Image Hunter at ImageCLEF 2012 Personal Photo Retrieval Task

Roberto Tronci[1,2], Luca Piras[1], Gabriele Murgia[2], and Giorgio Giacinto[1]

[1] DIEE - Department of Electric and Electronic Engineering
University of Cagliari, Italy
{roberto.tronci,luca.piras,giacinto}@diee.unica.it
[2] AmILAB - Laboratorio Intelligenza d'Ambiente, Sardegna Ricerche, Italy
{roberto.tronci,gabriele.murgia}@sardegnaricerche.it

Abstract. This paper presents the participation of the Pattern Recognition and Application Group (PRA Group) and the Ambient Intelligence (AmILAB) in the ImageCLEF 2012 Personal Photo Retrieval Pilot Task. This is a pilot task that aims to provide a test bed for QBE-based retrieval scenarios in the scope of personal information retrieval based on a collection of 5,555 personal images plus rich meta-data. For this challenge we used *Image Hunter*, a content based image retrieval tool with relevance feedback previously developed by ourselves. The results show that we obtained good results by taking into account that we used only visual data, moreover we were the only one that used relevance feedback.

Keywords: photo retrieval, content based image retrieval, relevance feedback, SVM

1 Introduction

The personal photo retrieval is a pilot task introduced in the ImageCLEF 2012 competition. This pilot task provides a test-bed for query by example (QBE) image retrieval scenarios in the scope of personal information retrieval. In fact, instead of using images downloaded from Flickr or other similar web resources, the dataset proposed reflects an amalgamated personal image collection that has been taken by 19 photographers. The aim of this pilot task is to create an image retrieval scenario where a normal person (i.e., not expert in image retrieval tasks) searches in its personal photo collections some "relevant" images, i.e. the search for similar images or images depicting a similar event, e.g. a rock concert. This pilot task is divided into two tasks: retrieval of visual concepts and retrieval of events. A detailed overview of the dataset and the task can be found in [16].

We took part only to the *Task 1* of this competition. In this task we have a set of visual concepts with five QBE associated to be used in the retrieval process to retrieve relevant images. To perform the task we used Image Hunter, a content based image retrieval tool with relevance feedback previously developed at the AmILAB [15].

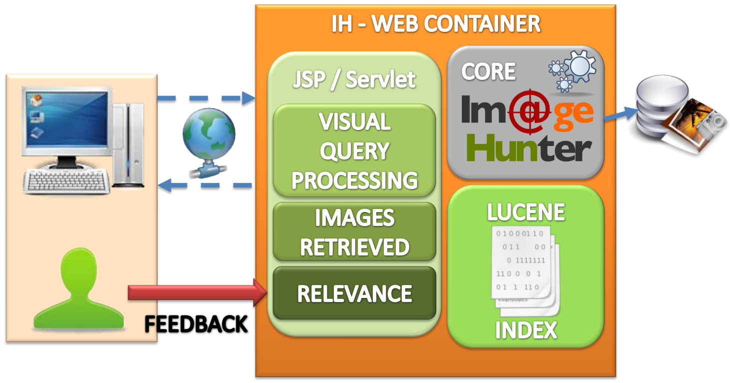

Fig. 1. Image Hunter: Web Application Architecture

2 Image Hunter: a brief description

With the aim of building a practical application to show the potentialities of Content Based Image Retrieval tools with Relevance Feedback, we developed *Image Hunter*. *Image Hunter* is a prototype that shows the capabilities of an Image Retrieval engine where the search is started by an image provided by the user, and the system returns the most visually similar images. The system is enriched by Relevance Feedback capabilities that let the user specify which results match the desired concepts through an easy user interface.

2.1 Architectural description

This tool is entirely written in JAVA, so that the tool is machine independent. For its development, we partially took inspiration from the LIRE library [11] (that is just a feature extraction library). In addition, we chose Apache Lucene for building the index of the extracted data.

The main core of *Image Hunter* is a full independent module, thus allowing the development of a personalized user interface. A schema of the whole system can be seen in Figure 1.

The core of the system is subdivided into three main parts:

– Indexing and Lucene interface for data storing;
– Feature extraction interface;
– Image Retrieval and Relevance Feedback.

The *Indexing* part has the role of extracting the visual features and other informations from the images. The visual features and other descriptors of the images are then stored in a particular structure defined inside *Image Hunter*. The tool can index different types of image formats and it can be built in an incremental way. Lucene turned out to be well suited for the storage needs of

Image Hunter. The core of Lucene's logical architecture is a series of document containing text fields, where we have associated different features (fields) to each image (document).

As we said before, for the feature extraction we took inspiration from the LIRE library that it is used as an external feature extraction library. We expanded and modified its functionalities by implementing or reimplementing in *Image Hunter* some extractors. The *Feature extraction interface* allows to extract different visual features based on different characteristics: color, texture and shape. They are:

- *Scalable Color* [3], a color histogram extracted from the HSV color space;
- *Color Layout* [3], that characterizes the spatial distribution of colors;
- *RGB-Histogram* and *HSV-Histogram* [11], based on RGB and HSV components of the image respectively;
- *Fuzzy Color* [11], that considers the color similarity between the pixel of the image;
- *JPEG Histogram* [11], a JPEG coefficient histogram;
- *Edge Histogram* [3], that captures the spatial distribution of edges;
- *Tamura* [12], that captures different characteristic of the images like coarseness, contrast, directionality, regularity, roughness;
- *Gabor*[7] that allows the edge detection;
- *CEDD* (Color and Edge Directivity Descriptor) [4];
- *FCTH* (Fuzzy Color and Texture Histogram) [5].

One of Image Hunter's greatest strengths is its flexibility: in fact, its structure was built in a way that it is possible to add any other image descriptor. The choice of the above mentioned set is due to the "real time" nature of the system with large database. In fact even if some local features such as SIFT or SURF could improve the retrieval performance for some particular kind of searches, on the other hand they are more time expensive in the evaluation of the similarity between images.

The core adopts three relevance feedback techniques [15]. Two of them are based on the nearest-neighbor paradigm (NN), while one of them is based on Support Vector Machines (SVM). The use of the nearest-neighbor paradigm has been driven by its use in a number of different pattern recognition fields, where it is difficult to produce a high-level generalization of a class of objects, but where neighborhood information is available [1, 8]. In particular, nearest-neighbor approaches have proven to be effective in outliers detection, and one-class classification tasks [2, 13]. Support Vector Machines are used because they are one of the most popular learning algorithm when dealing with high dimensional spaces as in the case of CBIR [6, 14].

The user interface is structured to provide just the functionalities that are strictly related with the user interaction (e.g., the list of relevant images found by the user).

Image Hunter employs a web-based interface that can be viewed and experienced at the address `http://prag.diee.unica.it/amilab/WIH`. This version is a web application built for the Apache Tomcat web container by using a mixture of JSP and java Servlet. The graphic interface is based on the jQuery framework, and has been tested for the Mozilla Firefox and Google Chrome browsers. The *Image Hunter* homepage let the user choose the picture from which starting the search. The picture can be chosen either within those of the proposed galleries or among the images from the user hard disk. In order to make intuitive and easy the features offered by the application, the graphical interface has been designed relying on the Drag and Drop approach. From the result page, the user can drag the images that her deems relevant to her search in a special box-cart, and then submit the feedback. Then the feedback is processed by the system, and a new set of images is proposed to the user. The user can iterate the feedback process as many times he/she wants. Figure 2 summarizes the typical user interaction within *Image Hunter*.

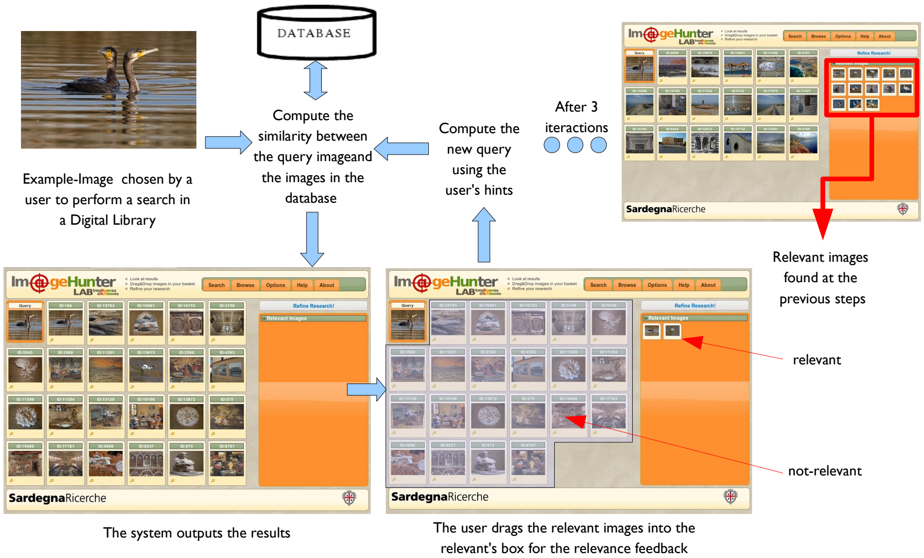

Fig. 2. Example of a typical user interaction with *Image Hunter*

2.2 Relevance Feedback techniques implemented in Image Hunter

In this section we briefly describe the two relevance feedback techniques implemented in the core that we have used in this competition. The use of the nearest-neighbor paradigm is motivated by its use in a number of different pattern recognition fields, where it is difficult to produce a high-level generalization of a class of objects, but where neighborhood information is available [1, 8]. In particular, nearest-neighbor approaches have proven to be effective in outliers detection, and one-class classification tasks [2, 13]. Support Vector Machines are used because they are one of the most popular learning algorithm when dealing with high dimensional spaces as in CBIR [6, 14].

k-NN Relevance Feedback In this work we resort to a technique proposed by some of the authors in [9] where a score is assigned to each image of a database according to its distance from the nearest image belonging to the target class, and the distance from the nearest image belonging to a different class. This score is further combined to a score related to the distance of the image from the region of relevant images. The combined score is computed as follows:

$$rel(\mathbf{I}) = \left(\frac{n/t}{1+n/t}\right) \cdot rel_{BQS}(\mathbf{I}) + \left(\frac{1}{1+n/t}\right) \cdot rel_{NN}(\mathbf{I}) \tag{1}$$

where n and t are the number of non-relevant images and the whole number of images retrieved after the latter iteration, respectively. The two terms rel_{NN} and rel_{BQS} are computed as follows:

$$rel_{NN}(\mathbf{I}) = \frac{\|\mathbf{I} - NN^{nr}(\mathbf{I})\|}{\|\mathbf{I} - NN^{r}(\mathbf{I})\| + \|\mathbf{I} - NN^{nr}(\mathbf{I})\|} \tag{2}$$

where $NN^r(\mathbf{I})$ and $NN^{nr}(\mathbf{I})$ denote the relevant and the non relevant Nearest Neighbor of $\mathbf{I}$, respectively, and $\|\cdot\|$ is the metric defined in the feature space at hand,

$$rel_{BQS}(\mathbf{I}) = \frac{1 - e^{1 - d_{BQS}(\mathbf{I}) \big/ \max_i d_{BQS}(\mathbf{I}_i)}}{1-e} \tag{3}$$

where e is the *Euler's number*, i is the index of all images in the database and d_{BQS} is the distance of image $\mathbf{I}$ from a reference vector computed according to the Bayes decision theory (Bayes Query Shifting, BQS) [10]. The aim of BQS approach is to "move" the query along the visual spaces by taking into account images marked as relevant and not-relevant within the visual concept searched to look for new images. The BQS query is computed as follows:

$$Q_{BQS} = \mathbf{m}_R + \frac{\sigma}{\|\mathbf{m}_R - \mathbf{m}_N\|} \cdot \left(1 - \frac{k_R - k_N}{\max\{k_R, k_N\}}\right) \cdot (\mathbf{m}_R - \mathbf{m}_N) \tag{4}$$

where $\mathbf{m}_R$ and $\mathbf{m}_N$ are the mean vectors of relevant and not-relevant images respectively, σ is the standard deviation of the images belonging to the neighborhood of the original query and k_R and k_N are the number of relevant and not-relevant images, respectively.

If we are using F feature spaces, we have different scores $rel(\mathbf{I})$ for each f feature space. Thus the following combination is performed to obtain a "single" score:

$$rel(\mathbf{I}) = \sum_{f=1}^{F} w_f \cdot rel^f(\mathbf{I}) \tag{5}$$

where the w_f is the weight associated to the f-space.

$$w_f = \frac{\sum_{i \in R} d^f_{min}(\mathbf{I}_i, R)}{\sum_{i \in R} d^f_{min}(\mathbf{I}_i, R) + \sum_{i \in R} d^f_{min}(\mathbf{I}_i, N)} \tag{6}$$

SVM based Relevance Feedback Support Vector Machines are used to find a decision boundary in each feature space $f \in F$. The SVM is very handy for this kind of task because, in the case of image retrieval, we deal with high dimensional feature spaces and two "classes" (i.e. relevant and not-relevant). For each feature space f, a SVM is trained using the feedback given by the user. The results of the SVMs in terms of distances from the hyperplane of separation are then combined into to a relevance score through the Mean rule as follows

$$rel_{SVM}(\mathbf{I}) = \frac{1}{F} \sum_{f=1}^{F} rel^f_{SVM}(\mathbf{I}) \tag{7}$$

3 *Image Hunter* at ImageCLEF

For the participation at the ImageCLEF competition we mainly used *Image Hunter* as it is. This means that as visual features we have used only those listed in the previous section, that are partially part of those provide for the competition [16].

We took part only to the task 1 (retrieval of visual concepts). In the task different visual concepts are provided and to each concept five QBE are associated. However *Image Hunter* is designed to trigger the image retrieval process only with one QBE. Thus, instead of performing five different runs for each concept starting with a different QBE and averaging the results, we slightly modified *Image Hunter* at the first interaction step (i.e., the first content based image retrieval before the relevance feedback steps) to take into account the five QBE. In this case we adopted two different techniques: the "mean" of the QBEs and a mixed multi query approach. In the first case, the query used to trigger *Image Hunter* is the mean vector of the five QBE in each visual feature space:

$$Q_{mean} = \mathbf{m}_{QBE} \tag{8}$$

that is the case of BQS presented in Equation (4) when there are only relevant images. In the second case we performed one content based image retrieval for each QBE, and then we mixed the results by assigning to each retrieved image the minimum distance from the five query images. After this first automatic interaction, the tool is ready to interact with real users by using one of the relevance feedback methodologies above.

The interaction with real users was performed by 10 different people. The only constraint that we gave to each person was in the minimum number of interactions (i.e., 3), but letting them free to choose when stop the retrieval process. In Figure 3 some snapshots of the tool in action are reported.

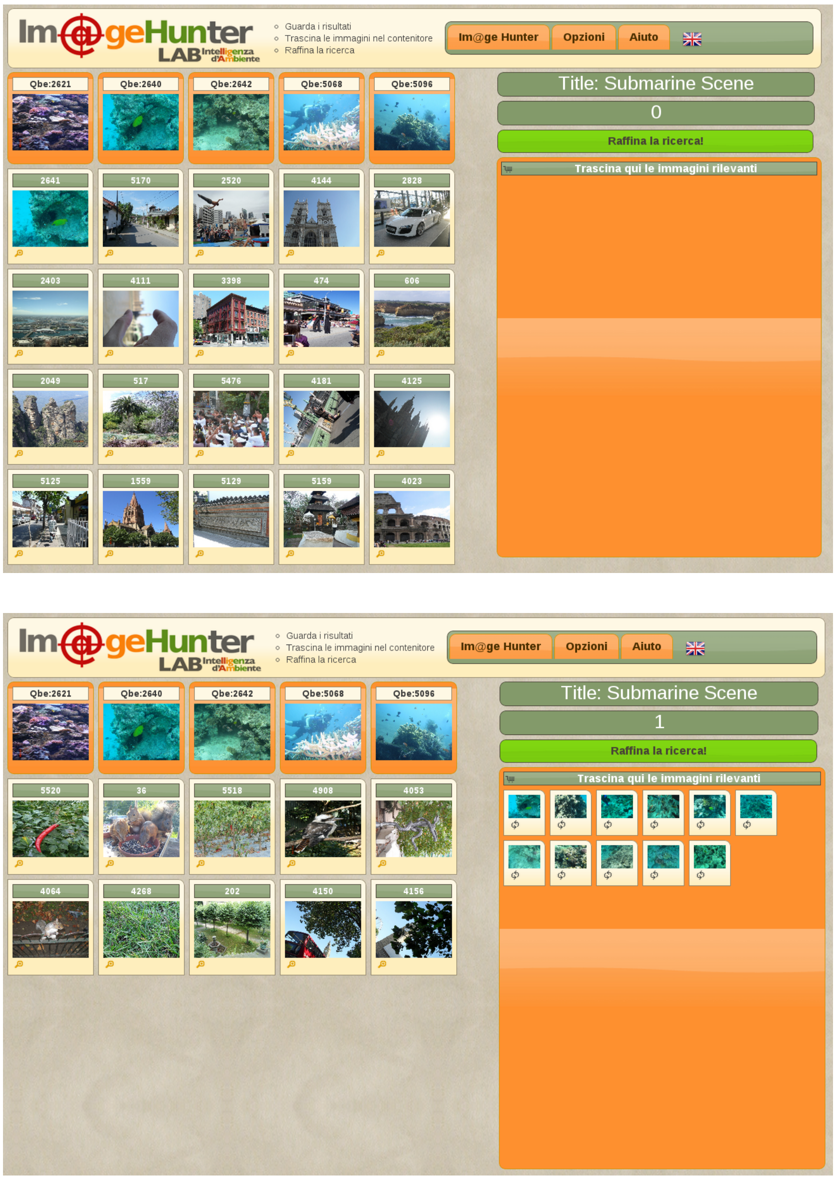

Fig. 3. *Image Hunter* in action with the Personal Photo Retrieval dataset

4 Results and Discussion

We submitted four runs to the competition combining the two methodologies for the first step and the relevance feedback methods used: Q_{mean} + kNN (*Run11* in tables), multi query + kNN (*Run12*), Q_{mean} + SVM (*Run31*), multi query + SVM (*Run32*). Unfortunately, *Run31* was not evaluated due some duplicates in the final file. In each run 100 retrieved documents are reported. In our case they are sorted with the relevance score obtained at the last step, these means that in some cases the first entry is not the first relevant image retrieved. This fact, derives from the methodologies used for relevance feedback. In particular the kNN, by means of the BQS, "moves" the query in the visual spaces, and with respect to the last BQS query the first relevant images retrieved could be far than other images.

In Table 1 the methodologies used by the competitors in the competition are presented. Among all the competitors we were the only one that used only the visual features for all the runs, and the only one that used a real user interaction relevance feedback.

Table 1. Competitors' runs with information on the methodologies used.

Group	Run ID	Run Type	Relevance Feedback	Retrieval Type
KIDS	IBMA0	Automatic	NOFB	IMGMET
KIDS	OBOA0	Automatic	NOFB	MET
KIDS	IOMA0	Automatic	NOFB	IMGMET
KIDS	OBMA0	Automatic	NOFB	MET
REGIM	run4	-	-	-
REGIM	run2	-	-	-
REGIM	run1	-	-	-
REGIM	run5	-	-	-
REGIM	run3	-	-	-
Image Hunter - Lpiras	Run12	User Feedback	BINARY	IMG
KIDS	IOOA4	Automatic	NOFB	IMG
Image Hunter - Lpiras	Run11	User Feedback	BINARY	IMG
Image Hunter - Lpiras	Run32	User Feedback	BINARY	IMG

In Table 2 the precision after N docs retrieved is reported. REGIM obtained the bet results until 20 retrieved documents, after the best results are obtained by the run IBMA0 of KIDS. Instead, our results are generally good if we take into account the kNN retrieval (i.e., *Run11* and *Run12*), and are mostly better than the only other method based only to visual features.

In Table 3 normalized discounted cumulative gain and mean average precision after N docs retrieved is reported. For these measures our run *Run11* obtained better results than those from REGIM (that were the best in the previous table), and we can claim that it is the best second run if we look at the overall of the measures presented in this table.

Table 2. Performance in terms of precision after N docs retrieved. In bold the best results, in italics the second best results per measure.

Group	Run ID	P_5	P_{10}	P_{15}	P_{20}	P_{30}	P_{100}
KIDS	IBMA0	*0,8333*	0,7833	0,7222	0,6896	**0,6347**	**0,4379**
KIDS	OBOA0	0,8000	0,7292	0,6667	0,6354	0,6083	*0,4117*
KIDS	IOMA0	0,7667	0,6583	0,6222	0,6104	0,5639	0,3925
KIDS	OBMA0	0,6500	0,6500	0,6083	0,5771	0,5611	0,3925
REGIM	run4	**0,9000**	0,8375	**0,7917**	**0,7333**	*0,6292*	0,3992
REGIM	run2	**0,9000**	*0,8417*	**0,7917**	*0,7292*	0,6278	0,3975
REGIM	run1	**0,9000**	*0,8417*	*0,7889*	*0,7292*	0,6278	0,3967
REGIM	run5	**0,9000**	**0,8458**	*0,7889*	*0,7292*	0,6278	0,3971
REGIM	run3	**0,9000**	**0,8458**	*0,7889*	*0,7292*	0,6278	0,3975
Image Hunter - Lpiras	Run12	0,7917	0,7667	0,7361	0,6938	0,6083	0,3417
KIDS	IOOA4	0,6750	0,6125	0,5778	0,5354	0,4486	0,3054
Image Hunter - Lpiras	Run11	0,8000	0,7083	0,6222	0,5646	0,4903	0,2825
Image Hunter - Lpiras	Run32	0,6583	0,5667	0,4667	0,3958	0,2972	0,1425

Table 3. Performance in terms of normalized discounted cumulative gain (ndcg) and mean average precision (map) after N docs retrieved. In bold the best results, in italics the second best results per measure.

Group	Run ID	$ndcg_5$	$ndcg_{10}$	$ndcg_{15}$	$ndcg_{20}$	$ndcg_{30}$	$ndcg_{100}$	map_{30}	map_{100}
KIDS	IBMA0	**0,6405**	**0,6017**	**0,5658**	**0,5459**	**0,5213**	**0,4436**	**0,1026**	**0,1777**
KIDS	OBOA0	0,5858	0,5348	0,5028	0,4836	0,4728	*0,4144*	0,0952	*0,1589*
KIDS	IOMA0	0,5800	0,5184	0,4951	0,4872	0,4615	0,3979	0,0906	0,1558
KIDS	OBMA0	0,4073	0,4268	0,4123	0,4066	0,4046	0,3717	0,0854	0,1518
REGIM	run4	0,4896	0,4703	0,4667	0,4563	0,4274	0,3572	0,0810	0,1224
REGIM	run2	0,4926	0,4722	0,4687	0,4561	0,4271	0,3566	0,0811	0,1224
REGIM	run1	0,4908	0,4730	0,4680	0,4560	0,4271	0,3561	0,0811	0,1222
REGIM	run5	0,4899	0,4742	0,4681	0,4551	0,4264	0,3563	0,0811	0,1222
REGIM	run3	0,4899	0,4742	0,4681	0,4551	0,4264	0,3564	0,0811	0,1222
IH - Lpiras	Run12	0,6122	*0,5777*	*0,5656*	*0,5457*	*0,5017*	0,3700	*0,0991*	0,1216
KIDS	IOOA4	0,5701	0,5062	0,4798	0,4545	0,4016	0,3303	0,0632	0,0930
IH - Lpiras	Run11	*0,6265*	0,5642	0,5165	0,4835	0,4342	0,3087	0,0692	0,0852
IH - Lpiras	Run32	0,4857	0,4404	0,3917	0,3466	0,2853	0,1795	0,0319	0,0363

5 Conclusions

In our participation to the personal photo retrieval pilot task of ImageCLEF, we tested the efficiency of our previous tool Image Hunter. As our intention was to benchmark this tool on this task, we did not make any modification. The only modification made was about the use of five QBE instead of one. The results obtained are encouraging, especially if we think that the results were obtained using only visual features. Future improvements of the tool will focus on the use of combination of the meta-data features with the visual ones, and on the improvement of our ranking system that is not actually designed for scientific evaluation.

References

1. Aha, D.W., Kibler, D., Albert, M.K.: Instance-based learning algorithms. Machine Learning 6(1), 37–66 (1991)
2. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: LOF: Identifying density-based local outliers. In: Chen, W., Naughton, J.F., Bernstein, P.A. (eds.) SIGMOD Conference. pp. 93–104. ACM (2000)
3. Chang, S.F., Sikora, T., Puri, A.: Overview of the mpeg-7 standard. IEEE Trans. Circuits Syst. Video Techn.
4. Chatzichristofis, S.A., Boutalis, Y.S.: Cedd: Color and edge directivity descriptor: A compact descriptor for image indexing and retrieval. In: Gasteratos, A., Vincze, M., Tsotsos, J.K. (eds.) ICVS. Lecture Notes in Computer Science, vol. 5008, pp. 312–322. Springer (2008)
5. Chatzichristofis, S.A., Boutalis, Y.S.: Fcth: Fuzzy color and texture histogram - a low level feature for accurate image retrieval. In: Image Analysis for Multimedia Interactive Services. pp. 191–196. IEEE (2008)
6. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press (2000)
7. Deselaers, T., Keysers, D., Ney, H.: Features for image retrieval: an experimental comparison. Inf. Retr. 11(2), 77–107 (2008)
8. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. John Wiley and Sons, Inc., New York (2001)
9. Giacinto, G.: A nearest-neighbor approach to relevance feedback in content based image retrieval. In: CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval. pp. 456–463. ACM, New York, NY, USA (2007)
10. Giacinto, G., Roli, F.: Bayesian relevance feedback for content-based image retrieval. Pattern Recognition 37(7), 1499–1508 (2004)
11. Lux, M., Chatzichristofis, S.A.: Lire: lucene image retrieval: an extensible java cbir library. In: MM '08: Proceeding of the 16th ACM international conference on Multimedia. pp. 1085–1088. ACM, New York, NY, USA (2008)
12. Tamura, H., Mori, S., Yamawaki, T.: Textural features corresponding to visual perception. IEEE Trans. Systems, Man and Cybernetics 8(6), 460–473 (June 1978)
13. Tax, D.M.: One-class classification. Ph.D. thesis, Delft University of Technology, Delft, The Netherlands (June 2001)
14. Tong, S., Chang, E.: Support vector machine active learning for image retrieval. In: Proc. of the 9th ACM Intl Conf. on Multimedia. pp. 107–118 (2001)

15. Tronci, R., Murgia, G., Pili, M., Piras, L., Giacinto, G.: Imagehunter: a novel tool for relevance feedback in content based image retrieval. In: 5th Int. Workshop on New Challenges in Distributed Information Filtering and Retrieval - DART 2011. Ceur (2011)
16. Zellhöfer, D.: Overview of the personal photo retrieval pilot task at imageclef 2012. Tech. rep., CLEF 2012 working notes, Rome, Italy, Rome, Italy (2012)