Ranking user-annotated images for multiple query terms

Moray Allan
http://lear.inrialpes.fr/people/allan/
Jakob Verbeek
http://lear.inrialpes.fr/people/verbeek/

INRIA Grenoble Rhône-Alpes
655 avenue de l'Europe
38330 Montbonnot, France

Abstract

We show how web image search can be improved by taking into account the users who provided different images, and that performance when searching for multiple terms can be increased by learning a new combined model and taking account of images which partially match the query. Search queries are answered by using a mixture of kernel density estimators to rank the visual content of web images from the Flickr website whose noisy tag annotations match the given query terms. Experiments show that requiring agreement between images from different users allows a better model of the visual class to be learnt, and that precision can be increased by rejecting images from 'untrustworthy' users. We focus on search queries for multiple terms, and demonstrate enhanced performance by learning a single model for the overall query, treating images which only satisfy a subset of the search terms as negative training examples.

1 Introduction

Current web search engines provide much less relevant results when asked to search for images than for textual content. When ranking search results they typically ignore the images' content, and instead rank the images based on the text associated with them – for example the textual content of the web page where each image was found, or the tags and textual description that the user of a photograph-sharing website has provided. While it makes sense to make use of this associated text, and other metadata such as the pagerank of a web page or the popularity of a shared photograph, we can generally improve the search results by looking at the images themselves. This paper examines the task of searching photographs shared on the Flickr website (http://www.flickr.com/) to find images which contain objects matching user queries. If, for example, the user enters the search terms 'cat' and 'dog', we can quickly find potentially-relevant images using the textual tags associated with the images. Our aim is to rank those potentially-relevant images and to return to the user the images which we are most confident do in fact contain a cat and a dog.

While Flickr tags are intended to describe the image they are attached to, they are provided by individual users without any checks and are very noisy. Users often upload a large set of photographs and apply the same tags to the whole set, when each tag may in fact be relevant only to a small subset of the images. Even when Flickr tags are accurate, one-word textual tags are frequently ambiguous: a 'dog' tag might indicate that the picture contains a

BMVC 2009 doi:10.5244/C.23.20

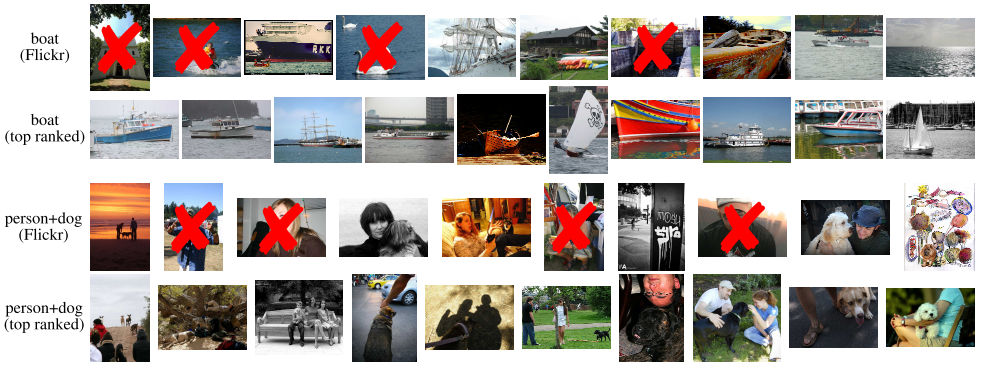

Figure 1: Sample of filtered Flickr images with matching tags, and top-ranked results by our method.

toy for a dog, a kennel for a dog, or damage done by a dog, rather than a dog itself. However, while for ordinary textual web searches users may be interested in pages relating to a weakly-defined topic, whether or not those pages mention the search terms directly, we consider that most image search users are interested in having the relevant objects directly visible in the images returned.

The system described in this paper identifies what is consistent across images to learn the visual meaning of user queries and return relevant images to the user. We improve on previous approaches to this task by taking into account the relationships between photographs and the users who uploaded them, to avoid being overly influenced by individual users' inaccurate or idiosyncratic use of textual tags. We focus on user searches containing multiple terms, and show that image ranking performance for multiple query terms can be improved by learning a combined model for the query and treating images which match only a subset of the query terms as negative rather than positive training examples.

Section 2 describes some related work, Section 3 gives details of our proposed method, Section 4 gives experimental results, and the paper closes with a discussion in Section 5.

2 Related work

While other approaches have been taken, for example using the constellation model [3] for object detection, work related to filtering and ranking web images generally falls into two categories: topic models and kernel methods. Most recent work on filtering and ranking web images has only examined single-term searches, but in this paper we focus on the case of searching for images that match multiple query terms.

Topic models such as Latent Dirichlet Allocation [2] can be applied either to textual documents or to images. While they can be extended to incorporate spatial information [4], the 'bag of words' has generally been retained for efficiency, treating images as collections of location-free visual features independently generated from some number of topics, where topics might represent, for example, different object classes. Berg and Forsyth [1] showed good results using LDA to learn topics and then filtering web images for a number of animal classes based on the most likely visual words for each topic. Li *et al*. [6] took an iterative approach to learning image categories from web images, at each step adding images which

are accepted by the current model. The strong independence assumptions in most topic models make them problematic for ranking according to multiple query terms. If we model two objects as separate topics, we need to learn a classifier on top of the topic mixing proportions to recognise when both objects occur, but, although a bag-of-words approach ignores geometry, in most implementations the topic mixing proportions vary roughly in proportion to the image area related to each topic. We also need to distinguish between images that give the topics of interest high mixing proportions because they fits the topics well and ones that happen to fit those topics least badly. If we model the desired objects together as a single topic, we can consider the probability of the image features under that topic, but then an image that contains two copies of the object with higher-probability features will rank better than an image containing both objects.

Kernel method approaches to this problem have most frequently used a Support Vector Machine to classify bag-of-words representations of images. We take a similar approach, though using kernel density estimation rather than an SVM; compare also Li *et al.*'s nearest-neighbour matching of images based on a global descriptor [7]. Schroff *et al.* [9] supplemented image histograms with some textual features to re-rank images found by Google image search or on web pages returned by Google text search. Vijayanarasimhan and Grauman [10] looked at using an SVM in a multiple-instance learning framework to deal with noisy data. Wang and Forsyth [11] used a generative model for text and an SVM for image histograms to model web data and rank images found on pages returned by Google text search. The method we propose below for Flickr images could similarly be extended to model page text if applied to general web images. All these approaches focused on single query terms; Grangier and Bengio [5] looked at ranking according to multiple query terms, but they assumed that a data set of correctly labelled images is available as training data for ranking unannotated images.

3 Ranking web images

We first state our baseline approach, then describe how we enhanced this method to give improved performance.

3.1 Baseline approach

We rank images by using a mixture of kernel density estimators to estimate the probability that each image belongs to the query set of interest.

For a new query, we begin by obtaining a set of images whose tags include the query terms. If we also have a non-query specific sample from the overall image distribution, we can treat an image $\mathbf{X}_j$ from the noisily-labelled query set as generated by a mixture model with components corresponding to the query relevant class C and all other classes O:

$$P(\mathbf{X}_j) = P(\mathbf{X}_j|C)P(C) + P(\mathbf{X}_j|O)P(O). \tag{1}$$

To rank the images, we want to estimate the probability that each image in the query set was generated from the query-related class:

$$P(C|\mathbf{X}_j) = \frac{P(\mathbf{X}_j|C)P(C)}{P(\mathbf{X}_j)} = \frac{P(\mathbf{X}_j|C)P(C)}{P(\mathbf{X}_j|C)P(C) + P(\mathbf{X}_j|O)P(O)}. \tag{2}$$

We can take a leave-one-out approach, working through the images in the query set, considering each in turn as the test image, with the rest of the image set treated as training data.

We use kernel density estimation to model the probability that an image $\mathbf{X}_j$ was generated from the non-query-specific class O containing images $\mathbf{Y}_i$:

$$P(\mathbf{X}_j|O) = \frac{1}{N_O}\sum_{i=1}^{N_O}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{d(\mathbf{X}_j,\mathbf{Y}_i)^2}{2\sigma^2}}, \tag{3}$$

where $d(\mathbf{X}_i,\mathbf{X}_j)$ is an appropriate distance function that measures the similarity of two images' visual content. In the experiments below, we fix $\sigma = \frac{1}{2}$ and use the chi-squared distance between bag-of-words representations of the two images: if two images are represented as histograms $\mathbf{X}_j$ and $\mathbf{X}_k$, of dimensionality D,

$$d(\mathbf{X}_j,\mathbf{X}_k) = \sum_{i=1}^{D}\frac{(X_{ji}-X_{ki})^2}{X_{ji}+X_{ki}}. \tag{4}$$

For each image we also fit a class-specific kernel density estimator to the rest of the image set, and model the image's probability of being generated from the query-specific class C as

$$P(\mathbf{X}_j|C) = \frac{1}{N_C-1}\sum_{i=1}^{N_C}\delta_{ij}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{d(\mathbf{X}_j,\mathbf{X}_i)^2}{2\sigma^2}}, \tag{5}$$

where δ_{ij} is zero for $i = j$ and one otherwise.

Using this pair of kernel density estimators for the class and non-class distributions we can calculate $P(C|\mathbf{X}_j)$ as in equation 2, comparing the probability that a test image is generated from the query-specific distribution with the probability that it is generated from the overall distribution of images for other queries.

3.2 User information

We can enhance this baseline model by taking into account user information relating to the images in the query set. Different users use the same textual tags to mean different things, and different users have very different levels of accuracy when the tags are interpreted as describing the visual content of the images.

The kernel density estimator in our baseline approach above may be excessively influenced by a collection of similar images which a user has labelled inaccurately or idiosyncratically. To avoid this, we can change from 'leave-one-out' to 'leave-some-out'. We first compile information about what images come from the same source – for the experiments below we retrieved from Flickr the username for the uploader of each image, and cross-referenced this information to make a list of images for each user within the query set. We can then replace equation 5 by

$$P(\mathbf{X}_j|C) = \frac{1}{Z_j}\sum_{i=1}^{N_C}\delta'_{ij}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{d(\mathbf{X}_i,\mathbf{X}_j)^2}{2\sigma^2}}, \tag{6}$$

where δ'_{ij} is zero for images belonging to the same user and one otherwise, and the normalising factor

$$Z_j = \sum_{i=1}^{N_C}\delta'_{ij}. \tag{7}$$

This means that images from a user cannot reinforce our confidence in other images from the same user. Instead before we will believe an image label we require agreement from the labels provided by other users.

Additionally, if we cross-reference image and user information, we are likely to discover that some users are responsible for many images which our model judges to have inaccurate tags. Our image rankings may be improved if we avoid trusting such users' images even when our model would otherwise rank them highly. In the experiments below we look at the effect of imposing a strict threshold on image confidence: if any image from a user has $P(C|\mathbf{X}_j)$ below the threshold we do not return images from that user in the initial search results. We set the threshold low enough to return at least the desired number of images (here, 100).

3.3 Multiple query terms

If the user searches for two query terms together, one approach would be to learn a separate model for images matching each of the query terms, and then evaluate our confidence that both relevant objects appear in a test image as the product of the individual classifiers' confidence, by the approximation

$$P(C_1, C_2|\mathbf{X}_j) \approx P(C_1|\mathbf{X}_j)P(C_2|\mathbf{X}_j). \tag{8}$$

In the experiments below we compare this approach with using a single model for the combination of query terms.

We also look at improving the ranking performance of the model for a combination of query terms by making use of class-specific negative data, in addition to the images sampled from the overall distribution for other classes. While in the product-of-classifiers approach images matching individual query terms are taken as positive training examples for the separate classifiers, here we consider these images as negative training data. This was motivated by finding that a common failure mode in response to searches for two objects of similar appearance was to give high rankings to images where only one of them is in fact visible.

4 Experiments

4.1 Data

For each example image search, we take from Flickr a data set of images whose textual tags match the query terms. Our method can be viewed as cleaning this noisy data, by removing irrelevant images and returning high-confidence search results to the user.

A first set of single-term search queries were chosen based on the names of the categories in the PASCAL Visual Object Classes Challenge 2008 (http://www.pascal-network.org/challenges/VOC/), omitting the compound terms 'dining table' and 'tv/monitor'. A second set of searches, with two query terms each, were chosen by finding the most frequent tag cooccurrences within the data Flickr provided for the first set of queries. Although 'sofa' and 'chair' cooccurred frequently, this combination was omitted as sofas are themselves chairs. The full list of queries is given in Tables 1 and 2. Up to 4000 images were fetched for each query; fewer images were obtained for the multiple term queries, as shown

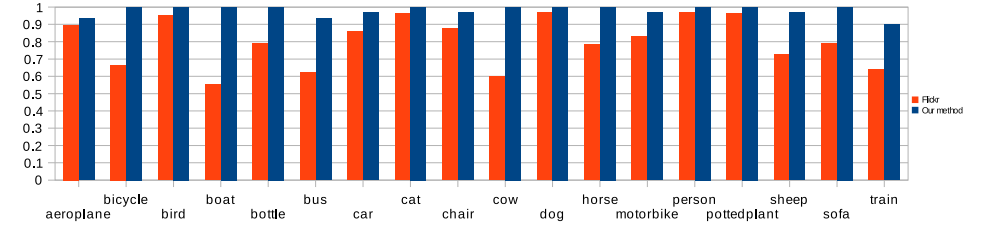

Figure 2: Precision of filtered Flickr input, and precision of our search results measured on the 30 highest-ranked images, for single-term search queries.

in these tables. An additional set of 4000 'random' images was obtained by fetching images from Flickr without any constraints on their tags, but ordered only by the time the photographs were taken; these were used as negative training examples.

As we do not have access to the full Flickr image database, the noisy data set for each search was downloaded through Flickr's public API. In each case a sample of images matching the specified tags was retrieved, ordered by the time they were taken, not by Flickr's own rankings of image 'relevance' or 'interestingness'. Only images with fewer than 10 tags were accepted, as where images have large numbers of tags the tags tend to have at best a very weak relationship to the image content. Harris features were detected in each image, and SIFT descriptions computed for the corresponding local regions. A sample of region descriptions were clustered using k-means to create 1000 visual words, and all images were then represented as 1000-dimensional bag-of-words histograms.

Only the images and their associated tags were used as input for the experiments, but a subset of the data was manually labelled to provide a quantitative evaluation of our method's performance. A sample of the data (where possible, 500 images) was annotated to assess the input data's relevance to the search query, shown in the 'Flickr' column of Tables 1 and 2, and the images which our method ranked highest for each query were annotated to measure how large an improvement it brought. The image annotators were instructed only to mark images as positive if the objects corresponding to the query terms were at least partially visible in the images (not just implied by context). Only the objects corresponding to the query terms' primary meanings were accepted. Drawings of the relevant objects, and photographs of toy versions, were put into a special category, which has been merged with the positive category in the results presented below. Writing visible in the images was ignored.

4.2 Single query term

Table 1 gives experimental results for the example searches using a single query term, for the full method, the baseline method without taking into account user information, and an intermediate approach which benefits from user information during matching, but does not reject images from users whose other images are judged to have inaccurate tags. We evaluate the various approaches by comparing the precision of the 30 highest-ranked images – this typically represents two or three pages of web image search results (as far as most search users go). For the full method we also show the precision of the 100 highest-ranked images. Note that the precision of the input Flickr images would be significantly lower if they had not been filtered to remove images with very large numbers of tags. The overall performance increase is shown in Figure 2, and some sample filtered Flickr images matching 'boat', and

Query	Images	Flickr	Final		–User	–UserFilter
aeroplane	4000	90%	80/100	28/30	26/30	30/30
bicycle	4000	67%	100/100	30/30	26/30	29/30
bird	4000	95%	97/100	30/30	29/30	30/30
boat	4000	55%	100/100	30/30	27/30	26/30
bottle	4000	79%	91/100	30/30	22/30	30/30
bus	4000	62%	93/100	28/30	28/30	29/30
car	4000	89%	98/100	29/30	28/30	28/30
cat	4000	96%	100/100	30/30	30/30	30/30
chair	4000	88%	98/100	29/30	30/30	29/30
cow	4000	60%	99/100	30/30	25/30	29/30
dog	4000	97%	100/100	30/30	30/30	30/30
horse	4000	78%	100/100	30/30	22/30	30/30
motorbike	4000	83%	97/100	29/30	30/30	28/30
person	4000	97%	98/100	30/30	30/30	30/30
pottedplant	746	97%	99/100	30/30	30/30	30/30
sheep	4000	72%	95/100	29/30	17/30	29/30
sofa	4000	79%	98/100	30/30	26/30	22/30
train	4000	64%	84/100	27/30	30/30	28/30

Table 1: Results for single query term: Flickr = filtered Flickr input; Final = full method; –User = ignoring user information completely; –UserFilter = not filtering based on users' overall annotation accuracy. Lower precision is highlighted with darker red.

the highest-ranking search results for that term, are shown in Figure 1.

Since this quantitative evaluation strictly required the primary relevant object to be visible in an image for the image to be accepted, only some of the images marked as incorrect will be completely irrelevant to the search term. Looking at the two queries with the lowest precision at the 100 image recall level, the high-ranking incorrect 'aeroplane' images (precision 80/100) include many views from aeroplanes and skies with aeroplane vapour trails where the aeroplanes themselves are not visible, and the high-ranking incorrect 'train' images (precision 84/100) include many pictures of railway lines and stations where no train is visible. A search user might well accept a high ranking for these images.

When we ignore user information completely, images from the same user can support each other, so sets of similar images from a single user can be ranked highly, whether or not they really correspond to the class of interest. Ignoring user information we end up ranking highly, for example, many repetitive pictures of a train from a single user, and many pictures of an event tagged with the acronym 'SOFA'. Comparing the 'Final' and '–UserFilter' columns in Table 1 shows that for some classes rejecting images from users whose other images are judged to have inaccurate tags brings a further gain in performance.

4.3 Multiple query terms

Table 2 gives experimental results for the example searches using multiple query terms. For many of these queries the input Flickr images have very low precision compared with the single term queries, making ranking them more important but also more difficult. In addition to the approaches used in the single query term experiments above, Table 2 also compares using a product of classifiers, a single classifier with additional negative training examples, and a single classifier without this additional training data.

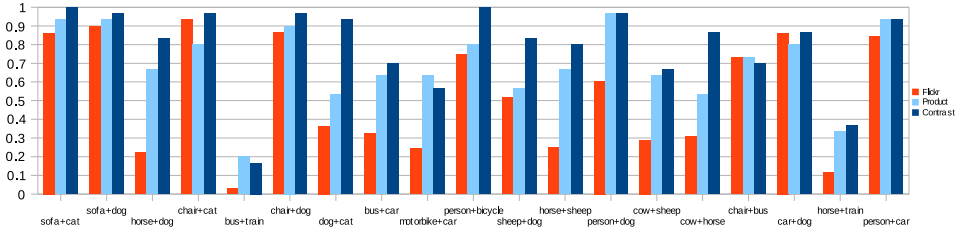

Figure 3: Precision of filtered Flickr input, and precision of our search results using a product of two classifiers, or the final combined classifier with additional negative training examples as a contrast to the positive examples (both measured on the 30 highest-ranked images).

Query	Images	Flickr	Final		–Contrast	–User	–UserFilter	Product
sofa+cat	1543	86%	99/100	30/30	28/30	29/30	30/30	28/30
sofa+dog	1159	90%	97/100	29/30	29/30	30/30	27/30	28/30
horse+dog	3075	22%	81/100	25/30	26/30	4/30	13/30	20/30
chair+cat	4000	94%	96/100	29/30	25/30	27/30	29/30	24/30
bus+train	3008	3%	18/100	5/30	5/30	0/30	0/30	6/30
chair+dog	1786	86%	97/100	29/30	27/30	30/30	26/30	27/30
dog+cat	4000	36%	87/100	28/30	18/30	10/30	23/30	16/30
bus+car	2603	32%	66/100	21/30	20/30	8/30	20/30	19/30
motorbike+car	458	24%	65/100	17/30	16/30	7/30	16/30	19/30
person+bicycle	107	75%	52/76	30/30	29/30	19/30	30/30	24/30
sheep+dog	1726	52%	84/100	25/30	21/30	22/30	24/30	17/30
horse+sheep	578	25%	65/100	24/30	19/30	1/30	4/30	20/30
person+dog	512	61%	94/100	29/30	30/30	13/30	18/30	29/30
cow+sheep	652	29%	66/100	20/30	20/30	7/30	9/30	19/30
cow+horse	1074	31%	74/100	26/30	20/30	11/30	24/30	16/30
chair+bus	122	73%	67/87	21/30	22/30	22/30	22/30	22/30
car+dog	3040	86%	92/100	26/30	24/30	26/30	27/30	24/30
horse+train	299	11%	29/100	11/30	12/30	0/30	1/30	10/30
person+car	305	85%	93/100	28/30	29/30	30/30	29/30	28/30
cow+cat	492	20%	32/100	11/30	3/30	3/30	8/30	3/30

Table 2: Results for multiple query terms: Flickr = filtered Flickr input; Final = full method; –Contrast = trained without additional negative training examples; –User = ignoring user information completely; –UserFilter = not filtering based on users' overall annotation accuracy; Product = using product of two classifiers (and user information). Lower precision is highlighted with darker red.

The improvement from taking advantage of user information is greater than for single term queries. In some cases precision drops off disastrously when it is not used: for example searching for 'horse' and 'sheep' precision falls from 24/30 to 1/30. For a few classes the final combined model performs much better than the product method (both benefit from user information). The overall performance increase is shown in Figure 3. The biggest increase comes when searching for two objects of similar appearance.

Adding the additional negative data, which matches only one of the query terms, helps for most classes, and brings a large improvement in a few cases. There are again large gains

Figure 4: Sample of filtered Flickr images with matching tags, and top-ranked results by our method, for difficult example queries (filtered Flickr input precision 52%, 25% and 3%).

on queries that ask for two similar-looking objects, but also for example on a query for 'chair' and 'cat'. This could be because in the relevant images chairs tend to have quite few informative features (they vary highly in appearance, and are usually only partly visible in the images). Using the partial query-matches as negative data prevents images with only a cat from getting high rankings, ensuring that the full query is taken into account.

Figures 1 and 4 show sample filtered Flickr images, and the highest ranking results, for three two-term search queries. The examples in Figure 4 were chosen to show especially difficult cases; the filtered Flickr images used as input have a precision of 52%, 25% and 3% respectively. For most of these examples images without the relevant objects visible are marked with a cross, while for 'bus+train' the two images where a bus and a train are visible are marked with a tick. In the last case the model still manages a precision of 18/100 for its highest-ranked images; the other high-ranked images are of related transport infrastructure.

5 Discussion

In this paper we described a model that uses a mixture of kernel density estimators to rank web images, and showed how it can be enhanced to take into account user information, and how performance can be improved when searching for multiple query terms. Our experiments above used image data, tags, and user information taken from Flickr. A similar approach could also be taken for general web pages, replacing the tags with the web page text and the user information with the website identity.

The model described above can be extended to include a latent variable representing the true presence or absence of the query-relevant objects, and a per-user class probability to denote reliability. We can iteratively reweight images by the apparent user reliability and reestimate the model. Initial experiments, not included here due to lack of space, show a small improvement on some test queries, but on other queries the precision of the highest-

ranked images fell. Qualitatively the high-ranking images tend to be less diverse, with an increased number of images from a few users: it seems the per-user class probability makes us rank too highly inaccurately labelled images from otherwise reliable users. It is possible to restore some diversity by including only one image per Flickr user in the initial search results, rather than simply presenting the images in the order they are ranked by the model.

It would be interesting to share the learnt 'trustworthiness' of each Flickr user across different queries. Besides potentially improving the accuracy of search results, this could be useful for real-world search applications by allowing the parts of a large data set which are most likely to be reliable to be searched first.

Our use of query-specific negative examples is similar to the use of additional query-related negative training data for finding specific faces by Mensink and Verbeek [8], except that here the added negative data contains the very objects we want to learn. We also considered using additional negative data from other confusable classes, but it is not straightforward to automatically pick good classes to use as a contrast for a query. Good contrasting classes may appear as frequently cooccurring tags, but it would be dangerous to see that, for example, 'cat' often appears for 'kitten' images and so add 'cat' images as negative examples when searching for 'kitten'. Choosing good contrast terms would require appropriate use of a thesaurus, itself ideally learnt from the overall cooccurrence patterns in the data.

References

[1] T. L. Berg and D. A. Forsyth. Animals on the web. In *CVPR*, volume 2, pages 1463–1470, 2006.

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

[3] R. Fergus, P. Perona, and A. Zisserman. A visual category filter for Google images. In *ECCV*, pages 242–256, 2004.

[4] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from Google's image search. In *ICCV*, volume 2, pages 1816–1823, 2005.

[5] D. Grangier and S. Bengio. A discriminative kernel-based model to rank images from text queries. *PAMI*, 30(8):1371–1384, 2008.

[6] L.-J. Li, G. Wang, and L. Fei-Fei. OPTIMOL: automatic online picture collection via incremental model learning. In *CVPR*, volume 2, pages 1463 – 1470, 2006.

[7] X. Li, C. G. M. Snoek, and M. Worring. Learning social tag relevance by neighbor voting. *IEEE Transactions on Multimedia*, 2009. In press.

[8] T. Mensink and J. Verbeek. Improving people search using query expansions: How friends help to find people. In *ECCV*, volume 2, pages 86–99, 2008.

[9] F. Schroff, A. Criminisi, and A. Zisserman. Harvesting image databases from the web. In *ICCV*, 2007.

[10] S. Viyanarasimhan and K. Grauman. Keywords to visual categories: Multiple-instance learning for weakly supervised object categorisation. In *CVPR*, 2008.

[11] G. Wang and D. Forsyth. Object retrieval by exploiting online knowledge resources. In *CVPR*, 2008.