Harvesting Image Databases from the Web

Florian Schroff, Antonio Criminisi, and Andrew Zisserman

Abstract—The objective of this work is to automatically generate a large number of images for a specified object class. A multimodal approach employing both text, metadata, and visual features is used to gather many high-quality images from the Web. Candidate images are obtained by a text-based Web search querying on the object identifier (e.g., the word penguin). The Webpages and the images they contain are downloaded. The task is then to remove irrelevant images and rerank the remainder. First, the images are reranked based on the text surrounding the image and metadata features. A number of methods are compared for this reranking. Second, the top-ranked images are used as (noisy) training data and an SVM visual classifier is learned to improve the ranking further. We investigate the sensitivity of the cross-validation procedure to this noisy training data. The principal novelty of the overall method is in combining text/metadata and visual features in order to achieve a completely automatic ranking of the images. Examples are given for a selection of animals, vehicles, and other classes, totaling 18 classes. The results are assessed by precision/recall curves on ground-truth annotated data and by comparison to previous approaches, including those of Berg and Forsyth [5] and Fergus et al. [12].

Index Terms—Weakly supervised, computer vision, object recognition, image retrieval.

1 Introduction

The availability of image databases has proven invaluable for training and testing object class models during the recent surge of interest in object recognition. However, producing such databases containing a large number of images and with high precision is still an arduous manual task. Image search engines apparently provide an effortless route, but currently are limited by poor precision of the returned images and restrictions on the total number of images provided. For example, with Google Image Search, the precision is as low as 32 percent on one of the classes tested here (shark) and averages 39 percent, and downloads are restricted to 1,000 images.

Fergus et al. [11], [12], Lin et al. [20], Li et al. [19], Collins et al. [8], Vijayanarasimhan and Grauman [33], and Fritz and Schiele [14] dealt with the precision problem by reranking the images downloaded from an image search. The method in [12] involved visual clustering of the images by using probabilistic Latent Semantic Analysis (pLSA) [16] over a visual vocabulary, [19] used a Hierarchical Dirichlet Process instead of pLSA, and [33] uses multiple instance learning to learn the visual models. Lin et al. [20] reranked using the text on the original page from which the image was obtained. However, for all of these methods, the yield is limited by the restriction on the total number of images provided by the image search.

Berg and Forsyth [5] overcome the download restriction by starting from a *Web* search instead of an *image* search. This search can generate thousands of images. Their method then proceeds in two stages: First, topics are discovered based on words occurring on the Webpages using Latent Dirichlet Allocation (LDA) [6] on *text* only. Image clusters for each topic are formed by selecting images where nearby text is top ranked by the topic. A user then partitions the clusters into positive and negative for the class. Second, images and the associated text from these clusters are used as exemplars to train a classifier based on voting on visual (shape, color, and texture) and text features. The classifier is then used to rerank the downloaded data set. Note that the user labeling of clusters avoids the problem of polysemy, as well as providing good training data for the classifier. The method succeeds in achieving a greater yield, but at the cost of manual intervention.

Our objective in this work is to harvest a large number of images of a particular class *automatically*, and to achieve this with high precision. Our motivation is to provide training databases so that a new object model can be learned effortlessly. Following [5], we also use Web search to obtain a large pool of images and the Webpages that contain them. The low precision does not allow us to learn a class model from such images using vision alone. The challenge then is how best to combine text, metadata, and visual information in order to achieve the best image reranking.

The two main contributions are: First, we show in Section 3 that metadata and text attributes on the Webpage containing the image provide a useful estimate of the probability that the image is in class, and thence, can be used to successfully rank images in the downloaded pool. Second, we show in Section 4 that this probability is sufficient to provide (noisy) training data for a visual classifier, and that this classifier delivers a superior reranking to that produced by text alone. Fig. 1 visualizes this two-stage improvement over the initially downloaded images. The class-independent text ranker significantly improves this unranked baseline and is itself improved by quite a margin when the vision-based ranker

- *F. Schroff is with the Department of Computer Science and Engineering, University of California, San Diego, CA 92093. E-mail: gschroff@cs.ucsd.edu.*
- *A. Criminisi is with the Microsoft Research—Cambridge, 7 J.J. Thomson Avenue, Cambridge CB3 0FB, UK. E-mail: antcrim@microsoft.com.*
- *A. Zisserman is with the Robotics Research Group, Department of Engineering Science, University of Oxford, Parks Road, Oxford, OX1 3PJ, UK. E-mail: az@robots.ox.ac.uk.*

Manuscript received 26 Nov. 2008; revised 13 May 2009; accepted 11 July 2009; published online 19 July 2010.
Recommended for acceptance by B. Schiele.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2008-11-0816.
Digital Object Identifier no. 10.1109/TPAMI.2010.133.

0162-8828/11/$26.00 © 2011 IEEE Published by the IEEE Computer Society

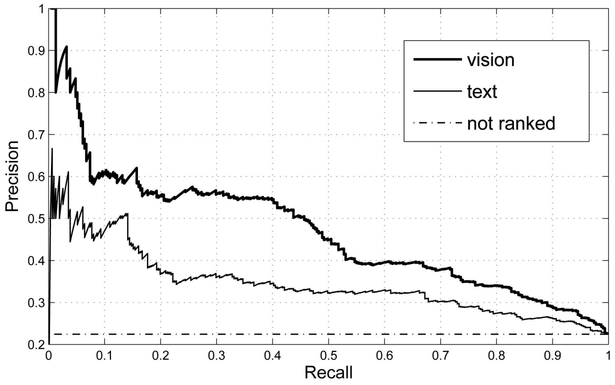


Fig. 1. **Text and visual ranking versus unranked baseline:** precision recall plot for the text reranking, the visual ranking trained on the text ranking, and the (unranked) original downloaded images, for the "shark" query.

(trained on the text ranker results) is employed. We compared our proposed discriminative framework (SVM) to unsupervised methods (topic models), concluding that the discriminative approach is better suited for this task, and thus, the focus of this work.

Others have used text and images together, however, in a slightly different setting. For example, Barnard et al. [2] use ground-truth annotated images as opposed to noisy annotation stemming from Webpages, as in our case. Other work by Berg et al. [4] uses text from the Internet, but focuses on identifying a specific class rather than general object classes.

We show in Section 5.1 that our automatic method achieves superior ranking results to those produced by the method of Berg and Forsyth [5] and also to that of Google Image Search.

This paper is an extended version of [28]. The extensions include: a comparison of different text ranking methods, additional visual features (HOG), an investigation of the cross validation to noise in the training data, and a comparison of different topic models (for the visual features).

2 The Databases

This section describes the methods for downloading the initial pool of images (together with associated metadata) from the Internet, and the initial filtering that is applied. For the purpose of training classifiers and for assessing precision and recall, the downloaded images are annotated manually for 18 classes: airplane (ap), beaver (bv), bikes (bk), boat (bt), camel (cm), car (cr), dolphin (dp), elephant (ep), giraffe (gf), guitar (gr), horse (hs), kangaroo (kg), motorbikes (mb), penguin (pg), shark (sk), tiger (tr), wristwatch (ww), and zebra (zb).

Data collection. We compare three different approaches to downloading images from the Web. The first approach, named `WebSearch`, submits the query word to Google Web search and all images that are linked within the returned Webpages are downloaded. Google limits the number of returned Webpages to 1,000, but many of the Webpages contain multiple images, so in this manner, thousands of images are obtained. The second approach, `ImageSearch`, starts from Google image search (rather than Web search). Google image search limits the number of returned images to 1,000, but here, each of the returned images is treated as a "seed"—further images are downloaded from the Webpage where the seed image originated. The third approach, `GoogleImages`, includes only the images directly returned by Google image search (a subset of those returned by `ImageSearch`). The query can consist of a single word or more specific descriptions such as "penguin animal" or "penguin OR penguins." Images smaller than 120 × 120 are discarded. In addition to the images, text surrounding the image HTML tag is downloaded, together with other metadata such as the image filename.

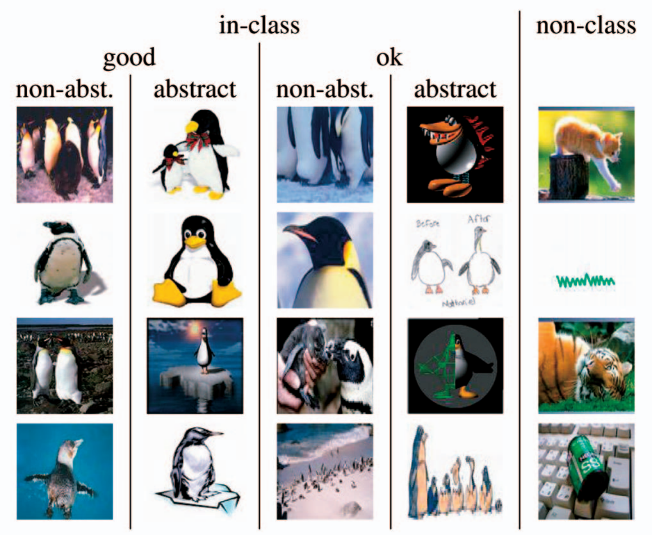


Fig. 2. **Image annotations:** example images corresponding to annotation categories for the class penguin.

Ground-truth annotation. In a similar manner to Fergus et al. [12], images are divided into three categories:

in-class-good. Images that contain one or many class instances in a clearly visible way (without major occlusion, lighting deterioration, or background clutter, and of sufficient size).

in-class-ok. Images that show parts of a class instance, or obfuscated views of the object due to lighting, clutter, occlusion, and the like.

nonclass. Images not belonging to *in-class*.

The good and ok sets are further divided into two subclasses:

abstract. Images that do not resemble realistic natural objects (e.g., drawings, nonrealistic paintings, comics, casts, or statues).

nonabstract. Images not belonging to the previous class.

Example annotations for the class penguin are shown in Fig. 2. The full data set is published in [29]. As is usual in annotation, there are ambiguous cases, e.g., deciding when occlusion is sufficiently severe to classify as *ok* rather than *good*, or when the objects are too small. The annotations were made as consistent as possible by a final check from one person. Note that the *abstract* versus *nonabstract* categorization is not general but is suitable for the object classes we consider in this paper. For example, it would not be useful if the class of interest was "graph" or "statue" or a similar more abstract category.

Table 1 details the statistics for each of the three retrieval techniques (`WebSearch`, `ImageSearch`, and

TABLE 1
Statistics by Source: The Statistics of Downloaded Images for Different Retrieval Techniques for 18 Classes

Service	in-class	non-class	precision
WebSearch	8773	25252	26%
ImageSearch	5963	135432	4%
GoogleImages	4416	6766	39%

TABLE 2
Image Class Statistics of the Original Downloaded Images Using WebSearch and GoogleImages Only, and After Applying the Drawing and Symbolic Images Removal Filter

	downloaded images			after drawing&symbolic filtering			
Class	in-cl.	non-cl.	prec.	in-cl.	non-cl.	prec.	false pos.
airplane (ap)	904	1659	35.27%	635	1007	38.67%	91
beaver (bv)	236	3121	7.03%	160	2195	6.79%	4
bikes (bk)	1268	1931	39.64%	983	1082	47.60%	111
boat (bt)	856	2175	28.24%	726	1354	34.90%	70
camel (cm)	594	1808	24.73%	485	1274	27.57%	46
car (cr)	1128	1042	51.98%	938	568	62.28%	92
dolphin (dp)	791	1416	35.84%	533	906	37.04%	81
elephant (ep)	937	1558	37.56%	763	1007	43.11%	11
giraffe (gf)	945	1267	42.72%	802	763	51.25%	32
guitar (gr)	1219	2035	37.46%	873	832	51.20%	248
horse (hs)	1229	1720	41.68%	975	1043	48.32%	78
kangaroo (kg)	418	1763	19.17%	329	1161	22.08%	14
motorbikes (mb)	732	953	43.44%	607	582	51.05%	86
penguin (pg)	748	1400	34.82%	447	794	36.02%	33
shark (sk)	583	1710	25.43%	413	1089	27.50%	60
tiger (tr)	379	2068	15.49%	311	1274	19.62%	17
wristwatch (ww)	941	957	49.58%	710	549	56.39%	220
zebra (zb)	483	1662	22.52%	416	987	29.65%	19
total	14391	30245	32.24%	11106	18467	37.55%	1313

GoogleImages). Note that some images are common between the methods. ImageSearch gives a very low precision (only about 4 percent) and is not used for the harvesting experiments. This low precision is probably due to the fact that Google selects many images from Web-gallery pages which contain images of all sorts. Google is able to select the in-class images from those pages, e.g., the ones with the object-class in the filename; however, if we use those Webpages as seeds, the overall precision greatly decreases. Therefore, we only use WebSearch and GoogleImages, which are merged into one data set per object class. Table 2 lists the 18 categories downloaded and the corresponding statistics for *in-class* and *nonclass* images. The overall precision of the images downloaded for all 18 classes is about 29 percent.

Due to the great diversity of images available on the Internet and because of how we retrieve the images, it is difficult to make general observations on how these databases look. However, it is clear that polysemy affects the returned images. Interestingly, this is not a problem that could be predicted directly from the English word since most of the classes we search for don't have direct polysemous meanings, i.e., they are not polysemous in the sense of "bank" (as in place to get money or river bank) for example. It is rather that the words correspond to brands or product names ("leopard tank") or team names (the NHL ice hockey team "San Jose Sharks") or are used as attributes ("tiger shark"). Apart from that, the *in-class* images occur in almost all variations imaginable, as sharks crashed into houses or other oddities. Even though context [31] can clearly be important in reranking the images (e.g., camel and kangaroo in desert-like images), it will have its limitations due to the variety of occurrences of the object.

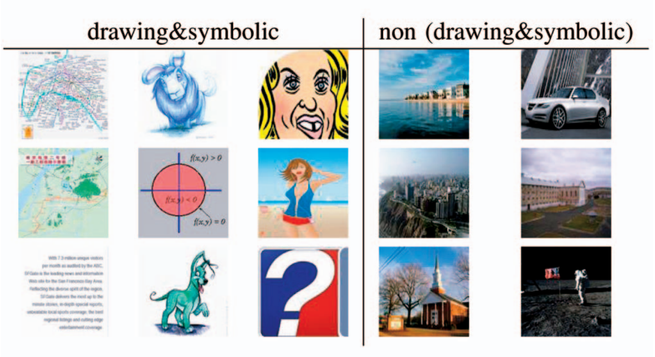


Fig. 3. **Drawings and symbolic images:** examples of positive and negative training images.

2.1 Removing Drawings and Symbolic Images

Since we are mostly interested in building databases for natural image recognition, we ideally would like to remove all *abstract* images from the downloaded images. However, separating *abstract* images from all others automatically is very challenging for classifiers based on visual features. Instead, we tackle the easier visual task of removing drawings and symbolic images. These include: comics, graphs, plots, maps, charts, drawings, and sketches, where the images can be fairly simply characterized by their visual features (see below). Example images are shown in Fig. 3. Their removal significantly reduces the number of *nonclass* images, improving the resulting precision of the object class data sets as shown in Table 2 (overall precision goes from 29 to 35 percent). Filtering out such images also has the aim of removing this type of *abstract* image from the *in-class* images.

Learning the filter. We train a radial basis function Support Vector Machine (SVM) on a hand-labeled data set (examples in Fig. 3). After the initial training, no further user interaction is required. In order to obtain this data set, images were downloaded using ImageSearch with one level of recursion (i.e., Webpages linked from "seed" Webpages are also used) with queries such as "sketch" or "drawing" or "draft." The goal was to retrieve many images and then select suitable training images manually. The resulting data set consists of approximately 1,400 drawings and symbolic images and 2,000 nondrawings and symbolic images.

Three simple visual only features are used: 1) a color histogram, 2) a histogram of the L2-norm of the gradient, and 3) a histogram of the angles $(0 \dots \pi)$ weighted by the L2-norm of the corresponding gradient. In all cases, 1,000 equally spaced bins are used. The motivation behind this choice of features is that drawings and symbolic images are characterized by sharp edges in certain orientations and or a distinctive color distribution (e.g., only few colors in large areas). The method achieves around 90 percent classification accuracy on the drawings and symbolic images database (using two-fold cross-validation).

This classifier is applied to the entire downloaded image data set to filter out drawing and symbolic images, before further processing. The total number of images that are removed for each class is shown in Table 2. In total, 39 percent of *nonclass* images are removed over all classes. The remaining images are those used in our experiments. As well as successfully removing *nonclass* images, the filter also succeeds in removing an average of 60 percent (123 images) *in-class abstract* images, with a range between 45 percent (for motorbikes, 40 images) and 85 percent (for wristwatch, 11 images). There is some loss of the desired *in-class nonabstract* images, with, on average, 13 percent (90 images) removed, though particular classes lose a relatively high percentage (28 percent for shark and wristwatch). Even though this seems to be a high loss, the precision of the resulting data sets is improved in all cases except for the class shark.

3 Ranking on Textual Features

We now describe the reranking of the returned images based on text and metadata alone. Here, we follow and extend the method proposed by Frankel et al. [13] in using a set of textual attributes whose presence is a strong indication of the image content.

Textual features. We use seven features from the text and HTML tags on the Webpage: *contextR, context10, filedir, filename, imagealt, imagetitle*, and *Websitetitle*.

Filedir, *filename*, and *Websitetitle* are self-explanatory. *Context10* includes the 10 words on either side of the image link. *ContextR* describes the words on the Webpage between 11 and 50 words away from the image link. *Imagealt* and *imagetitle* refer to the "alt" and "title" attribute of the image tag. The features are intended to be conditionally independent given the image content (we address this independence below). It is difficult to compare directly with the features in [13] since no precise definition of the features actually used is given.

Context here is defined by the HTML source, not by the rendered page, since the latter depends on screen resolution and browser type and is an expensive operation. In the text processing, a standard stop list [24] and the Porter stemmer [25] are used. In addition, HTML-tags and domain-specific stop words (such as "html" or "&nbsp;") are ignored.

We also experimented with a number of other features, such as the image MIME type ("gif," "jpeg," etc.), but found that they did not help discrimination.

3.1 Image Ranking

Using these seven textual features, the goal is to rerank the retrieved images. Each feature is treated as binary: "True" if it contains the query word (e.g., penguin) and "False" otherwise. The seven features define a binary feature vector for each image $\mathbf{a} = (a_1, \ldots, a_7)$, and the ranking is then based on the posterior probability, $P(y = in\text{-}class|\mathbf{a})$, of the image being *in-class*, where $y \in \{in\text{-}class, nonclass\}$ is the class label of an image.

We learn a class *independent* ranker in order to rerank the images based on the posterior $P(y|\mathbf{a})$. To rerank images for one particular class (e.g., penguin), we do not employ the ground-truth data for that class. Instead, we train the Bayes classifier (specifically we learn $P(\mathbf{a}|y)$, $P(y)$, and $P(\mathbf{a})$) using all available annotations *except* the class we want to rerank. This way, we evaluate performance as a *completely automatic class independent* image ranker, i.e., for any new and unknown class, the images can be reranked without *ever* using labeled ground-truth knowledge of that class.

3.1.1 Ranking Models

We compare different Bayesian posterior models for $P(y = in\text{-}class|\mathbf{a})$. Specifically, we looked at the suitability of the following decompositions:

Chow-Liu dependence tree decomposition [7]:

$$P(\mathbf{a}|y) \propto \prod_{1}^{8} P(x_i|x_{m(i)}) \tag{1}$$

with $x = (a_1, \ldots, a_7, y)$ and m being a permutation of $(1, \ldots, 8)$. The Chow-Liu model approximates the full joint dependency graph as a tree by retaining the edges between variables with the highest mutual information.

Naive Bayes model:

$$P(\mathbf{a}|y) \propto \prod_{1}^{7} P(a_i|y). \tag{2}$$

Given the class label for an image, the text features are assumed to be independent. For our application, this is "obviously" not the case, e.g., filename and image alternative tag are highly correlated.

Pairwise dependencies:

$$P(\mathbf{a}|y) \propto \prod_{i,j=1}^{7} P(a_i, a_j|y). \tag{3}$$

Only pairwise dependencies are modeled. This is similar to the Chow-Liu model, but less sparse.

Full joint:

$$P(\mathbf{a}|y) \propto P(a_1, \ldots, a_7|y). \tag{4}$$

The full joint probability distribution is learned. If the amount of available training data is too small, the learned model can be inaccurate.

Mixed naive Bayes:

$$P(\mathbf{a}|y) \propto P(a_1, \ldots, a_4|y) \prod_{5}^{7} P(a_i|y), \tag{5}$$

where $P(a_1, \ldots, a_4|y)$ is the joint probability of the first four textual features (*contextR, context10, filedir, and filename*).

Logistic regression. Additionally, we evaluate the performance using the discriminative logistic regression model, where

$$P(y|\mathbf{a}) = \frac{1}{1 + e^{-\mathbf{w}^T\mathbf{a}}}. \tag{6}$$

3.1.2 Text Reranking Results

We assess the performance by reporting precision at various points of recall as well as average precision in Table 3. Figs. 4 and 5 give an overview of the different methods and their performance. Fig. 6 shows the precision-recall curves for selected classes using the mixed naive

TABLE 3
Precision of Textual Reranking: The Performance of the Textual Reranking for All 18 Classes over Different Models: Precision at 15 Percent Recall, Precision at 100 Images Recall, and Average Precision

prec.	ap	bv	bk	bt	cm	cr	dp	ep	gf	gr	hs	kg	mb	pg	sk	tr	ww	zb	avg.
	Mixed Naïve Bayes																		
15%	40.61	30.00	**70.44**	52.22	49.65	61.16	70.27	70.70	82.39	63.18	57.72	53.93	67.94	68.42	**54.05**	**43.81**	78.03	48.41	59.05
100	39.00	31.00	76.00	52.00	47.00	**66.00**	69.00	**77.00**	**87.00**	58.00	58.00	**53.00**	67.00	**69.00**	**55.00**	**43.00**	79.00	50.00	**59.78**
avg.	41.67	31.17	58.66	48.97	48.17	66.46	59.18	**64.64**	**76.10**	65.71	57.21	40.95	**66.62**	61.53	42.55	37.61	70.59	52.97	**55.94**
	Pairs																		
15%	41.52	34.29	67.45	**54.64**	**50.00**	61.43	68.42	64.91	82.98	61.95	57.72	**57.14**	66.42	69.15	43.17	43.40	**81.10**	53.04	58.82
100	**43.00**	**36.00**	66.00	51.00	53.00	61.00	69.00	62.00	79.00	59.00	**61.00**	**53.00**	66.00	**69.00**	38.00	41.00	81.00	49.00	57.61
avg.	46.64	33.47	60.53	45.19	45.63	65.24	59.83	62.28	74.40	65.23	55.18	43.35	65.55	62.74	39.30	38.26	**71.69**	53.52	54.89
	Chow-Liu tree																		
15%	38.11	19.35	62.45	53.27	47.97	61.16	61.42	64.91	72.22	**68.65**	55.91	32.88	**69.53**	57.52	41.10	38.33	66.03	40.67	52.86
100	**43.00**	19.00	69.00	55.00	50.00	59.00	63.00	65.00	68.00	**70.00**	58.00	33.00	**72.00**	56.00	41.00	34.00	80.00	33.00	53.78
avg.	41.11	21.08	56.48	45.30	45.50	66.55	57.18	58.36	73.47	61.44	55.29	38.65	66.13	55.48	42.57	36.01	63.39	45.58	51.64
	Naïve Bayes																		
15%	**41.89**	**38.71**	67.45	54.36	**50.00**	**61.71**	70.27	64.91	81.25	61.95	57.49	**57.14**	66.42	**70.65**	43.17	41.07	**81.10**	**58.65**	**59.34**
100	**43.00**	**36.00**	66.00	51.00	**54.00**	61.00	68.00	62.00	79.00	59.00	**61.00**	**53.00**	66.00	**69.00**	37.00	40.00	81.00	**58.00**	58.00
avg.	**46.67**	**34.21**	60.58	45.22	45.49	65.11	59.85	62.32	74.36	65.52	55.24	**43.46**	65.55	**62.93**	39.79	38.43	**71.69**	52.13	54.92
	Full joint																		
15%	41.33	32.00	69.76	53.54	47.02	59.83	**71.56**	**72.55**	**85.40**	61.95	**58.92**	39.67	65.44	67.71	45.45	41.07	79.84	55.96	58.28
100	37.00	32.00	**77.00**	**56.00**	45.00	62.00	**71.00**	74.00	84.00	57.00	55.00	42.00	63.00	66.00	45.00	41.00	**82.00**	55.00	58.00
avg.	42.13	32.19	**60.59**	**49.26**	**49.09**	**66.64**	**62.60**	63.18	75.88	**65.75**	**57.64**	37.64	65.59	62.12	**44.92**	**38.64**	71.33	**54.13**	55.52

The precision is given as a percentage. The last column gives the average over all classes. Mixed naive Bayes performs best and was picked for all subsequent experiments.

model. It can clearly be seen that precision is highly increased at the lower recall levels compared to the average precision of Table 2.

The mixed model (5) gave slightly better performance than other factorizations *on 100 images recall*, and reflects the fact that the first four features are less independent of each other than the remaining three. Overall, all models perform comparably and the differences are negligible, except for the Chow-Lui dependence tree, which performs slightly worse. We chose the mixed naive model for our experiments as the 100 images recall is more related to our selection of training data than average precision or precision at 15 percent recall.

A separate set of experiments was carried out to measure how the performance of the text ranker varies with the number and choice of classes used for training. Ideally, we would like to compare $P(\mathbf{a}|y)$, $P(y)$, and $P(\mathbf{a})$ learned using different numbers of training classes. However, given our goal of ranking images, we instead compare these probabilities indirectly by assessing precision at 15 percent recall. We find that the performance is almost unaltered by the choice of training classes provided more than five classes (chosen randomly) are used for training.

Discussion. As can be seen in Fig. 6, the text reranker performs well, on average, and significantly improves the precision up to quite a high recall level (see Figs. 8 and 9 for top-ranked images). In Section 4, we will show that this is sufficient to train a visual classifier. For some classes, the text ranker performs very well (e.g., wristwatch, giraffe); for others, it performs rather poorly (e.g., airplane, beaver,

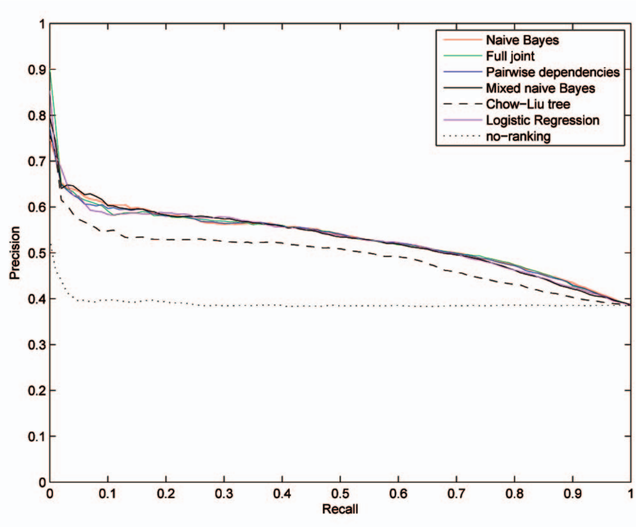


Fig. 4. **Average precision-recall of text rankers:** The precision-recall curve averaged over all 18 classes for each of the described ranking methods.

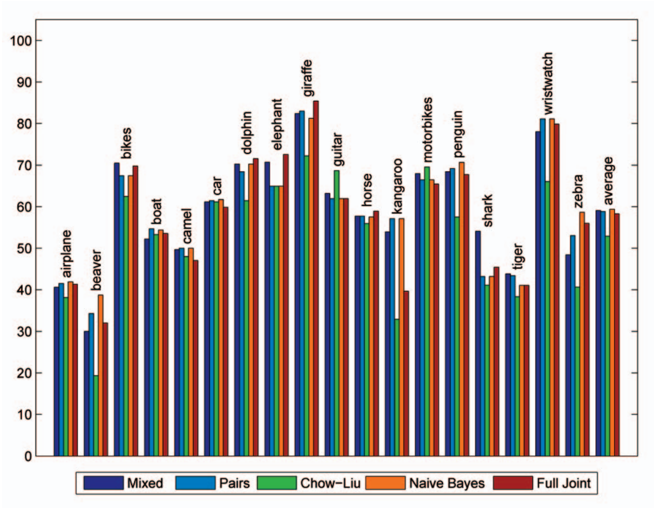


Fig. 5. **Comparison of text rankers:** The precision at 15 percent recall for all 18 classes and four different models. See Table 3 for details and the average precision over all classes.

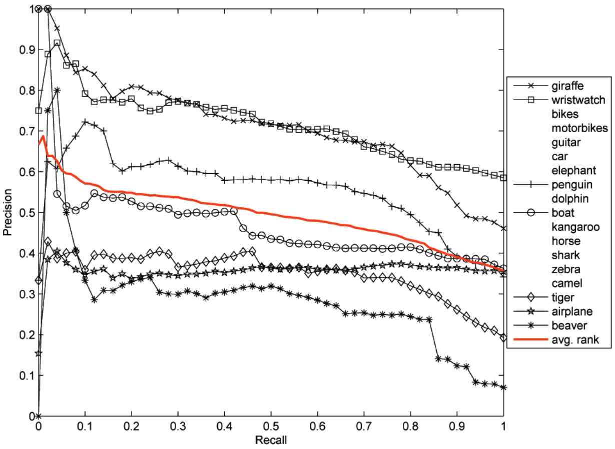


Fig. 6. **Text-based reranking:** precision versus recall estimated for each class with *abstract* images considered *in-class* using the mixed naive Bayes model. The labels are shown in decreasing order of precision at 15 percent recall. The recall precision curves are only shown for selected classes for clarity. The average over *all* 18 classes is also shown.

camel, and tiger). Visual inspection of the highly ranked "outlier" (*nonclass*) images in the text-ranked lists gives some explanation for these performances. Classes that perform well (wristwatch, giraffe) generally have outliers that are unrelated to each other. In contrast, for the classes that perform poorly, the outlier images are related and result from lack of discriminativity of the query word—for example, for airplanes, there are images of airplane food, airports, toy airplanes, paper airplanes, airplane interiors, and advertisements with comic airplanes. Other classes suffer from the type of polysemy described in Section 2: For camels, there are brand and cigarette-related outliers, and for tiger, there is the attribute problem with images of helicopters, tanks, fish (sharks), boxing, golf, stones, and butterflies.

These examples cover very different classes, from animals to various man-made objects. Generalization to other well-defined object classes is demonstrated in the experiments of Section 5.1 on the object classes of [5] and [12].

We investigated two alternative text-based classifiers, pLSA and SVM, in addition to the Bayes estimator finally adopted, but found that they had inferior performance. For the SVM, the same binary text features $\mathbf{a}$ were used. It is possible that the binary features led to the poor performance of the SVM. For the pLSA, we used *context10* and *contextR* (similar to [5]). Due to the problems in the pLSA clustering, the problem of how to select the right topic without user interaction as in [5], and the question of how to use the additional binary features (e.g., *filename*) in a principled manner, we adopted the Bayes estimator instead.

The text reranking is not meant to compete with the performance of Internet image search engines, which could also be used in our algorithm and, given their recent performance improvements, would be a reasonable choice. Instead, we decided to use this simple setup to gather the training data for our visual models, with the focus of independence and a more controlled setup, considering the fast changing quality of image search engines.

4 Ranking on Visual Features

The text reranking of Section 3 associates a posterior probability with each image as to whether it contains the query class or not. The problem we are now faced with is how to use this information to train a visual classifier that would improve the ranking further. The problem is one of training from noisy data: We need to decide which images to use for positive and negative training data and how to select a validation set in order to optimize the parameters of the classifier.

We first describe the visual features used and then how the classifier is trained.

Visual features. We follow the approach of [12] and use a variety of region detectors with a common visual vocabulary in the bag of visual words model framework (BOW). All images are first resized to 300 pixels in width. Regions are detected using difference of Gaussians, Multiscale-Harris [22], Kadir's saliency operator [18], and points sampled from Canny edge points. Each image region is represented as a 72-dimensional SIFT [21] descriptor. A separate vocabulary consisting of 100 visual words is learned for each detector using k-means, and these vocabularies are then combined into a single one of 400 words. Finally, the descriptor of each region is assigned to the vocabulary. The software for the detectors is obtained from [32]. Fuller implementation details are given in [12] and are reproduced in our implementation. The 72-dimensional SIFT is based on the work in [12] and driven by the motivation that a coarser spatial binning can improve generalization, as opposed to a finer binning that is more suitable for particular object matching.

In addition to the descriptors used in [12], we add the widely used HOG descriptor [9], computed over the whole image to incorporate some spatial layout into the model. It was also used in a similar setting in [14]. We use a cell size of 8 pixels, a block size of one cell, and 9 contrast invariant gradient bins. This results in a 900-dimensional feature vector, as the images were resized to 80×80 pixels. The two descriptors are concatenated resulting in a 1,300-dimensional feature vector per image.

4.1 Training the Visual Classifier

At this point, we can select n_+ positive training images from the top of the text-ranked list, or those that have a posterior probability above some threshold, but a subset of these positive images will be "noisy," i.e., will not be *in-class*. Table 4 (text) gives an idea of the noise from the proportion of outliers. It averages 40 percent if $n_+ = 100$. However, we can assume that the *nonclass* images are *not* visually consistent—an assumption verified to some extent by the results in Section 4.2. The case of negative images is more favorable: We select n_- images at random from all downloaded images (i.e., from all 18 classes, tens of thousands of images) and the chance of any image being of a particular class is very low. We did not choose to select the n_- images from the low-ranked images of the text ranker output because the probability of finding *in-class* images there is higher than finding them in the set of *all* downloaded images.

Given this situation, we choose to use an SVM classifier since it has the potential to train despite noise in the data. The SVM training minimizes the following sum [23]:

TABLE 4
Comparison of Precision at 15 Percent Recall

prec. 15%	ap	bv	bk	bt	cm	cr	dp	ep	gf	gr	hs	kg	mb	pg	sk	tr	ww	zb	avg.	std
text	40.61	30.00	70.44	52.22	49.65	61.16	70.27	70.70	82.39	63.18	57.72	53.93	67.94	68.42	54.05	43.81	78.03	48.41	59.05	–
t+v (250/250)	54.0	36.4	64.8	60.9	53.5	78.5	61.9	77.5	86.5	65.4	60.6	53.5	78.2	55.5	56.8	45.7	81.6	**96.8**	64.9	9.3 (2.5)
t+v (150/500)	59.1	33.8	**69.7**	63.4	56.8	91.8	61.6	70.7	83.2	**67.3**	67.0	50.0	77.4	70.4	69.3	59.6	86.3	85.5	67.9	8.3 (2.0)
t+v (250/500)	58.6	35.0	64.6	**65.1**	47.8	79.0	60.1	75.4	86.9	51.7	68.4	50.2	77.7	69.7	64.3	39.8	90.4	87.9	65.2	8.8 (2.0)
t+v (250/1000)	**63.5**	32.3	65.9	62.4	51.6	93.0	61.7	80.2	87.8	62.6	**71.2**	45.5	**84.6**	69.6	64.9	53.0	86.6	95.8	68.4	7.9 (1.9)
t+v (150/1000)	52.3	39.3	68.6	**66.2**	57.3	87.9	66.5	78.7	83.8	63.0	57.3	48.6	64.0	72.5	**82.9**	**62.6**	**93.2**	79.8	68.0	7.2 (2.0)
t+v (150/1000) C	49.8	42.7	**71.0**	**66.2**	55.1	88.8	64.7	78.0	85.7	62.3	49.2	**56.2**	67.0	72.6	69.8	58.5	88.9	86.1	67.4	8.3 (2.1)
HOG t+v	45.1	21.1	53.2	62.0	49.1	80.4	51.6	71.6	78.7	62.9	64.1	43.0	68.2	77.8	47.7	29.8	87.6	48.0	57.9	6.6 (1.8)
HOG+BOW t+v	42.8	**46.4**	70.4	60.9	54.5	**93.5**	67.7	**84.8**	88.3	62.5	68.8	55.2	72.1	78.9	80.7	57.3	89.7	81.4	69.8	8.1 (2.1)
HOG+BOW t+v C	51.3	42.3	68.2	60.3	**63.2**	91.1	**69.7**	78.7	**88.7**	66.3	70.1	53.9	76.6	**90.2**	66.1	50.2	92.5	91.7	**70.6**	7.2 (1.7)
gt (150/1000)	83.1	90.8	75.8	76.1	78.0	98.6	78.2	96.0	91.4	88.8	90.2	69.0	95.5	82.7	91.8	94.3	96.1	93.3	87.2	8.4 (2.0)
(B) (150/1000)	52.4	12.9	55.4	63.6	54.1	94.9	42.6	47.9	83.7	61.4	52.8	29.8	65.2	53.7	42.9	28.5	82.7	78.3	55.7	6.7 (1.7)

"text" refers to text reranking alone, "t+v" is text+vision reranking using different training ratios n_+/n_-, "gt" is ground-truth (only positive images) training of the visual classifier, and (B) is the baseline where the visual classifier is trained on $n_+ = 150$ images uniformly sampled from the filtered images of one class instead of the text reranked images, and $n_- = 1{,}000$ background images as before. The second to last column (avg.) gives the average over all classes. The last column states the mean of the classwise standard deviations over five runs of cross validation, as well as the standard deviation of the means over all classes, in parentheses.

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + C_+\sum_{i:y_i=1} \xi_i + C_-\sum_{j:y_j=-1} \xi_j \tag{7}$$

$$\text{subject to} \quad y_l\big(\mathbf{w}^T\Phi(\mathbf{x}_l) + b\big) \geq 1 - \xi_l, \tag{8}$$

$$\xi_l \geq 0, l = 1, \dots, (n_+ + n_-), \tag{9}$$

where $\mathbf{x}_l$ are the training vectors and $y_l \in \{1, -1\}$ the class labels. C_+ and C_- are the false classification penalties for the positive and negative images, with $\boldsymbol{\xi}$ being the corresponding slack variables.

To implement the SVM, we use the publicly available SVM$^{\text{light}}$ software [17] (with the option to remove inconsistent training data enabled). Given two input images I_i and I_j and their corresponding normalized histograms of visual words and HOG, S_i and S_j, this implementation uses the following χ^2 radial basis function (RBF) kernel: $K(S_i, S_j) = exp(-\gamma \cdot \chi^2(S_i, S_j))$ [35], with γ the free kernel parameter.

Thus, γ, C_+, and C_- are the three parameters that can be varied. The optimal value for these parameters is obtained by training the SVM using 10-fold cross validation. Note that we do not use the ground-truth at any stage of training, but we split the *noisy* training images into 10 training and validation sets. That is, the validation set is a subset of the n_+ top text-ranked images as well as the n_- background images. We require a performance measure for the cross validation and use precision at 15 percent recall, computed on the validation subset of the n_+ images (treated as positive) and the n_- images (treated as negative). This parameter selection is performed automatically for *every* new object class.

Sometimes, C_+ and C_- are used to correct unbalanced training data [23]. In our case, however, the SVM is very sensitive to these parameters, probably due to the huge amount of noise in the data, and the optimal value does not directly correspond to the ratio of positive to negative images.

Finally, the trained SVM is used to rerank the filtered image set based on the SVM classification score. The entire image harvesting algorithm is summarized in Fig. 7.

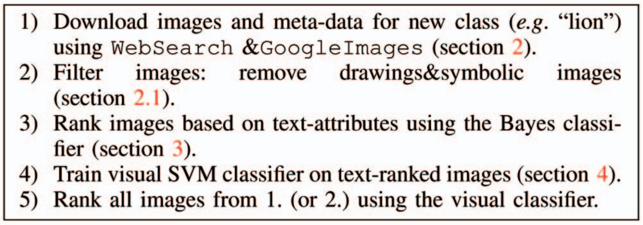
4.2 Results for Textual/Visual Image Ranking

In this section, we evaluate different combinations of training and testing. If not stated otherwise, the text+vision system of Fig. 7 was used. Results are given in Table 4 for various choices of n_+ and n_-. For each choice, five different random selections are made for the sets used in the 10-fold cross validation, and mean and standard deviation are reported. The clear improvement brought by the visual classifier over the text-based ranking for most classes is obvious. Figs. 8 and 9 compare the top-ranked images from the text and vision steps.

We first investigate how the classification performance is affected by the choice of n_+ and n_-. It can be seen that increasing n_- tends to improve performance. It is, however, difficult to select optimal values for n_+ and n_- since these numbers are very class dependent. Table 4 indicates that using more images in the background class n_- tends to improve the performance but there is no real difference between using $150/1{,}000$ and $250/1{,}000$ (n_+/n_-), which perform at $68.4\% \pm 1.9$ and $68.0\% \pm 2.0$, and thus are not significantly different. All numbers in this section report precision at 15 percent recall.

It can be seen (Table 4) that HOG alone performs significantly worse than the bag of visual words $57.9\% \pm 1.8$, but the combination of BOW and HOG improves the overall performance to $69.8\% \pm 2.1$, compared to BOW alone $68.0\% \pm 2.0$.

In order to select the appropriate parameter values, we use cross validation, where the validation set is part of the n_+ and n_- images as described in Section 4.1, together with precision at 15 percent recall as selection criterion. There are two possible cases that can occur: 1) a parameter setting that overfits to the training data. This problem is detected on the



1) Download images and meta-data for new class (*e.g.* "lion") using `WebSearch` &`GoogleImages` (section 2).
2) Filter images: remove drawings&symbolic images (section 2.1).
3) Rank images based on text-attributes using the Bayes classifier (section 3).
4) Train visual SVM classifier on text-ranked images (section 4).
5) Rank all images from 1. (or 2.) using the visual classifier.

Fig. 7. **Overview** of the text+vision (t+v) image harvesting algorithm.

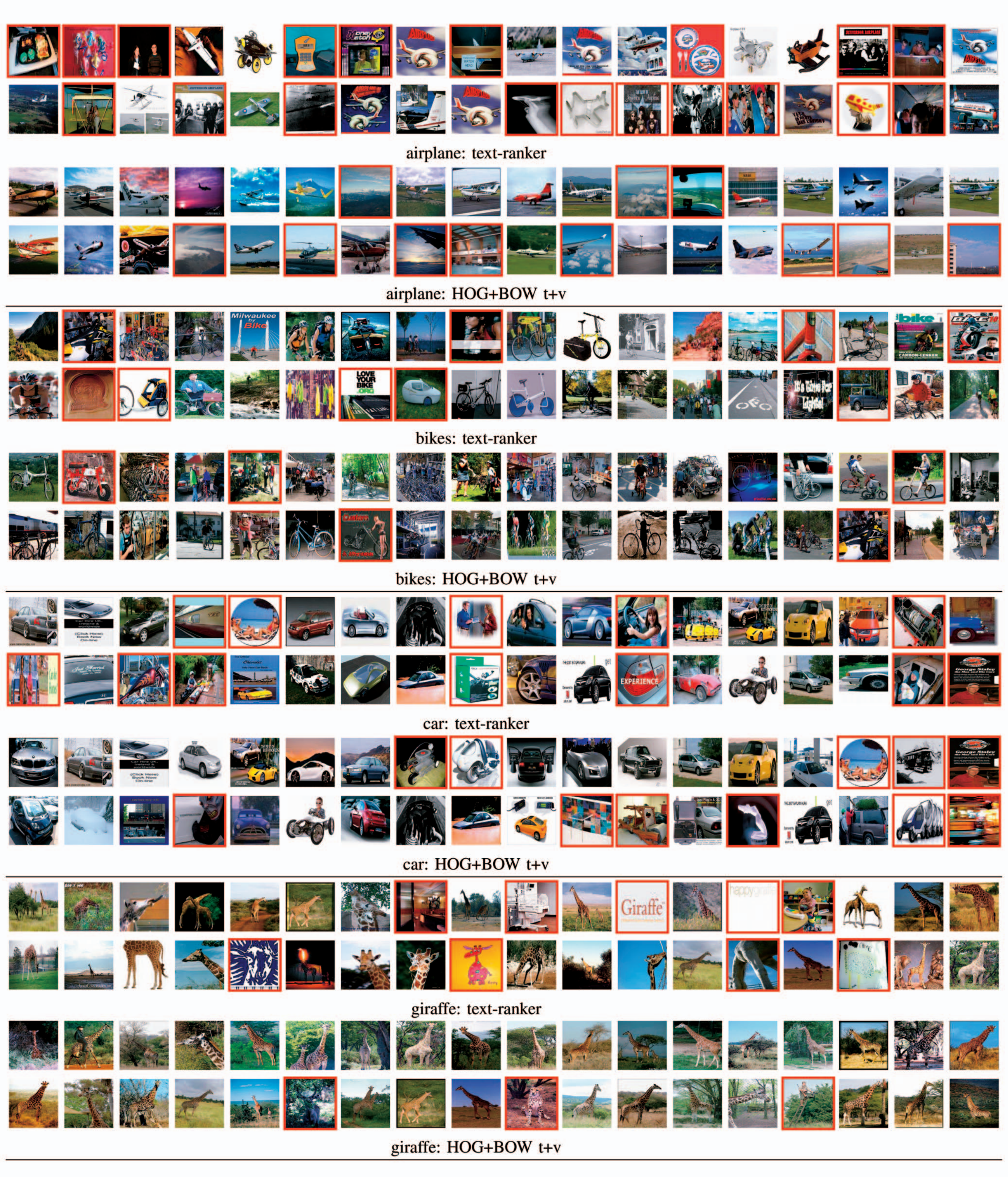


Fig. 8. **Comparing the top-ranked 36 images using the text ranker and the full system:** Red boxes indicate false positives.

validation set due to a low precision at 15 percent recall. 2) All images (training and validation sets) are classified as background. This leads to bad, but detectable, performance as well.

Here, we describe a slight adjustment to this method, which ignores "difficult" images. Parameter settings that classify (almost) all images as fore or background are not useful; neither are those that overfit to the training data. We reject those parameter settings. We then use the "good" parameter settings to train and classify all images. By looking at the distribution of SVM responses (over all parameter settings), we are able to eliminate "intermediate" images, i.e., images that are not classified as positive or negative images in the majority of cases. We assume that

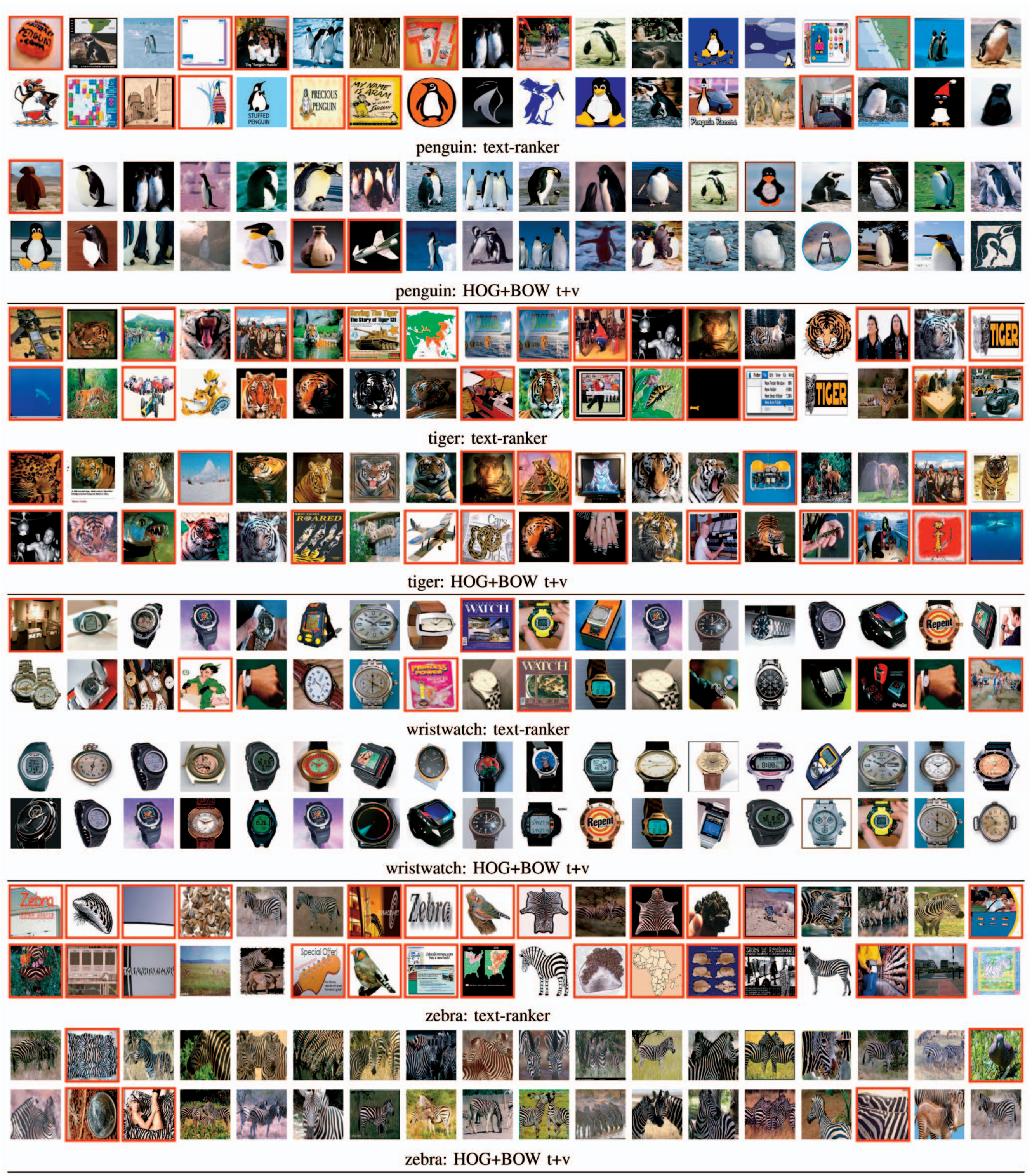


Fig. 9. **Comparing the top-ranked 36 images using the text ranker and the full system:** Red boxes indicate false positives.

those images are difficult to classify and we don't use those in our model selection step because we cannot rely on their class labels being correct due to the noise. Note that training is still performed on *all* images, and the "difficult" images are only discarded during the computation of the precision at 15 percent recall during the cross validation.

This method does not give a significant improvement over all classes, but improves the performance dramatically for some classes, e.g., penguin $90.2\% \pm 5.0$ from $78.9\% \pm 13.2$. This modified version of the cross validation is denoted by "C" in Table 4.

We next determine how much the performance is affected by the noise in the training data by training the SVM on ground-truth positive data, i.e., instead of selecting n_+ images from the text-ranked images, we select n_+ *in-class* images using the ground-truth labeling. We find that the text+vision

TABLE 5
Comparing Filtered versus Nonfiltered Images and Baseline

prec. 15&	ap	bv	bk	bt	cm	cr	dp	ep	gf	gr	hs	kg	mb	pg	sk	tr	ww	zb	avg.	std.
tf rf	42.8	46.4	70.4	60.9	54.5	93.5	67.7	84.8	88.3	62.5	68.8	55.2	72.1	78.9	80.7	57.3	89.7	81.4	**69.8**	8.1 (2.1)
tf ru	38.7	39.0	68.7	62.3	55.5	93.4	63.6	81.8	84.4	64.3	68.9	53.0	74.7	80.0	79.1	58.2	92.5	80.2	68.8	6.6 (1.8)
tu ru	49.9	32.1	61.4	64.1	55.1	82.0	63.9	62.2	90.1	46.6	41.6	49.3	78.8	80.6	62.0	34.7	91.4	78.2	62.4	9.2 (2.5)
tf rf (B)	52.4	12.9	55.4	63.6	54.1	94.9	42.6	47.9	83.7	61.4	52.8	29.8	65.2	53.7	42.9	28.5	82.7	78.3	55.7	6.7 (1.7)

tf and (tu) denote training on filtered (unfiltered) images, rf and (ru) ranking on filtered (unfiltered) images. The first row gives the results for the whole system (HOG+BOW (t+v) in Table 4). The second row shows the case where all *images were reranked, including the ones that were filtered out in Section 2.1. The third row shows training and ranking on unfiltered images. The last row gives results for the baseline (B) method, where the visual classifier was trained on images selected from the initial database of that class, i.e., no text ranking applied, and tested on unfiltered images. The last column states the mean of the classwise standard deviations over five runs of cross validation, as well as the standard deviation of the means over all classes, in parentheses.*

system performs well for the classes where the text ranking performs sufficiently well (see Figs. 8 and 9 for the top images returned by the text ranker). HOG+BOW versus gt, e.g., car $93.5\% \pm 7.1$ versus $98.6\% \pm 1.1$, or giraffe $88.3\% \pm 2.7$ versus $91.4\% \pm 4.8$. If the text ranking fails, the ground-truth performs, as is to be expected, much better than the text ranked-based training, e.g., for airplane, camel, and kangaroo. These experiments show that the SVM-based classifier is relatively insensitive to noise in the training data as long as a reasonable noise level is given.

As a baseline comparison, we investigate the performance if no text reranking is used, but the n_+ images are sampled uniformly from the filtered images. If the text reranking works well and hence provides good training data, then text+vision improves over the baseline, e.g., elephant $84.8\% \pm 3.3$ versus $47.9\% \pm 4.0$, penguin $78.9\% \pm 13.2$ versus $53.7\% \pm 3.4$, or shark $80.7\% \pm 4.2$ versus $42.9\% \pm 14.4$. In cases where the text ranking does not perform well, the baseline can even outperform text+vision. This is due to the fact that bad text ranking can provide visually consistent training data which does *not* show the expected class (e.g., for airplanes, it contains many images showing airplane food, inside airplanes/airports, taken out of the window of an airplane). However, the uniformly sampled images still consist of about 35 percent *in-class* images (Table 2) and the n_- are very unlikely to contain *in-class* images.

In addition to reranking the filtered images, we applied the text+vision system to all images downloaded for one specific class, i.e., the drawings and symbolic images were included. It is interesting to note that the performance is comparable to the case of filtered images (Table 5). This means that the learned visual model is strong enough to remove the drawings and symbolic images during the ranking process. Thus, the filtering is only necessary to train the visual classifier and is not required to rank new images, as evident from rows 2 and 3 "tf ru" and "tu ru" in Table 5. "tf ru" performs almost as well as "tf rf," $68.8\% \pm 1.8$ versus $69.8\% \pm 2.1$, i.e., using unfiltered images during testing does not affect the performance significantly. However, using unfiltered images during training "tu ru," decreases the performance significantly, down to $62.4\% \pm 2.5$. The main exception here is the airplane class, where training with filtered images is a lot worse than with unfiltered images. In the case of airplane, the filtering removed 91 good images and the overall precision of the filtered images is quite low, 38.67 percent, which makes the whole process relatively unstable, and therefore can explain the difference.

Discussion. Training and evaluating the system with the goal of building natural image databases, i.e., treating *abstract* images as *nonclass* in all stages of the algorithm, as opposed to treating *abstract* images as *in-class*, gives similar performance; however, it is slightly worse. The slight drop in performance can be explained by the fact that the text ranker inevitably returns *abstract* images, which are then used as training images. That our method is applicable to both of those cases is further supported by the results we retrieve on the Berg et al. data set (see Section 5.1).

We also investigated alternative visual classifiers, topic models (see Section 5.2), and feature selection. For feature selection, our intention was to find discriminative visual words and then use these in a Bayesian framework. The discriminative visual words were obtained based on the likelihood ratio of a visual word occurring in the foreground to background images [10]. However, probably due to the large variation in both the foreground and background images, together with the noisy training data, we weren't able to match the performance of the SVM ranker.

We found that combining the vision and text-ranked lists using Borda count [1] or similar methods gave a slight improvement, on average, but results were very class dependent. Combining the probabilistic outputs of text and SVM as in [34] remains an interesting addition for future work. The advantage of our system is that, once trained, we can rank images for which no metadata are available, e.g., the Fergus and Berg images, see the next section.

5 Comparison with Other Work and Methods

We compare *our* algorithm with three other approaches. We report the results for $n_+ = 150, n_- = 1{,}000$, and HOG+BOW (see highlighted versions in first column of Table 4). Again, we report mean and standard deviation over five runs of 10-fold cross validation.

5.1 Comparing with Other Results

Comparison with Google image search. Here, we rerank the images downloaded with `GoogleImages` with our fully automatic system, i.e., text-based training of visual classifier followed by visual reranking of `GoogleImages`. Comparative results between our approach and `GoogleImages` are shown in Fig. 10. As can be observed, our approach achieves higher average precision for 16 out of 18 classes with only airplane and boat being outperformed by the Google results. Those are cases where our text ranker doesn't perform that

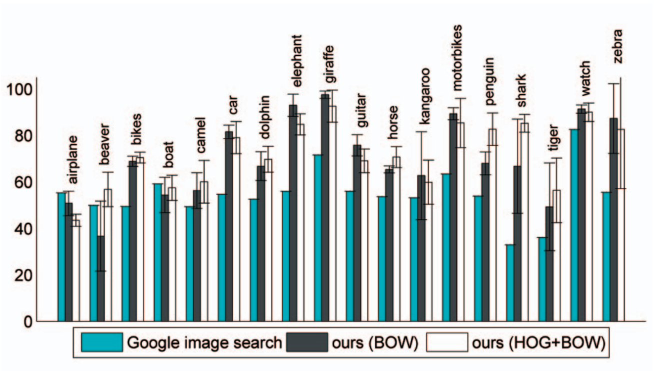


Fig. 10. **Comparison with Google image search:** Precision at 100 image recall. Compare to highlighted versions in first column of Table 4 for *our* algorithm.

well, which increases the noise in the training data and thus explains the decreased visual performance.

Images for the class penguin returned from both Google and our reranking (using the learned visual classifier applied to the Google images) are shown in Fig. 11.

Comparison with Fergus et al. [12] data. In this experiment, we rerank the Google images provided by Fergus et al. [12]. In order to do so, we train on the downloaded images and their corresponding metadata. We downloaded "leopard" in addition to the previously described classes. It is difficult to directly compare results in [12] to our text+vision algorithm as [12] treats *ok* images as nonclass, whereas our system is not tuned to distinguish *good* from *ok* images. Due to this, our system performs slightly worse than [12] when measured only on *good* images. However, it still outperforms Google image search on most classes, even in this case. The same holds for the comparison with [14], which performs very well on this data set. Table 6 also shows (first row) the results when *ok* images from the [12] data are treated as in-class. As expected, the performance increases significantly for all classes. It needs to be noted that this data set is relatively small and the distribution of images is not always diverse (e.g., many close duplicates in the airplane category).

TABLE 6
Comparison with Fergus et al. [12] Data: Average Precision at 15 Percent Recall with One Standard Deviation

	airplane	guitar	leopard	motorbike	wristwatch
our	58.5 ± 25.8	70.0 ± 3.9	49.6 ± 9.9	74.8 ± 7.3	98.1 ± 3.0
our (øk)	41.8 ± 18.4	31.7 ± 4.1	22.6 ± 10.2	50.5 ± 8.0	94.4 ± 3.2
[12] (øk)	57	50	59	71	88
Google (øk)	50	30	41	46	70
Fritz [14]	100	91	65	97	100

The images are from Google image search and were provided by the author. (øk) uses the same annotation as [12]. The first row of the table treats Fergus' ok class as in-class, unlike [12].

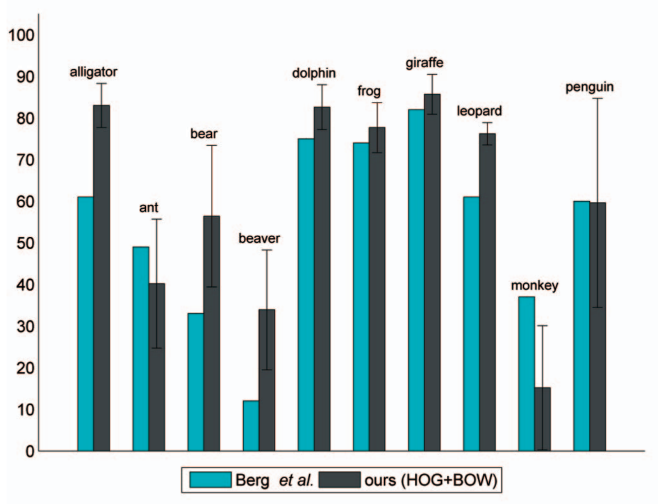


Fig. 12. **Comparison with Berg and Forsyth [5]:** Precision at 100-image recall level for the 10 animal classes made available by the authors of [5]. Note that our automatic algorithm is superior in many cases, even though the method of [5] involves manual intervention.

Comparison with Berg and Forsyth [5]. Here, we run our visual ranking system on the data set provided by Berg and Forsyth [5]. In order to do so, we downloaded an additional set of six classes (alligator, ant, bear, frog, leopard, and monkey) for which no manual annotation was obtained. In our experiments, we only use the test images from [5] for testing and training is performed on our downloaded images and the corresponding metadata. Fig. 12 compares the results reported in [5] to reranking of the test images available from [3] using our visual classifier. Note that we are training on our set of images, which might stem from a different distribution than the Berg test set. We compare with the "classification on test data" category of [5], not to

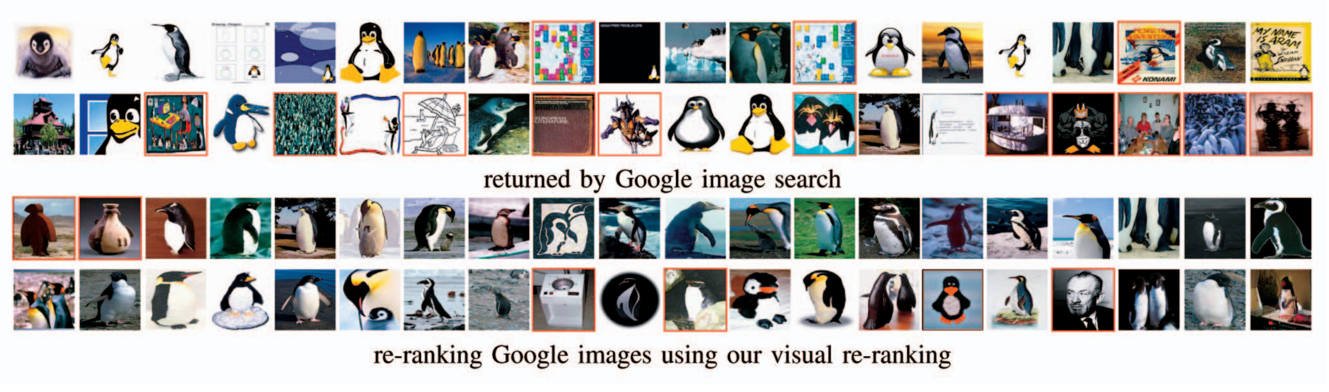

returned by Google image search
re-ranking Google images using our visual re-ranking

Fig. 11. **Reranking Google images:** The 40 top-ranked penguin images returned by Google image search (in their original order), as well as the 40 top-ranked images after reranking Google images using our visual (only) ranking. Red boxes indicate false positives. Google image search returns 14 *nonclass* images (false positives), compared to the six returned by our system.

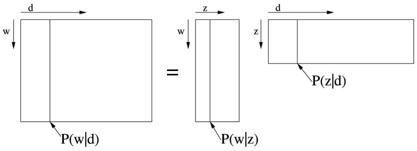


Fig. 13. **pLSA** models word frequencies in a document $P(w|d)$ as topics in a document $P(z|d)$ given the word distribution for a topic $P(w|z)$. pLSA then minimizes this equation with respect to KL divergence.

their "final data set," which includes ground-truth from their manual step. Their provided ground-truth, which treats *abstract* images as *nonclass*, was used. Note that our automatic algorithm produces results comparable or superior to those of Berg and Forsyth, although their algorithm requires manual intervention.

5.2 Comparison with Topic Models

In this section, we compare our SVM-based model to three widely used topic models (pLSA ([12], [15]), LDA [6], and HDP [30]). The underlying idea of all three topic models is to model each image as a distribution over topics whereby each topic is a distribution over textons. pLSA [15] models each entity (image) as a mixture of topics. Using the common notation for pLSA, we have $P(w|d) = \sum_z P(w|z)P(z|d)$. In our case, w are the visual words (described in Section 4), d are the images, and z are the classes (see Fig. 13). LDA [6] and HDP [30] are "probabilistic" extensions of pLSA, and HDPs model the data using hierarchical Dirichlet processes and avoid the need to explicitly specify the number of desired topics.

Given a set of topics and the test images ranked by their topic likelihood, the question is how to select the "right" topic which induces the final ranking. This problem of selecting the right topic is difficult and there exist various approaches (see, for example, [11]). To give an idea of the performance of the topic models, we avoid this topic selection stage and select the best topic based on ground-truth. The topic that gives the best 15 percent recall on the images evaluated on ground-truth is selected. This gives a huge advantage to the topic models in Table 7, as ground-truth is used during testing. Each of these models is trained on all images that we want to rank later on. This results in a ranking of all images for each topic (number of topics specified for pLSA and LDA and learned for HDP).

Discussion. The best performing topic models are the pLSA with 500 topics $68.7\% \pm 1.0$ closely followed by the HDP $68.6\% \pm 0.7$ that uses an average of 257 topics. Our SVM-based system performs similarly well with $68.0\% \pm 2.0$ *without* using ground-truth for model selection. There is also no single topic model that consistently outperforms the SVM on the majority of classes. As mentioned before, the results reported for the topic models use ground-truth to select the best topic and thus give an unfair advantage to topic models, but they also give evidence to the conclusion that the SVM is the "stronger" classifier. It could be possible to combine the high-ranked images from two topics and thereby improve the ranking, although there is no straightforward method to do this.

6 Conclusion

This paper has proposed an *automatic* algorithm for harvesting the Web and gathering hundreds of images of a given query class. Thorough quantitative evaluation has shown that the proposed algorithm performs similarly to state-of-the-art systems such as [12] while outperforming both the widely used Google Image Search and recent techniques that rely on manual intervention [5].

Polysemy and diffuseness are problems that are difficult to handle. This paper improves our understanding of the polysemy problem in its different forms. An interesting future direction could build on top of this understanding as well as the ideas in [26] and leverage multimodal visual models to extract the different clusters of polysemous meanings, i.e., for tiger: Tiger Woods, the animal. It would also be interesting to divide diffuse categories described by the word airplane (airports, airplane interior, and airplane food) into smaller visually distinctive categories. Recent work [26] addresses the polysemy problem directly and a combination with our work would be interesting.

TABLE 7
SVM versus Topic Models

prec. 15%	ap	bv	bk	bt	cm	cr	dp	ep	gf	gr	hs	kg	mb	pg	sk	tr	ww	zb	avg.	std
t+v (150/1000)	52.3	39.3	68.6	**66.2**	**57.3**	87.9	66.5	**78.7**	**83.8**	63.0	57.3	48.6	64.0	72.5	**82.9**	62.6	**93.2**	79.8	68.0	7.2 (2.0)
pLSA(5)	71.2	23.4	70.2	61.9	49.6	85.7	60.0	59.9	78.2	56.7	56.3	38.2	64.5	53.4	55.7	47.0	67.2	63.6	59.0	4.4 (1.1)
pLSA(10)	70.3	21.4	69.0	59.3	53.2	84.2	66.2	61.3	78.0	63.9	65.0	45.4	61.5	62.3	65.6	52.3	74.8	74.6	62.7	5.5 (1.4)
pLSA(50)	74.8	36.7	74.2	62.2	56.5	87.3	65.5	65.6	81.0	66.4	71.5	59.9	71.8	73.2	62.4	54.5	82.9	81.5	68.2	3.8 (0.9)
pLSA(100)	73.7	38.2	73.8	63.1	56.1	88.9	66.4	63.7	80.2	67.0	**72.3**	57.5	71.7	72.7	65.7	55.5	85.4	78.6	68.4	3.9 (0.9)
pLSA(200)	**76.1**	34.6	70.9	60.2	55.8	88.7	65.8	62.6	81.8	**67.6**	67.8	55.2	71.6	73.9	61.9	57.5	86.1	81.0	68.4	3.7 (0.9)
pLSA(500)	76.0	37.5	68.3	61.6	54.4	88.3	66.9	63.5	81.5	65.2	68.2	55.0	73.0	69.6	63.6	**64.6**	86.2	**88.4**	**68.7**	3.7 (1.0)
LDA(5)	70.6	22.4	65.9	60.9	46.0	84.2	62.0	55.8	77.9	57.3	55.7	42.7	63.7	54.2	52.9	52.9	72.5	59.8	58.7	4.5 (1.2)
LDA(10)	68.9	22.7	70.0	59.0	47.2	82.6	63.9	60.5	80.7	65.5	59.9	54.7	62.3	61.1	54.2	62.5	78.3	66.5	62.2	4.6 (1.1)
LDA(50)	72.4	37.8	74.3	64.4	56.8	**89.3**	64.1	63.8	83.6	66.7	66.6	**62.6**	69.7	68.7	50.3	60.5	86.3	82.6	67.8	3.7 (0.9)
LDA(100)	74.5	36.6	74.5	63.2	56.2	85.9	62.9	62.0	83.6	66.1	65.8	62.5	70.5	71.3	53.9	62.0	86.1	80.9	67.7	3.5 (0.9)
LDA(200)	72.9	38.9	69.7	65.1	57.1	86.7	61.5	59.4	82.9	65.4	63.8	61.3	72.2	68.9	48.4	60.8	86.0	83.8	66.9	2.8 (0.7)
LDA(500)	74.3	**42.2**	65.3	62.7	51.6	85.3	61.7	59.5	79.1	66.2	64.2	60.4	72.0	65.5	45.9	58.8	83.0	84.6	65.7	3.0 (0.7)
HDP	71.2	33.6	**76.2**	61.7	53.5	86.1	**71.7**	76.6	81.8	64.8	70.2	60.8	**72.6**	**74.3**	55.0	60.2	83.6	80.5	68.6	2.9 (0.7)

Given are the results as in Table 4 (i.e., 15 percent recall) for all object classes. We compare the best result achieved, using BOW only, by the SVM model to the performance achieved by using topic models (pLSA, LDA, and HDP), where the best topic is selected using ground-truth*. This gives the topic models a huge advantage, despite the fact that the topic models don't clearly outperform the* fully automatic *SVM model. HDP used an average of 257 topics. The last column states the mean of the classwise standard deviations over 10 runs of training the topic models, as well as the standard deviation of the means over all classes, in parentheses.*

Our algorithm does not rely on the high precision of top returned images, e.g., from Google Image Search. Such images play a crucial role in [12], [19], and future work could take advantage of this precision by exploiting them as a validation set or by using them directly instead of the text-based ranker to bootstrap the visual training.

There is a slight bias toward returning "simple" images, i.e., images where the objects constitute large parts of the image and are clearly recognizable. This is the case for object categories like car or wristwatch, where an abundance of such images occurs in the top text-ranked images. For other object classes, more difficult images are returned as well, e.g., elephant. The aim to return a more diverse set of images would require additional measures (see, for example, [19]). Although some classification methods might require difficult images, [27] gives an example of how a car model can be learned from these images. This automatically learned model is able to segment cars in unseen images.

Acknowledgments

The authors are grateful for financial support from Microsoft Research through a European PhD scholarship, EU Project CLASS, ERC grant VisRec no. 228180, and the Royal Academy of Engineering.

References

[1] J. Aslam and M. Montague, "Models for Metasearch," *Proc. ACM Conf. Research and Development in Information Retrieval,* pp. 276-284, 2001.

[2] K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, and M. Jordan, "Matching Words and Pictures," *J. Machine Learning Research,* vol. 3, pp. 1107-1135, Feb. 2003.

[3] T. Berg, "Animals on the Web Data Set," http://www.tamaraberg.com/animalDataset/index.html, 2006.

[4] T. Berg, A. Berg, J. Edwards, M. Mair, R. White, Y. Teh, E. Learned-Miller, and D. Forsyth, "Names and Faces in the News," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2004.

[5] T.L. Berg and D.A. Forsyth, "Animals on the Web," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2006.

[6] D. Blei, A. Ng, and M. Jordan, "Latent Dirichlet Allocation," *J. Machine Learning Research,* vol. 3, pp. 993-1022, Jan. 2003.

[7] C.K. Chow and C.N. Liu, "Approximating Discrete Probability Distributions with Dependence Trees," *IEEE Information Theory,* vol. 14, no. 3, pp. 462-467, May 1968.

[8] B. Collins, J. Deng, K. Li, and L. Fei-Fei, "Towards Scalable Data Set Construction: An Active Learning Approach," *Proc. 10th European Conf. Computer Vision,* 2008.

[9] N. Dalal and B. Triggs, "Histogram of Oriented Gradients for Human Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 886-893, 2005.

[10] G. Dorkó and C. Schmid, "Selection of Scale-Invariant Parts for Object Class Recognition," *Proc. Ninth Int'l Conf. Computer Vision,* 2003.

[11] R. Fergus, P. Perona, and A. Zisserman, "A Visual Category Filter for Google Images," *Proc. Eighth European Conf. Computer Vision,* May 2004.

[12] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman, "Learning Object Categories from Google's Image Search," *Proc. 10th Int'l Conf. Computer Vision,* 2005.

[13] C. Frankel, M.J. Swain, and V. Athitsos, "Webseer: An Image Search Engine for the World Wide Web," technical report, Univ. of Chicago, 1997.

[14] M. Fritz and B. Schiele, "Decomposition, Discovery and Detection of Visual Categories Using Topic Models," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2008.

[15] T. Hofmann, "Unsupervised Learning by Probabilistic Latent Semantic Analysis," *Machine Learning,* vol. 43, pp. 177-196, 2001.

[16] T. Hofmann, "Probabilistic Latent Semantic Analysis," *Proc. Conf. Uncertainty in Artificial Intelligence,* 1999.

[17] T. Joachims, "SVM^{light}," http://svmlight.joachims.org/, 2010.

[18] T. Kadir, A. Zisserman, and M. Brady, "An Affine Invariant Salient Region Detector," *Proc. Eighth European Conf. Computer Vision,* May 2004.

[19] J. Li, G. Wang, and L. Fei-Fei, "OPTIMOL: Automatic Object Picture Collection via Incremental Model Learning," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2007.

[20] W.-H. Lin, R. Jin, and A. Hauptmann, "Web Image Retrieval Re-Ranking with Relevance Model," *Proc. IADIS Int'l Conf. WWW/Internet,* 2003.

[21] D. Lowe, "Object Recognition from Local Scale-Invariant Features," *Proc. Seventh IEEE Int'l Conf. Computer Vision,* pp. 1150-1157, Sept. 1999.

[22] K. Mikolajczyk, C. Schmid, and A. Zisserman, "Human Detection Based on a Probabilistic Assembly of Robust Part Detectors," *Proc. Eighth European Conf. Computer Vision,* May 2004.

[23] K. Morik, P. Brockhausen, and T. Joachims, "Combining Statistical Learning with a Knowledge-Based Approach—A Case Study in Intensive Care Monitoring," *Proc. 16th Int'l Conf. Machine Learning,* 1999.

[24] Onix, "ONIX Text Retrieval Toolkit," http://www.lextek.com/manuals/onix/stopwords1.html, 2010.

[25] M. Porter, R. Boulton, and A. Macfarlane, "The English (Porter2) Stemming Algorithm," http://snowball.tartarus.org/, 2002.

[26] K. Saenko and T. Darrell, "Unsupervised Learning of Visual Sense Models for Polysemous Words," *Proc. Conf. Advances in Neural Information Processing Systems,* 2008.

[27] F. Schroff, "Semantic Image Segmentation and Web-Supervised Visual Learning," DPhil thesis, Univ. of Oxford, 2009.

[28] F. Schroff, A. Criminisi, and A. Zisserman, "Harvesting Image Databases from the Web," *Proc. 11th Int'l Conf. Computer Vision,* 2007.

[29] F. Schroff, A. Criminisi, and A. Zisserman, "Harvesting Image Databases from the Web," http://www.robots.ox.ac.uk/~vgg/data/mkdb, 2007.

[30] Y. Teh, M. Jordan, M. Beal, and D. Blei, "Hierarchical Dirichlet Processes," Technical Report 653, 2003.

[31] A. Torralba, "Contextual Priming for Object Detection," *Int'l J. Computer Vision,* vol. 53, no. 2, pp. 153-167, 2003.

[32] VGG, "Affine Covariant Features," http://www.robots.ox.ac.uk/~vgg/research/affine/index.html, 2010.

[33] S. Vijayanarasimhan and K. Grauman, "Keywords to Visual Categories: Multiple-Instance Learning for Weakly Supervised Object Categorization," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2008.

[34] G. Wang and D. Forsyth, "Object Image Retrieval by Exploiting Online Knowledge Resources," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1-8, 2008.

[35] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, "Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study," *Int'l J. Computer Vision,* vol. 73, no. 2, pp. 213-238, 2007.

Florian Schroff received the diploma degree in computer science from the University of Karlsruhe in 2004, where he was working with Professor H.-H. Nagel on camera calibration and focused on artificial intelligence, cryptography, and algebra, the master of science degree in computer science from the University of Massachusetts—Amherst, in 2003, where he started his studies under the Baden-Württemberg exchange scholarship in 2002, and the DPhil degree from the University of Oxford in 2009, funded by Microsoft Research through the European PhD Scholarship Program. He is currently a postdoctoral researcher at the University of California, San Diego. He is working together with Serge Belongie and David Kriegman on face recognition and projects related to object recognition.