

# What Can Pictures Tell Us About Web Pages? Improving Document Search using Images

Sergio Rodriguez-Vaamonde, Lorenzo Torresani, *Member, IEEE*,  
and Andrew W. Fitzgibbon, *Senior Member, IEEE*

**Abstract**—Traditional Web search engines do not use the images in the HTML pages to find relevant documents for a given query. Instead, they typically operate by computing a measure of agreement between the keywords provided by the user and only the text portion of each page. In this paper we study whether the *content* of the pictures appearing in a Web page can be used to enrich the semantic description of an HTML document and consequently boost the performance of a keyword-based search engine. We present a Web-scalable system that exploits a pure text-based search engine to find an initial set of candidate documents for a given query. Then, the candidate set is reranked using visual information extracted from the images contained in the pages. The resulting system retains the computational efficiency of traditional text-based search engines with only a small additional storage cost needed to encode the visual information. We test our approach on one of the TREC Million Query Track benchmarks where we show that the exploitation of visual content yields improvement in accuracies for two distinct text-based search engines, including the system with the best reported performance on this benchmark. We further validate our approach by collecting document relevance judgements on our search results using Amazon Mechanical Turk. The results of this experiment confirm the improvement in accuracy produced by our image-based reranker over a pure text-based system.

**Index Terms**—Image Content, Web Search, Multimedia Search, Ranking.

## 1 INTRODUCTION

“A picture is worth a thousand words.” Despite this old saying, modern Web search engines ignore the pictures in HTML pages and retrieve documents merely by comparing the query keywords with the text in the documents [1]. Of course this text includes the words in image captions and markup tags, but does not look at the pixels themselves. This lack of attention to the visual information contrasts with the current state of the Web, which over the last 20 years has evolved from a collection of mostly textual documents to the modern fast-growing large-scale repository of multimedia where nearly every page contains several pictures or videos. The exclusive reliance on text-based technology to search the Web is explained by the challenges posed by the handling of image data: automatic image understanding is still today computationally expensive and prone to mistakes.

In this paper we propose a novel document retrieval approach that uses the content of the pictures in the Web pages to boost the accuracy of pure text-based search engines. At a high-level we expect that, for example, for the query “Ferrari Formula 1”, users

will judge documents containing pictures of Ferrari cars to be more relevant than pages with unrelated images. Consequently our hope is that a search system combining the textual information with the visual information extracted from the pictures will yield improved accuracy. While there is a large literature on combining text and image data for *image* search, we know of only a couple of prior attempts to improve Web *document* search using image content. An example is represented by the system of Yu et al. [2] who demonstrate improved ranking by using simple image measures such as aspect ratio, size, and high-level features such as blurriness. In contrast, we use a modern image recognition system to provide rich data on the picture content. Another related approach is the work of Zhou and Dai [3]. While we leverage text-based image search in order to obtain training data to learn a visual model of the query, this prior system offers the advantage of being fully unsupervised. However, in our experiments we demonstrate that this unsupervised learning of the visual model for a given query is much more computationally expensive and results in lower accuracy compared to our system.

In order to design an image-based search engine that can scale to Web-size databases we are posed with two fundamental challenges. First, the descriptor extracted from the pictures must be semantically rich but also extremely compact so that the overall size of the document is sufficiently small for fast search in billions of pages. Second, we must devise a way to efficiently translate the keywords provided by the user into a visual model (i.e., an image classifier) that

- 
- S. Rodriguez-Vaamonde is with Tecnalia, Zamudio, Bizkaia, Spain.  
E-mail: sergio.rodriguez@tecnalia.com
  - L. Torresani is with the Department of Computer Science, Dartmouth College, Hanover, NH 03755, USA.  
E-mail: lorenzo@cs.dartmouth.edu
  - A. Fitzgibbon is with Microsoft Research Cambridge, Cambridge CB1 2FB, UK.  
E-mail: awf@microsoft.com

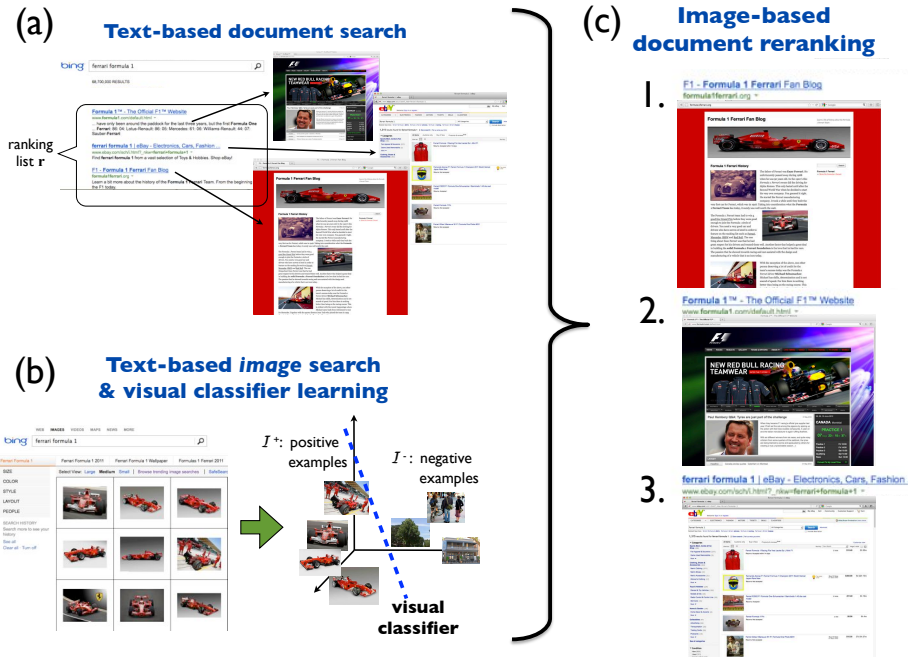


Fig. 1. Method overview: the query  $q$  is issued (a) to a document search engine producing a ranked list  $r$  of Web pages and (b) to a text-based image search engine yielding positive training image examples to learn a query-specific visual classifier. Finally, (c) the visual classifier is used to rerank the pages in the ranking list  $r$ .

can be used to measure the compatibility between the text query and the photos in a Web page. We address the first requirement by utilizing a compact attribute-based image descriptor—the classeme vector [4]—which has been shown to yield good results on high-level image understanding tasks even with simple linear classifiers, which are efficient to train and test. The second requirement is met by learning “on the fly” the visual model associated to the query keywords using as positive training examples the top image results of a text-based image search engine, such as Google Images [5] or Bing Images [6]. The learned visual classifier can then be used in conjunction with the text-matching techniques of traditional search engines to measure the compatibility between the query keywords expressed by the user and the content—now both *visual* as well as *textual*—of each Web page.

## 2 RELATED WORK

Related work falls into a number of categories: automatic textual annotation of images; the combination of image and text features to improve image retrieval; retrieval of “multimedia” documents using image and text; and the use of image features to boost relevance in text document retrieval.

A representative work belonging to the last of these genres, is the approach of Yu *et al.* [2], who collect a feature vector for each image in a document

which includes metadata such as aspect ratio, width and height, as well as looking at the pixels to compute “blurriness”, “colourfulness”, a flag indicating presence/absence of faces, and a graphic/photo flag. Then, from training data where users rate images by “importance” within a document, they learn an “image importance” classifier, which is applied to each image in the document. They show that adding this feature improves judged relevance in a document search task. In contrast to their work, our paper builds a *query-dependent* document representation which uses the image content at a semantic level. The system proposed by Yeh *et al.* [7] is another example of multimedia search. However, their method requires additional user input, in the form of an image accompanying the text query. The approach that most closely relates to our own is the work of Zhou and Dai [3], who were the first to show that content extracted from the pictures of Internet pages can be used to improve Web document search. Their system relies on an *unsupervised* method to discover a visual representation of the query from the images of Web pages retrieved via text search. The visual model of the query is computed via an iterative technique for density estimation aimed at finding the region of the visual feature space that contains the highest concentration of image examples associated to the query. These image examples are then averaged to form a single prototypical representation of the

query. Then, an image-based rank of candidate Web pages is computed by measuring the distance between the pictures in the page and the visual prototype of the query. This image-based rank is fused with a traditional keyword-based rank to form the final sorted list of documents. In our approach we replace the costly and brittle unsupervised method of this prior system with the supervised learning of a visual classifier by exploiting as training data the photos retrieved by a text-based image search engine. In our experiments we show that this yields much higher accuracy compared to the system of Zhou and Dai. Furthermore, we demonstrate that our approach has much lower runtime compared to the algorithm of this prior work. Finally, while the image-based system of [3] was tested only on 15 hand-selected visual queries, we report results on a large-scale independent benchmark for Web retrieval (the TREC 2009 Million Query Track (MQ09) [8]).

As mentioned, there is a vast amount of work which attempts to retrieve *images* using textual query terms. To summarize the state of the art (in terms of methodology rather than benchmark results), the recent paper of Schroff *et al.* [9] serves as an adequate exemplar. This work combines text, metadata and visual features in order to achieve a completely automatic ranking of the images pertaining to a given query. Their approach begins from Web pages, recovered by text search for the query. Then images in the pages are reranked using text and metadata features, and finally a form of pseudo-relevance feedback (PRF) [1] is used: a classifier is trained to predict high rankings, and re-rank the image list. This could well be a useful addition to our system, perhaps improving the training set for our image model, but as with any PRF system, the results are an amplification of successes and failures of the base algorithm, so we prefer to test the baseline systems without PRF. Among the methods for content-based image search using text keywords, we would like to mention the work of Krapac *et al.* [10] since, similarly to our approach, it also uses a query-relative representation. However, as already discussed above, using text features to improve image search is a qualitatively different problem than the one we introduce here.

Our work may be viewed as part of a more general endeavor: using images to help with problems in language. Barnard and Johnson [11] address the problem of word sense disambiguation in the context of words in image captions, and thus could hope to segment the results for ambiguous query terms. This might again be useful in a PRF addendum to our class of system.

Another sweep of related work is on automatic image annotation. Typically, classifiers are trained to label images with the object classes represented within. The key limitation of such methods from our point of view is that the number of classes is fixed in advance. Even the most ambitious current work

looks at only thousands of classes [12]. However, in the context of search, there are millions of possible queries, and because of the “long tail” it is unsatisfactory to focus only on the most common ones. Furthermore, even if 10000 classes were pre-trained, this would add thousands of bytes to each document, while our method enables search of *all* possible classes with less per-document data.

### 3 APPROACH OVERVIEW

The architecture of our system is schematically illustrated in Fig. 1. Let  $\mathcal{D}$  be the database of Web pages. In order to produce the list of relevant documents for an input query  $q$ , we use a reranking strategy combining traditional text-retrieval methods with the visual classifier learned for query  $q$ :

- 1) The query  $q$  is provided as input to a text-based search engine  $\mathcal{S}$  operating on  $\mathcal{D}$  to produce a ranking list  $\mathbf{r}$  of  $K$  candidate pages (Fig. 1(a)).
- 2) In parallel, the query  $q$  is issued to a keyword-based *image* search engine (in this work we use the visual search service of Bing Images); the top  $M$  image results  $\mathcal{I}^+$  are used as positive examples to train a visual classifier recognizing the query in images (Fig. 1(b)).
- 3) The list of pages  $\mathbf{r}$  is resorted (Fig. 1(c)) by taking into account several image features including the classification scores produced by evaluating the visual classifier on the pictures of the  $K$  candidate pages.

The key intuition is that when the query represents a visual concept, i.e., a concept that can be recognized in images, the learned visual classifier can be applied to increase or decrease the relevancy of a candidate page in the ranking list depending on whether the document contains pictures exhibiting that visual concept.

Our approach uses as image representation the binary “classeme” vector of Torresani *et al.* [4]. The  $C$  binary entries in this descriptor are the binarized outputs of  $C$  nonlinear object classifiers evaluated on the image. These base classifiers are pre-trained on an independent dataset corresponding to  $C$  distinct hand-selected object categories. This idea is evocative of the use of attributes [13], [14], [15] which are fully-supervised classifiers trained to recognize certain properties in the image such as “has beak”, “near water”. While attributes have been used as features for recognition in specialized domains (e.g., animal recognition [15] or face identification [13]), classemes are obtained by choosing a very large ( $C=2659$ ) and varied set of *visual* categories as basis classes: these are the so-called visual concepts of the Large Scale Concept Ontology for Multimedia (LSCOM) [16], which are concepts selected to be useful, observable and feasible for automatic visual detection, and as such are likely to form a good basis for image retrieval and object recognition tasks. The resulting classeme

descriptor was shown to be an effective universal feature representation for general object categorization, even classes different from the LSCOM categories [4].

In our context, the classeme vector provides two key benefits. First, it is very compact in size (the dimensionality of the binary vector is 2659, corresponding to only 333 bytes/image). Second, it has been shown to produce good classification accuracy with linear classifiers. These two properties enable efficient query-time learning and testing of the visual classifier for large databases. In our system we employ a linear Support Vector Machine (SVM) as visual classifier: for each input query  $q$ , we train an SVM to discriminate the set  $\mathcal{I}^+$  of Bing images retrieved for query  $q$  (we use  $M = 50$  positive examples) from a fixed collection of images representative of many object classes ( $\mathcal{I}^-$ ). Specifically, we use as negative set  $\mathcal{I}^-$  Bing images retrieved for many queries, thus essentially implementing a one-vs-the-rest strategy.

## 4 AN IMAGE-BASED MODEL FOR DOCUMENT RERANKING

We now describe in detail our reranking model, including our learning approach and the features used by it. Let  $q$  denote the input query, corresponding to the keywords issued by the user for the search. We use a query-relative representation of the documents: let  $\mathbf{x}^{(q,i)} \in \mathbb{R}^d$  be the feature vector describing the  $i$ -th document in the database  $\mathcal{D}$  relative to query  $q$ . Given an input query  $q$ , our approach enables real-time computation of the vector  $\mathbf{x}^{(q,i)}$  for each document  $i$ .

Our objective is to learn a query-independent reranking function  $f : \mathbb{R}^d \rightarrow \mathbb{R}$  such that  $f(\mathbf{x}^{(q,i)})$  provides a numerical estimate of the final relevancy of document  $i$  for query  $q$ , where  $i$  is one of the top- $K$  documents retrieved by  $\mathcal{S}$ . We denote with  $\theta$  the learning parameters of the function, i.e.,  $f(\mathbf{x}^{(q,i)}) = f(\mathbf{x}^{(q,i)}; \theta)$ . We learn the parameters  $\theta$  from a labeled training set  $\mathcal{T} = \{(q_1, \mathbf{r}_1, \mathbf{y}_1), \dots, (q_N, \mathbf{r}_N, \mathbf{y}_N)\}$  where  $\mathbf{r}_j$  is the sorted ranking list of  $K$  documents produced by the text-based search engine  $\mathcal{S}$  for input query  $q_j$ , i.e.,  $r_{jk} \in \mathcal{D}$  denotes the ID of the document ranked in the  $k$ -th position; the vector  $\mathbf{y}_j$  encodes the corresponding ground-truth relevance labels. In our implementation we use binary relevance labels with  $y_{jk} = 1$  denoting that document  $r_{jk}$  is relevant for query  $q_j$ , and value 0 indicating “non-relevant”.

### 4.1 The query-document features

Next, we present our choice of query-document features for image-based reranking. We would like to point out that we considered and tested several additional features not described here. However, for clarity we present only those that we found to be beneficial in terms of improving the ranking accuracy.

The vector  $\mathbf{x}^{(q,i)}$  for query-document pair  $(q, i)$  comprises the following 12 features.

- **Text features** ( $\mathbf{x}_{1,2}^{(q,i)}$ ): ‘*relevance score*’ and ‘*ranking position*’ of document  $i$  in the ranking list  $\mathbf{r}$  produced by the text-based engine  $\mathcal{S}$  for input query  $q$ . The ‘*relevance score*’ feature is a numerical value indicating the relevancy of the document  $i$  for query  $q$ , as estimated by  $\mathcal{S}$ , purely based on textual information. The ‘*ranking position*’ is the position of  $i$  in the ranking list  $\mathbf{r}$ . By including these two features we can leverage the high-accuracy of modern text-based search. Because our reranking function uses the ranking scores and positions generated by  $\mathcal{S}$ , it can be viewed as an extended version of  $\mathcal{S}$ , where visual information is exploited in addition to the traditional text features.
- **Visual metadata features** ( $\mathbf{x}_{3,4}^{(q,i)}$ ): ‘*# linked images*’ and ‘*# valid images*’. These attributes are used to describe whether the document contains many images. We expect that this information can be useful to the image-based reranker as it reveals whether the page contains a lot of visual information. The feature ‘*# linked images*’ is simply the number of images linked in the Web page. A potential problem is that Web pages often include a large number of small images corresponding to banners, clipart, icons and graphical separators. These images typically do not convey any information about the semantic content of the page. To remove such images from consideration, we extract the classeme descriptor only from pictures having at least 100 pixels per side. The feature ‘*# valid images*’ gives the total number of images in the page for which the classeme descriptor was computed. The ‘*# linked images*’ and ‘*# valid images*’ jointly inform the image-based reranker on whether the document is likely to contain advertisement or rather pictures potentially useful to check the semantic agreement between the query and the content of the page.
- **Query visualness features** ( $\mathbf{x}_{5,6}^{(q,i)}$ ): ‘*visual classifier accuracy*’ and ‘*visual concept frequency*’. These entries are features dependent only on the query (i.e., they are constant for all documents) and describe the general ability of the visual classifier learned for query  $q$  to recognize that concept in images. In particular, ‘*visual classifier accuracy*’ gives the cross-validation accuracy of the classifier trained on the examples retrieved by Bing Images for query  $q$ . We use 5-fold cross validation to determine the SVM hyperparameter and then store the best cross-validation accuracy over all hyperparameter values in the feature ‘*visual classifier accuracy*’. While this feature provides us with an estimate of how reliably the

classifier recognizes visual concept  $q$  in images, it does not convey how frequently this visual concept is present in pictures of Web pages. This information is captured by feature ‘*visual concept frequency*’ which is computed as the fraction of times the visual classifier for query  $q$  returns a positive score on the images of the database  $D$ . The intuition is that the joint analysis of the two query visualness features may provide the reranker with an indication of the usefulness of employing the visual classifier for query  $q$  to find relevant pages.

- **Visual content features** ( $\mathbf{x}_{7-12}^{(q,i)}$ ): the visual content features consist of the ‘*histogram of visual scores*’ and the ‘*document relevancy probability*’. The ‘*histogram of visual scores*’ is a five-bin histogram ( $\mathbf{x}_{7-11}^{(q,i)}$ ) representing the quantized distribution of the classification scores (i.e., the SVM outputs) produced by the visual classifier of query  $q$  on the images of document  $i$ . The histogram is unnormalized and thus the sum of histogram values is equal to ‘*# valid images*’. We set the bin bounds to correspond to the following percentiles of classification scores, estimated from a large number of queries: 30%, 45%, 60% and 80%. Thus, the histogram gives us the number of images in the document that yield classification score exceeding these thresholds. The histogram captures a measure of the semantic compatibility between the images in  $i$  and the query  $q$ . The ‘*document relevancy probability*’ ( $\mathbf{x}_{12}^{(q,i)}$ ) is an estimate of the posterior probability that the document  $i$  is relevant for query  $q$  given the observed classification scores of the images contained in the page, i.e.,  $p(i \text{ is relevant} | s_1, \dots, s_{n_i})$ , where  $s_1, \dots, s_n$  are the binarized scores that the SVM for query  $q$  produces on the  $n_i$  (valid) images of document  $i$ . This probability is computed via standard application of Bayes’s rule under the assumption of conditional independence (also known as the Naïve Bayes assumption) [17]. In our case, conditional independence means that the classification scores are independent given the relevancy status of the document. In other words we assume that

$$p(s_u | i \text{ is relevant}, s_v) = p(s_u | i \text{ is relevant})$$

and that

$$p(s_u | i \text{ is not relevant}, s_v) = p(s_u | i \text{ is not relevant})$$

for  $u \neq v$ . Under this hypothesis, the *posterior* probability that document  $i$  is relevant can be computed as follows:

$$p(i \text{ is relevant} | s_1, \dots, s_{n_i}) = \frac{p(i \text{ is relevant}) TP^{m_i} (1 - TP)^{n_i - m_i}}{p(s_1, \dots, s_{n_i})} \quad (1)$$

where  $m_i$  is the number of images of  $i$  having positive classification score while  $TP$  denotes the true positive rate of the classifier, i.e.,  $TP = p(s_u = 1 | i \text{ is relevant})$ . Finally, note that the denominator in Eq. 1 can be evaluated via application of Bayes rule:

$$p(s_1, \dots, s_{n_i}) = p(i \text{ is relevant}) TP^{m_i} (1 - TP)^{n_i - m_i} + p(i \text{ is not relevant}) FP^{m_i} (1 - FP)^{n_i - m_i} \quad (2)$$

where  $FP$  is the false positive rate. We assume that the rates  $TP$ ,  $FP$  are query-independent and we estimate them empirically over a large number of labeled training queries. In conclusion, the ‘*document relevancy probability*’ feature provides us directly with an estimate of the relevancy of the document purely based on the visual content of the images in the page. Note that, while it may appear that the ‘*document relevancy probability*’ and the ‘*histogram of visual scores*’ capture similar information, they actually represent the outputs of different classification models and we empirically found the inclusion of both these features to be beneficial to improve the reranking accuracy.

Finally, if a document does not contain any valid image, features  $\mathbf{x}_{3,4}^{(q,i)}$  and  $\mathbf{x}_{7-11}^{(q,i)}$  are set to zero.

## 4.2 Learning to rerank using visual content

We now describe the training procedure to learn the image-based reranking function  $f$  from the labeled training set  $\mathcal{T}$ . Note that this function is query independent, learned only once during an offline training stage. In our work we have experimented with the following different ranking models:

- 1) **Ranking SVM**. This algorithm learns a linear model of the input features, i.e.,  $f(\mathbf{x}^{(q,i)}; \boldsymbol{\theta}) = \boldsymbol{\theta}^T \mathbf{x}^{(q,i)}$ . Intuitively, the parameter  $\boldsymbol{\theta}$  is optimized so as produce a ranking function that preserves the ordering of the training examples in  $\mathcal{T}$ , i.e., such that

$$\boldsymbol{\theta}^T \mathbf{x}^{(q_j,k)} > \boldsymbol{\theta}^T \mathbf{x}^{(q_j,l)} \iff y_{jk} > y_{jl} .$$

Specifically, we use the learning objective of SVM ordinal regression proposed by Herbrich et al. [18]:

$$\min_{\boldsymbol{\theta}, \boldsymbol{\xi}} \frac{1}{2} \boldsymbol{\theta}^T \boldsymbol{\theta} + C \sum_{j=1}^N \sum_{k,l=1}^K \xi_{jkl} \quad \text{subject to:} \\ \forall q_j, k, l : \xi_{jkl} \geq 0 , \\ \forall q_j, k, l \text{ s.t. } y_{jk} > y_{jl} : \\ \boldsymbol{\theta}^T \mathbf{x}^{(q_j,k)} > \boldsymbol{\theta}^T \mathbf{x}^{(q_j,l)} + 1 - \xi_{jkl} . \quad (3)$$

It can be shown that this objective is convex. We optimize it using the efficient cutting plane algorithm described in [19].

- 2) **Random Forest**. This method learns a random forest [20] with each tree greedily optimized

to predict the relevancy labels  $y_{jk}$  of the training example. The resulting hypothesis computes an average of  $P$  independently trained regression trees  $f^{(1)}, \dots, f^{(P)}$ , i.e.,  $f(\mathbf{x}^{(q,i)}; \theta) = \frac{1}{P} f^{(P)}(\mathbf{x}^{(q,i)})$ . The  $P$  trees are diversified by considering at each split only  $d' < d$  randomly chosen features (we use the common rule of thumb of setting  $d'$  to 10% of the number of features). We let each tree to grow to full depth and avoid overfitting by using a large number of trees. The hyperparameter  $P$  is determined as the number of trees after which the cross-validation error stops decreasing.

- 3) **Gradient Boosted Regression Trees (GBRT).** Gradient Boosted Regression Trees (GBRT) were first introduced in [21] and have been shown to be among the best known models for document ranking (e.g., the best performing systems in the recent Yahoo Learning to Rank Challenge [22] use some form of GBRT). This model also predicts by averaging the outputs of  $P$  regression trees. However, unlike in case of the random forest where the trees are high-variance classifiers independently learned, the GBRT trees are trained in sequence and are constrained to have small depth so that each individual tree has a high bias (in our experiments we set the depth to 4). Each tree is optimized to correct the prediction of the training documents responsible for the current regression error (for further details on the learning procedure see [23]). In this case  $P$  is chosen via grid-search on the cross-validation error. The random forest and the GBRT are trained with the code from [24].

## 5 DISCUSSION OF COMPUTATIONAL AND STORAGE COSTS

In the paragraphs below we describe in detail the additional computational and storage cost posed by our image-based reranking scheme. Our analysis shows that our system is sufficiently efficient to be deployed on existing search engines for Web-document search without causing any significant delay in response time.

**Gathering image examples for a query.** Note that while we exploited an existing image-search engine (Bing Images [6]) to retrieve images from which classeme features were then extracted, in a real application scenario the classeme vectors (which are query-independent) would be precomputed at the time of the creation of the index by the image-search service. Then the image and document queries would be issued *in parallel*, and the image index would return only the 333 bytes classeme vector per image. As a reference, Google Images [5] and Bing Images [6] report search times of a few tens of a second per query.

This would be the effective time needed to retrieve precomputed classemes for the top images of a query.

**Training the visual classifier.** One of the fundamental advantages of the classeme vector is that it enables excellent recognition accuracy even with *linear* classifier, which are very efficient to train [4]. For example, using for each query a positive training set  $\mathcal{I}^+$  of 50 Bing Images and a fixed negative set  $\mathcal{I}^-$  of 30,000 images, the learning of a linear SVM on classeme features with the LIBLINEAR software [25] takes on average 0.056 seconds (runtimes were measured on a standard budget PC with an Intel Core i7-930 CPU @ 2.80GHz). As a comparison, note that running the unsupervised image-based learning method of [3] on the photos of the top-200 documents for a given query takes much longer: 3 minutes and 9 seconds on average, using the same machine reported above.

**Testing the visual classifier on images of Web pages.** We would like to emphasize that our reranking scheme requires evaluation of the visual classifier only on the images of the top-200 documents retrieved by the text-search engine. Thus, under the assumption that the classeme feature vectors are precomputed for all images of Web pages (see paragraph below for an analysis of the added storage cost), the testing of the linear visual model for each query becomes negligible: only 0.0192 milliseconds on average in our tests performed on the same computer listed above.

**Storage cost of the visual representation.** As for the storage cost, our system requires saving the classeme vectors of the valid images in each Web page. In the Category B Clueweb09 dataset [26] used for our experiments, each document contains on average 1.44 valid images. Thus, the added storage cost due to the use of images is less than 500 bytes per document, which can be easily absorbed by modern retrieval systems.

## 6 EXPERIMENTS

### 6.1 Training and evaluation setup

We evaluate our system on the benchmark of the TREC 2009 Million Query Track (MQ09) [8], which involves ad-hoc retrieval over a large set of queries. The benchmark is based on the “Category B” ClueWeb09 dataset [26] which includes roughly 50 million English pages crawled from the Web. The publicly available distribution of this dataset includes the original HTML pages collected by the ClueWeb09 team in January and February 2009, but not the images linked in them. In order to run our image-based system on this collection, in September 2011 we attempted to download all pictures linked in these documents. Unfortunately many of the original pages and images were no longer available on the Web. Thus here we restrict our experimental analysis only to the pages for which we successfully downloaded *all* images linked

		statMPC@10 (%)	statMPC@30 (%)
$S=UDMQ$	Ranking using text only ( $S$ )	48.2	38.8
	Image-based reranking of Zhou and Dai [3]	49.8	40.4
	Our image-based reranking w/ Ranking SVM	48.3	38.7
	Our image-based reranking w/ Random Forest	53.2	32.5
	Our image-based reranking w/ GBRT	<b>64.5</b>	<b>40.5</b>
$S=Indri$	Ranking using text only ( $S$ )	27.7	27.7
	Image-based reranking of Zhou and Dai [3]	27.2	27.7
	Our image-based reranking w/ Ranking SVM	27.8	27.3
	Our image-based reranking w/ Random Forest	31.6	23.4
	Our image-based reranking w/ GBRT	<b>37.3</b>	27.2

TABLE 1

Mean precision @ 10 and 30 on the TREC 2009 Million Query benchmark using different ranking models. Top: search engines based on UDMQ. Bottom: search engines based on Indri. Our image-based GBRT reranker achieves by far the best precision @ 10 by greatly outperforming the text-based search engines (UDMQ and Indri). The image-based reranking schemes do not provide significant advantages in precision @ 30.

in the original document. This amounts to 41% of the pages in the Category B ClueWeb09 collection.

In order to train and test our reranking system, we use the publicly available MQ09 queries and human relevance judgements [27]. These are suitable labels to train an image-based system as the judgements were collected by showing the assessors the full Web pages, with pictures included. In all, judgements are available for 684 queries, with each query receiving either 32 or 64 document assessments. The possible relevance values are “not relevant” ( $y_{jk} = 0$ ), and “relevant” ( $y_{jk} = 1$ ). The report of the MQ09 competition [8] asserts that “it is possible to reuse MQ09 topics and judgments for within-site comparisons, that is, comparisons between new runs that are developed by the same sites that contributed to the track”. Thus, in order to meet the conditions for reusability of the MQ09 topics and judgements, we selected as one of our base text-search engines  $S$  the UDMQxQEWB system [28], which was one of the systems participating in the MQ09 competition and actually the one achieving the highest accuracy on this benchmark. The ranking lists of UDMQxQEWB on the MQ09 queries are publicly available at [29]. The UDMQxQEWB engine uses an axiomatic retrieval model to rank documents based on a semantic term matching method (for further details on the approach we refer the reader to [28]). For brevity, we refer to this system as UDMQ.

In order to test the flexibility of our approach to work with different text-search systems  $S$ , we also present results using the popular Indri search engine [30]. We selected this engine as it implements state-of-the-art ranking methods and provides a batch query service on the Category B ClueWeb09 dataset through which we were able to obtain the ranking lists for the MQ09 queries. Unlike UDMQ, Indri did not participate to the MQ09 competition. Thus, while the estimate of the *absolute* accuracy of Indri according to the MQ09 relevance judgements may be unreliable,

here we use it just as a baseline to judge the *relative* improvement produced by reranking its search results with our system.

We generate the vectors  $\mathbf{r}$  by truncating the ranking lists of both search engines at  $K = 200$ . We employ 10-fold cross validation over the queries, thus using in each run 9/10th of the queries for training and the remaining 1/10-th of the queries for validation. Performance is measured as estimated precision at 10 and 30, which give the proportion of relevant documents in the top 10 and 30, respectively. We focus on these performance measures as our main goal is to improve the relevancy of the documents in the top part of the ranking list. The precision is estimated using the “statistical evaluation method” [8], which draws and judges random samples of documents from the given ranked lists and produces unbiased, low-variance estimates. We use the official MQ09 evaluation scripts and denote the resulting performance measures as statMPC@10 and statMPC@30.

## 6.2 TREC results

We begin by comparing the accuracy of the text-based search engines (UDMQ and Indri) to the different image-based ranking models introduced in section 4.2. In addition, we include results obtained with the image-based system of Zhou and Dai [3]. Note that for this algorithm we also use classemes as representation for the images, so as to compare all image-based reranking schemes on equal ground, i.e., on the same visual input. We compute the prototypical representation of the query proposed in [3] using the images in the top-200 documents retrieved by the text-search engine and then fuse their image-based rank with the text-based rank according to the model described in their paper. As for our system, even for this algorithm we tuned all hyperparameters using cross-validation.

Table 1 summarizes the results. First, we see that all image-based rerankers yield higher values of

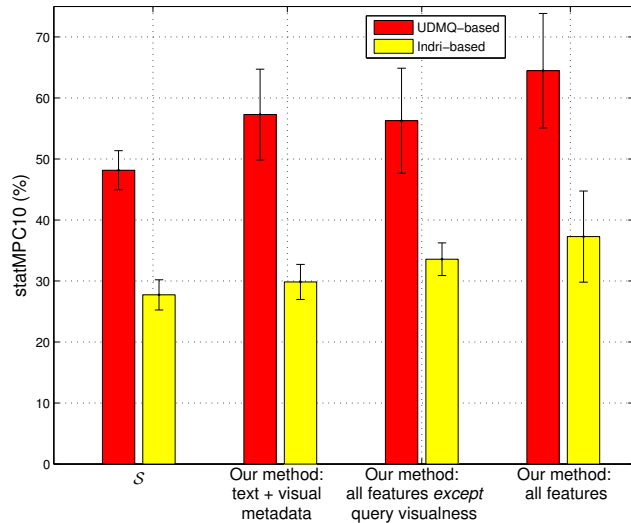


Fig. 2. Mean precision @ 10 using different image features with the GBRT reranker based on UDMQ (red) and Indri (yellow). Removing the visual content features (“text + visual metadata”) or the visualness features from our descriptor causes a drop in performance, which suggests that all our features contribute to the improvement over the pure text-based system  $S$ .

statMPC@10 than the base search engines using text only. Among these models, the GBRT visual reranker is by far the best, improving by over 33% the precision of UDMQ, which achieved the highest accuracy among all search engines participating in the MQ09 competition. This is a clear indication that our image-based features provide new and relevant information compared to that captured by traditional text-based engines. Instead, no much gain is achieved in terms of statMPC@30. Empirically we found that our reranker tends to apply fairly small displacements to the positions of documents in the original ranking list. While these small rearrangements have a positive impact on the top-10 lists examined by statMPC@10, they are too small to change sensibly the statMPC@30. Furthermore, we note that, since we adopt a reranking strategy, the statMPC@30 of our system is limited by the number of relevant documents among the  $K$  candidate pages retrieved by the text-search engine. In fact, we found that the performance of our algorithm is not far off from that produced by an *ideal* reranker that places the ground-truth relevant documents of the candidate list at the top ranking positions: such perfect system would give a value of 43.6% in statMPC@30. This indicates that in order to further increase the accuracy in the top 30 positions, it is necessary to improve the recall of relevant documents in the initial candidate list.

Next, we would like to investigate which of our features contribute to the statMPC@10 improvement. For this purpose we select the GBRT-based visual reranker

Single feature used	statMPC@10 (%)
$x_1^{(q,i)}$ ('relevance score')	48.2
$x_2^{(q,i)}$ ('ranking position')	48.2
$x_3^{(q,i)}$ ('# linked images')	15.8
$x_4^{(q,i)}$ ('# valid images')	10.3
$x_5^{(q,i)}$ ('visual classifier accuracy')	N/A
$x_6^{(q,i)}$ ('visual concept frequency')	N/A
$x_{7-11}^{(q,i)}$ ('histogram of visual scores')	21.7
$x_{12}^{(q,i)}$ ('document relevancy probability')	15.9

TABLE 2

Reranking using only one feature, rather than the full set (results based on the candidate list given by UDMQ). The text-based features ( $x_1^{(q,i)}$  and  $x_2^{(q,i)}$ ) obviously provide the best performance when used individually, but the ‘histogram of visual scores’ is also an effective single feature for reranking.

as our model since it produced clearly superior results over the other models. We retrain the GBRT model using two different variants of our feature vector: 1) “text + visual metadata” (i.e., we use only the subvector  $x_{1-4}^{(q,i)}$  consisting of the first four features, which do not capture the content of the images); 2) the vector “all features – visualness” (i.e., all our features *excluding*  $x_{5,6}^{(q,i)}$ , which capture the document-independent visualness of the query in terms of the visual classifier accuracy and the visual concept frequency). The results are presented in Figure 2 using UDMQ (red bars) and Indri (yellow bars) as base retrieval models  $S$ . From this plots we see that, although GBRT with the “text + visual metadata” descriptor achieves accuracy slightly superior to the text-based search engines, the performance is not as good as when our approach uses all features, *including* the visual content. This suggests that despite the noisy nature of the Bing images utilized as training examples, the resulting visual classifier does capture information that is useful to predict whether a document is relevant with respect to the input query. Finally, we would like to point out that excluding the query visualness features from our feature vector also causes a drop in accuracy. We believe that this happens as these features allow the reranker to determine whether the input query is visually recognizable and specific so as to properly modulate the contribution of the visual content features in the reranking function.

Table 2 provides interesting information about the importance of each individual feature by showing the statMPC@10 accuracy achieved by using a single feature at a time from our set. Note that for this experiment we treat the entries  $x_{7-11}^{(q,i)}$  as a single feature, since they collectively represent a histogram of scores. We omit from this analysis  $x_5^{(q,i)}$  and  $x_6^{(q,i)}$  since these features are constant for all documents of a query and thus not useful individually for ranking. As expected,



		% queries	median gain in prec@10 (%)	median image classif. error (%)
$\mathcal{S}$ =UDMQ	queries where $\mathcal{S}$ wins	15.3	20.0	29.4
	queries where our method wins	12.6	33.1	25.7
	queries with a tie	72.1	n/a	27.6
$\mathcal{S}$ =Indri	queries where $\mathcal{S}$ wins	12.6	20.0	27.7
	queries where our method wins	14.5	29.5	25.2
	queries with a tie	72.9	n/a	28.4

TABLE 3

A comparison across queries between the text-based engines and the GBRT image-based reranker: the “queries where  $\mathcal{S}$  wins” are those for which the text-based search engine provides higher prec@10 than our approach. The “median image classification error” is computed from the cross-validation error of the visual classifier. Note that this error is higher for the queries where  $\mathcal{S}$  wins compared to the queries where our approach is superior: this suggests that our method does better when the query is more easily recognizable in images.

the textual features ( $\mathbf{x}_1^{(q,i)}$  and  $\mathbf{x}_2^{(q,i)}$ ) provide the best performance as single features. However, the results also indicate that  $\mathbf{x}_{7-11}^{(q,i)}$  (“*histogram of visual scores*”), which captures image *content*, is an effective feature for document reranking. Surprisingly,  $\mathbf{x}_3^{(q,i)}$  (“*# linked images*”) is also a fairly good individual feature. This suggests that, among the pages in the candidate list, the documents that contain many images tend to be judged more relevant than those with fewer pictures, perhaps indicating a bias of the human subjects in favor of pages that are visually rich.

We have also performed the “dual” experiment in which we measured the effective drop in performance caused by the removal of each individual feature from the complete set. Again, we found that all features contribute to increasing accuracy. Even the removal of simple features such as  $\mathbf{x}_3^{(q,i)}$  (“*# linked images*”) or  $\mathbf{x}_4^{(q,i)}$  (“*# valid images*”) causes a drop in accuracy (a decrement of 3.4% and 0.1% in statMPC@10, respectively).

In order to verify that our performance gain is truly due to visual recognition rather than matching of image-duplicates, we checked for the presence of identical copies of the Bing images in the pictures of the top-10 documents. We found zero duplicates. In addition, we tested for “near duplicates”, such as images differing only in terms of size, compression, or cropping: for each query, we manually inspected the image pair with the smallest L2 classeme distance (among all pairs of Bing images and photos in the top-10 documents) and we found a single near-duplicate out of more than 600 queries (a photo of the 1789 US judiciary act saved at two different resolutions). This stresses that the improvement enabled by the features  $\mathbf{x}_{7-12}^{(q,i)}$  in our vector is truly due to the extraction of semantic information from the images.

In Table 3 we report the percentage of queries for which our image-based GBRT reranker provides a higher value of prec@10 than  $\mathcal{S}$ , i.e., “wins” over the text-based engine. Our method and  $\mathcal{S}$  are tied

for roughly 72% of the queries, while the number of times one wins over the other are fairly evenly divided. However, in the cases where our system wins, it provides a much higher boost in prec@10, compared to the cases when  $\mathcal{S}$  wins (+33.1% vs +20% and +29.5% vs +20% when comparing to UDMQ and Indri, respectively). Finally, it is interesting to observe that the median cross-validation error of the visual classifier is lower for the queries where our system improves over  $\mathcal{S}$  compared to the queries where  $\mathcal{S}$  does better.

In Figure 3 we show a few examples of queries where our approach does better than UDMQ and queries corresponding to the opposite case.

### 6.3 Validation using crowdsourcing

In this section we describe an exhaustive validation of our search results using a crowdsourcing platform. The purpose of this experiment is to perform a direct comparison between our system and the text-based search engine that is used by our method to produce the initial candidate list. While the TREC MQ09 relevance judgements allow us to obtain an *absolute* measure of precision, in this crowdsourcing experiment we are focusing exclusively on the task of verifying the beneficial effect of our image-based reranking method with respect to the text-based retrieval baseline. Thus, we perform our own independent evaluation by collecting relevance labels using the crowdsourcing platform of Amazon Mechanical Turk (MTurk) [31] for both our search results (using UDMQ as text-search engine  $\mathcal{S}$ ) and those directly produced by UDMQ.

Specifically, for each MQ09 query, we generated a list of potentially relevant documents by merging the top 10 documents retrieved by UDMQ and the top 10 returned by our system using GBRT as reranking method. The Web pages in this list were then inspected for relevance by MTurk workers. To avoid biases we reshuffled the order of the documents in


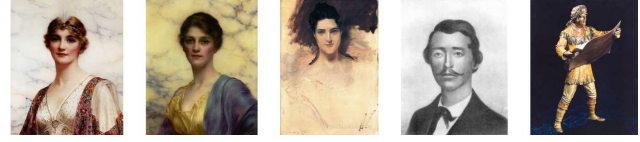


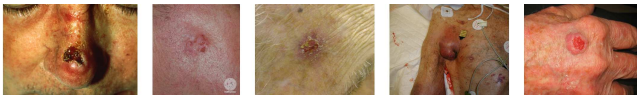

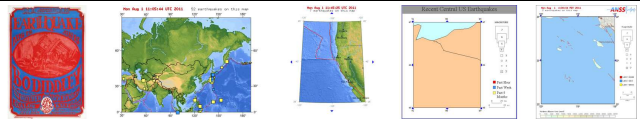
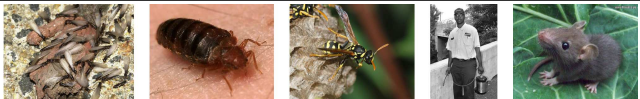




	Query	Image error	Bing training images
(a)	raw juices	0.21	
	william clark	0.24	
	rio grande	0.25	
	foodnetwork	0.28	
	squamous cell carcinoma	0.15	
	elliptical trainer	0.12	
(b)	all earthquake list	0.30	
	pest control education	0.46	
	sterling planet	0.36	
	piggly wiggly store ads	0.29	
	alcoholism genetics	0.37	
	wounded warrior act	0.34	

Fig. 3. Visualization of a few queries where our system produces (a) higher and (b) lower values of prec@10 compared to UDMQ. For each query we show the 5 Bing training images that receive the largest classification score by the learned visual classifier. The image error is the cross-validation error rate of the visual classifier trained on Bing images. Note how the image error tends to be lower for the queries in (a) compared to those in (b): indeed, our approach tends to do better when the query corresponds to a visual concept (see results in Table 3).

	MP@10 (%)	higher prec@10 (% of queries)	MAP@10 (%)	higher AP@10 (% of queries)
UDMQ	64.7	13.5	54.5	28.9
Our method	66.8	26.7	56.6	47.6

TABLE 4

Exhaustive manual assessment of the top-10 search results for all MQ09 queries using crowdsourcing. Our method is found to yield higher “Mean Precision at 10” (MP@10) and higher “Mean Average Precision at 10” (MAP@10) compared to UDMQ. Furthermore, our system provides better ranking results on many more queries than UDMQ in terms of both “Precision at 10” (prec@10) and “Average Precision at 10” (AP@10).

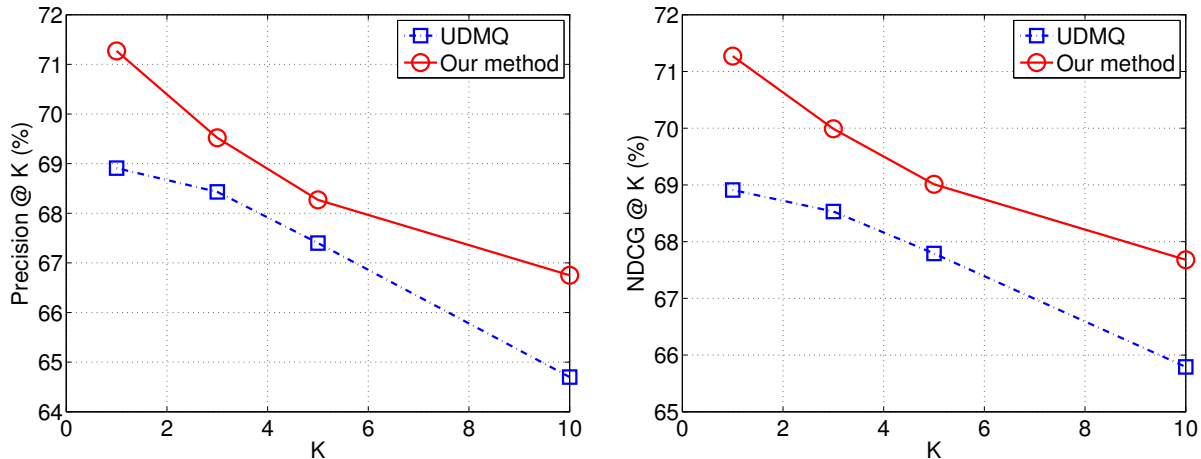


Fig. 4. Precision and Normalized discounted cumulative gain (NDCG) obtained with UDMQ and our approach for a varying number of top documents retrieved ( $K$ ). Here we focus on performance when  $K$  is small, as typically in Web search we are mostly interested in producing accurate results for the first few documents retrieved. Our method outperforms the text-based search engine UDMQ according to both accuracy metrics.

the list before presenting the pages to the assessors. For each query-document pair we requested a binary relevance label by 5 distinct MTurk workers. We computed the final relevance label for each document-query pair as the median of the 5 individual judgements (i.e., the majority label), since this labeling strategy was found to be the least error-prone among those evaluated in [32]. We found that on average for each document-query pair 3.7 workers (out of 5) agreed and voted for the majority label (the average is 3.8 for the documents labeled as relevant and 3.5 for those labeled as not relevant). These numbers indicate that the labels provided by the untrained MTurk workers are quite noisy, probably too noisy to be used to estimate absolute values of precision. However, they are still useful for a direct comparison between the two systems in order to determine if the image-based reranking is beneficial (see [32] for a discussion of the reliability of judgements collected via crowdsourcing).

The results of this manual assessment are summarized in Table 4. Using these labels we computed the Mean Precision at 10 (MP@10) for both UDMQ and our system: the resulting accuracy of our system was found to be 66.8% versus a value of 64.7%

for UDMQ. Thus, this independent test confirms the accuracy improvement of our system over UDMQ, albeit with a smaller margin than that estimated by the statMPC@10 measure of MQ09 (again, we stress that the MTurk-based numbers should not be interpreted as absolute accuracy values but rather as indicators of relative performance). Furthermore, according to these human-judgement labels, our system provides better precision at 10 than UDMQ on 26.7% of the queries, while UDMQ wins on only 13.5% of the queries (the two systems are tied on 59.8% of the queries).

We also measured the performance of the two systems in terms of Mean Average Precision at 10 (MAP@10), which is the sample mean over all queries of the Average Precision at 10 (AP@10) computed for each query. The AP@10 is an appropriate evaluation measure as it considers for each query not only how many relevant documents are in the top-10 but also how these are ranked within the top-10 list, by rewarding rankings where the relevant documents occupy the first few positions. From Table 4 we see that even according to this performance measure our system yields better results on many more queries than UDMQ (47.6% versus 28.9% of the queries) and

	UDMQ	Our method
MP@10 (%) on Visual Queries	66.2	66.8
MP@10 (%) on Non-Visual Queries	64.3	66.8

TABLE 5

Mean precision @ 10 of UDMQ and our approach on *visual* and *non-visual* queries. In this experiment we declare a query to be visual if the cross-validation error of the visual classifier trained on Bing images of that query is below a certain threshold. Our approach outperforms UDMQ even on non-visual queries (although the improvement is much larger for visual queries).

provides overall higher MAP@10.

In Figure 4 we plot Mean Precision and Normalized discounted cumulative gain (NDCG) as a function of the number of documents retrieved for both UDMQ and our image-based document reranking. This figure shows that our use of image content consistently improves accuracy of the very top documents.

Because our system reranks using image content, we expect that it should yield improvements only on queries that have a strong visual meaning. This suggests that perhaps improved performance can be obtained with a hybrid approach that first determines whether the query is *visual* or *non-visual*; then the system would use our reranking approach only if the query is visual, while it would handle non-visual queries via traditional text-based search (i.e., using UDMQ). We implemented such a hybrid architecture using the cross-validation image error of the visual classifier trained on Bing Images to decide whether the query is visual (note that we optimized the threshold on the visual classifier error to yield the best possible accuracy with this hybrid method). However we found that the resulting MP@10 of the hybrid algorithm is 66.6%, i.e. lower than the 66.8% obtained by using our image-based reranking method on every query. The reason is revealed by Table 5, which shows the accuracy of UDMQ and our method separately on visual queries and non-visual queries: our approach does better than UDMQ on both visual and non-visual queries, although the improvement is particularly noticeable on visual queries. We have obtained the same qualitative result by manually classifying the queries into visual and non-visual. Even in such case our method performed better than UDMQ on both kinds of queries.

In Figure 5 we show the relevant Web pages *added* to the top-10 list by our method compared to the ranking produced by UDMQ for the best 5 queries. This figure provides an intuitive insight about the accuracy improvements enabled by our system: note how nearly all the documents inserted in the top-10 list contain pictures semantically related to the query.

## 7 CONCLUSIONS

In this paper we have investigated the largely unexplored topic of how to use images to improve Web document search. We have demonstrated that by using modern methods and representations for image understanding, it is possible to enrich the semantic description of a Web page with the content extracted from the pictures appearing in it. We have shown that this yields a 33% relative improvement in accuracy over a state-of-the-art text-based retrieval baseline. All this is achieved at the small cost of a few additional hundred bytes of storage for each page. While in this work we have focused on a reranking strategy, we believe that our framework is sufficiently efficient to support in the near future the application of a single joint search model over text and images in the Web collection.

## 8 ACKNOWLEDGEMENTS

This work was supported in part by Microsoft Research, NSF CAREER award IIS-0952943 and NSF award CNS-1205521. This research was performed while the first author was a visiting student at Dartmouth College partly funded by the Basque Government under grant number IE11-316.

## REFERENCES

- [1] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.
- [2] Q. Yu, S. Shi, Z. Li, J.-R. Wen, and W.-Y. Ma, "Improve ranking by using image information," in *Advances in Information Retrieval, 29th European Conference on IR Research, ECIR 2007, Rome, Italy, April 2-5, 2007, Proceedings*, 2007, pp. 645–652.
- [3] Z.-H. Zhou and H.-B. Dai, "Exploiting image contents in web search." in *IJCAI*, 2007, pp. 2922–2927.
- [4] L. Torresani, M. Szummer, and A. W. Fitzgibbon, "Efficient object category recognition using classemes," in *Proc. of the European Conference on Computer Vision (ECCV)*, 2010, pp. 776–789.
- [5] "Google images," Website, <http://images.google.com>.
- [6] "Bing images," Website, <http://www.bing.com/images>.
- [7] T. Yeh, J. J. Lee, and T. Darrell, "Photo-based question answering," in *Proceedings of the 16th ACM international conference on Multimedia*, ser. MM '08. New York, NY, USA: ACM, 2008, pp. 389–398.
- [8] B. Carterette, V. Pavlu, H. Fang, and E. Kanoulas, "TREC Million Query Track 2009 Overview," in *TREC*, 2009.
- [9] F. Schroff, A. Criminisi, and A. Zisserman, "Harvesting image databases from the web," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 4, pp. 754–766, 2011.
- [10] J. Krapac, M. Allan, J. J. Verbeek, and F. Jurie, "Improving web image search results using query-relative classifiers," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 1094–1101.
- [11] K. Barnard and M. Johnson, "Word sense disambiguation with pictures," *Artificial Intelligence*, vol. 167, no. 1, pp. 13–30, 2005.
- [12] K. L. Jia Deng, Alexander C. Berg and L. Fei-Fei, "What does classifying more than 10,000 image categories tell us?" in *Proc. of the European Conference on Computer Vision (ECCV)*, vol. LNCS 6315, 2010, pp. 71–84.
- [13] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," in *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2009, pp. 365–372.

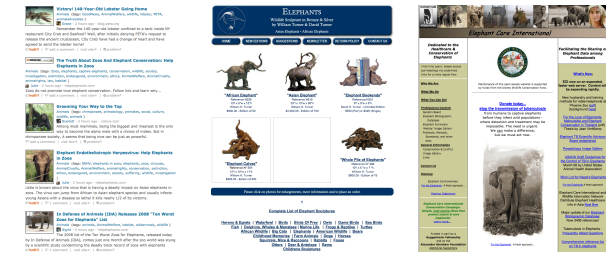


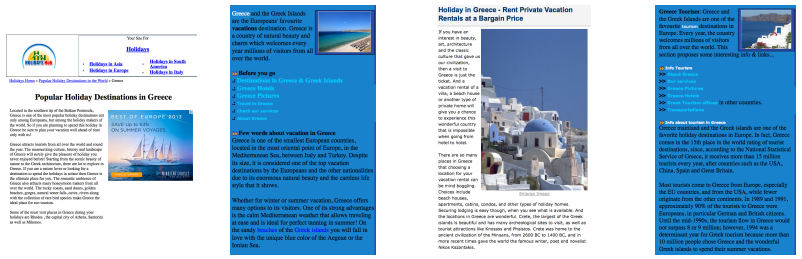

Query	Image error	Relevant Web pages that are in the top-10 list of our method and <i>not</i> in the top-10 list of UDMQ
elephant list	0.17	
united post office	0.17	
western quilt patterns	0.16	
greece	0.18	
erigonum jamesii	0.08	

Fig. 5. Visualization of the 5 queries where our system yields the highest gain in prec@10 according to the crowdsourced relevance labels. For each query we show the relevant Web pages that are ranked in the top-10 list of our system and that are *not present* in the list of UDMQ. Note that nearly all documents contain pictures that are highly representative of the concept expressed by the query.

- [14] A. Farhadi, I. Endres, D. Hoiem, and D. A. Forsyth, "Describing objects by their attributes," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 1778–1785.
- [15] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 951–958.
- [16] "Lscom: Cyc ontology dated (2006-06-30)," <http://lastlaugh.inf.cs.cmu.edu/lscm/ontology/LSCOM-20060630.txt>, <http://www.lscm.org/ontology/index.html>.
- [17] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, August 2006.
- [18] R. Herbrich, T. Graepel, and K. Obermayer, "Large margin rank boundaries for ordinal regression," in *Advances in Large Margin Classifiers*, 2000.
- [19] T. Joachims, "Training linear svms in linear time," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '06. New York, NY, USA: ACM, 2006, pp. 217–226.
- [20] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [21] J. H. Friedman, "Greedy Function Approximation: A gradient boosting machine," *The Annals of Statistics*, 2001.
- [22] "Yahoo learning to rank challenge," Website, <http://learningtorankchallenge.yahoo.com/>.
- [23] Z. Zheng, H. Zha, T. Zhang, O. Chapelle, K. Chen, and G. Sun, "A General Boosting Method and its Application to Learning Ranking Functions for Web Search," in *NIPS*, 2007.
- [24] A. Mohan, Z. Chen, and K. Q. Weinberger, "Web-search ranking with initialized gradient boosted regression trees," *Journal of Machine Learning Research - Proceedings Track*, vol. 14, pp. 77–89, 2011.
- [25] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.
- [26] "Carnegie Mellon University, Language Technologies Institute. The ClueWeb09 Dataset," Website, 2009, <http://lemurproject.org/clueweb09.php/>.
- [27] TREC, "TREC 2009 Million Query Track - Prels relevance judgements," Website, <http://trec.nist.gov/data/million.query/09/prels.20001-60000.gz>.
- [28] W. Zheng and H. Fang, "Axiomatic Approaches to Information Retrieval- University of Delaware at TREC 2009 Million Query and Web Tracks," in *TREC*, 2009.
- [29] TREC, "UDMQAxQEWeb ranking lists at TREC 2009 Million Query Track," Website, <http://trec.nist.gov/results.html>.
- [30] T. Strohan, D. Metzler, H. Turtle, and W. B. Croft, "Indri: a language-model based search engine for complex queries," in *International Conference on Intelligent Analysis*, 2005.
- [31] "Amazon mechanical turk," Website, <https://www.mturk.com>.
- [32] O. Alonso and S. Mizzaro, "Using crowdsourcing for trec relevance assessment," *Information Processing and Management*, vol. 48, no. 6, pp. 1053–1066, Nov. 2012.



**Lorenzo Torresani** is an Associate Professor in the Computer Science Department at Dartmouth College. He received a Laurea Degree in Computer Science with summa cum laude honors from the University of Milan in 1996, and an M.S. and a Ph.D. in Computer Science from Stanford University in 2001 and 2005, respectively. He has worked at several industrial research labs including Microsoft Research, Like.com and Digital Persona. His research is in computer vision and machine learning. In 2001, Torresani and his coauthors received the Best Student Paper Award at the IEEE Conference On Computer Vision and Pattern Recognition (CVPR). He is the recipient of a National Science Foundation CAREER Award and a Google Faculty Research Award.



**Andrew W. Fitzgibbon** is a principal researcher at Microsoft Research Cambridge, where he heads the computer vision group. He is best known for his work on 3D vision, having been a core contributor to the Emmy-award-winning 3D camera tracker "boujou" ([www.boujou.com](http://www.boujou.com)) and Kinect for Xbox 360, but his interests are broad, spanning computer vision, graphics, machine learning, and even a little neuroscience. He has published numerous highly-cited papers, and received

many awards for his work, including several "best paper" prizes, the Silver medal of the Royal Academy of Engineering, and the BCS Roger Needham award. He is a fellow of the British Computer Society, and of the International Association for Pattern Recognition. Before joining Microsoft in 2005, he was a Royal Society University Research Fellow at Oxford University, having previously studied at Edinburgh University, Heriot-Watt University, and University College, Cork.



**Sergio Rodriguez-Vaamonde** is a Ph.D. student in Computer Science at the University of the Basque Country (Spain). He holds a B.S. and an M.S. in Telecommunications Engineering as well as an M.S. in Information Technologies from the University of the Basque Country. He was a visiting researcher in the Visual Learning Group at Dartmouth College (USA), led by Professor Lorenzo Torresani, from April to July 2012.

Since 2008 he has been working as a researcher in the Computer Vision Business Area of Tecnalia, Spain. Since 2012 he has been involved in several publicly-funded national and European research projects in the field of information retrieval, machine learning and computer vision.