Object image retrieval by exploiting online knowledge resources

Gang Wang
Dept. of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
gwang6@uiuc.edu

David Forsyth
Dept. of Computer Science
University of Illinois at Urbana-Champaign
daf@uiuc.edu

Abstract

We describe a method to retrieve images found on web pages with specified object class labels, using an analysis of text around the image and of image appearance. Our method determines whether an object is both described in text and appears in a image using a discriminative image model and a generative text model.

Our models are learnt by exploiting established online knowledge resources (Wikipedia pages for text; Flickr and Caltech data sets for image). These resources provide rich text and object appearance information. We describe results on two data sets. The first is Berg's collection of ten animal categories; on this data set, we outperform previous approaches [7, 33]. We have also collected five more categories. Experimental results show the effectiveness of our approach on this new data set.

1. Introduction

Image retrieval is an established research task. There are several major strategies. One could match representations of image appearance and sketches (e.g. [6, 19]). However, content based methods do not appear to be able to meet user needs [3, 15], and the major search engines use text-based methods, relying on cues such as file name, alt tags, user labeling and nearby text.

So one may ask: Does the combination of image cues and associated text suggest the image contains the relevant concept? In this framework, one works with a pool of web pages, typically collected by querying on an object name (e.g. "frog"), and must then identify images that depict frogs. This problem is challenging, because both text and images display richly varied properties. Words usually have multiple senses. For example, "frog" can be a bicycle band, an article of dress, a video game or a films, meaning that many web pages collected with a "frog" query have nothing to do with the animal "frog" . To overcome this problem, Berg *et al* [7] applied latent Dirichlet allocation [9] to discover word clusters that are relevant to the desired sense and rank images according to these words. They must select the word cluster by hand, and LDA may fail to discover meaningful clusters from noisy web texts. Schroff *et al.* [33] used text features with strong semantics (image filename, image alt text and website title) rather than just nearby text. This form of text semantics offers strong constraints, but does not guarantee avoiding sense ambiguity. Web images of concepts are often extremely complex, with variations in pose, location, species, etc. Nonetheless, both [7] and [33] showed visual information makes a substantial contribution to the final retrieval results.

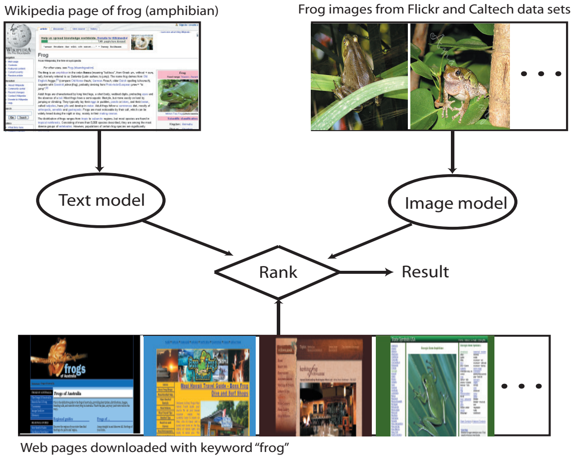
Wikipedia page of frog (amphibian)
Frog images from Flickr and Caltech data sets
Text model
Image model
Rank
Result
Web pages downloaded with keyword "frog"

Figure 1. The framework of our approach. Query "frog" is taken as an an example in this figure. We collect a pool of noisy web pages by inputting "frog" to Google. Wikipedia page of frog (amphibian) is extracted and a text model is built with its textual description. Similarly, image model is trained with Caltech and Flickr "frog" images. By combining text and image cues, images from web pages are ranked.

A shortage of good training data is an important challenge for this problem. One strategy is to build text model from collected noisy web pages. Then images with high rank based on the text model are used to train image model(e.g [7, 33]). The strategy is reasonable, but makes methods sensitive to text ranking results, and forces a com-

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

promise between the number of training images and their quality. Instead, we use web derived **online knowledge resources** (Sec.2). With enough human compiled text and image data, we build text and image models separately.

An important feature of our image model is its discriminative form (systems for object recognition that are trained on data where location is not known are more typically generative in form; see, for example, [28]). We use an SVM, because the method has behaved well on visual classification in the past [23, 33]. However, because images obtained in the wild are extremely complex, we need to stabilize the feature space; we use a novel form of the method advocated by [31, 30].

Our approach is automatic — meaning one could build a search engine that took text queries, extracted information from the knowledge resources, and identified relevant images — except for a step where we identify the sense of the query word (which we do by offering a user a set of senses from Wikipedia). We perform experiments on the same animal data set of [7], which includes ten animal categories. Our method outperforms reported results [7, 33]. We collect five new categories: “binoculars”, “fern”, “laptop”, “motorbike” and “rifle”. The experimental result shows our algorithm can also get high precision over these categories.

1.1. Background

Words and Pictures: There are many data sets of images with associated words. Examples include: collections of museum material [5]; the Corel collection of images ([4, 12], and numerous others); any video with sound or closed captioning [35]; images collected from the web with their enclosing web pages [7]; or captioned news images [34]. It is a remarkable fact that, in these collections, pictures and their associated annotations are complementary. The literature is very extensive, and we can mention only the most relevant papers here. For a more complete review, we refer readers to [13, 24]. Joint image-keyword **searches** are successful [6], and one can identify images that illustrate a story by search [21]. **Clustering** images and words jointly can produce useful, browsable representations of museum collections [5].

Linking keywords to images: One could: predict words associated with an image (*image annotation*); or predict words associated with particular image structures (*image labelling*). Because words are correlated, it can be helpful to cluster them and predict clusters, particularly for annotation ([25]). Labeling methods are distinguished by the way correspondence between image structures and labels is inferred. Methods include: clustering or latent variable methods [5, 4]; using multiple-instance learning [27, 12]; explicit correspondence reasoning with generative model ([14]; model from [10]); latent dirichlet allocation [8]; cross-media relevance models [20]; continuous relevance models [22]; and localization reasoning [11]. Barnard *et al.* demonstrate and compare a wide variety of methods to predict keywords, including several strategies for reasoning about correspondence directly [4]. Most methods attempt to predict noun annotations, and are more successful with mass nouns — known in vision circles as “stuff”; examples include *sky*, *cloud*, *grass*, *sea* — than with count nouns (“things”; *cat*, *dog*, *car*). For these methods, evaluation is by comparing predicted annotations with known annotations. Most methods can beat a word prior, but display marked eccentricities. One could then propagate text labels from labelled images to unlabeled images, making keyword based searches of large image collections possible.

2. Online Knowledge Resources

The web is rich in pools of information carefully compiled and edited by humans, typically volunteers. We call these pools of information **knowledge resources**. They are convenient to access and rich in context. In this paper, we use text and image knowledge resources for the image retrieval task.

For text, we employ Wikipedia[1], which is the biggest free encyclopedia on internet nowadays. It had over 2,104,000 articles on 902,000,000 words by December 2007. Besides abundant information, Wikipedia can disambiguate: for objects with multiple senses, it provides separate descriptions for each sense. This is very useful for our task. We select the desired sense and build a text model using its description. Then the resulting model can filter web pages from other senses and avoid ambiguity. For example, a text model trained with the “frog (amphibian)” Wikipedia page could filter text about a horror film called “frog”.

Wikipedia has a hierarchical taxonomy of classes, which could help to find classes close to the object in semantics. For example, from “frog”, we can go to its child class “ascaphidae ”. A better text model is built by combining “ascaphidae ” with “frog” since this captures specific information about specific frogs. Wikipedia pages of other semantically close classes which are not descendant such as “amphibian”,“snake” and “caecilian” are used to smooth the “frog” model parameters.

To train the image model, we exploit Caltech data sets (Caltech 101 [16] and Caltech 256 [18]) and Flickr [2] as image knowledge resources (Labelme [32] is another possibility; we have not used it to date). Caltech images depict objects cleanly because they are collected for research purpose, but the number is limited. For object classes which appear in Caltech data sets, we use Caltech images as positive training examples; for object classes which don’t appear in Caltech, we use Flickr images as positive training examples. Flickr has immense numbers of images (several thousands uploaded each minute; 2.2 million geotagged last month), but the labels are usually noisy. We use query extension to get a cleaner set of images. We query Flickr with an object name as well as its parent class name, obtained from the Wikipedia taxonomy. For example, we use “frog amphibian” to extract frog images.

3. Approach

Our goal is to retrieve object images from noisy web page with image and text cues. We have a query q which is the object class name, for example, “frog”. We also have a collection of web pages which are collected by inputting q and some extensions to Google text search engine. The ith web page is represented as a packet $\{W_i, I_i\}, i = 1, \cdots, N$, where I_i denotes image and W_i denotes text nearby I_i. We write $c_i = 1$ if I_i is relevant to q; otherwise $c_i = 0$. We write θ_t for the text model parameter and θ_v for the image model parameter when $c_i = 1$; write θ_b for the text model parameter when $c_i = 0$. We rank images according to:

$$p(c_i = 1 \mid W_i, I_i, q; \theta_t, \theta_v, \theta_b) \tag{1}$$

We adopt a generative text model and a discriminative image model. Eq.1 is written as:

$$\frac{p(W_i \mid c_i = 1, q; \theta_t) p(c_i = 1 \mid I_i, q; \theta_v)}{p(W_i \mid I_i, q)} \tag{2}$$

$p(W_i \mid I_i, q)$ is:

$$\begin{aligned} &p(W_i \mid c_i = 1, q; \theta_t) p(c_i = 1 \mid I_i, q; \theta_v) + \\ &p(W_i \mid c_i = 0, q; \theta_b) p(c_i = 0 \mid I_i, q) \end{aligned} \tag{3}$$

Where $p(c_i = 0 \mid I_i, q)$ equals to $1 - p(c_i = 1 \mid I_i, q)$.

θ_t and θ_v are trained on text and image **knowledge resources**. Fig.1 takes query “frog” as an example to illustrates our approach. We show how to learn $p(W_i \mid c_i = 1, q; \theta_t)$ and $p(W_i \mid c_i = 0, q; \theta_b)$ in Sec.3.1. $p(c_i = 1 \mid I_i, q; \theta_v)$ is studied in Sec.3.2.

3.1. Text model

We adopt a generative text model. W_i is a sequence of words $\{w_i^j, j = 1, \cdots, L\}$. θ_t is multinomial parameter over words and is estimated from text knowledge resource. Assume words are independent from each other in W_i:

$$p(W_i \mid c_i = 1, q; \theta_t) = \prod_{j=1}^{L} p(w_i^j \mid c_i = 1, q; \theta_t) \tag{4}$$

But Eq.4 tends to underweight the contribution of long text. For example, a short sentence may be accidental, but a paragraph is not. So we use the following formula:

$$p(W_i \mid c_i = 1, q; \theta_t) = (\prod_{j=1}^{L} p(w_i^j \mid c_i = 1, q; \theta_t))^{\frac{1}{L}} \tag{5}$$

which weights longer sets of relevant text more heavily in posterior inference (Eq.2).

The text knowledge resource is denoted K. It is a simple combination of all the Wikipedia pages (just body text) from queried object class (with desired sense) and its descendant classes in Wikipedia taxonomy. In the simplest case, θ_t could be estimated from K by maximum likelihood, which estimates θ_t^j (the jth component of the multinomial parameter) as a ratio of word counts. However, word set of K is limited, meaning that zero counts are a problem. “smoothing” is necessary. In this paper, we adopt Dirichlet smoothing [36] since it is simple and effective.

A much richer Wikipedia page collection A is extracted. The pages are from a number of semantically close classes (except children classes) of the object. With A as smoothing data, θ_t is estimated as:

$$\theta_t^j = \frac{N_K^j + \lambda \eta^j}{N_K + \lambda} \tag{6}$$

Where N_K^j denotes the counts of the jth word in K and N_K denotes the counts of all the words in K. Similarly, $\eta^j = \frac{N_A^j}{N_A}$. λ is a parameter to control the contribution of the prior. Words are set to be independent and of uniform probability when $c = 0$. Then $p(W_i \mid c_i = 0, q; \theta_b)$ is calculated similar to Eq.5.

3.2. Image model

We use a discriminative method to learn $p(c_i = 1 \mid I_i, q; \theta_v)$ directly. An SVM is employed because it has been proven to be effective and highly robust to noise in image classification [23, 33]. We exploit Caltech or Flickr images of the queried object class as positive training examples; the “clutter” category from Caltech 256 is used as negative examples. Each image is represented as a normalized histogram of visual words with dimension l. Training examples for this task are denoted as $\{(x_r, y_r), r = 1, \cdots, R, y_r = 1, -1\}$.

The original SVM classifier just outputs a hard decision. In this paper, we adopt the method of [29] to fit a posterior probability with a sigmoid function.

l is usually high dimensional. It is difficult to learn the model in the high dimensional space. [31, 30] have shown that using subsidiary tasks can produce a low dimensional feature space, which is stable and effective for the problem at hand. But instead of using unlabeled data as the subsidiary task, we propose a novel method to exploit highly relevant images in the knowledge resources, which are more helpful for the main task.

We first represent the object class we want to query as a normalized histogram of codewords f_o by using all positive training images. Other categories from the Caltech data set (except the queried object class) are also represented as histograms $f_m, m = 1, \cdots, M$. We calculate the difference between queried object class and Caltech classes using χ^2:

$$\frac{1}{2} \sum_{j=1}^{l} \frac{[f_o^j - f_m^j]^2}{f_o^j + f_m^j} \tag{7}$$

The T_s most similar categories are chosen from Caltech and act as positive examples in the subsidiary tasks. We download T_n sets of background images from web as negative examples. By pairwise matching, there are $T = T_s T_n$ subsidiary tasks overall. Each image in subsidiary tasks is also represented as a normalized histogram with dimension l. Similar to [30], for each auxiliary task t, we learn a linear function w_t^* which is most discriminative between positive and negative training images with a linear SVM.

We concatenate all w_t^* (each w_t^* is a column) to form a matrix W with dimension $l \times T$. We obtain a projection matrix P with dimension $h \times l$ by taking the first h eigenvectors ($h \ll l$) of matrix WW'. The training examples for the main task are now represented in the new feature space as $\{(P \cdot x_r, y_r), r = 1, \cdots, R, y_r = 1, -1\}$, where $P \cdot x_r$ is with dimension h. A kernel SVM classifier with optimal parameter w^* is trained:

$$\min \quad \|w\|^2 + C \sum_r \xi_r \tag{8}$$

$$\text{subject to} \quad y_r(w \cdot \Phi(P \cdot x_r) - b) \geq 1 \tag{9}$$

Where Φ denotes the kernel function. We use radial basis function in this paper. To calculate $p(c_i = 1 \mid I_i)$, I_i is represented in the low dimensional feature space and the learnt kernel SVM classifier is applied. SVM decision is converted to a probability with the method of [29].

The overview of our learning algorithm is presented in Alg.1

4. Implementation

In this section, we give implementation details for the text and image models. In Eq.6, λ is set to be one tenth of the words number of K. We remove stop words from Wikipedia and collected web pages.

Both training and testing images are converted to gray scale and resized to a moderate size. We use a canny edge detector to extract edge points from image. A set of points are randomly selected and regions are extracted at these points. As in [17], scale is uniformly sampled from a sensible range (10-30 pixels in this paper). Around 400 regions from each image are extracted. We represent these regions with SIFT [26] feature. Features from 150 Caltech categories (100 categories from Caltech101 plus 50 from Caltech 256) are quantized with Kmeans. The number of clusters is 500. So each image and class is represented with a 500 dimensional histogram.

When constructing subsidiary tasks, the 10 most similar categories are selected out, and 3 sets of background images are downloaded from web. So there are 30 subsidiary tasks by pairwise matching. We reduce the 500 dimensional feature to be 20 dimensional.

As considered by [33], there are "abstract" images which don't look realistic such as comics, graphs, plots, charts. In order to get natural images, it's better to remove them. [33] learnt a SVM classifier between "abstract" and "non-abstract" with extra training images. In this paper, we simply remove the non-color images since most of "abstract" images are black and white.

Algorithm 1 The overview of image model learning.

For a given query:

1. **Obtaining training examples**: Use Caltech or Flick images of the queried object class as positive training examples; use "clutter" category from Caltech 256 as negative training examples.
2. **Representation**: Represent image as a normalized histogram of codewords with dimension l; represent queried object class and other categories in Caltech data sets as normalized histograms with dimension l too.
3. **Constructing subsidiary tasks**: By Chi-square measure over histograms, find the T_s most similar categories from Caltech data set and set them to be positive examples in subsidiary tasks. Download T_n sets of background images from web as negative examples. By pairwise matching, there are $T = T_s T_n$ subsidiary tasks overall.
4. **Learning feature projection**: For each subsidiary task t, learn linear function w_t^* with linear SVM. Concatenating all w_t^* to form a matrix W with dimension $l \times T$. By taking the first h eigenvectors of WW', get a projection matrix P with dimension $h \times l$.
5. **Training SVM classifier**: Convert training examples of the main task to low dimensional space with projection matrix P, train a kernel SVM.

5. Experiments

We perform two experiments in this paper. The first one is on the data set of [7], which includes ten animal classes as shown in Fig.3. The second experiment is performed on five newly collected categories.

Besides the combined model in Eq.1, we also perform retrieval experiments with a pure text model and a pure image model. The text model ranks images according to:

$$\frac{p(W_i \mid c_i = 1, q; \theta_t) p(c_i = 1, q)}{p(W_i \mid c_i = 0, q; \theta_b) p(c_i = 0, q)} \tag{10}$$

$p(c_i = 1, q)$ and $p(c_i = 0, q)$ are simply set to be equal.

The image model ranks images with $p(c_i = 1 \mid I_i, q; \theta_v)$.

5.1. Experiment 1

For each animal class, we use its Wikipedia pages and Caltech or Flickr images as knowledge resources to train the text and image models. Then images in the returned web pages by "google" are ranked for each class. There is no "monkey" in Caltech data sets, so we use Flickr images

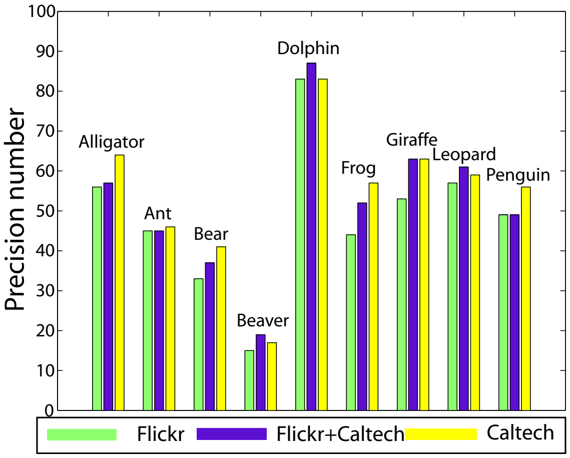

Figure 2. Precision at 100 image recall by image model. "Flickr" denotes the model is trained with Flickr images as positive training examples. Similarly, "Caltech" denotes the model is trained with Caltech images; "Flickr and Caltech" denotes the model is trained with both Flickr and Caltech images. In most categories, clean Caltech images produce better results. But results by Flickr are comparable and acceptable. This shows we can build image model with Flickr if there are not clean Caltech images for the queried object class.

to train "monkey" image model. For the other nine categories, we use Caltech images. We also compare the performance with different types of training images in Fig.2, which shows precision at 100 image recall by the pure image model. "Flickr" denotes the Image model is trained with noisy Flickr images as positive training examples. Similarly, "Caltech" denotes the model is trained with Caltech images; "Flickr and Caltech" denotes the model is trained with both Flickr and Caltech images. In most categories, clean Caltech images produce better results. Results using Flickr images are comparable and acceptable, which shows we can use Flickr images to train the image model if there are not clean Caltech images available.

In Fig.3, we present precision recall curves with different models. In all figures, the x axis denotes recall while the y axis denotes precision. "Text" is the result with text model; "Image" is the result with image model. "Text+Image" shows the result with the combined model. Note that we don't remove "abstract" images here.

We compare our ranking results produced by the combined model with the work of [33] and of [7] in Fig.4. This is based on the precision of 100 image recall. Note that we use the result of "classification on test data" in [7]. We outperform [7] on all the categories and outperform [33] except "Alligator" and "Beaver". Improvement is significant for categories such as "Bear", "Dolphin", "Monkey" and "Penguin". We also make a comparison with different measures on the whole data set as shown in Table.1.

We show the top ranked images for "Alligator", "Bear", "Frog", "Dolphin", "Giraffe", "Penguin" and "Leopard" in

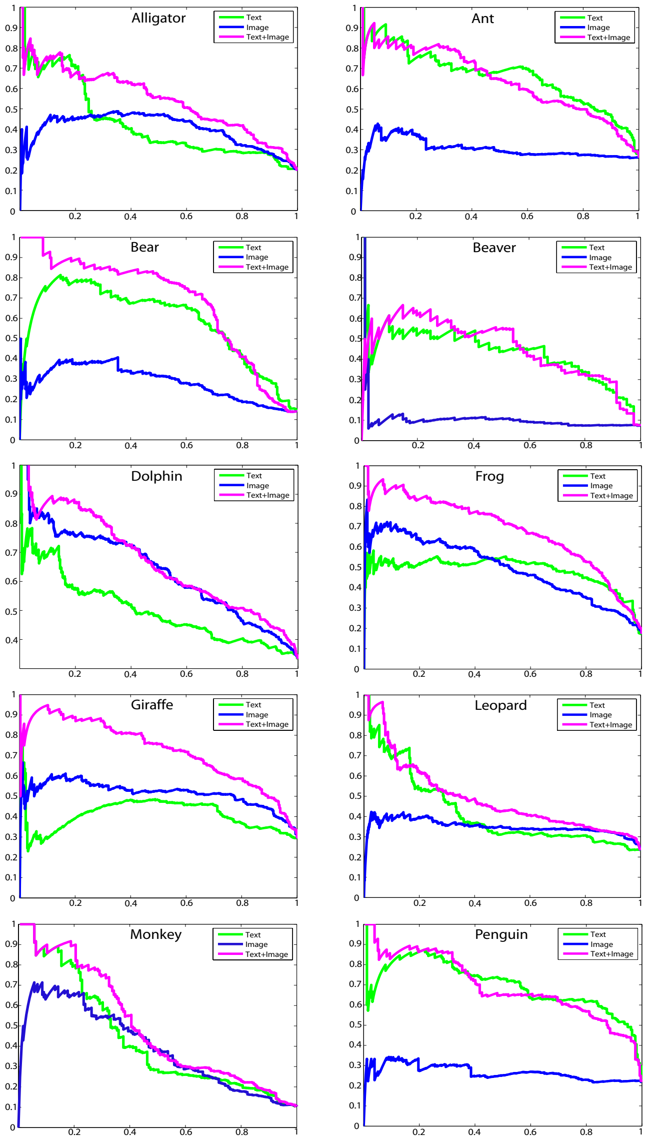

Figure 3. Precision recall curves with different models. In all figures, x axis denotes recall while y axis denotes precision. "Text" is the result with text model; "Image" is the result with image model. "Text+Image" shows the result by combining text and image models. Note that we don't remove "abstract" images here. The combined model usually works better than separate models. Image models can be quite discriminative such as the "dolphin" image model and the "frog" image model.

	Mean	Median	Minimum	Maximum
[7]	55.1	61	15	83
[33]	63.3	64	36	88
Our result	**79.4**	**84**	**41**	**94**

Table 1. Overall comparison withSchroff *et al* [33] and Berg *et al* [7] on the ten animal categories. This is based on the precision at 100 image recall. Our method outperform them on all the four measures: "Mean", "Median", "Minimum" and "Maximum".

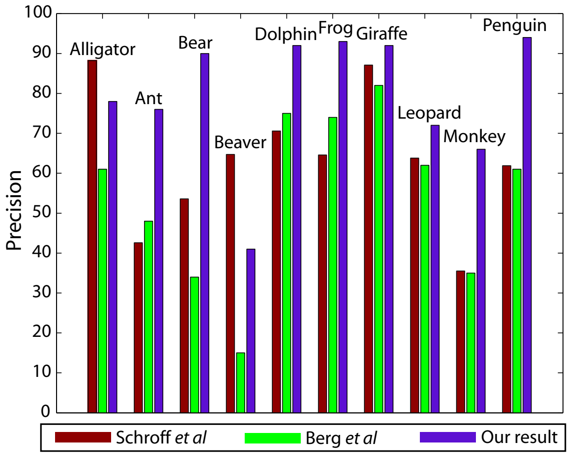

Figure 4. Results comparison with Schroff *et al* [33] and Berg *et al* [7] for each category. This is based on the precision on 100 image recall. Note that we compare with the "classification on test data" of [7]. We outperform [7] over all the categories and outperform [33] except "Alligator" and "Beaver". Improvement over many categories are significant such as "Bear", "Dolphin", "Monkey" and "Penguin".

Fig.6. Images in the red squares are false positives. Most of these images are correct.

5.2. Experiment 2

Experiment 1 is carried only on animal categories. In this section, we collect five diverse object classes ("binoculars", "fern", "laptop", "motorbike" and "rifle"). Similar to [7], we query google with the object name and some extensions. The top returned web pages are collected. We restricted downloaded images to be ".jpg" format. Finally, we get 732 "binoculars" images, 323 of which are correct images; 501 "Laptop" images, 158 of which are correct; 636 "Fern" images, 190 of which are correct; 801 "Motorbike" images, 276 of which are correct; 921 "Rifle" images, 195 of which are correct.

	Text	Image	Text+Image
Binoculars	76	90	93
Laptop	58	41	67
Fern	72	68	80
Motorbike	57	34	63
Rifle	55	21	57

Table 2. Precision at 100 image recall. "Text" is the result with text model; "Image" is the result with image model. "Text+Image" shows the result with combined model.

Similar to Experiment 1, our algorithm is applied to these categories and the precision recall curves is shown in Fig.5. In Table.2, we show the precision at 100 image recall with different models. Highly ranked images are exhibited in Fig.7. False positive images are marked with red squares.

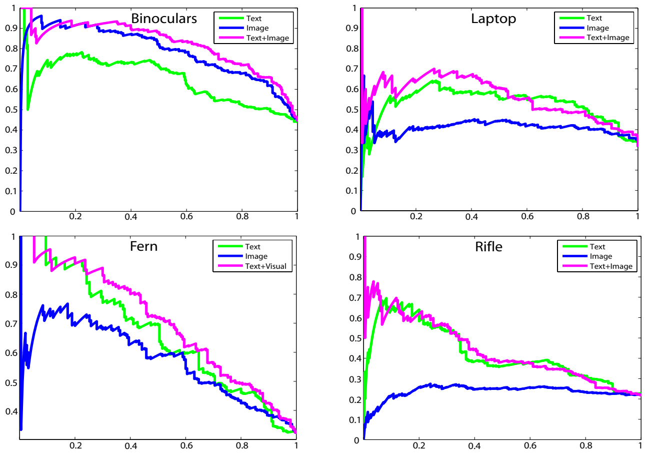

Figure 5. Precision recall curves with different models. In all figures, x axis denotes recall while y axis denotes precision. "Text" is the result with text model; "Image" is the result with image model. "Text+Image" shows the result with the combined model.

6. Conclusion

In this paper, we present a novel idea to exploit online knowledge resource for object image retrieval, which provides human compiled data to build object models. We perform experiments on two data sets. The results show the effectiveness of this approach.

7. Acknowledgement

We would like to thank Tamara Berg for providing the animal dataset and Derek Hoiem for the comments and suggestions. Funding for this research is provided by ONR-MURI Grant N00014-06-1-0734.

References

[1] http://en.wikipedia.org/wiki. 2

[2] http://www.flickr.com/. 2

[3] L.H. Armitage and P.G.B. Enser. Analysis of user need in image archives. *Journal of Information Science*, 23(4):287–299, 1997. 1

[4] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. Blei, and M. Jordan. Matching words and pictures. *'Journal of Machine Learning Research*, 3:1107–1135, 2003. 2

[5] K. Barnard, P. Duygulu, and D.A. Forsyth. Clustering art. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages II:434–441, 2001. 2

[6] S. Belongie, C. Carson, H. Greenspan, and J. Malik. Color and texture-based image segmentation using EM and its applications to content based image retrieval. In *Int. Conf. on Computer Vision*, 1998. 1, 2

[7] T.L. Berg and D.A. Forsyth. Animals on the web. In *Proc. Computer Vision and Pattern Recognition*, 2006. 1, 2, 4, 5, 6

[8] David M. Blei and Michael I. Jordan. Modeling annotated data. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 127–134, New York, NY, USA, 2003. ACM Press. 2

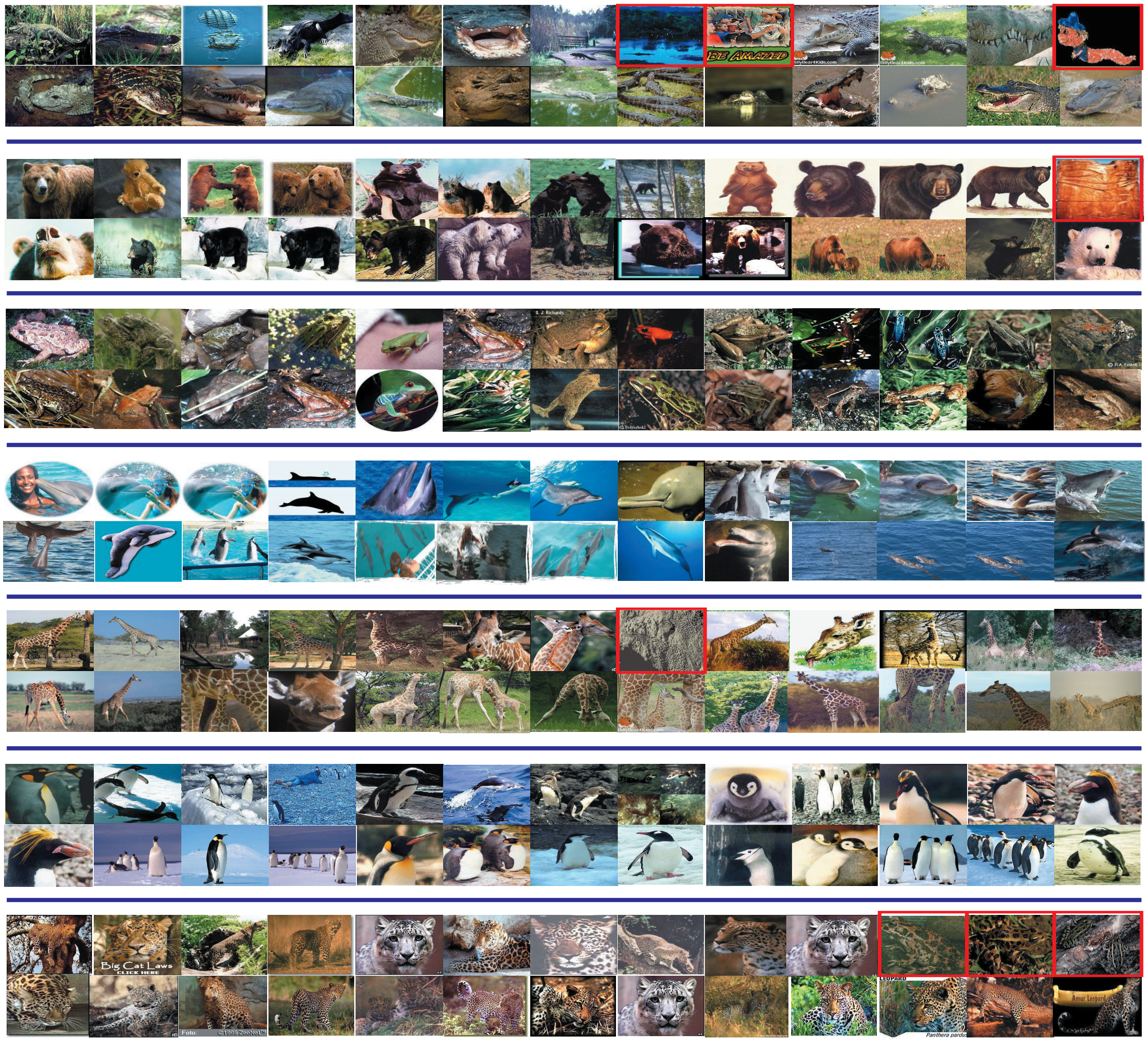

Figure 6. Top ranked images for “Alligator”, “Bear”, “Frog”, “Dolphin”, “Giraffe”, ”Penguin“ and “Leopard”. Images in red squares are false positives. Most of the images are correct.

[9] D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003. 1

[10] P. Brown, S.A. Della Pietra, V.J. Della Pietra, and R.L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 32(2):263–311, 1993. 2

[11] G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *IEEE T. Pattern Analysis and Machine Intelligence*, 29(3):394–410, 2007. 2

[12] Yixin Chen and James Z. Wang. Image Categorization by Learning and Reasoning with Regions. *J. Mach. Learn. Res.*, 5:913–939, 2004. 2

[13] Ritendra Datta, Jia Li, and James Z. Wang. Content-based image retrieval: approaches and trends of the new age. In *MIR ’05: Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval*, pages 253–262, New York, NY, USA, 2005. ACM Press. 2

[14] P. Duygulu, K. Barnard, N. de Freitas, and D.A. Forsyth. Object recognition as machine translation. In *Proc. European Conference on Computer Vision*, pages IV: 97–112, 2002. 2

[15] P.G.B. Enser. Visual image retrieval: seeking the alliance of concept based and content based paradigms. *Journal of Information Science*, 26(4):199–210, 2000. 1

[16] L. Fei-Fei, R. Fergus, and P. Perona. Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories. *Computer Vision and Image Understanding*, 106(1):59–70, 2007. 2

[17] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning Object Categories from Googles Image Search. *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, 2, 2005. 4

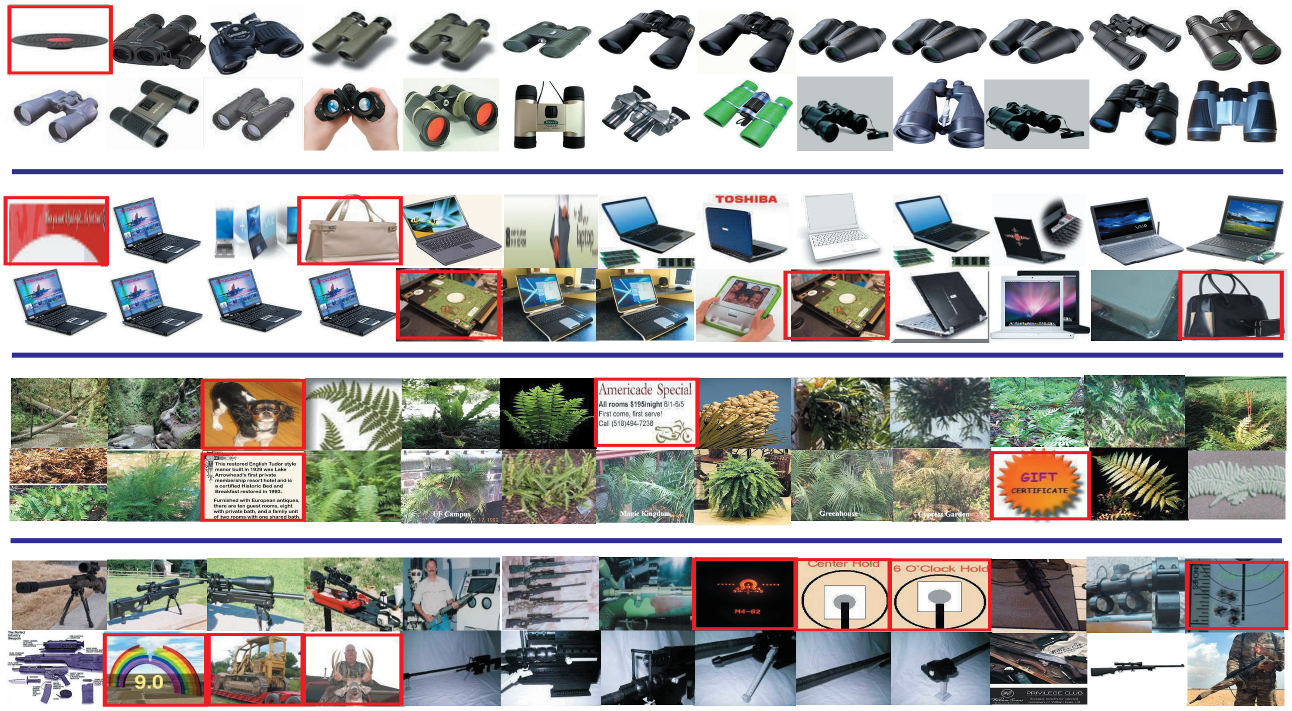

Figure 7. Top ranked images for “Binoculars”, “Laptop”, “Fern” and “Rifle” . Images in red squares are false positives.

[18] G. Griffin, A. Holub, and P. Perona. Caltech-256 Object Category Dataset. 2007. 2

[19] C.E. Jacobs, A. Finkelstein, and D.H. Salesin. Fast multiresolution image querying. In *Proc SIGGRAPH-95*, pages 277–285, 1995. 1

[20] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using crossmedia relevance models. In *SIGIR*, pages 119–126, 2003. 2

[21] Dhiraj Joshi, James Z. Wang, and Jia Li. The story picturing engine: finding elite images to illustrate a story using mutual reinforcement. In *MIR ’04: Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, pages 119–126, New York, NY, USA, 2004. ACM Press. 2

[22] V. Lavrenko, R. Manmatha, and J. Jeon. A model for learning the semantics of pictures. In *Neural Information Processing Systems*, 2003. 2

[23] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 2, 3

[24] Michael S. Lew, Nicu Sebe, Chabane Djeraba, and Ramesh Jain. Content-based multimedia information retrieval: State of the art and challenges. *ACM Trans. Multimedia Comput. Commun. Appl.*, 2(1):1–19, 2006. 2

[25] Jia Li and James Z. Wang. Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(10), 2003. 2

[26] D. Lowe. Object recognition from local scale-invariant features. In *Proc. International Conference on Computer Vision*, pages 1150–1157, 1999. 4

[27] O. Maron and A.L. Ratan. Multiple-instance learning for natural scene classification. In *The Fifteenth International Conference on Machine Learning*, 1998. 2

[28] A. Pinz. Object categorization. *Foundations and Trends in Computer Graphics and Vision*, 1(4):255–353, 2006. 2

[29] J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, 10(3), 1999. 3, 4

[30] A. Quattoni, M Collins, and T. Darrell. Learning visual representations using images with captions. In *Proc. CVPR 2007*. IEEE CS Press, June 2007. 2, 3, 4

[31] R. Raina, A. Battle, H. Lee, B. Packer, and A.Y. Ng. Self-taught learning: transfer learning from unlabeled data. *Proceedings of the 24th international conference on Machine learning*, pages 759–766, 2007. 2, 3

[32] B.C. Russell, A. Torralba, K.P. Murphy, and W.T. Freeman. LabelMe: a database and web-based tool for image annotation. 2005. 2

[33] F. Schroff, A. Criminisi, and A. Zisserman. Harvesting image databases from the web. In *Proceedings of the 11th International Conference on Computer Vision, Rio de Janeiro, Brazil*, 2007. 1, 2, 3, 4, 5, 6

[34] H. Wactlar, T. Kanade, M. Smith, and S. Stevens. Intelligent access to digital video: The informedia project. *IEEE Computer*, 29(5), 1996. 2

[35] Y. Wang, Z. Liu, and J.-C. Huang. Multimedia content analysis-using both audio and visual clues. *Signal Processing Magazine*, 17(6):12–36, 2000. 2

[36] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems (TOIS)*, 22(2):179–214, 2004. 3