Combining image retrieval, metadata processing and naive Bayes classification at Plant Identification 2013

UAIC: Faculty of Computer Science, “Alexandru Ioan Cuza” University, Romania

Cristina Şerban, Alexandra Siriţeanu, Claudia Gheorghiu,
Adrian Iftene, Lenuţa Alboaie, Mihaela Breabăn

{cristina.serban, alexandra.siriteanu, claudia.gheorghiu,
adiftene, adria, pmihaela}@info.uaic.ro

Abstract. This paper aims to combine intuition and practical experience in the context of ImageCLEF 2013 Plant Identification task. We propose a flexible, modular system which allows us to analyse and combine the results after applying methods such as image retrieval using LIRe, metadata clustering and naive Bayes classification. Although the training collection is quite extensive, covering a large number of species, in order to obtain accurate results with our photo annotation algorithm we enriched our system with new images from a reliable source. As a result, we performed four runs with different configurations, and the best run was ranked 5th out of a total of 12 group participants.

Keywords: ImageCLEF, plant identification, image retrieval, metadata processing, naive Bayes

1 Introduction

ImageCLEF 2013[1] Plant Identification task[2] is a competition that aims to improve the state-of-the-art of the Computer Vision field by addressing the problem of image retrieval in the context of botanical data [1]. The task is similar to those from 2011 and 2012, but there are two main differences: (1) the organizers offered the participants more species (around 250, aiming to better cover the entire flora from a given region) and (2) they preferred multi-view plant retrieval instead of leaf-based retrieval (the training and test data contain different views of the same plant: entire, flower, fruit, leaf and stem).

In 2013, the organizers offered a collection based on 250 herb and tree species, most of them from the French area. The training data contains 20985 pictures and the test data contains 5092 pictures, divided in two categories: SheetAsBackground (42%) and NaturalBackground (58%). In **Fig. 1** we can see some examples from the pictures used in 2013 in the Plant Identification task.

[1] ImageCLEF2013: http://www.imageclef.org/2013

[2] Plant Identification Task: http://www.imageclef.org/2013/plant

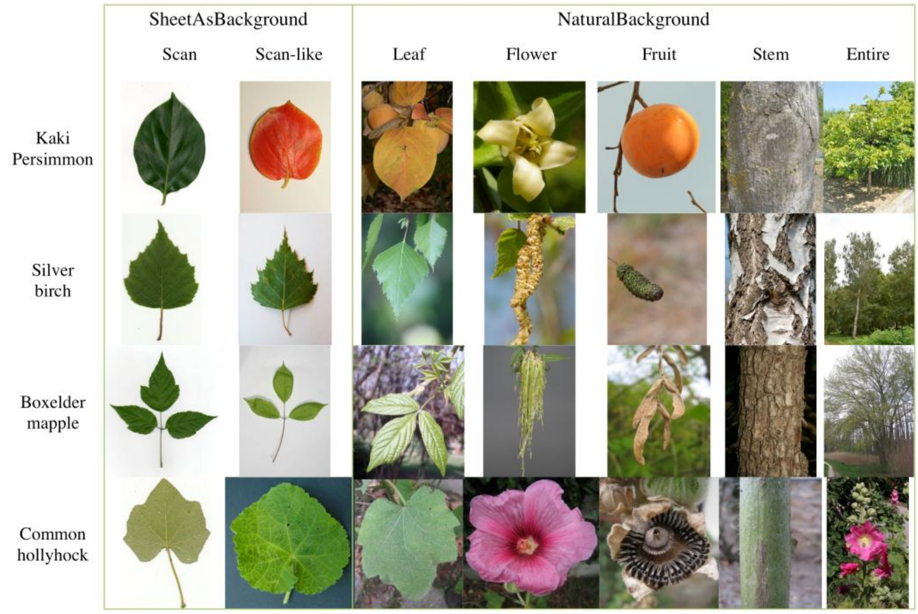

Fig. 1. Examples of train images (image was taken from http://www.imageclef.org/2013/plant)

The rest of the article is organized as follows: Section 2 presents the visual and textual features we extracted to describe the images, Section 3 covers the pre-processing and enrichment of the data, Section 4 describes the classification process, Section 5 details our submitted runs and Section 6 outlines our conclusions.

2 Visual and textual features

2.1 Visual features

The system we used for extracting and using image features is LIRe (Lucene Image Retrieval) [2]. LIRe is an efficient and light weight open source library built on top of Lucene, which provides a simple way for performing content based image retrieval. It creates a Lucene index of images and offers the necessary mechanism for searching this index and also for browsing and filtering the results. Being based on a light weight embedded text search engine, it is easy to integrate in applications without having to rely on a database server. Furthermore, LIRe scales well up to millions of images with hash based approximate indexing.

LIRe is built on top of the open source text search engine Lucene[3]. As in text retrieval, images have to be indexed in order to be retrieved later on. *Documents* consisting of *fields*, having a *name* and a *value*, are organized in the form of an index that is typically stored in the file system.

[3] Lucene is hosted at http://lucene.apache.org

Some of the features that LIRe can extract from raster images are:

- **Color histograms in RGB and HSV color space**;
- **MPEG-7 descriptors scalable color, color layout and edge histogram**. Formally known as Multimedia Content Description Interface, MPEG-7 includes standardized tools (descriptors, description schemes, and language) that enable structural, detailed descriptions of audio–visual information at different granularity levels (region, image, video segment, collection) and in different areas (content description, management, organization, navigation, and user interaction). [3];
- **The Tamura texture features coarseness, contrast and directionality**. In [4], the authors approximated in computational form six basic textural features, namely, *coarseness*, *contrast*, *directionality*, line-likeness, regularity, and roughness. The first three of these features are available in LIRe;
- **Color and edge directivity descriptor (CEDD)**. This feature incorporates color and texture information in a histogram and is limited to 54 bytes per image [5];
- **Fuzzy color and texture histogram (FCTH)**. This feature also combines, in one histogram, color and texture information. It is the result of combining three fuzzy systems and is limited to 72 bytes per image [6];
- **Joint Composite Descriptor (JCD)**. One of the Compact Composite Descriptors available for visual description, JCD was designed for natural color images and results from the combination of two compact composite descriptor, CEDD and FCTH [7];
- **Auto color correlation feature**. This feature distills the spatial correlation of colors, and is both effective and inexpensive for content-based image retrieval. [8].

2.2 Metadata

The ImageCLEF Pl@ntView dataset consists of a total of 20985 images from 250 herb and tree species in the French area. The dataset is subdivided into two main categories based on the acquisition methodology used: SheetAsBackground (42%) and NaturalBackground (58%), as stated in **Table 1Error! Reference source not found.**.

Table 1. Statistics of images in the dataset

Type	SheetAsBackground	NaturalBackground	Total
Training	9781	11204	20985
Test	1250	3842	5092
Total	11031	15046	26077

The training data is further divided based on the content as follows: 3522 "flower", 2080 "leaf", 1455 "entire", 1387 "fruit", 1337 "stem". The test dataset has 1233 "flower", 790 "leaf", 694 "entire", 520 "fruit", 605 "stem" items.

Each image has a corresponding .xml file which describes the metadata of the plant:

- **IndividualPlantId**: the plant id, which may have several associated pictures; it is an additional tag used to identify one single individual plant observed by one same person the same day with the same device with the same lightening conditions;
- **Date**: date of shot or scan;
- **Locality**: locality name (a district or a country division or a region);
- **GPSLocality**: GPS coordinates of the observation;
- **Author and Organisation**: name and organisation of the author;
- **Type**: *SheetAsBackground* (pictures of leaves in front of a uniform background) or *NaturalBackground* (photographs of different views on different subparts of a plant into the wild);
- **Content**:
  - *Leaf* (16%): photograph of one leaf or more directly on the plant or near the plant on the floor or other non-uniform background;
  - *Flower* (18%): photograph of one flower or a group of flowers (inflorescence) directly on the plant;
  - *Fruit* (8%): photograph of one fruit or a group of fruits (infructescence) directly on the plant;
  - *Stem* (8%): photograph of the stalk, or a stipe, or the bark of the trunk or a main branch of the plant;
  - *Entire* (8%): photograph of the entire plant, from the floor to the top.
- **ClassId**: the class label that must be used as ground-truth. It is a non-official short name of the taxon used for easily designating a taxon (most of the time the species and genus names without the name of the author of the taxon);
- **Taxon**: full taxon name (Regnum, Class, Subclass, Superorder, Order, Family, Genus, Species);
- **VernacularNames**: English common name;
- **Year**: ImageCLEF2011 or ImageCLEF2012;
- **IndividualPlantId2012**: the plant id used in 2012;
- **ImageID2012**: the image id.jpg used in 2012.

3 Pre-processing and enrichment

3.1 Training data and Wikimedia

In order to expand our image collection we searched for a public plant database. Having a larger training dataset to rely on would increase the performance of our photo annotation algorithm. Our choice was Wikimedia Commons[4], one of the various projects of the Wikimedia Foundation, which hosts a wide variety of photos, including plants. Wikimedia Commons provides a human annotated image repository,

[4] Wikimedia Commons, http://commons.wikimedia.org/wiki/Main_Page

uploaded and maintained by a collaborative community of users, thus ensuring a reliable content.

For downloading the images, we used the Wikimedia public API[5], by issuing a query for each plant ClassId found in the training dataset. To retrieve relevant results, we used several key API parameters. Each request had the *search* parameter set to the current ClassId value, the *fulltext* parameter set to “Search” and the *profile* parameter to “images”.

To determine the content of each new image, particular filters were used, as follows:

- *Flower* - flowers, flora, floret, inflorescence, bloom, blossom;
- *Stem* - bark, wood, trunk, branch substrings;
- *Leaf* - leaf, folie, foliage;
- *Fruit* - fruit, fruits, seed, seeds, fructus;

For every downloaded image, we created an .xml file having the PlantCLEF2013 file format. Thus, we managed to expand our image collection with 507 files.

3.2 Extraction of visual features and indexing

Using LIRe, we extracted the following features from the train images: color histogram, edge histogram, Tamura features [4], CEDD [5], FCTH [6], JCD [7]. Then we created an index in which we added a document for each train image, containing the previously mentioned features.

Using this index, we tried to see which feature is better for searches involving each type of plant (entire, flower, fruit, leaf or stem). We chose some images from the train set (for which we already knew the ClassId) and used them as queries to search the index. By comparing the results, we concluded that the JCD (Joint Composite Descriptor) gave us the most satisfying results in all cases.

The *Joint Composite Descriptor* [7] is one of the Compact Composite Descriptors available for visual description. It was designed for natural color images and results from the combination of two compact composite descriptors, namely the Color and Edge Directivity Descriptor [5] (CEDD) and the Fuzzy Color and Texture Histogram [6] (FCTH). JCD It is made up of 7 texture areas, with each area made up of 24 subregions that correspond to color areas. The texture areas are as follows: JCD(0) Linear Area, JCD(1) Horizontal Activation, JCD(2) 45 Degrees Activation, JCD(3) Vertical Activation, JCD(4) 135 Degrees Activation, JCD(5) Horizontal and Vertical Activation and JCD(6) Non Directional Activation. For more information on this visual feature, see [7].

[5] Wikipedia API, http://en.wikipedia.org/w/api.php

4 Classification

4.1 Using LIRe

Using the index described in Section 3.2, we searched for all the images in the test set. For each test image, we extracted from the index only those documents that contained images of the same type (entire, flower, fruit, leaf or stem) and then filtered the results by using the JCD feature [7]. Only the first 500 results were retained.

For the filtered results, LIRe computes a score from 0 to 1 (1 being the best match), based on the feature that is used to compare the query document to the documents in the index (in our case, JCD). Since the results we obtained were a list of train images that matched the query test image to a certain extent (given by the score), the output of the search was a list of ClassIds with attached scores, where a ClassId could appear multiple times with different scores, depending on how well the query test image and the train image corresponding to that ClassId matched. **Table 2** contains a sample of the result list obtained when using test image 1594.jpg as query.

Table 2. First 15 results for test image 1594.jpg

Train image	ClassId	Score
31580.jpg	Arbutus unedo	0.69314253
15871.jpg	Diospyros kaki	0.65694886
23509.jpg	Diospyros kaki	0.63348025
3445.jpg	Arbutus unedo	0.6267324
28871.jpg	Pittosporum tobira	0.6217524
22281.jpg	Viburnum opulus	0.6084521
30119.jpg	Papaver somniferum	0.60839295
36693.jpg	Punica granatum	0.6060035
1864.jpg	Pittosporum tobira	0.603533
26494.jpg	Pittosporum tobira	0.59786904
981.jpg	Arbutus unedo	0.5921401
27972.jpg	Viburnum opulus	0.58725476
36622.jpg	Olea europaea	0.5815127
13657.jpg	Arbutus unedo	0.57668316
23802.jpg	Pittosporum tobira	0.57403636

In order to obtain a single score for each ClassId from a result set, we tried three different approaches:

- Using only the **maximum score** of a ClassId in the result set;
- Computing the **sum for all the scores** of a ClassId and then dividing by the largest sum in that specific result set;
- Training a **naive Bayes classifier**[6]. For the training set, the first 100 ClassIds in the ranking given by LIRe are used as predictors for the (real) ClassId. Then the classifier is used to predict the ClassId on test data.

[6] Naive Bayes classifier, https://en.wikipedia.org/wiki/Naive_Bayes_classifier

4.2 Using training data set tags: author, organization, location

Image annotation task has usually relied on training data that has been manually, and thus reliably done but at an expensive and laborious endeavor. To automate this task, we have focused not only on visual information analysis, which is lacking in case of small data set training, but also on image associated meta tags: GPS coordinates, author and organization.

First, we clustered the training images based on their location to classify the test data using their geo-tags. This proved not to be as we have anticipated. The main reason is the fact that most of the training and test plants are located in France and so most species of the dataset are uniformly distributed in the covered area. Geo-tags would have been, probably, more useful at a global scale (continent, countries).

To cluster the images based on their geo-tags we used the quadtree data structure[7] [9], in which every internal node has exactly four sons and every child represents a quadrant (NE, NW, SE, and SW) of a given square at a corresponding level (see **Fig. 2**).

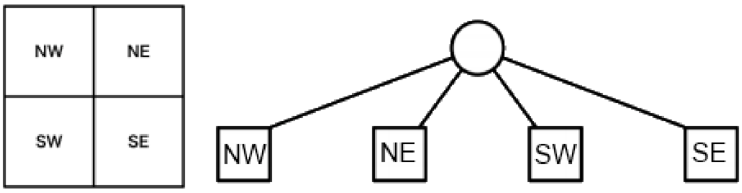

Fig. 2. An example of the quadrants of a plant GPS location and the corresponding quadtree

The quadtree[8] was built based on the training data geo location and was used to classify our test data by retrieving the nearest results in a 100 km area. To compute the distance between two locations, we used the Haversine formula[9], which computes the great-circle distance (the shortest distance over earth surface) between two points on a sphere from their latitude and longitude values:

$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$

- Where **d** is the distance between the two points;
- **r** is the radius of the sphere;
- ϕ_1, ϕ_2: latitude of point 1 and latitude of point 2;
- λ_1, λ_2: longitude of point 1 and longitude of point 2.

[7] Quadtree data structure, http://www.cs.umd.edu/~hjs/pubs/Samet85k.pdf

[8] QuadTree - Java implementation, http://openmap.bbn.com/svn/openmap/trunk/src/openmap/com/bbn/openmap/util/quadtree

[9] Haversine formula: http://en.wikipedia.org/wiki/Haversine_formula

A run example for test image 1000.jpg can be seen in **Fig. 3**, where the red marker is the test image and the green markers are the plant images located in the area within a 100 km radius of the red marker.

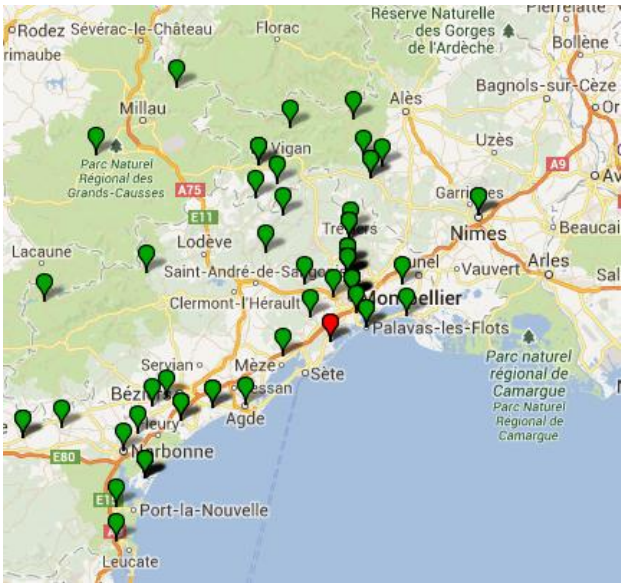

Fig. 3. Test results for image 1000.jpg

Next, the results were further refined and ranked based on author and organization tags, assuming that certain authors and organizations would have a greater interest in certain plant classes.

For every test image we have created a recommendation list of 20 items containing the author's most photographed plants. If no results were found, the algorithm continued searching for recommendations based on the author's organization tag.

5 Submitted runs and results

Our system (**Fig. 4**) has a modular and flexible structure and can easily be extended with some other feature extractors' algorithms. From the beginning, our intention was to use both features extracted from pictures and metadata information, and to find a way to combine them.

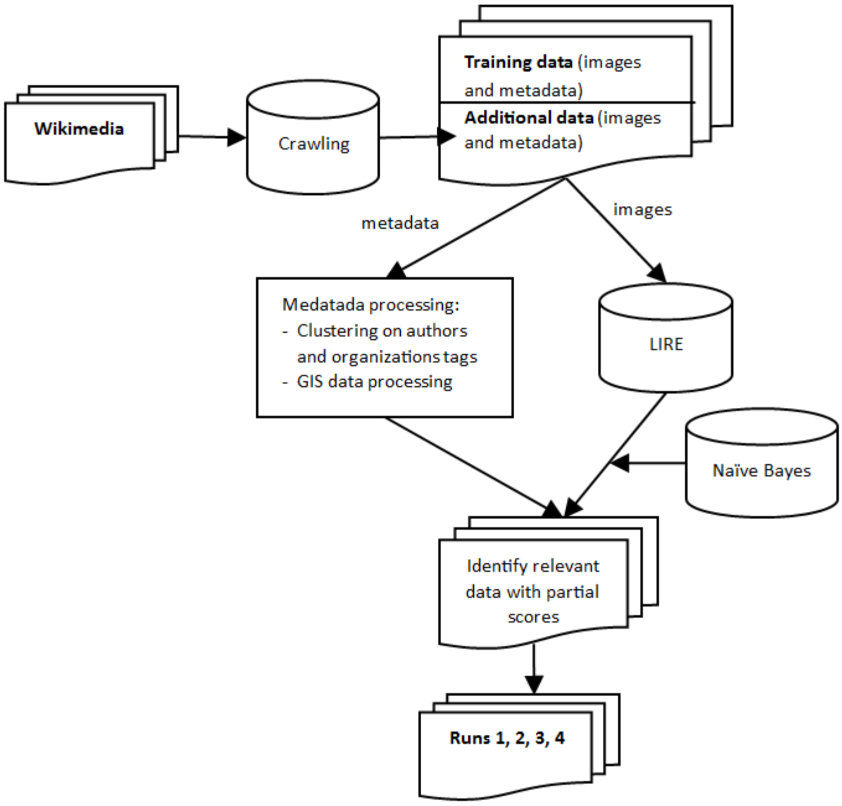
Wikimedia
Crawling
Training data (images and metadata)
Additional data (images and metadata)
metadata
images
Medatada processing:
- Clustering on authors and organizations tags
- GIS data processing
LIRE
Naïve Bayes
Identify relevant data with partial scores
Runs 1, 2, 3, 4

Fig. 4. UAIC Plant Identification System

We participated at ImageCLEF 2013 Plant Identification Task by submitting 4 runs with different configurations:

- *Run 1* (run_wiki_sum_3): We perform image retrieval using an extended collection of images obtained from the training collection and an extracted collection from Wikimedia[10]. For the final score of a ClassId we consider the sum of partial results obtained after retrieval.
- *Run 2* (run_author10_GSP10_lire80): We perform image retrieval using an extended collection of images obtained from training collection and an extracted collection from Wikimedia. For final score of a ClassId we consider different levels of importance for parameters: author (10%), GPS coordinates (10%) and image content (80%).
- *Run 3* (run_lire_naivebayes): For each image, nearest neighbors (from the training set) are computed based on image content. Nearest neighbors are replaced by their ClassIds that further become new attributes attached to the image. Naive Bayes is

[10] Wikimedia: http://commons.wikimedia.org

then used as classifier and the posterior probabilities are used to obtain the final ranking.

- *Run 4* (run_wiki_max_1): We perform image retrieval using an extended collection of images obtained from the training set and an extracted collection from Wikimedia. For final score of a ClassId we consider the maximum value from the partial results obtained after retrieval.

In the cases of runs 1, 3, 4, only visual information was used, while in the case of run 2 the textual information from the metadata was also used.

Table 3. Results of our submitted runs

#Run		Entire	Flower	Fruit	Leaf	Stem	Natural Background
4	run_wiki_max_1	0,09	**0,136**	0,12	0,08	**0,128**	**0,127**
1	run_wiki_sum_3	0,089	0,109	**0,132**	0,093	0,104	0,119
2	run_author10_GSP10_lire80	**0,092**	0,105	0,127	**0,096**	0,11	0,117
3	run_lire_naivebayes	0,068	0,055	0,111	0,049	0,102	0,081

Our best run was the one with *Run 4* configuration and it was ranked 5th of a total of 12 group participants (see Table 4). This run was based on the extended collection of training data obtained by getting new images from Wikimedia. *Run 3*, using naive Bayes, clearly suffered from overfitting: on the training data it recorded 100% accuracy while on the test data it shows to be the worst run.

Table 4. Best result of each participant in Plant Identification task at ImageCLEF 2013

#Run		Entire	Flower	Fruit	Leaf	Stem	Natural Background
1	NlabUTokyo Run 3	*0,297*	0,472	*0,311*	*0,275*	*0,253*	*0,393*
2	Inria PlantNet Run 2	0,274	*0,494*	0,26	0,272	0,24	0,385
3	Sabanci Okan Run 1	0,174	0,223	0,194	0,049	0,106	0,181
4	DBIS Run 2	0,102	0,264	0,082	0,034	0,095	0,159
5	**UAIC Run 4**	**0,09**	**0,136**	**0,12**	**0,08**	**0,128**	**0,127**
6	Liris ReVeS Run 2	0,026	0,102	0,082	0,161	0,166	0,092
7	Vicomtech Run 1	0,095	0,117	0	0	0,1	0,081
8	LAPI Run 1	0,026	0,073	0,025	0,084	0,043	0,058
9	Mica Run 2	0,016	0,086	0,048	0,014	0,014	0,053
10	SCG USP Run 3	0,017	0,025	0,042	0,047	0,054	0,03
11	I3S Run 1	0,017	0,023	0,041	0,038	0,025	0,026
12	AgSPPR Run 1	0	0	0	0	0	0

6 Conclusions

This paper proposes a modular system that allowed us to offer different solutions in the ImageCLEF 2013 Plant Identification task.

Even though we started with a large image collection, containing different views of the same plant (entire, flower, fruit, leaf, stem) with various backgrounds, we decided to add new images in the data test. We used Wikimedia Commons which offers human annotated images, thus ensuring a reliable content.

From this point on, we started to analyze and process our image repository in two directions. Firstly, in order to extract relevant features (see Section 2.1) from the available image set, we used LIRe. We performed different tests and we decided that JCD was the better choice for our image types, as far as visual features are concerned. Using JCD, we obtained an index from the train set and we started to search for the images in the test set. We obtained a list of train images (a list of ClassIds) corresponding to each query test image. At this point, in order to have a single score for each ClassId we decided on three approaches: using the maximum score for a ClassId, computing the sum of all scores for a ClassId and then dividing by the largest sum, or training a naive Bayes classifier.

Additionally, we performed various tests using the metadata of the images. Some results obtained using geo-tags metadata proved not to be relevant for our task. But during our tests we observed some patterns that allowed us to use author and GPS coordinates metadata in our final results.

After trying different configurations, we obtained four results depicted in Section 5. *Run 4* is our best result and was ranked 5th out of the 12 competitors.

Due to the increased complexity of the task from year to year, we believe that, in the future, our modular architecture will allow us to dynamically integrate new techniques and new algorithms to achieve suitable matches.

Acknowledgement. The research presented in this paper was funded by the project MUCKE (Multimedia and User Credibility Knowledge Extraction), number 2 CHIST-ERA/01.10.2012.

References

1. Goëau, H., Bonnet, P., Joly, A., Bakić, V., Barthelemy, D., Boujemaa, N., Molino, J. F.: The ImageCLEF 2013 Plant Identification Task. In CLEF 2013 Working Notes, Valencia, Spain (2013)
2. Lux, M., Savvas, A. C.: Lire: Lucene Image Retrieval – An Extensible Java CBIR Library. In proceedings of the 16th ACM International Conference on Multimedia, pp. 1085-1088, Vancouver, Canada (2008)
3. Chang, S. F., Sikora, T., Puri, A.: Overview of the mpeg-7 standard. IEEE Transactions on Circuits and Systems for Video Technology, vol. 11, no. 6, pp. 688–695 (2001)
4. Tamura, H., Mori, S., Yamawaki, T.: Textural features corresponding to visual perception. IEEE Transactions on Systems, Man, and Cybernetics, vol. 8, no. 6, pp. 460–472 (1978)
5. Chatzichristofis, S. A., Boutalis, Y. S.: Cedd: Color and edge directivity descriptor. a compact descriptor for image indexing and retrieval. In A. Gasteratos, M. Vincze, and J. Tsotsos, editors, Proceedings of the 6th International Conference on Computer Vision Systems, ICVS 2008, vol. 5008 of LNCS, pp. 312–322, Santorini, Greece, May, Springer (2008)

6. Chatzichristofis, S. A., Boutalis, Y. S.: Fcth: Fuzzy color and texture histogram a low level feature for accurate image retrieval. In Proceedings of the 9th International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS 2008, pp. 191–196, Klagenfurt, Austria, May, IEEE (2008)
7. Chatzichristofis, S.A., Arampatzis, A., Boutalis, Y. S.: Investigating the behavior of compact composite descriptors in early fusion, late fusion, and distributed image retrieval. Radioengineering vol. 19, no. 4, pp. 725–733 (2010)
8. Huang, J., Kumar, S. R., Mitra, M., Zhu, W. J., Zabih, R.: Image indexing using color correlograms. In Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition, CVPR ’97, vol. 00, pp. 762–768, San Juan, Puerto Rico, June, IEEE (1997)
9. Finkel, R., Bentley, J. L.: Quad Trees: A Data Structure for Retrieval on Composite Keys. Acta Informatica, vol. 4, no. 1, pp. 1–9 (1974)