Towards Scalable Dataset Construction: An Active Learning Approach

Brendan Collins, Jia Deng, Kai Li, and Li Fei-Fei

Department of Computer Science, Princeton University, New Jersey, U.S.A.
{bmcollin,dengjia,li,feifeili}@cs.princeton.edu

Abstract. As computer vision research considers more object categories and greater variation within object categories, it is clear that larger and more exhaustive datasets are necessary. However, the process of collecting such datasets is laborious and monotonous. We consider the setting in which many images have been automatically collected for a visual category (typically by automatic internet search), and we must separate relevant images from noise. We present a discriminative learning process which employs active, online learning to quickly classify many images with minimal user input. The principle advantage of this work over previous endeavors is its scalability. We demonstrate precision which is often superior to the state-of-the-art, with scalability which exceeds previous work.

1 Introduction

Though it is difficult to foresee the future of computer vision research, it is likely that its trajectory will include examining a greater number of visual categories (such as objects or scenes), that the complexity of the models employed on these categories will increase, and that these categories will include greater intraclass variation. It is unlikely that the researcher's patience for labeling images will keep pace with the growing need for annotated datasets. For this reason, this work aims to develop a system which can obtain high-precision databases of images with minimal supervision. The particular focus of this work is scalability, and its principle contribution is demonstrating the effectiveness of active learning for automatic dataset construction. With minimal supervision, we match or exceed the precision demonstrated by state-of-the-art technologies. However, using active learning, we are able to extend the performance of our system greatly as the user opportunistically labels additional images. Active learning focuses the attention of the user on those images which are most informative.

Computer vision research has always been heavily dependent on good datasets, many of which were hand-collected (e.g., Caltech-101 [1], Caltech-256 [2], PASCAL [3], LabelMe [4], Fink *et al.* [5], and LotusHill [6]). However, in the last several years, there have been a number of papers which attempt to automate this laborious task. Early work by Fergus *et al.* [7,8] re-ranked images obtained from Google Image Search using visual information.

D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part I, LNCS 5302, pp. 86–98, 2008.

Berg *et al.* [9] aim to automatically construct image datasets for several animal categories. They begin by searching the web using Google text search, obtaining images from the first 1000 web pages returned. Using Latent Dirichlet Allocation they identify a number of latent topics and corresponding exemplary images from this crawled set. These exemplary images are labeled by the user as relevant or background; this labeled set is used to train their voting classifier. Their classifier incorporates textual, shape, color, and texture features.

A slightly more automatic process, termed "OPTIMOL," is used by Li *et al.* [10]. Exploiting the fact that the first few results from search engines tend to be very good, they use the first 5-10 images returned by Google Image Search to train their classifier, built upon a Hierarchical Dirichlet Process [11]. OPTIMOL considers images sequentially, classifying them as either background or relevant. If the image is classified as relevant, the classifier uses incremental learning to refine its model. As the classifier accepts more images, these images allow the classifier to obtain a richer description of the category.

Most recently, the fully automatic "Harvesting Image Databases from the Web" by Schroff *et al.* [12] uses text information to re-rank images retrieved from text-based search. The top-ranked images form the training set of a support vector machine, which relies on visual information to re-rank images once again.

Finally, the Tiny Images [13] project aims to crawl a massive number of images from the internet. They have collected 80 million images to date, for about 75,000 keywords. The goal of this dataset is to collect an extensive collection of images, rather than to obtain high accuracy for each keyword.

2 Approach

Our general approach is motivated primarily by accuracy and scalability. We aim to deal with image categories with very many candidate images, exhibiting intraclass variation. High degrees of intraclass variability normally suggest that a large and accurate training set is necessary; our approach also aims to minimize the time required of the user while still capturing large amounts of diversity in the dataset.

2.1 Crawling

We rely on image search engines to obtain our noisy image set, leveraging the tremendous number of images available on the web. As noted by Schroff *et al.*, most image search engines restrict the number of images returned. To overcome this restriction, we generate multiple related queries using a standard lexical database [14]. We also translate our queries into several languages, accessing the regional website of our image search engines (e.g., *http://images.google.cn*).

2.2 Learning

We utilize a discriminative learning scheme with active, online learning. Our learning procedure begins as the user labels several randomly chosen images

(on the order of several dozen). Our classifier learns on these images, and then examines the set of unlabeled images to select an additional set of images to be labeled. Once these images are labeled, the classifier trains on the newly labeled images, and again selects more images to be labeled. This process proceeds iteratively until sufficient classification accuracy is obtained, as determined by the user.

At the outset, it may seem curious to choose a supervised approach in light of the recent research emphasizing nontraditional supervision (as in Berg *et al.*) or fully automatic operation (as in Li *et al.* and Schroff *et al.*). As shown in the experiments of this work and its predecessors, even state-of-the-art techniques do not always achieve the level of performance necessary for large-scale, high-precision datasets. As such, our aim is to demonstrate performance which is superior to recent research with minimal levels of supervision, while also allowing the user to apply greater levels of supervision in an opportunistic fashion.

Confidence-Weighted Boosting. The basis of our discriminative learning scheme is confidence-weighted boosting [15]. Confidence weighted boosting is similar to AdaBoost [16], but differs in that weak learners yield a real-valued vote instead of a binary vote. In our case, we take as our weak learner decision stumps. Each potential decision stump partitions the training data into two disjoint sets. For a given set $i \in \{1, 2\}$, we let W_+^i be the sum of the weight of positive instances in the set, and likewise for W_-^i. In each round of boosting, we select the decision stump which minimizes the quantity

$$Z = \sum_{i \in \{1,2\}} \sqrt{W_+^i W_-^i} \tag{1}$$

Using these quantities, we can also determine the manner in which the stump will vote (as its decision is no longer a binary +1 or -1). Given that a particular instance falls into partition i, the classifier's vote for that instance is given by

$$c_i = \frac{1}{2} \ln \left(\frac{W_+^i}{W_-^i} \right) \tag{2}$$

Once a weak learner is selected, the weights of each training instance are updated as in AdaBoost. Our choice of confidence-weighted boosting was motivated by its excellent speed and accuracy.

Active Learning. Because the number of unlabeled images greatly outnumbers the number of labeled ones, it is natural to try to exploit the unlabeled data in some way. Active learning is one technique to do so, as it allows the machine learning engine to select a subset of the unlabeled data to be labeled. Our approach is to apply the learned classifier to the set of unlabeled data, and select the subset of images for which the predicted class is least certain (i.e., those for which the sum of the votes of our classifiers is closest to zero). The active learning method has been applied in vision to classify videos [17,18]; our boosting-based approach is most similar to that applied by Tur *et al.* [19] in a speech-processing application.

Active learning is particularly well-suited to the sort of data we expect to see from an internet crawler, as there will be many images which are highly similar (if not near-duplicates). It serves little good to label images which are very likely positive given the training data; active learning allows us to focus the user's attention on the examples which add richness and diversity to the dataset.

Online Learning. Each time we obtain a new set of labeled images, a naive approach might be to take the set of all images labeled thus far as our training set, and run our algorithm as usual. However, this discards everything which has been learned from previous stages of our active learning process. Several sophisticated online learning schemes have been proposed for boosting ([20,21]), but we consider two simple, heuristic schemes.

In the first scheme, we simply set our set of weak classifiers to be those obtained on the smaller set of data. We weight each new training instance as though it had been present during the previous stage of learning, though the weighted votes of our classifiers do not reflect the presence of these datapoints. We then apply the AdaBoost algorithm as usual for several rounds of boosting, in order to learn modalities not present in the smaller dataset.

In a slightly more sophisticated scheme, learning proceeds in two stages. In the first stage, we restrict the universe of weak classifiers to those which were obtained on the smaller set of labeled images. Boosting proceeds as usual, and the real-valued votes of these classifiers is recomputed to reflect the new dataset at each round of boosting. However, learning operates much more quickly at this stage than it normally would, as far fewer weak classifiers need be considered. Once a number of weak classifiers have been obtained using this method, we apply the AdaBoost as usual. As our experiments show, this method gives superior performance to the naive scheme at only marginally greater computational cost.

3 System Overview: Walkthrough of the Ape Category

In this section, we provide a walkthrough of the ape image category. Our crawling technique yields 21526 images, of which 5292 actually contain an image of an ape. Throughout this paper, we consider abstract images (e.g., drawings of apes, pictures of toys in the form of an ape) to be background images.

First, a descriptor vector is computed. We begin by extracting descriptors of several types: SIFT codeword histograms, filterbank response histograms, color histograms, downsampled pixel values, and search engine rank. The complete descriptor vector for an image is the concatenation of the vector obtained through the following methods:

- **SIFT codeword histograms:** A codebook of SIFT descriptor vectors [22] is formed by first extracting SIFT descriptors from a random subset of the crawled images (on the order of 1000 images). We run Fast K-means [23] to obtain 300 128-dimensional vectors, which form our codebook. We then again run the SIFT algorithm on each image in the crawled dataset, and match each keypoint descriptor into the codebook by choosing the codeword which

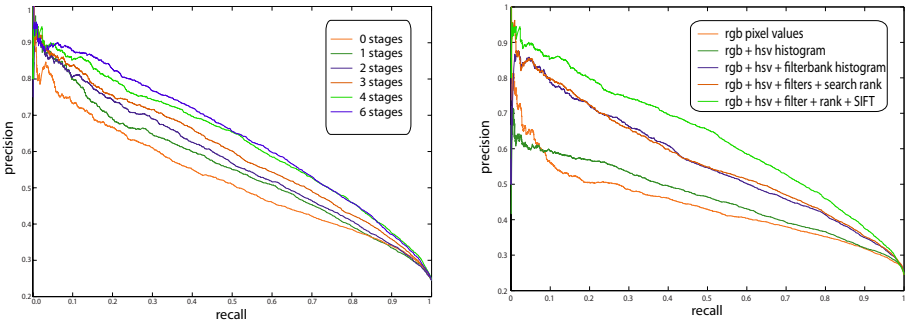
minimizes Euclidean distance. Finally, we compute a normalized histogram of codeword expression for each image.

- **Filterbank response histograms:** To obtain filterbank response histograms, we first convolve each image with each of 48 kernels, taken from the LM-48 filterbanks [24]. For each convolution response, we form an 11-bin histogram of responses.
- **Color histograms:** The color histogram descriptor for an image is simply a histogram of color expression in HSV-space.
- **Downsampled Pixel values:** Inspired by Torralba *et al.* [13], this descriptor is formed by simply applying histogram equalization, downsampling an image to 16x16 pixels, and giving the intensity of each pixel in RGB-space.
- **Image rank:** Finally, we preserve the rank order of each image in the search engines from which is was crawled. If an image was found on multiple search engines, this descriptor yields the minimum rank obtained. The motivation for this feature is that we expect our search engines to be effective at text processing, but to neglect image processing. We can incorporate the result of their text processing into our feature set without adding computational complexity to our pipeline.

Once these descriptors have been computed, learning begins. Figure 1 shows the process from the users' point of view: the initial set of randomly chosen images, the set of images chosen for active learning, and the classifier's top-ranked images after one stage of active learning. Note the types of images chosen for active learning: they can reflect some elements common in ape images (ape-like textures, natural scenery), but the positive images reflect elements less common for ape images, such as unusual viewpoints or scales.

Figure 2 shows how the performance of our classifier improves as stages of active learning progress, and shows the importance of each feature in our descriptor vector.

Fig. 1. Labeling process from user's point of view. **(left)** is a subset of 50 randomly chosen initial images. **(center)** shows some of the images selected for labeling after the first stage of learning. **(right)** shows highly-ranked images after 1 round of active learning.



(a) (b)

Fig. 2. Classification performance **(a)** by stages of active learning and **(b)** by feature set. This figure is best viewed in color.

4 Experiments and Results

In this section, we provide results of our learning algorithms in comparison to simpler alternatives, and also compare our whole-system performance to that of Berg *et al.*, Li *et al.*, and Schroff *et al.*

4.1 Performance of Learning Algorithms

Figure 3 provides convincing evidence that active learning dramatically reduces the number of labeled examples necessary to obtain high-precision classification. In each case we begin with 100 randomly chosen images; as active learning selects more images in a incremental fashion, the disparity between its performance benefit over passive learning increases. In particular, active learning with 250 labeled images outperforms passive learning with 400 labeled images.

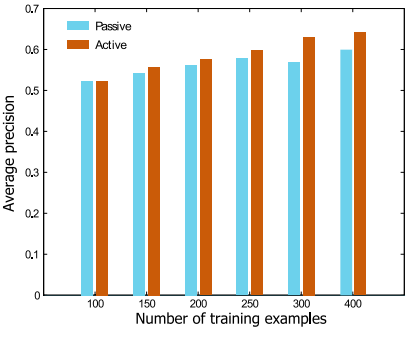


Fig. 3. Performance comparison of active learning and passive learning, under various numbers of training examples. In the case of active learning, we begin with 100 randomly chosen images, with the remaining labeled images chosen actively in increments of 50 images.

Table 1. Time and Classification Performance of Online Learning System. We evaluate on the ape dataset with 400 training examples, of which 300 were chosen actively. At each stage the number of rounds of boosting is proportional to the number of training images. For the online-naive approach, all weak learners from the previous stage are retained. For our online-relearning scheme, 50% of the weak learners at any stage are drawn from the weak learners of the previous stage, and the remainder are computed normally.

Algorithm	time	Area under precision-recall curve
Online-naive	91 s	.5743
Online-relearning	144 s	.6281
Batch relearning	221 s	.6415

Moreover, our online-learning approach yields improved time-complexity without hurting performance (table 1).

4.2 Comparison with OPTIMOL

In this section we compare our performance to that of Li *et al.*, using code provided by the authors. The training set for our algorithm consists of 200 images (100 images are chosen randomly; the remainder are chosen through two stages of active learning). OPTIMOL's supervision scheme is somewhat different, taking only a positive seed set. We thus feed OPTIMOL the positive images from our training set. The result of this comparison is presented in Figure 4.

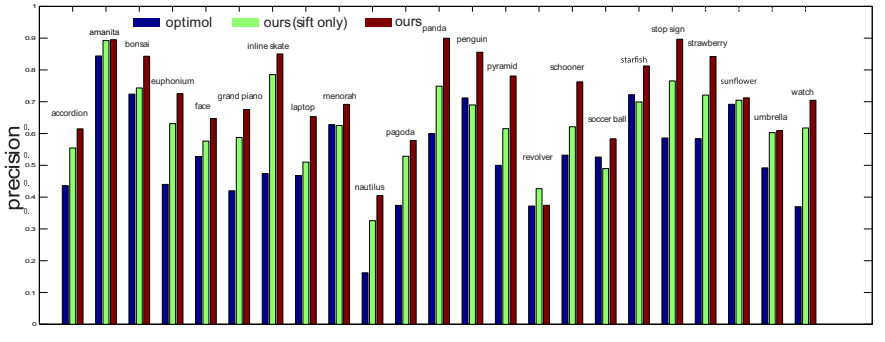


Fig. 4. Performance Comparison of OPTIMOL and our algorithm. We evaluate our algorithm using the complete feature set, and using only the SIFT codeword histograms. OPTIMOL is trained on the positive images from our training set. Our superior performance shows the effectiveness both of our learning mechanism and our feature set.

As a result of the sequential nature of OPTIMOLs classifier, there is no natural way to generate precision-recall curves. Instead, OPTIMOL returns a set of putatively positive images, the size of which cannot be specified *a priori*. We present the precision of this set of returned by OPTIMOL. To establish an

equal comparison, we obtain an equal-sized set of top-ranked images from our algorithm, and present the precision of this dataset. In order to evaluate the extent to which our performance difference is dependent on our feature set, we also present performance of our algorithm when our only feature is SIFT codeword histograms. As can be seen, in 22 of the 23 categories, the performance of our algorithm is superior to that of OPTIMOL. Moreover, in 20 of the 23 categories, our algorithm operating only on the SIFT histograms is superior, suggesting that our learning methodology is better-suited to this application than is that of OPTIMOL.

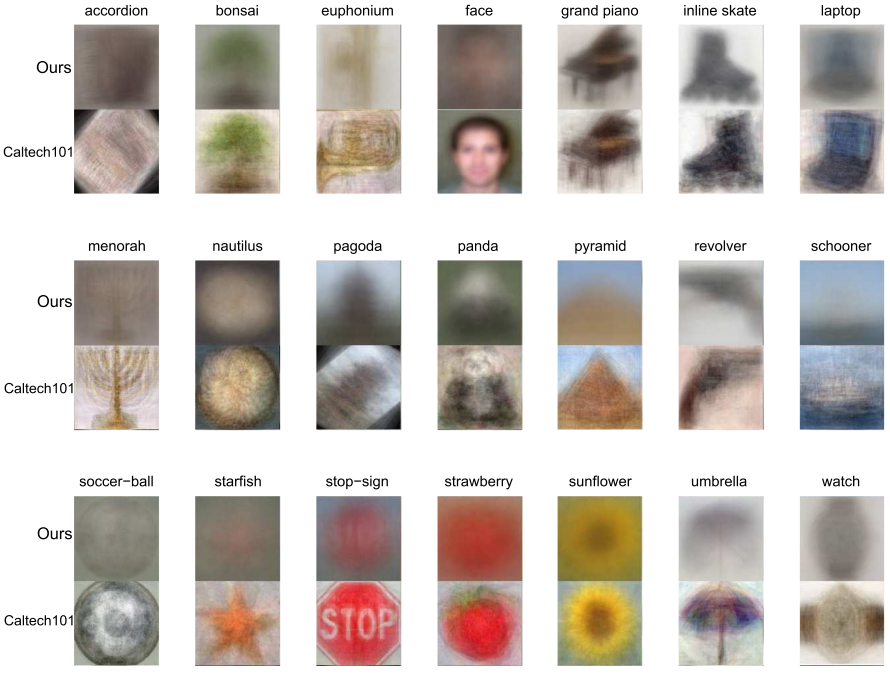


Fig. 5. Average image of our collected dataset, in comparison to that of Caltech-101. For each pair of rows, our average image is the top-most. False positives are not included in our average image. Our approach yields a diverse dataset, as active learning allows rapid learning of the most challenging cases.

We also present the relative runtimes of our systems in Figure 6. It is clear that our time performance is far superior, despite a greater feature set. Each round of boosting takes 5 ms per image, compared to 1.7 s for OPTIMOL. The significant speed advantage of our learning algorithm allows us to explore more descriptive feature sets; indeed, AdaBoost is "free" in comparison to feature extraction in terms of computational time.

To illustrate the diversity of the images we collect, Fig. 5 shows the average image comparison between the Caltech101 dataset [1] and our newly collected images. The images we collect show greater interclass variation. In short, our

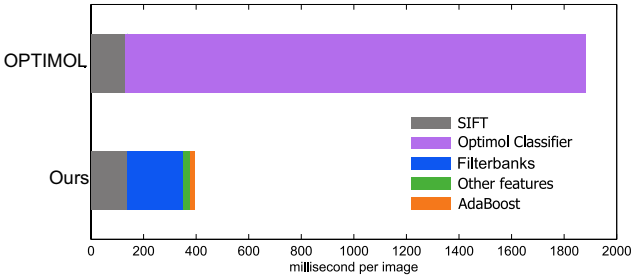


Fig. 6. Processing time of our algorithm and of OPTIMOL, per image, averaged over all categories. Because our learning algorithm is drastically faster than that of OPTIMOL, we are able to consider a richer feature set while still maintaining a high degree of scalability. This figure is best viewed in color.

approach has superior precision and superior time performance in comparison to OPTIMOL. As such, our approach shows much greater promise as a system scalable to very large datasets.

4.3 Evaluation on Animals on the Web Dataset

Two other major approaches, Berg *et al.* and of Schroff *et al.*, present results on the Animals on the Web (AoW) dataset; we consider our performance on this dataset.

Figure 7 presents the classification accuracy on the Berg dataset. Our results reflect only test data, given 150 labeled images. The first 50 images were selected randomly; the remaining 100 were selected using active learning in two stages. We present precision-recall curves for a number of these categories in Figure 8.

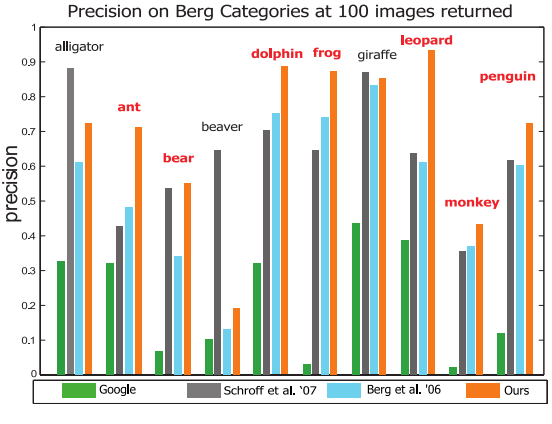
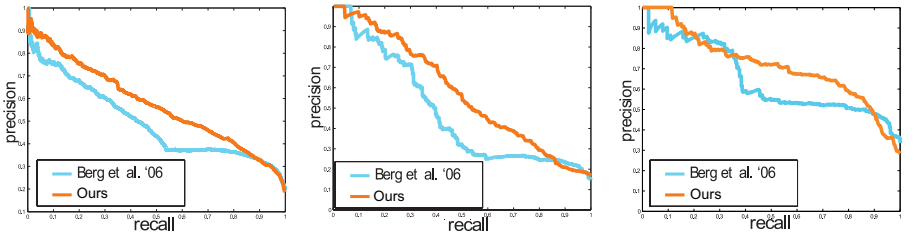


Fig. 7. Our precision on the Animals on the web dataset, in comparison to Schroff *et al.* and Berg *et al.* In seven of ten categories, our precision is superior; we mark these categories with red labels.



(a) monkey (b) frog (c) giraffe

Fig. 8. Precision-recall graphs for the categories of monkey, frog, and giraffe. Berg *et al.*'s performance is colored cyan; our results are in orange.

It is worth dwelling for a moment on the experimental conditions of Berg *et al.* and of Schroff *et al.*. Like Schroff, we compare our performance to that of the AoW test data, as results presented under "final dataset" include those directly labeled by the user. We believe that the amount of supervision is comparable between Berg *et al.* and this work; our user is asked to examine 150 images and will (assuming 25% precision of the crawled data) click on approximately 40 of them, as the user is only required to click on positive images. Similarly, Berg *et al.* present 10 topics, each of 30 images. The user in Berg *et al.* has the option of only clicking on 10 images. However, it is clear from their Figure 2 that much higher levels of high-precision recall are obtained when the additional step of swapping images between topics is performed. The Berg *et al.* performance we compare against reflect this additional supervision. The number of images examined by the user, as well as the number of images actually clicked, is comparable to that of Berg *et al.* [9].

It is more difficult to assess matters with respect to Schroff *et al.*. Though they indeed are fully automatic, they also avail themselves of a much larger, automatically harvested training set. It is likely that this aides them greatly in categories such as beaver, where the precision of the underlying dataset is very low. Though our training sets are smaller than those of Schroff *et al.*, minimal user supervision, targeted at the most informative examples using active learning, can provide performance which is comparable to the much larger training sets of their approach.

On seven of ten categories, our algorithm gives superior precision to both Schroff *et al.* and Berg *et al.*. In all ten categories, our performance is superior to that of Berg *et al.*. The precision-recall graphs of Figure 8 are also informative. Consider in particular the monkey dataset, as it is much larger than the others. The superior precision of our approach across the recall curve is testament to its scalability — users of our algorithm can expect to achieve very competitive performance regardless of the level of recall they desire.

4.4 Mammal Dataset

In order to demonstrate the scalability of our dataset, we aim to replicate the dataset of Fink and Ullman [5], which includes several esoteric mammal species. The Fink dataset consists of 400 mammalian image categories, with the candidate images obtained using Google Image Search. For each animal, the images are divided into five categories: irrelevant images, images which are not color photographs (which includes abstract images), color images which include a cropped version of the animal, color images with the full animal in a non-standard pose or view, and finally color images which include the animal in a standard pose.

We have replicated this dataset using the procedure presented in this paper, and made it available at `http://vision.cs.princeton.edu/easymammal.htm`. We do not separate our images into tiers as does Fink *et al.*; we consider his final three categories, plus black and white images, to be positive. Similarly, we do not provide the detailed annotation of his dataset. The labeling we employ consists of 100 randomly chosen images, followed by two stages of active learning with 50 images in each stage. Using an interface which requires the user to click on the positive images, it takes a user approximately 3 minutes to accomplish all three stages.

Table 2. For several mammal categories, we present the number of images present in the Fink *et al.* dataset, the number of images we obtain through crawling, and the precision we obtain at 200 images returned. For five of the ten categories, we obtain more images than Fink *et al.* within the first 200 images returned (precision bolded for these categories). We are very successful in the categories in which Fink obtains high recall ($n > 200$), with precision of 0.97 in four of five such categories.

Category	# of images in Fink	# of images crawled	Precision at 200 images returned
Aardvark	48	23152	**.885**
Camel	160	15951	**.95**
Fox	39	15842	**.925**
Chipmunk	301	14571	.975
Monkey	94	36765	**.94**
Hyena	293	19915	.97
Spotted Hyena	250	9170	.975
Brown Bear	213	25762	.91
Lion	60	27032	**.99**
Bengal Tiger	280	8981	.985

Here we highlight several categories. Table 2 gives the number of candidate images we obtained in comparison to Fink. Figure 9 shows the average image of the top 200 images returned. The superior number of crawled results (in comparison to the 1,000 maximum images returned by a single query to Google Image Search) is due to our use of query expansion (including latin scientific names, in most cases), and foreign language translation. Further, despite the limited

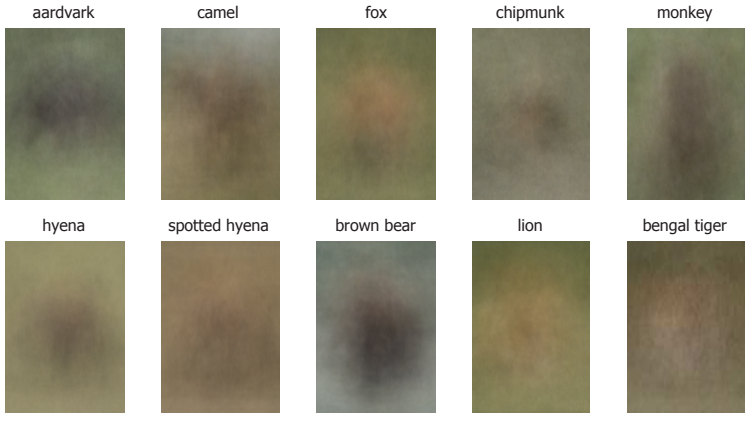


Fig. 9. Average image of the top 200 results for the ten animal categories in Table 2. False positives are not included in.

supervision we require, the precision we provide is quite high. A very rapid post-processing step could remove the false positives from this data, resulting in a large, diverse dataset with minimal effort on the part of the user.

5 Discussion

We have presented and evaluated a scalable and accurate dataset construction technique. Our diverse feature set and accurate machine learning technique allow for precision which is superior to the state-of-the-art. Moreover, the use of active learning serves to minimize the need for supervision, while allowing the user to opportunistically apply labels where necessary to improve the precision of the dataset. Finally, our use of features which are easy to compute efficiently and online learning allows for superior computational complexity. For the future, we intend to continue developing both the learning technique and the feature representations to improve our classification accuracy. It is also worthwhile to push for faster algorithms that can achieve real-time learning while users annotate.

References

1. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories (2004)
2. Griffin, G., Holub, A., Perona, P.: Caltech-256 object category dataset (2007)
3. Everingham, M., Zisserman, A., Williams, C.K.I., Van Gool, L.: The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results, `http://www.pascal-network.org/challenges/VOC/voc2006/results.pdf`
4. Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T.: Labelme: A database and web-based tool for image annotation. Int. J. Comput. Vision 77, 157–173 (2008)

5. Fink, M., Ullman, S.: From aardvark to zorro: A benchmark for mammal image classification. Int. J. Comput. Vision 77, 143–156 (2008)
6. Yao, B., Yang, X., Zhu, S.C.: Introduction to a large-scale general purpose ground truth database: Methodology, annotation tool and benchmarks, pp. 169–183 (2007)
7. Fergus, R., Perona, P., Zisserman, A.: A visual category filter for google images. In: Proceedings of the 8th European Conference on Computer Vision, Prague, Czech Republic, pp. 242–256 (2004)
8. Fergus, R., Fei-Fei, L., Perona, P., Zisserman, A.: Learning object categories from google's image search. In: Tenth IEEE International Conference on Computer Vision, 2005. ICCV 2005, 17-21 October 2005, vol. 2, pp. 1816–1823 (2005)
9. Berg, T.L., Forsyth, D.A.: Animals on the web. Computer Vision and Pattern Recognition, 1463–1470 (2006)
10. Li, J., Wang, G., Fei-Fei, L.: Optimol: automatic object picture collection via incremental model learning. Computer Vision and Pattern Recognition (2006)
11. Teh, Y.W., Jordan, M.I., Beal, M.J., Blei, D.M.: Hierarchical dirichlet processes. Journal of the American Statistical Association (2006)
12. Schroff, F., Criminisi, A., Zisserman, A.: Harvesting image databases from the web (2007)
13. Torralba, A., Fergus, R., Freeman, W.T.: Tiny images. Technical Report MIT-CSAIL-TR-2007-024, Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology (2007)
14. Princeton Cognitive Science Laboratory: Wordnet, `http://wordnet.princeton.edu`
15. Schapire, R.E., Singer, Y.: Improved boosting algorithms using confidence-rated predictions. Mach. Learn. 37, 297–336 (1999)
16. Schapire, R.: The boosting approach to machine learning: An overview. In: Hansen, M., Holmes, C., Mallick, B., Yu, B. (eds.) Nonlinear Estimation and Classification. Springer, Heidelberg (2003)
17. Yan, R., Yang, J., Hauptmann, A.: Automatically labeling video data using multi-class active learning. In: Eighth IEEE International Conference on Computer Vision. ICCV 2003, vol. 01, p. 516 (2003)
18. Abramson, Y., Freund, Y.: Semi-automatic visual learning (seville): a tutorial on active learning for visual object recognition. In: Computer Vision and Pattern Recognition (2005)
19. Hakkani-Tür, D., Riccardi, G., Tur, G.: An active approach to spoken language processing. ACM Trans. Speech Lang. Process 3, 1–31 (2006)
20. Oza, N.: Online bagging and boosting. In: IEEE International Conference on Systems, Man and Cybernetics, vol. 3, pp. 2340–2345 (2005)
21. Grabner, H., Bischof, H.: On-line boosting and vision. In: CVPR 2006: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, pp. 260–267. IEEE Computer Society Press, Los Alamitos (2006)
22. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision (2004)
23. Elkan, C.: Using the triangle inequality to accelerate k-means. In: Proceedings of the Twentieth International Conference on Machine Learning, pp. 147–153 (2003)
24. Leung, T., Malik, J.: Representing and recognizing the visual appearance of materials using three-dimensional textons (2001)