Learning-based Incremental Creation of Web Image Databases

Marian George
Computer Science Department
ETH Zurich
Zurich, Switzerland
Email: marian.george@inf.ethz.ch

Nagia Ghanem, M A Ismail
Computer and Systems Engineering
Alexandria University
Alexandria, Egypt
Email: {nagia.ghanem|maismail}@alex.edu.eg

***Abstract*—Manually creating an object category dataset requires a lot of hard work and wastes a large amount of time. Having an automatic means for collecting images that represent different objects is crucial for the scalable and practical expansion of these datasets. In this work, a methodology to automatically re-rank the images returned from a web search engine is proposed to improve the precision of the retrieved results. The proposed system works in an incremental way to improve the learnt object model and achieve better precision in each iteration. Images along with their meta data are ranked, then re-filtered based on their textual and visual features to produce a robust set of seed images. These images are used in learning weighted distances between the images which are used to incrementally expand the collected dataset. Using our method, we automatically gather very large object category datasets. We also improve the image ranking performance of the retrieved results over web search engines and other batch methods.**

***Keywords*-image retrieval; incremental learning; object recognition; visual object category datasets;**

I. Introduction

The evaluation of algorithms which aim at recognizing object categories, analyzing scenes, and performing content-based multimedia search, requires a well-built dataset which contains many and diverse images of the representative object category. Having such large databases serves the object category recognition problem in many different ways. For instance, simultaneous recognition and segmentation can be applied. When the database of images becomes large enough, it is even possible to directly match complete images with the expectation of finding a good match.

Many large databases currently exist to be used in the training and evaluation of object recognition algorithms, such as Caltech 101 [1], the UIUC car dataset [2], etc. Generally, all of these datasets are limited by the number of images representing each object with no other means to expand them except through costly human labor. Thus, recently, there has been an increasing need for an automatic way to collect large object databases which can scale to many object categories and large number of images with minimal cost.

Nowadays, large amounts of images are increasingly available to us through the World Wide Web. The Web provides an easy way to search for images through image search engines. However, we cannot just query an image search engine, download the returned results and use them as a visual object category dataset to the low precision of the results. Recently, researchers have developed approaches to utilize the images retrieved by image search engines to collect datasets automatically. However, current commercial image retrieval software is built upon text search techniques which contaminate the retrieved images with visually irrelevant images. Also, these systems face many challenges like intra-class appearance variance and polysemy. Visual polysemy means that a word has several dictionary senses that are visually distinct like when we query for the word "mouse". We draw inspiration from relevance feedback algorithms in iteratively improving the precision of retrieved images but without incurring any cost of manual interference. Our system works in a completely autonomous way, which proves to be more scalable and practical for the task of harvesting well-built object datasets.

The objective of this work is to extract a large number of images for a given object category (query). Our system starts by querying a web search engine which provides a very large number of images for any given object category(query), along with meta data describing the webpage from which each image was downloaded. We first rank the images based on their textual and visual features. This provides a robust set of seed images to be used in the next step which is learning weighted distances between the images. We use the top ranked images to learn a weighted L1 distance. This distance function is then used to classify the downloaded corpus of images as being relevant or irrelevant to the given query. We incrementally expand the collected dataset using the newly added images in each iteration, by tuning the learnt weights and iteratively retrieving more precise images.

II. Related Work

Most of the currently used databases by computer vision algorithms like Caltech 101[1], Tiny Images[3] and LabelMe[4] datasets rely on human effort to collect them. Recently, new trends in recognition databases emerged[5, 6, 7], like Web-based annotation and data collection tools. Many hours of work are dedicated to manually collecting

images, annotating them and labeling them which limits the ability of these databases to expand. An abundance of images is available through the Web and many computer vision algorithms[8, 9, 10, 11, 12] have been proposed to collect object datasets by querying image search engines. In [9], they argue that if the global web page in which the image is present is relevant to the query, the image would be also relevant and may be more informative than local textual content surrounding the image. An incremental Bayesian probabilistic model is learnt in [10], which represents the shape and appearance of a constellation of features belonging to the object. [12] also uses topic models but in different settings from the previous approach. For all these methods, the total number of retrieved images is restricted by the total number of results returned from the image search engine. Also, the results returned by the image search engine is much better in accuracy than those returned by web search engines. So, if we try to apply these methods to web search engines depending only on building good visual models to represent the objects, their performance will be highly degraded.

In recent years, attempts have been extended to overcome the limitations of the previous two approaches through ranking images returned from web search engines [13, 14, 15, 16]. All these techniques try to make use of the textual information available on the web pages containing the images to achieve a better dataset with better performance. [14] divides the re-ranking process into two stages, a collection stage and a selection stage. Their method is used primarily for annotating images while the main objective of our technique is determining the general category of the image. [15] builds a system to collect databases of animal images downloaded from Google text search. Their technique is also learnt for animal images only while we target a much larger and diverse set of object categories.

However, these batch methods do not benefit from incrementally improving the learnt model and increasing the re-ranking efficiency by applying iterative methods. Furthermore, many of the previous attempts in web image retrieval use semi-supervised learning techniques[13, 15] which require the user's interference in the learning process. Our system builds on the system proposed by Schroff et al. [16]. They employ the Web meta data to boost the performance of image dataset collection. The proposed system works in a completely automatic way.

III. Proposed Approach

A. Textual ranking of images

Images and the HTML documents containing them are downloaded. The method proposed by Frankel et al. [17] and extended by Schroff et al. [16] is followed in choosing indicative textual features of the image content. Seven features are used from the HTML document tags: contextR, context10, file name, file directory, image alternate text, image title and website title. Context10 represents the

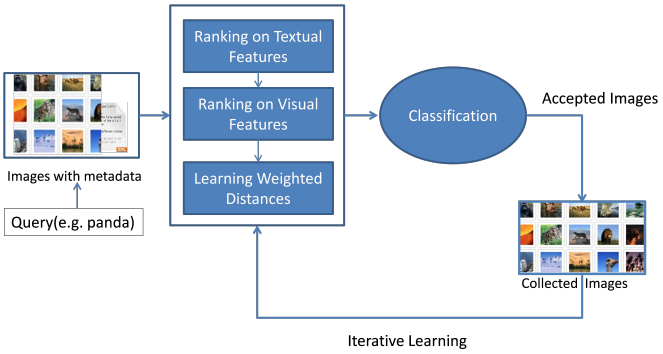

Figure 1. Overview of our system

ten words present before and after the image link in the document. ContextR includes the words on either side of the image between eleven and fifty words away from the image tag. The extracted textual attributes are used to code binary feature vectors which are used in ranking the retrieved images as described below.

A binary feature is defined for each textual field. If the field contains the query word, the binary feature has the value 1, else it has the value 0. The filtering is then applied using a Naive Bayes classifier, where images are ranked according to the posterior probability, $p(y = in - class|a)$, of the image being relevant to the query or not, where $y \in \{in - class, non - class\}$:

$$p(a|y) = P(a_1, ..., a_4|y) \prod_{i=5}^{7} P(a_i|y) \quad (1)$$

where $P(a_1, , a_4|y)$ is the joint probability of the first four textual features (contextR, context10, filedir, filename). The Bayesian model is learnt once and can be used with any new object class directly without further training. In the training phase, images need to be labeled in order to compute the probabilities needed for posterior estimation, $P(y|a) = P(a|y)P(y)/P(a)$, however this is done only once as the trained ranker is class independent.

B. Visual ranking of images

After the textual ranking phase, a set of images which are top-ranked according to the relevancy of their respective web pages to the query word are obtained. The next step is to re-rank these images based on their visual features to achieve higher precision.

In our proposed system, all images are resized to 150x150 pixels. Regions are detected using difference of Gaussians. Each image region is represented as a 64 dimensional SIFT descriptor. A visual vocabulary consisting of 512 visual words is learnt for the region detector using k-means clustering technique. Then, the descriptor of each region is assigned to the vocabulary to form the bag of words histograms that are used in the visual ranking with the SVM

classification technique. The aim of this stage is to improve the precision of the filtered images obtained from the textual ranking phase by filtering the retrieved set again based on the visual contents of images. Having a better ranked image set is vital to the next step of the proposed system which is learning weighted distances.

Visual ranking is carried out through the following procedure: n_+ positive training images are taken from the top ranked images of the text ranking phase. n_- negative training images are chosen at random from the whole downloaded image corpus from the web search engine. Following the approach in [16], we preferred to choose the negative training images randomly from the tens of thousands of downloaded images rather than take the low ranked results from the textual classifier output, as it is less likely to find relevant images among the set of all downloaded images than in the second case. We then train an SVM classifier, and re-rank the positive images based on the learnt SVM classification score. We chose the radial basis function (RBF) kernel on the normalized histograms of visual words. The values for the different parameters of the SVM classifier are determined by training using ten-fold cross validation.

C. Learning weighted distances

We learn a statistical model to retrieve images which are relevant to the highest ranked results from the visual classifier output. We achieve this through locally weighted learning which averages, or combines the training data through locally weighted training to bring together points which are considered neighbors of the query to produce the result.

We use a weighted version of the L1 distance to compare the visual features:

$$d(x,q) = \sum_{i=1}^{D} w_i |x_i - q_i| \tag{2}$$

where w_i is the weight for the i^{th} histogram bin of the downloaded image q_i and the top image from the visual ranking step x_i. If all w_i are chosen to be 1, this is the L1 distance. In this phase of the system, we are presented with n_+ positive training images which are chosen from the top ranked images from the visual ranking phase. This set is assumed to be noisy, however it is significantly more precise than the originally retrieved images due to the performed successive filtering. Another set of n_- training images is chosen randomly from all tens of thousands of downloaded images. Experiments were carried to choose the number of positive and negative training images for leaning the weighted distances and detailed in section 4.4.

The top ranked images from the visual ranking step are used as training images for the nearest neighbor system. We follow the approach proposed in [18] for learning weighted L2-distances to learn the weights w_i for the distance function. The criteria to learn the weights is to minimize the distance between positive images and maximize the distance between positive and negative images, This can be formulated in the following equation which is minimized with respect to w_i in the distance function d:

$$\sum_{x \in Q^+} \sum_{q_+ \in Q^+ \setminus \{x\}} \sum_{q_- \in Q^- \setminus \{x\}} \frac{d(x,q_+)}{d(x,q_-)} \tag{3}$$

and a similar term for all negative images has to be maximized [19]. Gradient descent is used to optimize equation 3. Accordingly, the learnt weights satisfy our need to minimize the distance between relevant images and the top ranked images and maximize the distance between relevant and irrelevant images, and thus is expected to improve retrieval accuracy.

D. Iterative retrieval of images

The main contribution of our work is that we incrementally collect visual object category datasets through iterative retrieval of images. The motivation draws from the observation that the precision of the retrieved images is highly affected by the amount of noise of the training images. Other batch approaches which collect images in one iteration try to overcome this limitation through utilizing user feedback in the training process. However, what we seek here is building a practical, scalable system that does not require any manual interference during its operation. Incremental collection of images for a given object category is achieved through the following procedure:

1) Classification of images: In our work, we follow the approach of [19] in using the combining classifiers technique which is extensively used in fusing the information from different cues [20] to rank the downloaded images based on their relevancy to the query. Each positive training image from the visual ranking step is used as basis for a nearest neighbor classifier with only one training sample. We then classify each image in the downloaded set as being relevant or irrelevant to the query using the above classifier. The probability that an image is relevant given the top ranked training image q^+ is $p_x(c = r|q_+)$ which is inversely related to the distance between both images as follows:

$$p_x(c = r|q_+) \quad \alpha \quad exp(-d(x,q_+)) \tag{4}$$

In a similar manner, irrelevant images to the query are related to the negative training images q_- such that the probability of an image being irrelevant $p(n|q_i)$ is:

$$p_x(n|q_-) \quad \alpha \quad exp(-d(x,q_-)) \tag{5}$$

The output of each classifier for all the positive and negative images are fused together using the sum rule [21] such that

the probability of an image being relevant to the query given both positive Q^+ and negative Q^- training images is:

$$p_x(r|(Q^+,Q^-)) = \frac{a}{|Q^+|}\sum_{q_+\in Q^+} p_x(n|q_+) + \frac{a}{|Q^-|}\sum_{q_-\in Q^-}(1-p_x(n|q_-)) \quad (6)$$

where α is used to change the impact of the negative and positive training images.

2) Tuning weights: Images with the highest relevance probabilities are considered relevant to the query and added to the collected dataset. Analogously, images whose computed relevance probabilities are low are discarded and considered irrelevant to the query. The newly collected images which are added to the object dataset, serve as new training data to update and improve the object model. The improved model can then be used to retrieve more images with better precision. This process continues until all downloaded images are considered or satisfactory results are achieved.

In tuning the learnt weights, only the newly added images are considered for the next step and not all the images in the currently collected dataset. This is motivated by the desire to build databases which are diverse and contain images which represent the considered object category in different shapes and conditions.

IV. Experiments

In this section, experiments are performed to assess the strengths and weaknesses of our proposed methodology. In the first section, we describe the image datasets used in our experiments. Then, we present the set of experiments carried to evaluate the performance of the algorithm. Finally, we assess the performance of our method against other related techniques and against Google Search.

A. Datasets

For evaluating our algorithm, we used two datasets:

"Web Queries" dataset: The "Web Queries" dataset contains 71,478 images and meta-data retrieved by 353 web queries. For 80% of queries there are more than 200 images It was proposed by Krapac et al. in CVPR 2010 [22]. The data set also includes a ground-truth relevance label for every image. Previous image re-ranking data sets [16, 23, 24] contain images for only a few classes, and in most cases provide image files without their corresponding meta-data.

For our experiments, we chose 23 web queries from the dataset. In choosing the queries, we picked words which represent concrete objects, celebrity names and abstract words to assess the performance of algorithm on different types of queries yielding different ranking performance on the search engine.

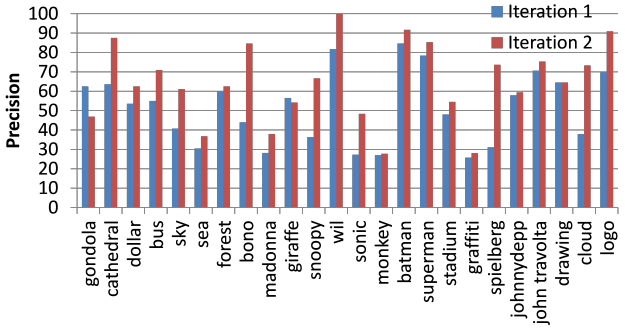

Figure 2. Precision of first iteration compared to second iteration at 15% recall on the "Web Queries" dataset.

The number of images in each category ranged from 197 to 277 images. The precision of the images in each category ranged from 19.11% to 71.8%.

Dataset by Fergus et al., ICCV 2005: The second dataset we use is the one used by Fergus, in [23]. It contains all images returned from Google Image Search (typically 500-700) for seven different query words and they are labeled into one of three different states: Good, Junk and Intermediate (some visual relation, e.g. a cartoon of the object). This dataset has been used to evaluate several previous query-specific ranking methods [24, 25, 26]. Since no textual meta-data is available, we rank the images using only visual features as detailed in section 4.5.

B. Classification performance

Figure 2 shows the precision of the first iteration results compared to the second iteration results at 15% recall on the "Web Queries" dataset. Our results prove that through incrementally collecting image databases, we improve the learning model and retrieve better images in each iteration. We have continued our experiments through the 3rd, 4th and 5th iterations. For some categories where there are relatively much more positive images than negative images (gondola, cathedral, dollar, madonna, batman, superman), performance keeps improving in each iteration. Other categories where there is a relatively small number of relevant images as compared to the number of negative images (sea, stadium, snoopy), the performance starts degrading after the 3rd iteration. This is due to the fact that a large percentage of the relevant images were already retrieved in the first three iterations. Using lower recall values (5%) in each iteration, showed that precision of the results returned in each iteration is improved over a larger number of iterations than when using higher recall values.

In Figure 3 we show to top-ranked 24 results returned by our algorithm in 4 of the categories. It is clear from the results that our method, in general, yields good ranking of the images. Visual inspection of the highly ranked "outlier" (non-class) images in the ranked lists gives some explanation

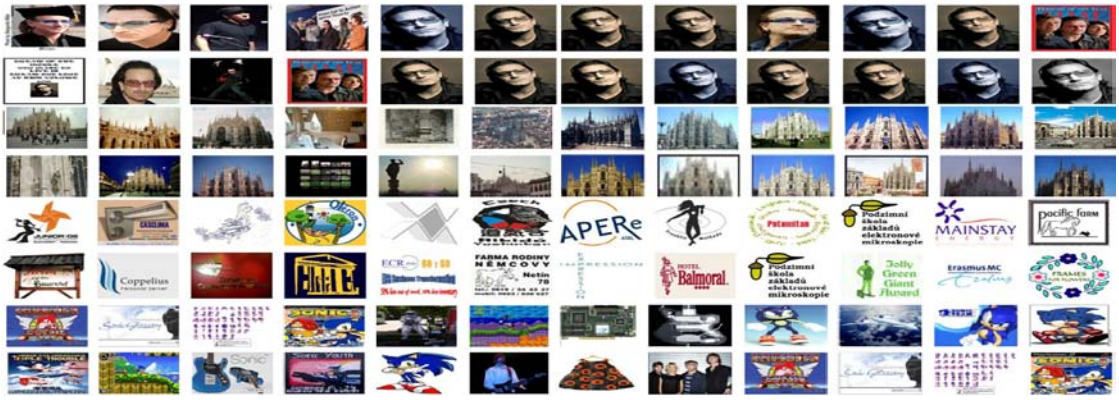

Figure 3. Top 24 images for "steven spielberg", "cathedral", "logo" and "sonic" categories in "Web Queries" dataset.

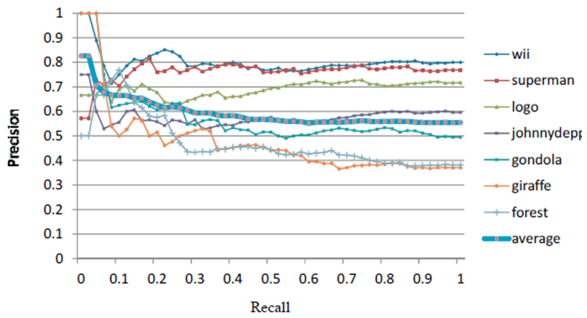

Figure 4. Precision vs. recall curve of 23 categories from the "Web Queries" dataset.

for the performances for the different categories displayed. Classes that perform well generally have outliers that are unrelated to each other. This gives better results when training the SVM classifier and optimizing the weights for the weighted L1 distance. In contrast for the classes that perform poorly, the outlier images are related and result from polysemy of the query word - for example for the sonic category, we have the cartoon character and the guitar brand name.

C. Precision and recall analysis

To analyze the performance of our algorithm over different recall values, we plotted the precision vs. recall curve estimated for each class in Figure 4. As can be seen, our reranker performs well on average and improves the precision up to a high recall level over the search engine's precision. The figure shows only the precision of 7 classes for visibility. However, the average of the precision of all the 23 classes is displayed. The curve shows that most classes perform relatively well at 15% recall.

D. Analysis of the effect of seed images on precision

Table 1 shows how the precision at 15% recall of the system is affected by the different number of positive/negative images for learning the weighted distances. It is noticed that, in general, increasing the number of negative images improves the detection performance. We also measured the precision when ranking the images using only textual and visual classification phases. The results displayed are those of the first iteration only of our system. From the results it is clear that learning weighted distances and using the learnt distance for classification greatly improves the performance. The average precision when using only the textual and visual classification phases is around 45%, while when adding the learning phase, the precision is averaged at around 51% for the worst choice of the number of seed images and reaches 54.3% on the best case which proves the efficiency of our method.

Table I
COMPARISON OF PRECISION AT 15% RECALL.

	text+ visual only	n+/n- =20/20	n+/n- =10/40	n+/n- =20/40	n+/n- =10/80	n+/n- 20/80
Average	45.17	51.99	52.37	51.35	54.29	51.31

E. Comparison with batch learning and search engine detection performance

Figure 5 shows the precision of the top 100 images on the chosen 23 categories from the "Web Queries" dataset. In all cases, n_+ = 50, n_- = 200 for the visual classifier and n_+ = 20, n_- = 80 for learning the weighted distances as this is the most stable setting. In all these categories, our system gives higher precision values than the batch method and Google Search results. We compared our method to Fergus [11] and Schroff [16] on the Fergus dataset. We ran our experiments to measure precision of the ranked results at 15% recall. This value was chosen to be able to compare our results to other approaches as in each of these approaches, precision at 15% recall is reported. Our method yielded better results than Google Search results and Schroff results in all categories as shown in table 2. Our results are not directly compared to the work of Fergus et al. in [11] for

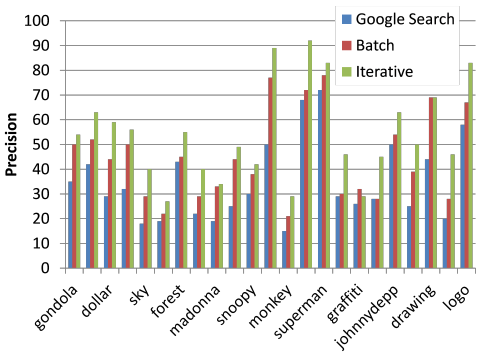

Figure 5. Precision of top 100 images on the 23 categories.

Table II
PRECISION AT 15% RECALL ON THE FERGUS DATASET.

	air-plane	cars rear	face	guitar	leo-pard	motor-bike	wrist watch
our	70.96	46.77	46.66	46.51	55.17	41.66	82.5
Schroff	54.54	37.17	23.86	28.98	50	35	71.7
Google	50	41	19	30	41	46	70

several reasons. The Fergus dataset does not provide any metadata, so our text + vision algorithm is compared based only on visual features which decreases its performance. Also, the dataset labels each image as (good, ok, junk) and [11] treats ok images as non-class and the algorithm is trained to distinguish between these three types of images, whereas our system is not tuned to distinguish good from ok images. Moreover, [11] starts their algorithm by training on a validation set of almost 100% accuracy whereas our system works in a completely autonomous way starting with a very noisy image set downloaded from the search engine. Accordingly, our system performs slightly worse than [11] when measured only on good images.

V. CONCLUSION AND FUTURE WORK

In this paper, we have presented an approach to collect a large number of images given a textual query to a web search engine in a completely automatic manner without any human intervention. Our system applies incremental learning to add relevant images to the query in each iteration. We eliminate the laborious human effort required to gather object datasets which makes them hard to expand. Also, our algorithm extracts images from text search engines which, unlike image search engines, provide a very large number of images along with useful meta data that can further improve precision. To further improve our system, a statistical method to determine the threshold that limits the number of retrieved images in each iteration can be developed. Also, incorporating the web search engine's ranking in the textual classification can improve its re-ranking performance.

REFERENCES

[1] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. Workshop on Generative-Model Based Vision, 2004.

[2] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(11):1475-1490, 2004.

[3] A. Torralba, W. T. Freeman, and R. Fergus. 80 million tiny images: a large dataset for non-parametric object and scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(11):1958-1970, 2008.

[4] B. C. Russell, A. Torralba, K. P. Murphy, W. T. Freeman. Labelme: A database and web-based tool for image annotation. Tech. Rep. MIT-CSAIL-TR-2005-056, Massachusetts Institute of Technology, 2005.

[5] J. Ponce, T. Berg, M. Everingham, D. Forsyth, M. Hebert et al. Dataset issues in object recognition. Toward Category-Level Object Recognition, pp. 29-48, Springer, 2006.

[6] L. von Ahn, and L. Dabbish. Labeling images with a computer game. In SIGCHI, 2004.

[7] L. von Ahn, R. Liu, and M. Blum. Peekaboom: A game for locating objects in images. In SIGCHI, 2006.

[8] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26(11):1475-1490, 2004.

[9] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang et al. Query by image and video content: The QBIC system. Computer, 28(9):23-32, 1995.

[10] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. Workshop on Generative-Model Based Vision, 2004.

[11] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from Google's image search. In ICCV, 2005.

[12] M. Fritz and B. Schiele. Decomposition, discovery and detection of visual categories using topic models. In CVPR, 2008.

[13] H. Feng and T. Chua. A bootstrapping approach to annotating large image collection. In SIGMM, 2003.

[14] K. Yanai and K. Barnard. Probabilistic web image gathering. In SIGMM, 2005.

[15] T. Berg and D. Forsyth. Animals on the web. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1-8, New York, NY, 2006.

[16] F. Schroff, A. Criminisi, and A. Zisserman. Harvesting image databases from the web. IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 754-766, 2011.

[17] C. Frankel, M. J. Swain, and V. Athitsos. Webseer: An image search engine for the world wide web. Technical Report TR-96-14, University of Chicago, 1996.

[18] R. Paredes and E. Vidal. Learning weighted metrics to minimize nearest neighbor classification error. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(7):1100-1110, 2006.

[19] T. Deselars, R. Paredes, E. Vidal, and H. Ney. Learning weighted distances for relevance feedback in image retrieval. In ICPR, 2008.

[20] H. Muller, W. Muller, S. Marchand-Maillet, and D. M. Squire. Strategies for positive and negative relevance feedback in image retrieval. In ICPR, 2000.

[21] J. Kittler. On combining classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(3):226-239, 1998.

[22] J. Krapac, M. Allan, J. Verbeek, and F. Jurie. Improving web image search results using query-relative classifiers. In CVPR, 2010.

[23] R. Fergus, P. Perona, and A. Zisserman. A visual category filter for Google images. In ECCV, 2004.

[24] J. Li, G.Wang, and L. Fei-Fei. OPTIMOL: automatic Object Picture collecTion via Incremental MOdel Learning. In CVPR, 2007.

[25] N. Vasconcelos. From pixels to semantic spaces: Advances in content-based image retrieval. Computer, 40(7):20-26, 2007.

[26] C.D. Ferreira, J.A. Santos, R. da S. Torres, M.A. Gonalves, R.C. Rezende, and Weiguo Fan. Relevance feedback based on genetic programming for image retrieval. Pattern Recognition Letters, 32(1):27 - 37, 2011.