Davide Desirello, Rosario Di Stabile, Luca Martini, and Federico Maria Pandolfi

> Rai Teche Via Giuseppe Verdi 31, 10124 Turin, Italy {davide.desirello,rosario.distabile, luca.martini,federico.pandolfi}@rai.it http://www.teche.rai.it/

**Abstract.** In today's digital age, the ability to access, analyze and (re)use ever-growing amounts of data is a strategic asset for the broadcasting and media industry. Despite the growing interest around new technologies, archive's search and retrieval operations are still usually done by means of text-based search over tags and metadata of manually pre-annotated material. This is particularly true because of its reliability and the broad availability of powerful full-text search platforms.

However, this approach still does not completely meet the requirements that a search over huge multimedia archives poses, such as the need for semantic-driven indexing and retrieval, or the possibility to access contents based on visual features.

In this paper, we describe a framework currently under development in Rai that enables visual search over the company's archive, which includes still images as well as annotated broadcast contents and raw footages, totaling over 1.5 million hours of video material. The current architecture's core is based on LIRe (Lucene Image Retrieval), an open source Java Library for content-based image retrieval, and Apache Solr, an enterprise full-text search platform. Possible extensions of the framework to include new technologies such as deep learning or semantic learning are also discussed.

Keywords: image search, video search, LIRe, Solr, CBIR

## 1 Introduction

For modern broadcast and media companies, the proper organization and management of contents, including archives of footage and production material, constitutes a strategic asset. Furthermore, efficient search and retrieval methodologies are equally important to quickly and effectively access those contents.

Multimedia asset management (MAM) systems attempt to address this problem by providing solutions to easily store and retrieve media files. Pioneer systems used by the industry employed text-based queries to search over textual information and metadata, typically associated to each stored file using either

semi-automatic or handmade annotations. While this procedure is still in practice these days, due to its overall reliability and robustness, it presents some critical weaknesses.

First, metadata extraction is an expensive and time consuming process, which requires human supervision and needs to be done both for audiovisual content that is produced digitally in the first place, as well as for vintage footage that is converted from analog to digital formats. Second, search and retrieval based on handmade metadata annotation usually does not involve semantics or analytical representations of the media contents, thus does not allow visual query tasks such as query-by-example (e.g. image queries) or near duplicate detection. A clever use of metadata helps to mitigate these issues, but does not solve the problem.

To address these and other shortcomings, Content-Based Image Retrieval (CBIR) systems have been developed. These systems tackle some of the issues related to the use of textual metadata by representing multimedia items in terms of features automatically extractable from the contents themselves, rather than in terms of metadata (manually) associated to the files. Nowadays, there is a considerable and always growing number of CBIR systems available on the market, with different features and licensing options tailored to address specific needs in image search. For a comprehensive review of state of the art in CBIR system, interested readers may refer to [1] and [2].

Despite the considerable effort, almost all the available CBIR systems still suffer from the semantic gap issue, being based on low-level features rather than on high level concepts. To overtake this issue, efficient algorithms for object recognition, such as SIFT and SURF, have been proposed in [3] and [4]. As an example, the MPEG Compact Descriptors for Visual Search (CDVS) framework provides a robust and inter-operable technology to create efficient visual search applications in image databases [5]. In the last years, as the number of index entries of image databases increases at a fast pace, the state-of-the-art paradigm is shifting from using features extracted by deterministic algorithms to using Deep Convolutional Neural Network features, as explained in [6].

The attention is also moving from still images to the video domain. The LIvRE project [7] represents an interesting attempt at exploring the expansion of Lucene Image Retrieval (LIRe) engine [8], an open-source CBIR system, for video retrieval on large scale video datasets. Furthermore, in order to meet industrial needs, the MPEG CDVA (Compact Descriptors for Video Analysis) call for proposal aims to enable efficient and inter-operable design of compact video description technologies for search and retrieval in video sequences [9].

In the broadcast domain in which we operate, the target tasks are mainly focused on image-to-video or video-to-video search. Since, as stated above, there is plenty of options to choose from for image search but there are fewer ready-tomarket solutions for video-to-video search, we started developing a new framework based on ready-to-use solutions, compatible with our enterprise infrastructure. This choice was motivated by the need of integrating such novel search and retrieval framework in the existing archival and production workflows while ensuring compatibility with the software used within our company. Since Apache Solr is widely adopted in Rai, among all the options we decided to use LIRe (Lucene Image Retrieval) [10], a simple but powerful and open-source (GNU-GPL) Java library, which is capable of retrieving images and photos based on visual characteristics and provides a plug-in for Solr integration.

The remain of the paper is organized as follows. Section 2 reveals more details about the use-case at the core of this paper. Section 3 describes the workflow at the heart of our framework. Section 4 provides some preliminary considerations about performance measurement. Section 5 concludes the paper with a brief sum up and future directions.

## 2 Case Study: Rai's production environment

Being Rai a broadcasting company, there are different scenarios within the company's departments that could greatly benefit from a proper visual information retrieval engine. To provide some examples, our real-world use-cases include:

- In the news department, being able to link an edited news/reportage to its raw footage and, viceversa, being able to retrieve all the news/reportages that used a specific footage (video-to-video search);
- In the archives department, aiding the employees during semi-automatic annotation tasks (video-to-video search) by correlating non-annotated material with similar pre-annotated contents;
- In the archives department, being able to retrieve a specific video or image in the multimedia catalog from a clip, a single frame or a similar image (image/video to image/video search);
- For online content, allowing the user to find a specific show from an image/clip (images/video to video search).

It can be noted that almost all of the use-cases mentioned above fall within two main categories: image-to-video search and video-to-video search. In this paper we will examine mainly the former category (image-to-video search) as it will serve as a starting point for the more complex video-to-video search.

Since our goal was to implement a CBIR framework, we decided to start the development process by adopting one of the already available image retrieval solutions and build our framework on top of that. From our preliminary research regarding the state-of-the-art, though, we spotted some possible obstacles that separate us from our goal. In fact, cutting-edge solutions usually offer solid absolute performance [11] at the price of very complex systems and/or non patent-free algorithms (especially regarding the descriptors employed [12]). Those factors are not ideal in an enterprise environment as they translate in a more expensive and difficult to maintain platform. Therefore, for a first approach, we decided to fall back on a simpler but more manageable option.

LIRe was our first choice because, as stated above, is a valid CBIR platform that can be integrated with Apache Solr, an enterprise search server widely used in Rai, by means of a ready-to-use plugin (currently used in this project) that

ensures compatibility to the Solr indexing format. The adoption of Solr allows distributed search and index replication and scalability, making it a much better and efficient enterprise solution.

## 3 Proposed workflow

In this section we will explain in details the workflow at the core of the whole project.

## 3.1 Modularity

One of the main advantages of the proposed framework is its modularity. The whole architecture was planned and designed to make the fundamental logic blocks of the workflow as independent as possible. This will enable us to easily develop code in parallel and swap the blocks in case we find out more efficient solutions in the future, other than making the whole framework easier to debug and maintain.

The main modules composing the framework (and their current implementations) are:

- 1. Listener (custom files and folders manager)
- 2. Scene detector/key-frames extractor (FFMpeg)
- 3. Feature extractor (LIRESOLR Plugin)
- 4. Indexer (LIRESOLR Plugin)
- 5. Retriever (LIRESOLR Plugin)

In Fig. 1 a diagram of the architecture is represented. It is worth noting that the scene detection and key-frames extraction blocks are completely separate from feature extraction and indexing blocks. This will allow us to replace the basic scene detection we used with more sophisticated algorithms (such as those described in [13]) and the key-frame extraction with motion-vector based approaches [14].

The starting point of the whole process is the creation/addition of a JSON token file within the watch folder, which triggers a Listener application that, in turn, acts like a supervisor of the whole chain.

### 3.2 Indexing and Listener flow

The first step in our workflow consists in indexing the reference videos for our database. Those videos are the references that will be matched during the re-trieval phase.

Rai owns a great amount of documents which differ both in format (image/video, analog/digital) and geographic storage location. To make our framework effective, we planned to provide various entry-points for video indexing and we opted to offer a two-way approach to input files into the chain:



Fig. 1. Framework architecture

- Shared folder: used to integrate easily our workflow inside pre-existing company's workflows such as the digitisation process of the DIGIMASTER [15] archive. This approach is mainly used to ingest files that, currently, are not stored in Rai's multimedia catalog.
- RESTful APIs: a well-known and solid standard for any modern distributed application. These APIs provide both an interface to write videos to be indexed into the shared folder via webservices and a way to index files already available in Rai's multimedia catalog without re-uploading them to the shared folder.

The Listener process is developed to run in background and watch a shared folder that acts as a container for the files to be indexed. When new files are added to that folder, the Listener is triggered and its execution follows the steps below:

- 1. Wait for a JSON token file creation/addition in the shared folder.
- 2. Create an output folder associated with each input video to be processed.
- 3. Perform scene detection with FFMPEG and save the selected frames with their time-stamp.
- 4. Extract CEDD [16] features with LIRESOLR plugin to provide an output compatible to the Solr indexing format.
- 5. Generate a JSON metadata file associated to the token (optional, if any metadata is available).
- 6. Index LIRe document in a Solr's index called ImageCore and JSON metadata in another index called MetaCore.

In the next subsections we will describe further those steps.

**JSON Token file:** The initial trigger of the whole indexing flow is the JSON token file, which should be added to the watch folder after the files it is related

to. This file contains an array of parameters needed by the indexing process to run properly and each element in the array is composed, in turn, by two main elements **VideoInfo** and **MetaData**.

To run the process in batch on multiple video files, these elements must be specified for each video and, for each video in the JSON file, a corresponding output folder is created at runtime using a structure based on the current datetime.

This configuration allows to easily manage video files with multiple formats and resolutions and to control the status of the execution via the JSON tokens. Moreover, this architecture allows to have multiple clients populating the folder simultaneously with heterogeneous sources.

Scene detection and subsampling: Step 3 of the workflow consists in the generation of the images whose features will form the retrieval index. Since the whole workflow is mainly targeted at image search on video files, a proper scene detection methodology has to be used to extract significant images from video files. In the current implementation scene detection and key-frame extraction are both performed using FFMpeg filters. This allows to execute these tasks with a good precision in acceptable processing times. To be more specific, the command chain currently used by FFMpeg is:

- 1. Selection of Intra-Frames with *select='eq(pict\_type,I)'*: this option makes the extraction phase much faster without penalizing the performance.
- 2. Selection of scene-change frames with that select='gt(scene,d)': selects the frames whose new scene probability value is greater than the threshold The scene probability value used by FFMpeg is evaluated using a LGPL algorithm within *libavfilter* library.
- 3. Extraction of time-stamps of selected frames with showinfo.

Feature extraction: After the key-frame extraction, the obtained images have to be indexed to allow visual search. During the indexing process, global features corresponding to the "Color and Edge Directivity Descriptor" (CEDD) are extracted using the corresponding LIREFEATURE class. This descriptor, despite being slightly obsolete and not state-of-the-art, was selected because it incorporates color and texture information in a histogram and performs well for many use cases, according to [8]. Two of the most important attributes of this descriptor are the low computational power needed for its extraction and its length which does not exceed 54 bytes, making it an advantage in terms of query time reduction. For each image, the following fields are stored in the index:

- $\mathbf{ID}:$  the identifier of the key-frame.
- $-~{\bf URI}:$  the key-frame's absolute path.
- **Feature vector**: the actual image features are stored in this field in their histogram and hash variants

The index creation is performed by LIRESOLR plugin to ensure compatibility with Solr indexing format. **Solr cores:** As just mentioned, our workflow is currently based on Apache Solr search platform for the indexing/retrieval module. In the current implementation we instantiated two separate Solr cores:

- ImageCore: it stores the index with the global features of the input frames (extracted using LIRESOLR plugin).
- MetaCore: it stores the metadata informations related to the videos indexed in the ImageCore (wherever available).

It is worth noting that this second core extends the capabilities of the ImageCore by giving the possibility to keep track of the source and original metadata of the input files besides allowing to update the metadata in a successive moment (after a manual annotation, for example). Moreover, theoretically, it enables the user to retrieve indexed videos using a more traditional text search approach.

#### 3.3 Retrieval process

During the retrieval process, the same features selected in the indexing phase are extracted from the query image. Results are then collected from the index after evaluating the distance between each entry and the query image using the distance metric specified for the selected feature (Tanimoto coefficient [16]).

At query time, LIRe allows to set two parameters to tweak the speed and accuracy of the retrieval process:

- Accuracy is a parameter used to choose a trade-off between runtime complexity and precision of results. An accuracy parameter below 1 means that the results are approximate but the search is performed faster.
- Number of candidates is another settings parameter aimed at reducing runtime complexity. Lower values means faster searches but less accurate results.

The results obtained after this operation are then sorted by relevance using the same distance measurement as score and presented to the user with a GUI.

The retrieval process described so far is very simple and a lot of effort has been made in the past to improve the indexing structures and retrieval performances (as it can be read in [17], [18] and [19]). It's worth noting, though, that this paper is describing a work in progress which is still in its early development stage. Better retrieval strategies will be further investigated and adopted in future releases.

# 4 Preliminary evaluation

In the preliminary stage of our work, image-to-video search was considered as the starting point. Regarding the datasets involved, Rai archives store a humongous amount of documents in different locations, with 1.540.032 hours<sup>1</sup> of video

<sup>&</sup>lt;sup>1</sup> Latest data as of 30 June 2015.

material only. Ideally, performing image search on all these files would be a massive long-term achievement. In this initial development phase, though, it would be impractical, to say the least, to process and perform image search on all those files. To tackle this problem we selected two specific datasets, that cover pretty well most of the use-cases mentioned above, in order to evaluate and test our platform. These datasets are:

- TG Leonardo (set of 2200 episodes, approx. 360 hours of material): a thematic, scientific focused, newscast, suitable for news/reportage and raw footage retrieval but also to find similar videos for the recommendation system.
- Medita (set of 2000 episodes, approx. 2000 hours of material): an educational show aired both on TV (Rai Edu 1) and online. It represents the greatest online educational media library and each episode is aimed to be a supporting material for teachers and students. This dataset is well suited to test pure image search and tagging-aid capabilities of our framework.

The proposed workflow has been tested using images extracted from the same videos used for indexing with two different key-frame extraction techniques. This type of visual search was chosen because is close to the use case of video search on video database, the next step in our roadmap. The dataset involved was the TG Leonardo, while the key-frame extraction techniques were:

- FFMpeg shot detection
- Rai's Shotfinder

The latter is a proprietary software, developed by Rai within a bigger framework aimed to aid news annotation and called Automatic Newscast Transcription System (ANTS) [20]. Shotfinder usually works pretty well for news-like format such as the TG Leonardo dataset as its scene-detection engine is tailored on newscasts editing style.

The reason why we decided to test the framework with two different keyframe extractors is because we wanted to test the robustness of the global descriptor of choice when the query image is slightly different to one or a small subset of indexed images.

Let's see now an example of a test performed using the different key-frame extractors mentioned above. In Fig. 2 is shown a comparison between the best match retrieved (on the right) and the query image (on the left). Is quite obvious that the reference video is the same but the images are slightly different because they were extracted using different algorithms. In this particular example, the descriptor seems to be robust enough, anyway, the retrieval performance is not always as good as in this case.



Fig. 2. Slightly different images extracted using different algorithms

In the following example (Fig. 3) we can see that the best match is not always found among the very first results: this could be related to the fact that CEDD is a very compact descriptor (good for fast retrieval times) and, hence, images with similar colours and textures may have very similar descriptors. Changing the accuracy does not guarantee a substantial improvement of the results but increases retrieval time.



Fig. 3. Slightly different images extracted using different algorithms

Regarding the quantitative evaluation of the framework, due to the nature of the datasets (high frame-to-frame difference, motion blur in shots, small number of shots representing a scene, etc) it was very difficult to give an evaluation in terms of *precision* and *recall* for query images different than the indexed images. In facts, when the query shot is the same as an already indexed shot, the  $p_{at}(1) \simeq 1$  and the correct shot is retrieved in the first position every time. Otherwise, the descriptor of choice does not prove to be robust enough and the first result has a distance value significantly higher than the matching case previously described.

This result, though, is satisfactory enough for our first use case (raw footage/final edit match) because, if the same piece of footage is present in two videos and the key-frames are extracted in the same way, there's a high chance that the query key-frame and indexed key-frame will be the same. Further considerations

regarding video quality differences between final edits and raw footages will be investigated in the future.

Our benchmarks also targeted retrieval times as we wanted to give an insight about the speed of the framework. The setup that we used was the following:

- Local web-server for request handling (based on java Spring framework).
- Single Solr index used for queries.
- Solr core and web-server both hosted on the same virtual machine with 4 cores and 8GB of RAM dedicated.

We tested query times using three different accuracy/candidates configurations, as it can be seen from the Tab.1.

Query time evaluation (ms)			
	rawDocsSearchTime	reRankSearchTime	totalTimeResponse
A=0.33, C=10000	91.2	90.5	181.7
A=0.5, C=50000	264	224.3	488,3
A=0.8, C=80000	355.4	375.8	731,2

Table 1. Avarage query time

The *accuracy* parameter (A) influences just the raw documents search time as well as the retrieval precision, while the *number of candidates* (C) affects the time needed to re-rank the results in a similar fashion.

# 5 Conclusions and future work

In the previous chapters we discussed briefly about CBIR system and where our work is trying to fit in today's scenario. We also described the current development stage of our framework and we presented some very early results to back up our approach.

In the current state our framework seems to confirm expectations that we are not able to find instances of same objects within different videos and under different conditions (e.g. different video quality, framing, etc..). One reason for this may be the choice of the CEDD descriptor, and, in general, global descriptors. On the other hand, those compact global descriptors may give good results for specific tasks like searching the exact same videos segments inside different dataset, useful in our case to match raw and edited footage.

The quantitative tests we presented are not mature yet, one reason for that is the lack of copyright-free datasets and evaluation framework that targets our specific use-case to use as reference. In fact, almost all open datasets available up to date are either very generic (ImageNet [21], CoPhIR [22]) or very applicationspecific (medical datasets, face recognition databases [23], ...) but any of them target a use-case like ours where the images indexed are actual video frames extracted from archive footage. Another reason is that making a proper dataset from scratch requires time and our framework is still in a very early stage of development. Those are common inconveniences and also other authors reported those problems and tried to propose various solutions [24]. To address this inconvenience and provide more scientific results, we are planning to build our own annotated dataset using the company's archive material.

For the future developments of the retrieval core we plan to evaluate the performance of more sophisticated feature extraction algorithms, including local features, bags of visual words and deep-learning generated feature vectors. Most likely, this could also lead to the adoption of different retrieval solutions than LIRe. Regarding the deep-learning, we also wish to integrate this technology within the framework, for example with DCNN features or enriching the MetaCore with automatically-extracted scene informations (e.g. object/face recognition, image captioning, ...).

Regarding the other functional blocks of the framework, our goal is to investigate further on key-frame extraction and shot detection algorithms in order to reduce the number of extracted key-frames and, possibly, weighting them according to their relevance within the related sequence. By doing this we hope to improve retrieval performances, decrease index size and, therefore, reduce disk occupation and speed-up search times.

## References

- Saurav Seth, Prashant Upadhyay, Ruchit Shroff, Rupali Komatwar. "Review of Content Based Image Retrieval Systems", International Journal of Engineering Trends and Technology (IJETT), V19(4), 178-181 Jan 2015
- Jigisha M. Patel, Nikunj C. Gamit. "A review on feature extraction techniques in Content Based Image Retrieval", Proceedings of the IEEE International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), 2016
- D. G. Lowe, "Object recognition from local scale-invariant features," Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, 1999, pp. 1150-1157 vol.2. doi: 10.1109/ICCV.1999.790410
- Ravi Kiran Boggavarapu, Pushpendra Kumar Pateriya, "A Study on Feature Detectors and Descriptors for Object Recognition", Third International Conference of Computing Sciences, 2016
- "Information technology Multimedia content description interface Part 14: Reference software, conformance and usage guidelines for compact descriptors for visual search", ISO/IEC 15938:14, Oct 2015
- Amato G., Debole F., Falchi F., Gennaro C., Rabitti F. (2016) Large Scale Indexing and Searching Deep Convolutional Neural Network Features. In: Madria S., Hara T. (eds) Big Data Analytics and Knowledge Discovery. DaWaK 2016. Lecture Notes in Computer Science, vol 9829. Springer, Cham
- Gabriel de Oliveira Barra, Mathias Lux and Xavier Gir Nieto, "Large scale contentbased video retrieval with LIvRE", Proceedings of the IEEE International Wokshop on Content-Based Multimedia Indexing (CBMI), 2016
- Lux, Mathias, and Savvas A. Chatzichristofis. "LIRe: Lucene Image Retrieval: an extensible java cbir library." Proceedings of the 16th ACM international conference on Multimedia. ACM, 2008.

- 12 A framework for visual search in broadcast archives
- "Call for Proposals for Compact Descriptors for Video Analysis (CDVA) Search and Retrieval", ISO/IEC JTC1/SC29/WG11/N15339, Warsaw, Jul 2015.
- 10. http://www.lire-project.net/
- L. Y. Duan et al., "Overview of the MPEG-CDVS Standard," in IEEE Transactions on Image Processing, vol. 25, no. 1, pp. 179-194, Jan. 2016. doi: 10.1109/TIP.2015.2500034
- S. S. Husain; M. Bober, "Improving large-scale image retrieval through robust aggregation of local descriptors," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.PP, no.99, pp.1-1 doi: 10.1109/TPAMI.2016.2613873, 2016
- Baraldi, Lorenzo, Costantino Grana, and Rita Cucchiara. "A deep siamese network for scene detection in broadcast videos." Proceedings of the 23rd ACM international conference on Multimedia. ACM, 2015.
- Qiang Z., Xu Q., Sun S., Sbert M. (2016) Key Frame Extraction Based on Motion Vector. In: Chen E., Gong Y., Tie Y. (eds) Advances in Multimedia Information Processing - PCM 2016. PCM 2016. Lecture Notes in Computer Science, vol 9917. Springer, Cham
- 15. http://www.crit.rai.it/eletel/2015-2/152-5.pdf
- Chatzichristofis, Savvas A., and Yiannis S. Boutalis. "CEDD: color and edge directivity descriptor: a compact descriptor for image indexing and retrieval." International Conference on Computer Vision Systems. Springer Berlin Heidelberg, 2008.
- Amato G., Savino P., Magionami V. (2007) Image Indexing and Retrieval Using Visual Terms and Text-Like Weighting. In: Thanos C., Borri F., Candela L. (eds) Digital Libraries: Research and Development. Lecture Notes in Computer Science, vol 4877. Springer, Berlin, Heidelberg
- Amato G, Savino P (2008) Approximate similarity search in metric spaces using inverted files. In: InfoScale '08: proceedings of the 3rd international conference on scalable information systems, ICST, pp 110
- Squire, D., Muller, W., Muller, H., & Raki, J. (1998). Content-based query of image databases, inspirations from text retrieval: Inverted files, frequency-based weights and relevance feedback. (025.063), Genve.
- Messina, Alberto, et al. "Ants: A complete system for automatic news programme annotation based on multimodal analysis." Image Analysis for Multimedia Interactive Services, 2008. WIAMIS'08. Ninth International Workshop on. IEEE, 2008.
- Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on (pp. 248-255). IEEE.
- Bolettieri, P., Esuli, A., Falchi, F., Lucchese, C., Perego, R., Piccioli, T., & Rabitti, F. (2009). CoPhIR: a test collection for content-based image retrieval. arXiv preprint arXiv:0905.4627.
- Phillips, P. J., Moon, H., Rizvi, S. A., & Rauss, P. J. (2000). The FERET evaluation methodology for face-recognition algorithms. IEEE Transactions on pattern analysis and machine intelligence, 22(10), 1090-1104.
- 24. Mller, H., Mller, W., Squire, D. M., Marchand-Maillet, S., & Pun, T. (2001). Performance evaluation in content-based image retrieval: overview and proposals. Pattern Recognition Letters, 22(5), 593-601.