LIRE - Open Source Visual Information Retrieval

Mathias Lux
Klagenfurt University
Universitätsstraße 65-67
Klagenfurt, Austria
mlux@itec.aau.at

Michael Riegler, Pål Halvorsen, Konstantin Pogorelov
SIMULA Research
Oslo, Norway
michael@simula.no

Nektarios Anagnostopoulos
Klagenfurt University
Universitätsstraße 65-67
Klagenfurt, Austria
nek.anag@gmail.com

ABSTRACT

With an annual growth rate of 16.2% of taken photos a year, researchers predict an almost unbelievable number of 4.9 trillion stored images in 2017. Nearly 80% of these photos in 2017 will be taken with mobile phones[1]. To be able to cope with this immense amount of visual data in a fast and accurate way, a visual information retrieval systems are needed for various domains and applications. LIRE, short for *Lucene Image Retrieval*, is a light weight and easy to use Java library for visual information retrieval. It allows developers and researchers to integrate common content based image retrieval approaches in their applications and research projects. LIRE supports global and local image features and can cope with millions of images using approximate search and distributing indexes on the cloud. In this demo we present a novel tool called F-search that emphasize the core strengths of LIRE: lightness, speed and accuracy.

Figure 1: Sample application built on LIRE. The image in the center is the query, the first six results of four queries based on four different features, three global, one local one, are shown around the query.

CCS Concepts

•Information systems → Multimedia information systems; Image search;

Keywords

Visual Information Retrieval; Search Engine

1. INTRODUCTION

Visual information retrieval and content based image retrieval have been around for years. In academia, it has been extensively reviewed (cp. [9]) and a lot of different approaches have been developed. However, early commercial software did not result in a broad application of visual information retrieval. Newer visual search engines took other approaches, like TinEye[2] with providing visual information retrieval technology as a service, or LegalZoom[3], which does a search for similar visual trademarks for the clients. Others focused on specific domains, like copyright infringement, medical retrieval, or near duplicate detection.

However, nowadays, visual information retrieval builds on the academic achievements of successful research and a lot of different approaches, techniques and methods are available. Applied research then adapts the methods to new data and new domains. For this, it is crucial to have a common foundation that agrees upon algorithms and software implementations. Such a foundation can prevent developers and researchers alike from re-developing well-known approaches. A common, free and easy to access knowledge base is the main goal of LIRE.

LIRE provides the most common and well working approaches to content based image retrieval. Implemented as a Java library, it allows easy integration in existing software environments. LIRE builds on Lucene[4], which is a well-known and well maintained text search engine. Furthermore,

[1]http://goo.gl/nJz8gJ
[2]http://tineye.com
[3]https://www.legalzoom.com
[4]https://lucene.apache.org/

MMSys'16 May 10-13, 2016, Klagenfurt, Austria

ACM ISBN 978-1-4503-4297-1/16/05.
DOI: http://dx.doi.org/10.1145/2910017.2910630

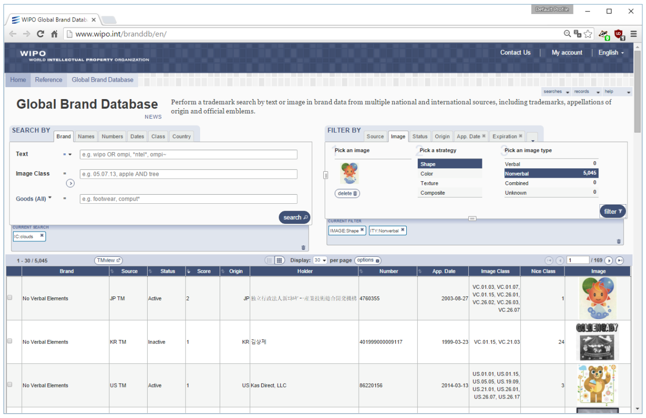

Figure 2: A screenshot of the UN WIPO Global Brand DB. The image filtering option is implemented using LIRE.

LIRE is the result of ongoing work of numerous contributors since February 2006. Since then, it is available as open source software under the GNU Public License. It has been hosted on sourceforge.net, Google Code and is currently maintained on Github[5]. Pre-compiled versions have been downloaded more than 51,000 times in 2015 alone. Major milestones were the release of the LIRE Solr Plugin in 2013 [16] and the version 1.0 beta release in 2015.

LIRE has been employed in academic research, teaching and real world scenarios alike. One major installation is at the UN headquarters in Geneve, Switzerland, running the visual trademark search at the World Intellectual Property Organization[6]. Fig. 2 shows a screen shot of the WIPO's Global Brand DB. There, a textual search for the term "clouds" is combined with a visual re-ranking based on a query image using PHOG [3]. Besides visual trademark search, LIRE has been employed for instance in asset management, copyright violation detection, and media monitoring. In the academic world, LIRE is used for feature extraction for classification, as base line for retrieval evaluation, for video search and summarization and as library providing image search for user interface and knowledge discovery projects.

2. LIRE

LIRE aims to be easy to use as well as easy to built new services on. If for instance new features are to be tested, developers and researchers only need to implement the feature interface including the serialization and extraction. Everything else then is done by LIRE, including parallel indexing, local feature aggregation, hashing, as well as approximate and linear search. This allows researchers and developers to focus on their features instead of having to implement the whole search engine.

LIRE supports multiple global and local features out of the box, to allow for easy comparison of new features to existing and well-known ones. Most notable global ones are CEDD [6] as well as the related features JCD [7] and FCTH [5], PHOG [3], the Auto Color Correlogram [11], Local Binary Patterns [18], CENTRIST [23], and the MPEG-7 features [4] Edge Histogram, Color Layout and Scalable Color.

[5]https://github.com/dermotte/lire
[6]http://www.wipo.int/branddb/en/

Local features are based on the OpenCV implementations of SIFT [15] and SURF [2]. For retrieval the bag of visual words approach [21] as well as VLAD aggregation of local features [14] are supported. In addition to that, LIRE fully implements the SIMPLE [12] approach to using global features on local image patches with configurable key point detectors.

For indexing, LIRE supports linear search as well as locality sensitive hashing [8] with a specific implementation of bit sampling. In addition to that, LIRE supports a permutation based approach called metric index [1], which adapts to image domains better than the hashing based approaches and employs inverted files for indexing [10].

3. PERFORMANCE

There are two main performance indicators for a image retrieval runtime: (i) performance on a single machine and (ii) scalability. For indexing, there are two main entry points. One is at the level of feature extraction, where indexing has to be handled by the users of LIRE. The more convenient approach is to use the parallel indexing routine provided by LIRE. It is configurable by supporting custom pre-processors, making use of multiple cores, and producing a Lucene index, which can easily be merged with indexes built with the same parameters. Thus, indexing is fully scalable.

For linear search, three optimizations are supported. These are, (i) memory cached search, where all image feature data is stored in memory, (ii) multi-core-search, where the search is run in parallel over index partitions, and (iii) DocValues based search using a mechanism of Lucene, where RAM and disk serialization are heavily optimized. With a GPU based approach, which is currently under development for indexing and searching video streams, indexes with up to one million images can be queried in 3ms for a resolution of $856x480$, and 18ms for images with a resolution of $1920x1080$. For more than a million images, LIRE provides approximate search techniques based on hashing [8] and permutation indexes [10]. Moreover, the index can be partitioned and search results can be merged to get more accurate results and at the same time increase speed [19].

Retrieval performance is shown in Table 3. The employed data sets are *SIMPLIcity* data set [22], the *UKBench Recognition Benchmark Images* data set [17], the *Uncompressed Colour Image Database* (UCID) [20], and the *INRIA Holidays* dataset [13]. While not being able to publish all possible feature and aggregation combinations, we aimed to give an overview on the performance. Retrieval features marked with a *(G)* in the Table 3 are global ones, i.e., Auto Color Correlogram, CEDD, Color Layout, Edge Histogram, JCD, Local Binary Patterns and Scalable Color. Global features marked with an *(SB)* are used on local image patches by employing the SIMPLE approach [12] with a bag of visual words aggregation. The number complementing the *SB* gives the number of visual words for this particular test. CVSIFT and CVSURF are the SIFT and SURF implementations from OpenCV, respectively. The *(B)* with the number indicates the use of the bag of visual words aggregation with the given number of visual words. *(V)* and *(SV)* denotes the use of the VLAD aggregation techniques for local and global features. In the latter case, the SIMPLE approach has been used to create local features first. The number of visual words is a lot smaller due to the VLAD aggregation.

	SIMPLICity [22]		UKBench [17]		UCID [20]		Holidays [13]	
	MAP	P@10	MAP	P@10	MAP	P@10	MAP	P@10
Auto Color Correlogram (SB, 128)	0.5380	0.7687	0.9082	0.3680	0.7752	0.2584	0.7914	0.2328
Auto Color Correlogram (G)	0.5099	0.7765	0.9253	0.3736	0.7488	0.2427	0.7986	0.2360
Auto Color Correlogram (SV, 16)	0.3920	0.7242	0.9009	0.3660	0.7513	0.2511	0.7602	0.2266
CEDD (SB, 2048)	0.5222	0.8030	0.8917	0.3596	0.7869	0.2611	0.7779	0.2284
CEDD (G)	0.5040	0.7410	0.8055	0.3324	0.6740	0.2229	0.7263	0.2114
CEDD (SV, 16)	0.4488	0.7333	0.8557	0.3504	0.7704	0.2542	0.7377	0.2154
CL (SB, 2048)	0.5211	0.7644	0.8399	0.3436	0.7079	0.2328	0.7385	0.2150
CL (G)	0.4506	0.6574	0.7035	0.2900	0.5675	0.1824	0.6480	0.1852
CL (SV, 64)	0.3747	0.6961	0.7844	0.3268	0.7068	0.2305	0.7060	0.2080
CVSIFT (B, 512)	0.3756	0.5620	0.6847	0.2808	0.6085	0.1954	0.6914	0.2016
CVSIFT (V, 64)	0.4489	0.6247	0.8047	0.3324	0.6933	0.2302	0.7581	0.2202
CVSURF (B, 2048)	0.3801	0.5555	0.6253	0.2644	0.5852	0.1885	0.6777	0.1954
CVSURF (V, 64)	0.4370	0.6111	0.6681	0.2900	0.6441	0.2145	0.7169	0.2092
Edge Histogram (G)	0.3454	0.5538	0.4832	0.2056	0.5019	0.1588	0.5551	0.1594
JCD (G)	0.5140	0.7498	0.8480	0.3464	0.6945	0.2279	0.7351	0.2162
Local Binary Patterns (G)	0.3699	0.6356	0.5302	0.2228	0.5325	0.1641	0.5575	0.1578
Scalable Color (G)	0.5222	0.7692	0.8990	0.3672	0.7116	0.2309	0.7454	0.2186

Table 1: Feature performance on four data sets. The X in (X) denotes: G for global, B for bag of visual words and V for VLAD aggregation. S for Simple, SB and SV denote bag of visual words or VLAD aggregation.

Figure 3: Sample application built on LIRE showing results for a different query image than Fig. 1.

4. DEMO

To show some of the aspects of LIRE, we present here a novel image retrieval and result browsing application. It utilizes the core strengths of LIRE: small footprint and minimal API, speed and accuracy. The difference to common image retrieval search engines is that it is a combination of browsing and searching, where users implicitly select the image features that match their sense of similarity best. At the start, the user provides a query image. Then, the search engine retrieves results using different pre-selected features. If users are for instance interested in similar colors and shapes, they can pre-select four different features that represent these attributes. After the users picked the features and used the query image to get the first results, they can explore the available results in four partitions, each representing the results for one feature. Fig. 1 and Fig. 3 show the desktop application. The query image is shown in the center, lines in the background of the results show the partitions. Users can navigate in the images and selecting an image results in a new search using the selected image as query. Therefore, users can browse the data set based on four different features. Artists and photographers for instance could find and browse images that share a either similar composition or color distribution at the same time. For example in Fig. 1 CEDD and SIMPLE CEDD give color based results with the latter providing different results as it is a localized version of CEDD, whereas PHOG and Edge Histogram (EH) based searches are returning images with similar composition. Fig. 3 shows the same composition of features for a different query image.

Moreover, we are testing the demo in a medical setting where it can help gastroenterologist (medical doctors specialized on the gastrointestinal tract of the human body) finding similar cases in their image databases. This is important since doctors are not likely to recall when and where a similar case happened, but they usually know if there was something similar in the past and how it approximately looked. The demo application is available for the desktop application written in *Processing 3* as well as for Android mobile phones and tablets.

Acknowledgments

We would like to thank at least some of the numerous people having contributed to LIRE: Anna-Maria Pasterk, Arthur Li, Arthur Pitman, Bart Van Bos, Bastian Hösch, Benjamin Sznajder, Berthold Daum, Carlos Perez Lara, Christian Penz, Christine Keim, Christoph Kofler, Chrysa Iakovidou, Dan Hanley, Daniel Pötzinger, Fabrizio Falchi, Franz Graf, Giuseppe Amato, Glenn MacStravic, James Charters, Ja-

nine Lachner, Katharina Tomanec, Konstantin Pogorelov, Lukas Esterle, Lukas Knoch, Manuel Oraze, Marco Bertini, Marian Kogler, Marko Keuschnig, Martha Larson, Michael Riegler, Nektarios Anagnostopoulos, Rodrigo Carvalho Rezende, Roman Divotkey, Roman Kern, Sandeep Gupta, and Savvas Chatzichristofis.

5. REFERENCES

[1] G. Amato and P. Savino. Approximate similarity search in metric spaces using inverted files. In *Proc. of InfoScale*, 2008.

[2] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Computer vision–ECCV 2006*, pages 404–417. Springer, 2006.

[3] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, CIVR '07, pages 401–408, New York, NY, USA, 2007. ACM.

[4] S.-F. Chang, T. Sikora, and A. Purl. Overview of the mpeg-7 standard. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11(6):688–695, 2001.

[5] S. Chatzichristofis, Y. S. Boutalis, et al. FCTH: Fuzzy color and texture histogram-a low level feature for accurate image retrieval. In *Image Analysis for Multimedia Interactive Services, 2008. WIAMIS'08. Ninth International Workshop on*, pages 191–196. IEEE, 2008.

[6] S. A. Chatzichristofis and Y. S. Boutalis. CEDD: color and edge directivity descriptor: a compact descriptor for image indexing and retrieval. In *Computer vision systems*, pages 312–322. Springer, 2008.

[7] S. A. Chatzichristofis, Y. S. Boutalis, and M. Lux. Selection of the proper compact composite descriptor for improving content based image retrieval. In *The Sixth IASTED International Conference on Signal Processing, Pattern Recognition and Applications SPPRA 2009*, 2009.

[8] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 253–262. ACM, 2004.

[9] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (CSUR)*, 40(2):5, 2008.

[10] C. Gennaro, G. Amato, P. Bolettieri, and P. Savino. An approach to content-based image retrieval based on the lucene search engine library. In *Research and Advanced Technology for Digital Libraries*, pages 55–66. Springer, 2010.

[11] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih. Image indexing using color correlograms. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 762–768. IEEE, 1997.

[12] C. Iakovidou, N. Anagnostopoulos, Y. Boutalis, S. Chatzichristofis, et al. Searching images with mpeg-7 (& mpeg-7-like) powered localized descriptors: the simple answer to effective content based image retrieval. In *Content-Based Multimedia Indexing (CBMI), 2014 12th International Workshop on*, pages 1–6. IEEE, 2014.

[13] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometry consistency for large scale image search-extended version. 2008.

[14] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3304–3311. IEEE, 2010.

[15] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[16] M. Lux and G. Macstravic. The LIRE request handler: A Solr plug-in for large scale content based image retrieval. In C. Gurrin, F. Hopfgartner, W. Hurst, H. Johansen, H. Lee, and N. O'Connor, editors, *Proceedings of the 20th MultiMedia Modeling Conference (MMM 2014)*, volume 8326 of *Lecture Notes in Computer Science*, pages 374–377, Dublin, IE, Jan 2014. Springer International Publishing.

[17] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2161–2168. IEEE, 2006.

[18] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, 2002.

[19] M. Riegler, M. Larson, M. Lux, and C. Kofler. How 'how' reflects what's what: Content-based exploitation of how users frame social images. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 397–406, New York, NY, USA, 2014. ACM.

[20] G. Schaefer and M. Stich. Ucid: an uncompressed color image database. In *Electronic Imaging 2004*, pages 472–480. International Society for Optics and Photonics, 2003.

[21] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1470–1477. IEEE, 2003.

[22] J. Z. Wang, J. Li, and G. Wiederhold. Simplicity: Semantics-sensitive integrated matching for picture libraries. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(9):947–963, 2001.

[23] J. Wu and J. M. Rehg. CENTRIST: A visual descriptor for scene categorization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(8):1489–1501, 2011.