



HAL
open science

Improving web-image search results using query-relative classifiers

Josip Krapac, Moray Allan, Jakob Verbeek, Frédéric Jurie

► **To cite this version:**

Josip Krapac, Moray Allan, Jakob Verbeek, Frédéric Jurie. Improving web-image search results using query-relative classifiers. CVPR 2010 - IEEE Conference on Computer Vision & Pattern Recognition, Jun 2010, San Francisco, United States. pp.1094-1101, 10.1109/CVPR.2010.5540092 . inria-00548636

HAL Id: inria-00548636

<https://inria.hal.science/inria-00548636>

Submitted on 20 Dec 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

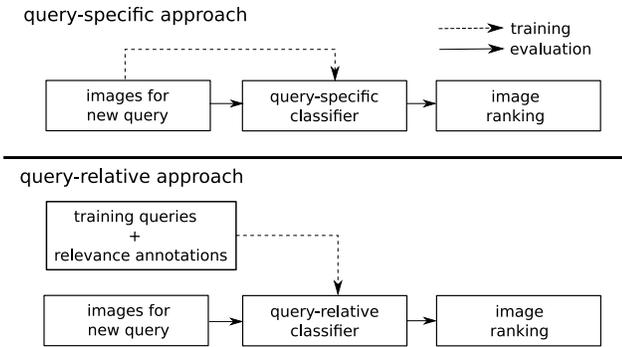


Figure 2. Image re-ranking approaches. The query-specific approach trains on the image set to be re-ranked. Images are described by the query-independent features; the models do not generalize to other queries. The query-relative approach trains on images with relevance annotations for queries. Images are described by query-relative features; the model generalizes to new queries.

dependent; see Figure 2 for an illustration. Our approach is inspired by text-based image search techniques [6, 13]. These methods compute an image’s relevance by weighting various meta-data fields where the query terms can appear, e.g. the web page title and image filename. For a particular query each image can be represented with a binary query-relative feature vector. Each feature codes for absence/presence in a particular text field, and its meaning is query-relative since the value of each feature depends on the match between the text field and the query terms.

If we wish to take into account the image contents as well as the text associated with each image, the situation is more complex, because image content and textual query terms cannot be matched directly. In our approach we overcome this problem as follows. The training stage makes use of sets of images retrieved by a text-based image search engine for a set of queries, annotated with ground-truth information about whether the images are relevant to the query terms. Images are represented by histograms of visual words, and the average histogram is computed over the image set retrieved for each query. Using these average histograms, we can rank the visual words for each query according to their frequency among the retrieved images. The final image representation is derived from the histograms of visual words, reordered for each query according to this ranking. After computing this representation for each image, it becomes possible to train a universal classifier (common to all queries) using feature vectors obtained by concatenating query-relative text and image features. This classifier can be used to re-rank images for new queries without requiring additional training, because while the exact image or text content represented by each feature depends on the search query for which it was computed, a feature’s meaning in relation to the query is always the same.

We believe this contribution to be significant as, to our

knowledge, we are the first to propose a re-ranking method for images that uses their visual content without requiring learning of query-specific models. The only per-query information we exploit is the average visual word histogram. This approach makes it possible to use image re-ranking in search engines with minor additional computational cost.

Until now there has been no large public data set appropriate for evaluating image re-ranking methods’ performance on the images found by text-based search engines. The data sets that exist [5, 11, 16] contain images for only a few classes and/or lack meta-data associated with the images (the text surrounding the images on the web page, file-names, URLs, page titles, etc.). Therefore, we introduce a large public data set of 353 search queries for which we provide the top-ranked images returned by an image search-engine along with the associated meta-data. Each image has a ground-truth relevance label, indicating whether or not it is relevant to the query. Using average precision we evaluate how much our models improve over the search engine’s results, and compare with an approach that learns a separate classifier for each query. Our generic models using query-relative features improve significantly over the original search engine ranking, and also outperform models that are trained separately for every search query.

The plan of the remaining sections is as follows. Section 2 gives an overview of related work. Then Section 3 presents our method for representing text and images by query-relative features. Section 4 describes experiments performed on the new data set, giving an experimental validation of our approach, and Section 5 concludes the paper.

2. Related work

In recent years there has been considerable interest in learning from the images and associated text that can be found on the web. Some authors have focused on images and their associated tags on photo sharing websites like Flickr, see e.g. [12, 17], while others have focused on general web images gathered using existing text or image search engines [2, 4, 5, 11, 14, 16]. Most of these methods rely on visual consistency to identify images that are relevant to the query terms, among a set of several hundreds to thousands of images obtained using the search engine.

Generative approaches learn a model on the images obtained from the search engine and then rank them by the likelihood of the images under the model. Constellation models are used in [5], where an initial model is trained on all images and RANSAC is used to iteratively remove outliers. In [18] Gaussian mixture models are trained on features of image regions as well as LSI vectors coding the words surrounding the images. The model is refined using EM and used to determine which regions of the images retrieved for the query are most likely to correspond to the query object. Topic models such as PLSA [8] and LDA [3]

have been used by several authors. In [4], PLSA models are used as well as variants that encode some of the spatial layout of objects. Ranking of images is done on the basis of one of the learned topics, selected using the topic mixing proportions of the first few images returned by the search engine. Hierarchical Dirichlet processes, an extension of LDA that do not require the number of topics to be fixed, were used in [11]. The model is learned in an incremental manner, where at each iteration several images positively classified by the current model are added to the training set. Topic models are also used in [7], but are applied on a lower level by modelling gradient orientations in a spatial grid over the image. Images are represented by the estimated topic mixing proportions. To rank the images k -means is applied to the mixing weight representation, and images in bigger clusters are ranked higher. A clustering approach was also proposed in [1], where images are first segmented into coherent regions and clustered based on HSV histograms using mean shift. The largest cluster found is assumed to contain the object of interest, and images are ranked by their distance to the centre of this cluster.

A semi-supervised approach using LDA was taken in [2] to retrieve images of animals from the web. They learned an LDA model for the 100 word context around images found using Google, and manually selected the topics that correspond to the query. In a second step local image features are compared between images, and scores are determined by matches to images associated with good topics. The final ranking of images is based on a sum of scores obtained from visual feature matching and the LDA model. A different semi-supervised learning approach was taken in [14]. Here a hybrid generative-discriminative model is learned based on a small set of positive images provided by a user. A discriminative model predicts the image relevance given its visual word histogram. Given the image relevance, a generative model is defined over binary variables that code the presence of the query term in the image filename and URL, and over words that appear near the image on the web page.

A discriminative approach was taken in [16], where images are filtered in a three-step process. First, a visual classifier is applied to remove drawings, logos, etc. Secondly, a generative model over textual features is used to rank the images, where the features encode the presence of the query terms in several fields such as the web page title and image filename. In the last step a visual classifier is learned, taking the images highest ranked by text as positive examples, and a random set of other images as negative examples.

We note that all this related work is based on learning specific models for each query, except for the text-based models used in [6, 16] which are trained on a set of queries and applied to others. In our work we explore the approach of learning query-independent classifiers, but applied to visual features, as well as to more complex textual features.

3. Query-relative features

We wish to train a generic classifier to re-rank *query sets*: the set of images returned by a text-based search engine. In this section we describe how to construct query-relative text and image features. All supervised learning is based on these query-relative features which are extracted in a fast, unsupervised, manner. This allows us to learn a generic model once, using whatever annotated data is available, and then use it to make relevance predictions for new queries.

3.1. Textual features

Our base query-relative text features follow [6, 16]. Nine binary features indicate the presence or absence of the query terms in various fields associated with the image: the text surrounding the image (10 words before and 10 words after), the image’s alternative text, the web page’s title, the image file’s URL’s hostname, directory, and filename, and the web page’s hostname, directory, and filename.

These base features are first supplemented with nine partial match features, each of which is active if some of the query terms, but not all, are present in the relevant field.

We further add *contextual features* which represent in more detail the text related to the image, beyond simple presence or absence of the query terms. We divide the image’s text annotation in three parts: the text surrounding the image, the image’s alternative text, and words in the web page’s title. For each of the three parts of text annotation we define contextual features by computing word histograms using all the images in the query set. Two words are considered the same if their lower-cased forms match, after removing punctuation. We ignore the query terms, as well as words included in a stop word list. Given the histogram of word counts t_i for each image i , we sort the words indexed by k such that

$$\sum_i t_{i,k} \geq \sum_i t_{i,(k+1)}. \quad (1)$$

The ordered histogram of word counts is used to define a set of additional context features, where the k th binary feature represents the presence or absence of the k th most common word in this source of context for this query:

$$T_{i,k} = \begin{cases} 1 & \text{if } t_{i,k} \geq 1, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

The earlier additional context features are more informative for classification than the later ones, so we can trim these features down to the first N elements for each data item without a large loss in performance. These contextual features can be understood as a form of pseudo-relevance feedback, compare [9, 13]. Taking the base features, partial match features, and N contextual features for each text field, we have $9 + 9 + 3N$ binary features.

3.2. Visual features

Local image descriptors are extracted from each image on a dense multi-scale grid, and normalized so that the L2 norm is equal to one. Each local region is described by a 16-dimensional histogram of oriented gradients – preliminary experiments showed that this performed better than 128-dimensional SIFT. Spatial binning at the patch level appears to be less important when using dense sampling than when using interest point detectors.

Our image representation is based on local appearance and position histograms. Local appearance is quantized using hierarchical k -means clustering as in [15], learnt from a set of patches sampled from all images in the dataset, with 11 levels of quantisation, where $k = 2$ at each level. For position quantisation we used quad-trees [10], with three levels of quantisation. For each combination of appearance and position quantisation an image is described by an appearance-position histogram: each bin in the joint histogram represents a patch appearance at a certain position in the image. For appearance quantisation, only the levels with at least 128 quantisation bins (levels 7–11) are used, as at lower levels the bins are not sufficiently discriminative. The image is represented by a concatenation of all 14 appearance-position histograms with dimensionality less than 10000; the histograms with higher dimensionality are extremely sparse. Within a search engine system, these features could be computed once, when each image is first crawled.

Query-relative visual features Our visual features follow the same query-relative design as the contextual text features. For visual data we cannot directly construct binary features for presence or absence of the query terms, like the base text features, as there is no direct correspondence between query terms and image appearance. However, as with the contextual text features, we can find which visual words are strongly associated with the query set, and define a set of visual features to represent their presence or absence in a given image.

After calculating normalised visual word histograms \mathbf{h}_i for all images i , we begin computing visual features by calculating the average visual word histogram $\mu(A)$ over the images in the query set A . Like words in text, different visual words do not have uniform prior probability. For example, visual words which correspond to textureless image regions or to simple image gradients appear more often, while some visual words that relate to specific image structure are quite rare. We therefore order the visual features not by raw counts but by the ratio between the mean visual word histogram for the current image set A and the overall mean

visual word histogram on a large training set T :

$$\rho(A, T)_k \geq \rho(A, T)_{k+1}, \text{ where } \rho(A, T)_k = \frac{\mu(A)_k}{\mu(T)_k}. \quad (3)$$

This gives us an ordered set of query-relative visual features, where the k th feature relates to the visual word k th-most related to this query. In our experiments the ordering is performed after the appearance-position histograms from different levels of quantisation have been concatenated, so a truncated version of the sorted histogram may mix features from many levels.

Using this ordering of the visual features, we compared three ways of representing each visual word’s presence or absence: the visual word’s normalised count for this image, $h_{i,k}$, the ratio between its normalised count for this image and its mean normalised count within this group of images,

$$r_{i,k} = \frac{h_{i,k}}{\mu(A)_k}, \quad (4)$$

and a binary version of this ratio, thresholded at 1:

$$V_{i,k} = \begin{cases} 1 & \text{if } r_{i,k} \geq 1, \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

The typical counts for the most related visual words vary considerably across different image classes, so making the features binary may give better performance when we use the image features to learn a generic model in a query-relative manner, even though it discards some information.

4. Experimental evaluation

New data set We introduce a new large public data set of 353 image search queries, for each of which we provide the top-ranked images returned by a web search engine along with their associated meta-data¹. The data set also includes a ground-truth relevance label for every image. Previous image re-ranking data sets [5, 11, 16] contain images for only a few classes, and in most cases provide image files without their corresponding meta-data.

For each of the 353 search queries, the data set includes the original textual query, the top-ranked images found by the web search engine, and an annotation file for each image. For 80% of queries there are more than 200 images; in total there are 71478 images in the data set. The annotation files contain manual labels for image relevance to the search query, and other meta-data obtained from the web: the image URL, the URL of the web page where it was found, the page title, the image’s alternative text, the 10 words before the image on the web page, and the 10 words after. The images themselves have been scaled to fit within a 150×150 pixel square while maintaining the original aspect ratio.

¹See <http://lear.inrialpes.fr/pubs/2010/KAVJ10/>

	Overall	LP	HP	SEP	SEG
Number of queries	353	25	25	25	25
Fraction relevant	44.3	12.1	78.0	52.5	31.5

Table 1. Data set properties by group.

Model training In our main experiments below we hold out each query in turn and train a binary logistic discriminant classifier from annotated data for all other queries.² Query-relative features of relevant images are used as positive examples, and query-relative features of irrelevant images are used as negative examples. We use the learned model to rank images for the held-out query by the probability that they are relevant according to the model. Note that in a practical application the logistic discriminant parameters would only need to be learnt once, and could then be re-used for any new query. For methods which take a long time to train we used 10-fold cross-validation for evaluation.

Evaluation We evaluate a model’s re-ranking performance by calculating the average precision (AP) for the scores it assigns to the images for each query, and taking the mean across all queries. These results can be compared with the precision of the images and with the mean average precision of the search engine’s own rankings. To allow a more detailed evaluation, we chose four groups of queries with extreme behaviour on the search engine, which uses an approach focused on textual cues:

- Low Precision (LP): 25 queries where the search engine performs worst, e.g. ‘will smith’, ‘rugby pitch’, ‘bass guitar’, ‘mont blanc’, ‘jack black’
- High Precision (HP): 25 queries where the search engine performs best, e.g. ‘batman’, ‘aerial photography’, ‘shrek’, ‘pantheon rome’, ‘brazil flag’
- Search Engine Poor (SEP): 25 queries where the search engine improves least over random ordering of the query set, e.g. ‘clipart’, ‘cloud’, ‘flag’, ‘car’
- Search Engine Good (SEG): 25 queries where the search engine improves most over random ordering, e.g. ‘rugby pitch’, ‘tennis court’, ‘golf course’, ‘ben stiller’, ‘dustin hoffman’.

Note that there is some overlap between these groups. In our results tables below, as well as the overall mean average precision on all 353 queries, we show the mean average precision for queries in each of these groups of 25 queries. The fraction of relevant images, averaged over the queries in each group, is shown in Table 1; this gives an indication

²We also tried support vector machines instead of logistic discriminant classifiers; the results were almost the same but training was slower.

mAP \times 100	Overall	LP	HP	SEP	SEG
Search engine	56.9	26.8	83.0	49.5	63.4
Base text features	53.7	24.0	82.0	58.9	49.3
+ partial match	54.8	23.5	82.4	60.3	51.2
+ $N = 10$ context	56.5	22.6	83.4	61.6	53.3
+ $N = 20$ context	56.5	24.0	83.0	61.4	54.4
+ $N = 50$ context	56.8	22.7	83.6	62.2	53.2
+ $N = 100$ context	57.0	24.3	84.1	62.4	54.8

Table 2. Performance using only textual features.

of the difficulty of each group. For random ranking the AP score is close to the fraction of relevant images, while perfect ranking yields an AP score of 100.

Ranking images by textual features Our first set of experiments examines the image ranking performance of our textual features. Table 2 shows the mean average precision achieved by the search engine, and by several text-only models using different feature sets. Note that we do not have access to the ranking system used by the web search engine, and it may use additional features beyond those in the data set annotation files, such as click-through data or PageRank-like scores.

Each set of text features compared in Table 2 includes all the features from the previous set. We start with the base set of text features similar to those used in [6, 16], and first add to this additional partial match features. This set of features is then supplemented with increasing numbers of additional context features. The overall performance increases as more context features are added, but with a diminishing gain per additional feature.

Our text features alone match the performance of the search engine. Although the overall performance using our additional context features is roughly the same as the search engine’s, the methods perform differently, as can be seen looking at the extreme groups in Table 2. We beat the search engine slightly on the ‘high precision’ group where text gives the best performance, and do slightly worse on the ‘low precision’ group where we do not expect a text-only algorithm to do well. Although we do worse on the ‘search engine good’ group where their algorithm works well, we perform significantly better on the ‘search engine poor’.

Ranking images by visual features We compare the performance of different visual feature representations described in section 3.2. The first set of features, $h_{i,k}$, uses the visual word histograms reordered by relative frequency in the class compared to general images. The second set of features, $r_{i,k}$, uses the ratio between the visual word frequency in an image and its frequency in the current class. The third set of features, $V_{i,k}$, thresholds these ratios at one

mAP \times 100	Overall	LP	HP	SEP	SEG
$h_{i,k}$	60.5	21.2	89.0	70.3	53.5
$r_{i,k}$	59.7	21.0	87.6	67.7	53.1
$V_{i,k}$	64.4	23.8	90.7	72.0	57.8

Table 3. Performance of different feature representations, using visual features of dimensionality 200, without text features.

mAP \times 100	Overall	LP	HP	SEP	SEG
Search engine	56.9	26.8	83.0	49.5	63.4
50 features	61.1	20.3	90.4	70.0	52.2
100 features	63.1	21.9	90.6	71.3	56.2
200 features	64.4	23.8	90.7	72.0	57.8
400 features	64.9	24.1	91.0	71.9	58.4

Table 4. Performance using only visual features.

to give binary indicators.

Table 3 shows that the performance is increased when we move to binary features, probably because the relative frequencies of visual words vary considerably between queries. In the following experiments we always use these binary query-relative visual features.

Table 4 shows the mean average precision achieved by using increasing numbers of visual features. Each set of features includes all the features from the previous set. Adding more visual features increases the overall performance, but with diminishing gains. The overall performance is better than that achieved by text features alone.

Looking at the extreme groups, we find that the ‘low precision’ group is also hard to rank using visual features, and that the ‘high precision’ group gives even better results using image features: the extremely low and comparatively high precision of the data for the queries in these two groups also hinders and aids ranking when visual features are used. For the ‘search engine good’ group, where the search engine gave the biggest improvement, it still beats our results using visual features, while on the ‘search engine poor’ group visual features give us a very large lead over the search engine’s performance.

Combining textual and visual features In Table 5 we look at the performance of various combinations of textual and visual features. We combine between 50 and 400 visual features with the 18 base and partial match text features and between 20 and 100 additional features for each text field. Using larger numbers of features gives better performance, though the performance is not much worse with the smaller feature sets. The largest feature set included in the table (still only 718 dimensions) beats the search engine on all four groups of extreme queries, and has an overall mean average precision more than 10% better than the search engine or than our results using textual features alone. Figure

mAP \times 100	Overall	LP	HP	SEP	SEG
Search engine	56.9	26.8	83.0	49.5	63.4
$a = 50, b = 20$	64.3	25.6	90.4	70.6	60.3
$a = 100, b = 20$	65.7	26.9	90.7	71.9	62.5
$a = 200, b = 20$	66.8	29.1	90.9	72.7	63.7
$a = 400, b = 20$	67.3	28.9	91.2	73.1	65.0
$a = 400, b = 100$	67.3	29.2	91.3	73.7	65.5

Table 5. Performance for concatenated text and visual features, using a visual features, the base and partial match text features, and b additional context features for each text field.

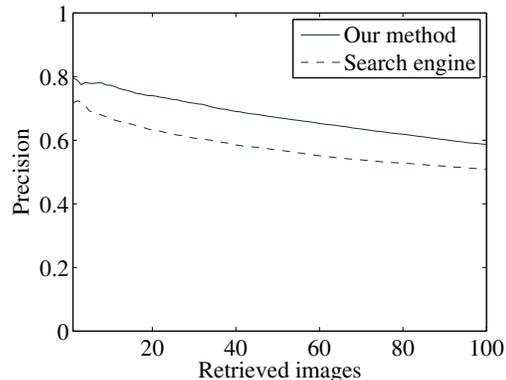


Figure 3. Precision averaged over all queries when retrieving n images.

3 shows the overall mean precision when a given number of images is retrieved for each query, using this feature set.

Figure 5 shows the top-ranked images given by the search engine for the queries ‘car’, ‘flag’, and ‘will smith’, and the top-ranked images after re-ranking. These results, and the ‘eiffel tower’ results in Figure 1, used 400 visual features and 100 textual features for each text field. Duplicate images were suppressed by comparing image distances between the top-ranked images. Our method does well on these queries, down-ranking irrelevant images, except on ‘will smith’ where it fails. In this case the model seems instead to detect portraits.

It might appear that our method will only work when the majority of the images in query set are relevant. In Figure 4 we show per query differences in performance between our method and the search engine, and the precision of the query sets. These results show that (i) for about 62% of the queries the precision in the query set is under 50%, (ii) for about 78% of all queries our model outperforms the search engine, and (iii) for 45% of all queries we improve results despite a query set precision under 50%. To understand why the method still works in such cases, note that the irrelevant images tend to be more diverse than the relevant ones. Even if only a minority of the images is relevant, the visual words in relevant images can still be the most frequent ones, as

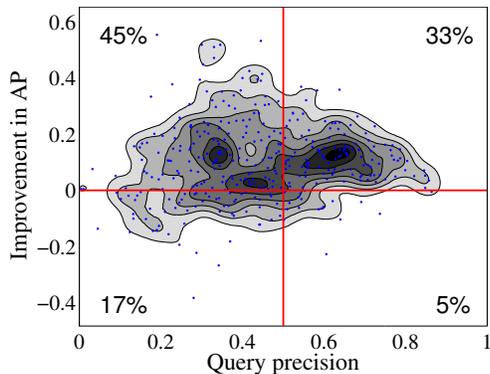


Figure 4. Improvement in AP of our method over the search engine, as a function of the fraction of relevant images in the query set.

compared to the database statistics.

Comparison with query-specific classifiers As a comparison, we also trained a query-specific support vector machine for each query in the data set. We followed the approach of Schroff et al. [16] to learn a query-specific classifier from noisy data. The images in the query set were treated as noisy positive training examples, while negative examples were sampled randomly from other queries. We ran experiments with two feature sets: the query-relative visual features used above, and a 512-dimensional histogram of visual word counts. A linear kernel was used with the binary query-relative features, and a squared exponential kernel with χ^2 distance measure with the histograms.

Table 6 shows results for these query-specific classifiers. Surprisingly, they give worse overall performance than the generic query-relative classifier, even with the same features (compare the third experiment in Table 4). Using a query-specific classifier with histograms of visual word counts also does less well overall than the query-relative classifier, though it performs slightly better on the ‘high precision’ and ‘search engine poor’ groups where the precision of the training data is higher. The SVM performance could be enhanced by optimising the parameter settings per class. Schroff et al. [16] use ten-fold cross validation to learn appropriate parameter settings for each class, while in our experiments we used a single set of parameters for all classes. The best parameters will depend, among other factors, on the proportion of irrelevant images in the input data, which varies across queries.

Performance on Fergus data set We also tested our method on the data set of Fergus et al. [4], which has been used to evaluate several previous query-specific ranking methods [4, 7, 16]. This data set has images from Google for seven object categories. Like Fergus et al. [4] we treat the “good” label as positive, and “intermediate” and “junk”

mAP \times 100	Overall	LP	HP	SEP	SEG
200 binary features	62.0	24.5	91.0	70.6	49.9
Histogram	62.6	20.1	92.4	73.4	49.4

Table 6. Performance of query-specific classifiers, using 200 visual features as in the third experiment in Table 4, or a 512-dimensional visual word histogram.

P \times 100	Mean	1	2	3	4	5	6	7
Google	43	50	41	19	31	41	46	70
Ours	56	65	55	72	28	44	49	79
[16]	54	35	-	-	29	50	63	93
[4]	69	57	77	82	50	59	72	88
[7]	91	100	83	100	91	65	97	100

Table 7. Comparison to Fergus data-set, precision at 15% recall: airplane, cars (rear), face, guitar, leopard, motorbike, wrist watch.

as negative. Since no textual meta-data is available we rank the images using only visual features. To compare with previous results, in this case we report precision at 15% recall.

This data set contains very few queries compared to our new data set, which limits the performance of our query-relative method. However, our method improves on the precision of the Google ranking for every class, in some cases by a large margin. We increase the mean precision at 15% recall by 13%. Moreover, our mean precision across the classes beats the query-specific method of Schroff et al. [16], even though they trained their models on a separate dataset which also has textual meta-data, while we used only the provided visual data. The methods of Fergus et al. [4] and Fritz et al. [7] perform better, however these are based on query-specific topic models which are not practical to use in a typical image search application as they require time-consuming training for every new query.

5. Conclusion

In this paper we showed how to construct query-relative features that can be used to train generic classifiers, and rank images for previously unseen search queries without additional model training. The features combined textual information about the occurrence of the query terms and other words found to be related to the query, and visual information derived from a visual histogram image representation.

We applied our image re-ranking method to the top-ranked images returned by a web search engine. We found that our query-relative models gave a considerable improvement over the raw search engine ranking, showing a more than 10% increase in mean average precision. Perhaps surprisingly, we also found that our models performed comparably to existing approaches that require training a visual classifier per query.

The second contribution of the paper is that we present



Figure 5. Top-ranked images using (a) the search engine and (b) our model, ordered left-to-right, for two queries which we answer well, and one – ‘will smith’ – where we fail.

a new public data set of images returned by a web search engine for 353 search queries, along with their associated meta-data, and ground-truth annotations for all images. We hope that this new data set will facilitate further progress in improving image search.

Acknowledgements. We thank ANR project R2I for financial support, and Exalead for image collection and labeling.

References

- [1] N. Ben-Haim, B. Babenko, and S. Belongie. Improving web-based image search via content based clustering. In *SLAM Workshop CVPR*, 2006.
- [2] T. Berg and D. Forsyth. Animals on the web. In *CVPR*, 2006.
- [3] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *JMLR*, 3, 2003.
- [4] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from Google’s image search. In *ICCV*, 2005.
- [5] R. Fergus, P. Perona, and A. Zisserman. A visual category filter for Google images. In *ECCV*, 2004.
- [6] C. Frankel, M. Swain, and V. Athitsos. WebSeer: an image search engine for the world wide web. In *CVPR*, 1997.
- [7] M. Fritz and B. Schiele. Decomposition, discovery and detection of visual categories using topic models. In *CVPR*, 2008.
- [8] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1/2), 2001.
- [9] V. Lavrenko and W. Croft. Relevance based language models. In *ACM SIGIR*, 2001.
- [10] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.
- [11] L.-J. Li, G. Wang, and L. Fei-Fei. OPTIMOL: automatic object picture collection via incremental model learning. In *CVPR*, 2007.
- [12] X. Li, C. Snoek, and M. Worring. Learning social tag relevance by neighbor voting. *IEEE Transactions on Multimedia*, 11(7), November 2009.
- [13] W.-H. Lin, R. Jin, and A. Hauptmann. Web image retrieval re-ranking with relevance model. In *Web Intelligence*, 2003.
- [14] N. Morsillo, C. Pal, and R. Nelson. Semi-supervised learning of visual classifiers from web images and text. In *IJCAI*, 2009.
- [15] D. Nistér and H. Stewénus. Scalable recognition with a vocabulary tree. In *CVPR*, 2006.
- [16] F. Schroff, A. Criminisi, and A. Zisserman. Harvesting image databases from the web. In *ICCV*, 2007.
- [17] K. Wnuk and S. Soatto. Filtering internet image search results towards keyword based category recognition. In *CVPR*, 2008.
- [18] K. Yanai and K. Barnard. Probabilistic web image gathering. In *ACM MIR*, 2005.