

[18] C. Schmid. Constructing models for content-based image retrieval. In *CVPR*, June 2001.

[19] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman. Discovering objects and their location in images. In *ICCV*, Oct. 2005.

[20] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky. Learning hierarchical models of scenes, objects, and parts. In *ICCV*, Oct. 2005.

[21] K. Yanai and K. Barnard. Probabilistic web image gathering. In *Workshop on MIR*, Nov. 2005.

[22] H. Zhang, A. Berg, M. Maire, and J. Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *CVPR*, June 2006.