A Modern Web Interface for Medical Image Retrieval

Roger Schaer, Henning Müller

University of Applied Sciences Western Switzerland (HES-SO),
Techno-Pôle 3 · CH-3960 Sierre
roger.schaer@hevs.ch

Abstract. This paper discusses the need for retrieval systems and interfaces in today's rapidly growing medical imaging datasets. It then presents Shambala, a web-based front-end for an image retrieval system called ParaDISE (Parallel Distributed Image Search Engine). The Shambala application aims to provide a user-friendly interface for both novice and expert users and takes advantage of modern Web technologies to make the user experience interactive and engaging. The interface can be extended to other retrieval systems and is in the process of being integrated with alternative Human-computer interaction methods (such as motion sensors or speech recognition) to make it suitable for use in a sterile environment.

Keywords. Medical Image Retrieval; Search Interfaces; Web Applications;

1 Introduction

A vast amount of imaging data is produced daily in hospitals [1]. These images can contain valuable information for aiding medical professionals in diagnosis, treatment planning and research. There is therefore a need for enabling retrieval in these enormous image collections to better use the available knowledge.

In past years, several efforts have been made in order to provide textual retrieval as well as CBIR (**C**ontent-**B**ased **I**mage **R**etrieval) systems and search interfaces to users in the medical domain, such as the Khresmoi project [2], a multilingual, multimodal search system for biomedical information and documents. Other examples include GoldMiner[1] (which provides filtering based on imaging modality), RadMiner[2] (a search tool aimed towards radiologists for text and semantic information), OpenI[3] (allows search based on text and on image examples) as well as Yottalook[4], which supports searching across multiple types of sources.

This paper presents a novel web-based search interface called Shambala[5], which aims to be easy to use, interactive and modern.

The remainder of the paper is organized as follows. Section 2 describes the technologies used in the development of the Shambala interface, as well as the underlying retrieval system, called ParaDISE (**Para**llel **D**istributed **I**mage **S**earch **E**ngine), described in [3]. Section 3 presents the

---

[1] http://goldminer.arrs.org
[2] http://www.averbis.de/public/download/RadMiner_brochure_en.pdf
[3] http://openi.nlm.nih.gov/
[4] http://yottalook.com
[5] http://faster.hevs.ch:8080/shambala/

features of the Shambala interface and Section 3.4 concludes the paper and lists some future development ideas.

2 Methods

The core concept and the various technologies used in the development of the interface are presented in this section, as well as a description of the retrieval system (ParaDISE) that is utilized by Shambala.

2.1 Concept and Technologies

Shambala started out as a simple sandbox interface for easy testing of the ParaDISE system, but it evolved over time to provide more features and ease of use. It is built on the following principles:

- **Easy -** The application should be straight-forward to use and not overwhelm the user with many options and parameters. It should also be easy to deploy and not require the installation of new software such as an application server, a database management system, PHP (**P**HP: **H**ypertext **P**reprocessor), etc.
- **Interactive -** The application should provide the user with features such as drag'n'drop to make searching more engaging and similar to desktop interfaces. It should also react quickly to user input and refresh search results continuously and transparently as the user refines his query.
- **Modern -** The application should integrate modern Web technologies, like the HTML5 (**H**yper**T**ext **M**arkup **L**anguage **5**) File & History APIs (**A**pplication **P**rogramming **I**nterfaces), as well as HTML5's Local Storage feature.

With these principles in mind, Shambala was developed as a client-side only application, based entirely on HTML5 and Javascript. All interactions with the ParaDISE system (which can be hosted on a completely different server, as it is totally independent) use AJAX (**A**synchronous **J**avaScript **A**nd **X**ML) calls to the Global ParaDISE web service, described in section 2.3.

2.2 ParaDISE Backend

The ParaDISE system, used by the Shambala interface, allows indexing and retrieving images using visual features as well as text. The design of the system was based on three main aspects:

- **Modularity -** The system was developed as a series of independent components, increasing the flexibility of the system and making it easier to distribute.
- **Expandability -** Expandability deals with the need to be able to add new features, representations, distances etc. This aspect is obtained by using object-oriented programming and having a plugin organization of the component libraries. The system should also be easy to integrate with other components or user interfaces (the use of the standard HTTP (**H**yper**T**ext **T**ransfer **P**rotocol) transmission protocol makes this easy). The use of JSON (**J**ava**S**cript **O**bject **N**otation), a standardized and light-weight data transfer format allows for interoperability with other retrieval systems.
- **Scalability -** ParaDISE also aims to be a scalable system, both for offline operations (such as indexing a set of images) as well as online operations (searching for images, for

example). On the offline part, scalability is made possible by the use of Hadoop[6], a distributed computing framework developed by the Apache Software Foundation. Its use and benefits are discussed in [4]. For online optimizations, parallel computing and efficient indexing techniques for fast similarity search are used to cope with an ever-increasing amount of images.

The architecture of the ParaDISE backend is component-based. The main components of the backend are described below and shown in Figure 1:

- **Extractor -** Performs the extraction of local visual features from the image.
- **Descriptor -** creates a global representation of the image.
- **Storer -** accesses visual indices (read/write operations).
- **Fusor -** combines multiple result lists from various queries based on a given fusion rule. It can also be used for combining ParaDISE with other systems (advanced text search with semantic aspects, for example).

These four core components are integrated into two composite components (**Indexer** and **Seeker**) to perform the fundamental retrieval operations: indexing and similarity search (see Figure 1).

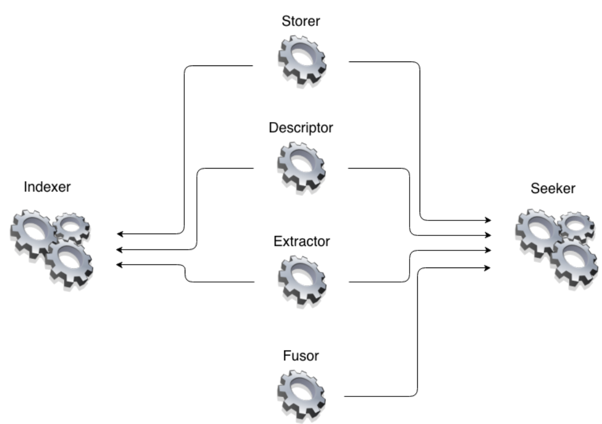

Figure 1 - Structure of the ParaDISE backend

2.3 ParaDISE Web Service Layer

The service layer of the ParaDISE system, built on top of the backend described in section 2.2, is made up of multiple web services that use a REST-style (**RE**presentational **S**tate **T**ransfer) architecture. It also provides additional features such as the text search.

Standard HTTP GET and POST requests are used to communicate with the web services, which simplifies interaction between the system and various client applications (Web-based or desktop applications that can be written in any language capable of making HTTP requests).

The various web services are listed and described below:

- **Caption web service -** Responsible for search by keyword. Uses the Lucene[7] search engine library.

---

[6] http://hadoop.apache.org/
[7] http://lucene.apache.org/

- **Visual web service -** Uses the visual information of the images in order to retrieve similar matches using CBIR techniques or index a new set of images. This service is backed by ParaDISE's Seeker and Indexer components.
- **Compound figure web service -** Used to retrieve links between compound figures (multiple images in a single block) and their children (obtained by separating the compound figure into its subparts), as well as getting the coordinates of each child within its parent image (see section 3.3 for more details and an example).
- **Fusion web service -** Responsible for combining results from different sources (text search, image search) into a single list (based on a given fusion rule). This service is backed by ParaDISE's Fusor component.
- **Global web service -** Provides a facade for client applications, which hides some of the complexity of calling the individual web services in succession by providing general methods such as "search images", which can use text, image examples, or both.

The structure of the web services is shown in Figure 2.

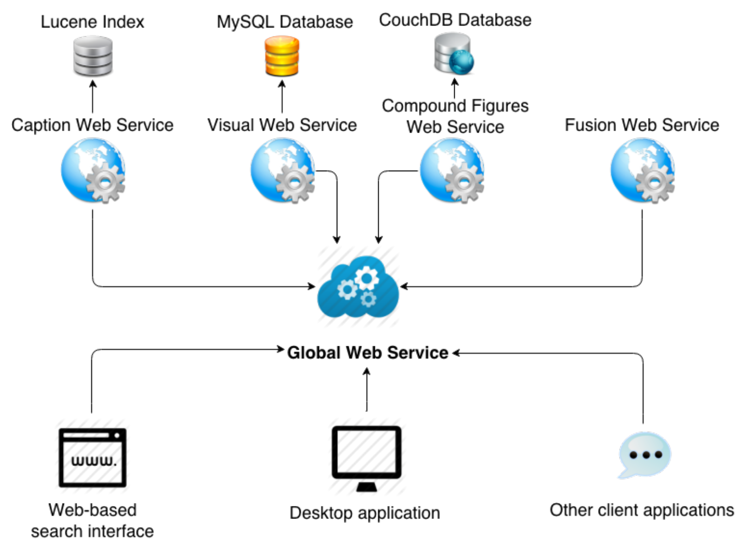

Figure 2 - Structure of the ParaDISE web service layer

2.4 Datasets

Currently, there are two datasets that can be accessed through the Shambala interface.

- **ImageCLEFmed 2012 dataset** - ImageCLEF[8] is a part of the Cross-Language Evaluation Forum (CLEF), an information retrieval competition. The medical task of ImageCLEF in 2012 used a dataset of over 300'000 images taken from 75'000 articles of the biomedical open access literature.
- **PubMedCentral dataset** - This dataset consists of the Open Access Subset provided by the PubMedCentral[9] digital repository, which contains over 1.5 million images from the biomedical literature. The status of PubMedCentral in 2013 was taken as the resource constantly increases its size.

[8] http://www.imageclef.org/
[9] http://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/

3 Results
This section details the features provided by Shambala and contains screenshots of the application to illustrate its use.

3.1 Interface Overview
An overview of the Shambala interface is shown below in Figure 3:

Figure 3 - Overview of the Shambala interface

The interface is split up into 5 main zones:

1. **Relevant zone** - The user can add relevant textual terms and image examples here.
2. **Irrelevant zone** - The user can filter out textual terms from the results and provide irrelevant image examples here.
3. **Results zone** - The search results are displayed here.
4. **Image sample zone** - This pull-up menu allows the user to get random sample images or upload his images that can be added to the relevant or irrelevant zones.
5. **Search option zone** - This pull-down menu allows the user to customize the search behavior.

These zones and the features they provide are described in more detail in the next subsections.

3.2 Relevant & Irrelevant Zones
On the left-hand and right-hand sides are containers for textual terms and image examples. The user can type terms he is interested in on the left and terms he wants to exclude on the right. The

queries can be simple words or more complex queries that use Lucene's query parser syntax[10], supporting wildcard searches, various Boolean operators, etc.

Additionally, the user can drag images from the **results** zone or from the **image samples** pull-up menu to the relevant or irrelevant container. By adding multiple images, he can refine his query to get more relevant results (using Rocchio relevance feedback [5]).

3.3 Result Zone

The result zone shows a grid of the returned search results, but also provides several interactions for the user. The zone is completely dynamic and will layout the images in a way that optimizes the number of shown results given the available vertical space. Additionally, the number of columns changes depending on the size of the browser window in order to accommodate devices with smaller screen resolutions (see Figure 4).

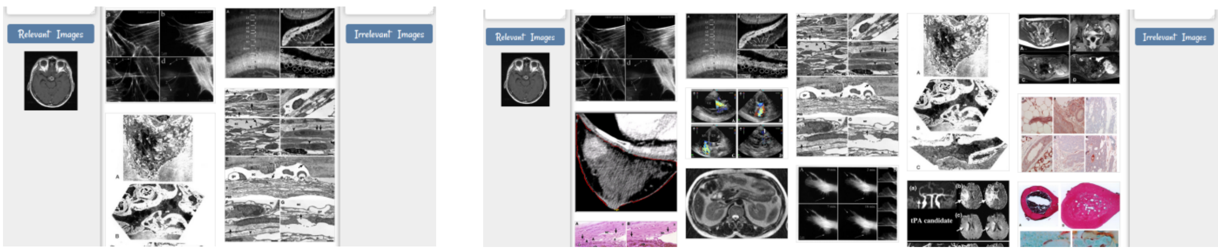

Figure 4 - Left: results zone on a small display (2 columns), Right: results zone on a large display (5 columns)

Furthermore, the user can click on each result to see a zoomed in view, the image caption and imaging modality (if detected automatically), along with several basic image editing tools that allow adjusting the image contrast/brightness (useful for radiology). The contrast and brightness can also be adjusted directly by dragging inside the image horizontally or vertically, as in radiology viewing stations. The detailed result view is shown in Figure 5.

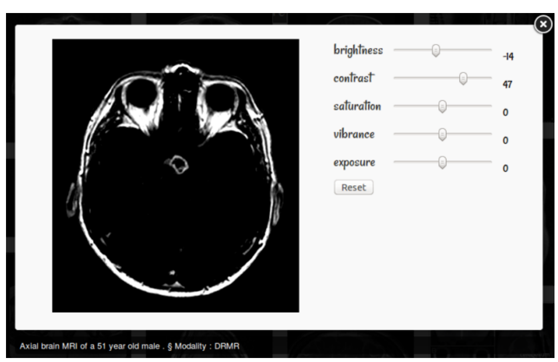

Figure 5 - Detailed view of a search result: caption and imaging modality at the bottom, image editing tools on the right

Finally, the interface treats compound figures, which are images containing multiple subfigures of the same or varying modality (mixed MRI and graphs, for example) and represent a large portion of images in the biomedical literature [6]. These compound figures are separated into their subcomponents by ParaDISE, using a technique described in [7]. All the information concerning the subfigure coordinates within the original parent image are stored and can be

---

[10] http://lucene.apache.org/core/2_9_4/queryparsersyntax.html

accessed, allowing the user to view a separated figure in its original context, as well as navigating between the multiple subfigures of a compound figure (see Figure 7). Subfigure results are highlighted in the result list with a visual cue (see Figure 6).

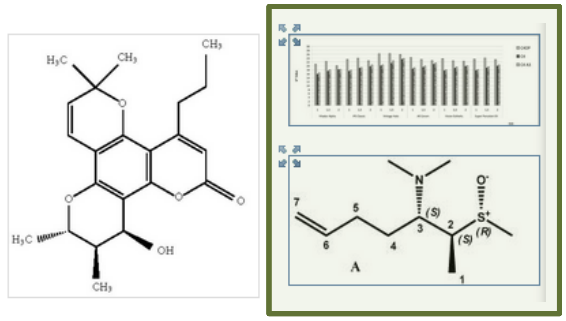

Figure 6 - Two subfigures shown in the result list: notice the blue border and icon in the top-left corner

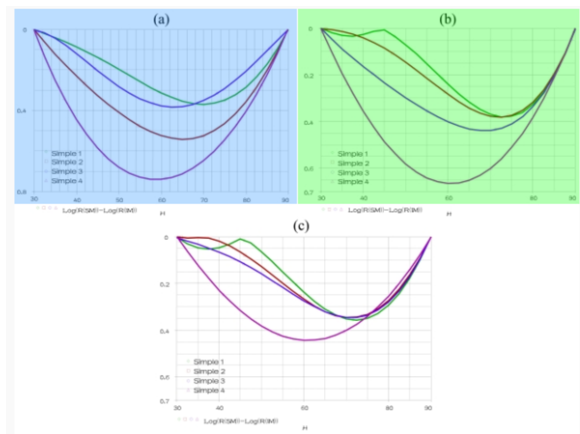

Figure 7 - Navigation interface for a compound figure: the selected subfigure from the results is highlighted in green, the subfigure where the mouse cursor is currently hovering is highlighted in blue

3.4 Image Sample Zone

The lower zone can be pulled up by clicking on the red bar at the bottom of the screen (shown in Figure 8). Two tabs are then shown: one allows the user to upload image examples from his computer and the other provides random image examples from the ImageCLEF2012 dataset.

Figure 8 - Image samples menu showing radiology image examples

The upload tab contains a drop zone for images, which can be dragged directly from the computer into the browser. Figure 9 shows the zone filled with a few uploaded images that can then be added to the relevant/irrelevant image containers.

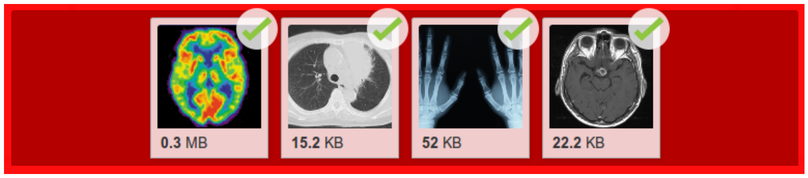

Figure 9 - Image upload drop zone with a few image examples

3.5 Search Option Zone

The upper option zone (Figure 10) can be pulled down by clicking on the red "Search options" tab. A few options are then presented to the user:

- **Index** - The user can switch between two available datasets described in Section 2.4.
- **Modalities** - The user can choose to filter the results by modality, either keeping all modalities, diagnostic images only, radiology images only or filter the results automatically based on the modalities of given relevant image examples.
- **Use image captions** - The user can set whether the captions of relevant/irrelevant image examples are also taken into account for the text search or not.
- **Use semantic terms** - Moreover, the user can choose to keep each caption as-is or to replace it with semantic terms extracted from the caption by an external web service.

Figure 10 - Search option panel

4 Conclusions and Future Work

Shambala is a modern and easy-to-use web interface for searching images. It takes advantage of trends in Web development, particularly HTML5 and its APIs. It currently functions in tandem with the ParaDISE retrieval system, but can easily be adapted to other systems that offer web services communicating with the JSON data exchange format. It has a minimalistic yet feature-rich user interface that helps engage users to discover and interact with the system. Plans for the future include more focused user tests to determine if the interface can help in reducing the time needed to find relevant information about a given pathology, as well as tightly integrating the interface with alternative Human-computer interaction tools such as motion sensors (ex: Leap Motion™[11], described in [8]) or speech recognition, in order to facilitate its use in a sterile clinical setting, such as an operating room.

5 Acknowledgments

This work was supported by the EU 7th Framework Program in the context of the Khresmoi project (FP7 grant 257528).

[11] https://www.leapmotion.com/

6 References

[1] C. Akgül, D. Rubin, S. Napel, C. Beaulieu, H. Greenspan and B. Acar, "Content--Based Image Retrieval in Radiology: Current Status and Future Directions," *Digital Imaging,* pp. 208-222, 2011.

[2] A. Hanbury, C. Boyer, M. Gschwandtner and H. Müller, "KHRESMOI: Towards a Multi--Lingual Search and Access System for Biomedical Information," Luxembourg, 2011.

[3] R. Schaer, D. Markonis and H. Müller, "Architecture and applications of the Parallel Distributed Image Search Engine (ParaDISE)," in *FoRESEE*, Stuttgart, Germany, submitted.

[4] D. Markonis, R. Schaer, I. Eggel, H. Müller and A. Depeursinge, "Using MapReduce for Large-scale Medical Image Analysis," in *HISB*, La Jolla, California, 2012.

[5] J. J. Rocchio, "Relevance Feedback in Information Retrieval," in *The SMART Retrieval System: Experiments in Automatic Document Processing*, Prentice-Hall Inc., 1971, pp. 313-323.

[6] H. Müller, A. Garcìa Seco de Herrera, J. Kalpathy-Cramer, D. Demner Fushman, S. Antani and I. Eggel, "Overview of the ImageCLEF 2012 Medical Image Retrieval and Classification Tasks," Rome, Italy, 2012.

[7] A. Chhatkuli, D. Markonis, A. Foncubierta-Rodríguez, F. Meriaudeau and H. Müller, "Separating Compound Figures in Journal Articles to allow for Subfigure Classification," in *SPIE Medical Imaging*, Orlando,FL,USA, 2013.

[8] A. Widmer, R. Schaer, D. Markonis and H. Müller, "Gesture interaction for content-based medical image retrieval," in *ICMR*, Glasgow, UK, 2014.