

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/279208617>

Content Based Image Retrieval for Large Medical Image Corpus

Conference Paper · June 2015

DOI: 10.1007/978-3-319-19644-2_59

CITATIONS

3

READS

308

4 authors:



Gjorgji Strezoski

University of Amsterdam

23 PUBLICATIONS 401 CITATIONS

SEE PROFILE



Dario Stojanovski

Ludwig-Maximilians-University of Munich

22 PUBLICATIONS 233 CITATIONS

SEE PROFILE



Ivica Dimitrovski

Ss. Cyril and Methodius University in Skopje

61 PUBLICATIONS 704 CITATIONS

SEE PROFILE



Gjorgji Madjarov

Ss. Cyril and Methodius University in Skopje

41 PUBLICATIONS 1,012 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



VISTORY [View project](#)

Content Based Image Retrieval for Large Medical Image Corpus

Gjorgji Strezoski¹, Dario Stojanovski¹, Ivica Dimitrovski¹, and Gjorgji Madjarov¹

Faculty of Computer Science and Engineering, Skopje, 1000, Macedonia,
strezoski.g@gmail.com, stojanovski.dario@gmail.com,
ivica.dimitrovski@finki.ukim.mk, gjorgji.madjarov@finki.ukim.mk
<http://www.finki.ukim.mk>

Abstract. In this paper we address the scalability issue when it comes to Content based image retrieval in large image archives in the medical domain. Throughout the text we focus on explaining how small changes in image representation, using existing technologies leads to impressive improvements when it comes to image indexing, search and retrieval duration. We used a combination of OpponentSIFT descriptors, Gaussian Mixture Models, Fisher kernel and Product quantization that is neatly packaged and ready for web integration. The CBIR feature of the system is demonstrated through a Python based web client with features like region of interest selection and local image upload.

Keywords: image processing, opponent SIFT, medical image retrieval, fisher vectors, PCA, product quantization

1 Introduction

As we explore new boundaries and create new technologies we thrive towards bettering our lives in a variety ways. These technological advances are highly dependent on the data that is constantly being collected from any field imaginable. From simple movie databases, to complex medical imaging, vast quantities of data is being stored and processed on daily basis. Throughout time very large collections of data have been generated which are hard to manage and extract information from, therefore an efficient technique of information retrieval is crucial.

In the medical domain, imaging plays a fundamental role in research, education, diagnostics and even treatment. Almost every type of medical condition can be in a way expressed through an image, so there is a lot of information in the image itself. The sole tangible nature of an image is probably the most important factor in choosing images over other types of data. Until recent years medical image retrieval systems have been text based, which is not that convenient especially because for the text based image retrieval system to work, all of the images that would be used need to be annotated. Having in mind that there are over 12400 different categories of medical conditions[1] for which images can

be collected, an efficient way of searching through this massive collection of data is also required. This presents a major problem for the medical community which grows over time and needs to be addressed.

Hospitals, clinics and other medical institutions have adopted technologies like Picture Archiving and Communication Systems (PACS) [2], Radiological Information Systems (RIS) [2] and Hospital Information Systems (HIS) [2]. The purpose of these systems is entering, storing and managing the patient data, but the sheer volume of data overwhelms the current information indexing and retrieval algorithms. As mentioned before, the text based nature of these systems limits their users to some kind of keyword based searches (image header keywords, patient ids, diagnosis ids. . .), which does not necessary capture the scope of the required query, some valuable visual characteristics that are proprietary to the image may be missing.

The CBIR systems create the possibility for searching these vast image collections, with visual queries that consist of sketches, textures, color characteristics, textures or cropped patches from existing images. Therefore the query gains contextual significance and captures those evading features that escaped the text based query approach.

In this paper we present a complete system for medical image retrieval that uses state-of-the-art techniques for feature representation, indexing and querying a large image archives. The remainder of this paper is organized as follows. Section 2 presents related systems and solutions in the CBIR field. Section 3 elaborates our system's architecture and implementation in detail, while section 4 describes the dataset, results and the experimental setup. Finally in section 5 we expressed our future intentions regarding this system and we gave our concluding thoughts.

2 Related work

The earliest text-based image retrieval systems can be traced back to the 1970s [3] so we can safely say that image retrieval systems have always been a very popular research topic. So far text based search has proven satisfactory results and is widely used today. There are a lot of systems that operate in the Medical field that use text based retrieval.

One of those systems is Goldminer¹ which retrieves images by looking at figure captions of journal articles from the Radiological Society of North America (RSNA). The idea of this system is that the keywords of the captions can be extracted and then mapped to UMLS concepts.

Yale have created the Yale Image Finder² (YIF) which through optical character recognition recognizes text in the image. The recognized text is then combined with features from captions, titles and abstracts from medical journal articles.

¹ <http://goldminer.arrs.org/home.php>

² <http://krauthammerlab.med.yale.edu/imagefinder/>

Stathopoulos et al. make a field-based representation of the images and index it using Lucene³. In the retrieval phase they add different weights to the fields of representation based on the part of the article they were extracted from.

All of the above systems rely on text based features to retrieve a list of relevant images to each query. But in medical imaging a more precise type of retrieval is required, especially because not all annotations can be interpreted in a same way. We need to retrieve images with relevant visual features to those of the query image. These visual features at specific points in each image can be described with image descriptors. Initially content based retrieval systems implemented a more global approach on image description using global descriptors like color histograms and texture values. Thus, results have shown that global descriptors do not apply well in medical image retrieval. This is because the key feature of the image, the segment of the image with the most clinically relevant information is usually located in a very small area and global descriptors fail to grasp this detail. Eakins [2] has divided image features into three levels:

Level 1 - This level deals with primitive features like color, texture, shape or some spatial information about the objects in the picture. This way we can filter images on a more global scale based on form or color. This can be used for finding images that are visually similar to the query image.

Level 2 - This level introduces the logical features or derived attributes which involve some degree of inference about the identity of the objects depicted in the image. So a typical query in a medical scope would be "Find images of a kidney"

Level 3 - Most complex of all levels, as it requires complex reasoning about the significance of the objects depicted. In this case the query would look like "Find image of an infected kidney"

Most of the developed systems offer us level 1 retrieval. There are some systems in an experimental faze that allow level 2 retrieval but there are none level 3 systems so far. Recent systems for content based image retrieval in the medical domain are ASSERT system for high resolution computed tomography (CT) images of lungs, the Flexible Image Retrieval Engine (FIRE) which is used for a wide variety of medical images and the Image Retrieval for Medical Applications (IRMA) system that classifies images in viewpoints, modalities and anatomical areas.

The ASSERT system lets users extract valuable pathological data from lung images in specific pathology-bearing regions. The images are described by their texture, gray-scale and shape attributes.

All of the above mentioned systems have the common fault of low scalability. Searching a large image database is in it self a slow process, combined with the fact that biomedical image archives grow all the time, scalability is becoming a great concern. In our system the major improvement is in fact the scalability factor. We implement a very efficient image querying method that is almost invariant to the size of the data-set.

³ <http://lucene.apache.org/>

3 Content based image retrieval system for large biomedical archives

Our system can be logically separated into two phases: an online phase and an offline phase. The offline phase is consisted of various methods of feature extraction, dimension reduction and quantization that increase system efficiency, while the online phase is basically a image search engine. Also because our system effectively powers a web client, the online phase can also be divided into two functional segments: a front end and a back end. The back end is basically a real time, slightly modified, on-call offline phase, while the front end delivers a user friendly interface. Once an image query is uploaded to the system, it immediately passes through the whole pipeline and a suitable representation is generated. Then the search is performed using a pre-indexed structure from the image archive in question. The results for the uploaded query image are then ordered by similarity and returned to the web client as a list from which image paths and relevant medical articles are constructed. Every query gets up to a 1000 result images. The similarity coefficient is also included in the result display.

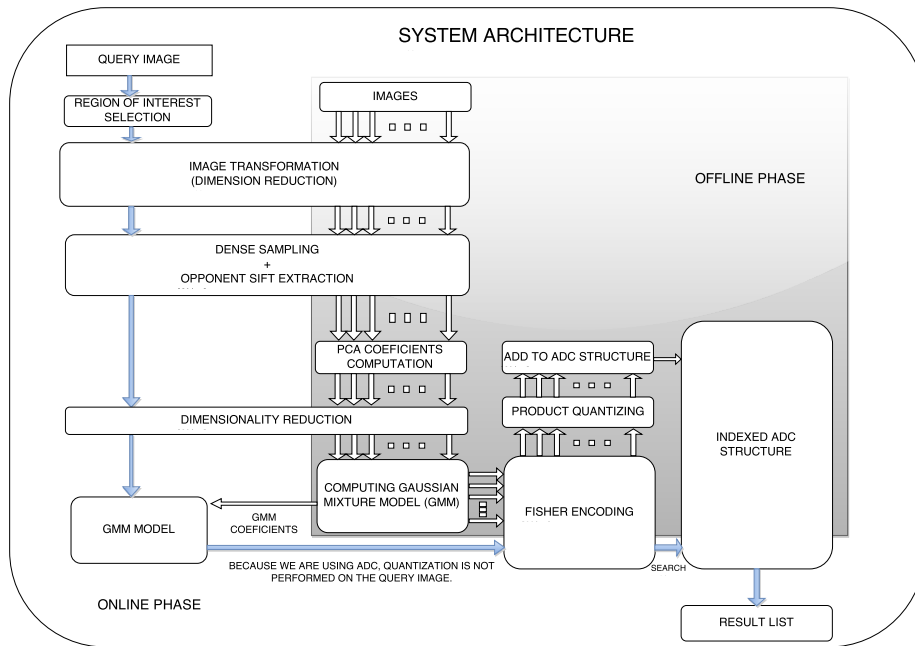


Fig. 1. Full system architecture

3.1 Offline phase

As mentioned above, the offline phase carries the whole system logic. It contains the image transformation, feature extraction, dimension reduction, distribution analysis, representation altering, product quantization and search functions. Each of these steps is vital to the end result and will be thoroughly described in the sections below. First the images from the training set are fed to the system because they are a fair representation of the whole image archive and a viable feature distributions can be extracted from them. With the help of the feature distributions we apply the Fisher vector encoding over the generated descriptors, resulting in one vector per picture representation of the image descriptors. Fisher vectors help us to significantly reduce the size of the information kept for each image and thus speed up search and indexing in the latter part of the process. Once the data analysis is complete, and all required coefficients for Fisher vector generation are retrieved we generated Fisher vectors on the whole 305 638 image set. We then fed the whole 305 638 image set once to the Product Quantization (PQ) pipeline extracted the structures needed for quick retrieval. Once we extracted the PQ structures the system was ready for integration with the online client for CBIR.

Image Transformation In order to improve the scalability of the system, the images are preprocessed before feature extraction and indexing is performed. As illustrated on Figure 1, the first step in the image processing pipeline is a physical transformation of the image itself. The transformation consists of reducing the dimensions of the image to the x768 dimension while keeping the aspect ratio. The aspect ratio is important so that the image does not appear warped and distorted upon processing. Also sometimes valuable information can be lost if the aspect ratio is not kept intact. This transformation of the images is done via the djpeg package from the Ubuntu software repositories. After the scaling is complete, the image continues to be processed in Portable Pixel Map (PPM) or jpeg format.

Both images from the training set, indexing set and the user uploads undergo this process so that the consistency of the data is preserved.

OpponentSIFT Van de Sande et al. (2010) [4] performed a study of the invariance properties of different color representations under different types of illumination transformations, encompassing light intensity changes, light intensity shifts, light color changes and light color shifts. We chose this descriptor over the regular grey-scale SIFT descriptor because even though SIFT is invariant to rotation, translation or scaling it does not capture the color features of the image. OpponentSIFT combines the best of both worlds in this case, it keeps the spatial invariance and adds color information. In medical imaging often times color can be the key semantic difference between two images (Example: infected vs healthy) so it was very important to capture this information.

The OpponentSIFT descriptor is 384 dimensional. This means that 384 values describe each point of interest or key point detected on the image. There are 128

values describing each of the opponent color channels - red vs green, blue vs yellow and black vs white. We used this descriptor in combination with dense sampling in order to extract as much information from the images as possible. All images were sampled with a window of 6x6 pixels on one scale and we saved the descriptors in a binary format proprietary for the ColorDescriptor tool we used for extraction.

Dimension reduction Once the opponent SIFT features are extracted, we have a matrix in which every key-point in the image is described by a 384 dimensional vector. Because of the dense extraction, the number of key-points is directly related to the size of the image and the step of key-point extraction. In our case the dimension of the image descriptors (384) is not optimal for working with a large number of images. The resource consumption would be too high and the performance will suffer. That is why we reduce the dimension of the descriptors by half using Principal Component Analysis (PCA). Principal component analysis (PCA) is a standard tool [5] for dimensionality reduction: the eigen vectors associated with the most energetic eigenvalues of the empirical vector covariance matrix are used to define a matrix M mapping a vector $x \in R^{128}$ to a vector $x = M_x$ in a lower-dimensional space [6].

Dimension reduction using PCA is a two step process. First random key-points from each image descriptor file are extracted and concatenated in a matrix. It is computationally inefficient to use all key-points from every image, because we are interested in a general distribution of features and that can be accomplished with as low as 800 key-points per image (depending on the image size). This gives us an object on which a suitable distribution analysis can be performed. The analysis starts with mean extraction from each of the 384 descriptor vectors for each randomly selected key-point. Every mean vector is then added to a matrix that will further be used in the dimension reduction process. After the mean extraction is complete, a principal component for each descriptor vector is also computed. The principal components actually represent the coefficients by which we multiply the original 384 dimensional descriptor in order to reduce its size. The multiplication of these matrices is the actual dimension reduction part. There are 384 principal components generated for each of the 384 descriptor vectors. We can choose dimension of the resulting descriptor by limiting the number of principal component used in the multiplication.

In our case we reduce the descriptors dimension in half, so we multiply by a 192x384 dimensional PCA matrix. This kind of dimension reduction can be performed on the resulting fisher vector representation of the images, but the trade off between vector dimension and accuracy does not favor accuracy.

Fisher vector representation The Fisher kernel combines all the benefits of generative and discriminative approaches when it comes to transforming an incoming variable size set of independent samples into a fixed sized vector representation [7][8]. The only assumption that has to be made during this process is that the samples in question follow some kind of parametric generative model

estimated on a training set. In the world of image classification and retrieval the Fisher kernel extends the popular Bag of Visual Words (BOV) and has shown to improve state-of-the-art results. Because of the convenient size of the end representation the Fisher kernel is also quite suited for large scale image classification and retrieval. Having this in mind we chose the Fisher kernel as a suitable representation for our data.

Another advantage of the Fisher kernel is that it naturally gives a similarity measure that takes into account the underlying probability distribution. For example in our case we use each image of the indexed set to generate a Gaussian Mixture Model (GMM). This way we extract weights of the mixture elements, number of centroids and the diagonal of the covariance matrix. After this it is easy to calculate how much a new data entry would stretch the parameters of the trained model. It also gives the direction in parameter space into which the learned distribution should be modified to better fit the observed data. This representation in comparison with the BOF representation shows that fewer visual words are required by this more sophisticated representation.

Product quantization This section addresses the problem of coding an image vector. Given a D-dimensional input vector, we want to produce a code of B bits encoding the image representation, such that the nearest neighbors of a (non-encoded) query vector can be efficiently searched in a set of n encoded database vectors. With our vectors we have 12288 (192x64) dimensions and we encode them into 8 bit codes.

The whole procedure takes place in two steps, first a suitable projection that reduces dimensionality of the vector is generated and second, quantization is used to index the resulting vectors. The best choice of action for this problem with large image corpora are approximate nearest neighbor search methods. Most of the popular approximate nearest neighbor search techniques require several hash tables to be present in memory all the time. With a body of images this size, that is not efficient. So we chose the Spectral Hashing [9] [10] method which embeds the vector into binary space and better adapts to memory requirements. Spectral hashing, similarly to Semantic Hashing, has each image in the data-set represented by a compact binary code. Similar images in the data-set have similar codewords, and similarity is measured by Hamming distance.

This method has several approaches, from which we chose the asymmetric distance computation (ADC) approach. Jegou et al. [11] have provided a compact coding scheme which is combined with an inverted file system and the ADC approach significantly improves results in terms of the trade-off between search quality and memory usage. Proprietary for this method is the fact that it encodes all the vectors in the database but it does not encode the query vector. So let $x \in \mathcal{R}^d$ be the query vector for which we need to find the nearest neighbors $NN(x)$ in the data-set $Y = \{y_1, \dots, y_n\}$. The ADC approach consists in encoding each vector y_i by a quantized version $c_i = q(y_i)$. For the quantization we use a quantizer $q(\cdot)$ of k centroids [12]. This means that the vector will be encoded by

$\log_2(k)$ bits, where k is a power of 2. After the quantization is complete, finding the a nearest neighbors of x is just a matter of computing this simple equation:

$$NN_a = a - \arg \min \|x - q(y_i)\|^2$$

From this equation we also notice that because there is no quantization of the query vector, there exists no approximation error on the query side. The ADC search function is called every time a query is made. So it is very important that the function duration is as short as possible.

3.2 Online phase

The online phase of this pipeline creates a user friendly environment for querying the CBIR system. The pipeline changes track after the indexing of the large data-set so the new query images are processed much faster. This is due to the pre indexed structures and distributions that do not need to be recomputed for each query. The Gaussian mixture model is constructed on the indexed data set, the PCA coefficients required for dimensionality reduction are also precomputed with the indexed set images.

When it comes to the Product Quantization we also have a significant speed up. Because we use the Asymmetric distance computation method we have no quantization on the query vector, so no additional time is spent. The only time consuming procedure that is left in the whole query process is the generation of descriptors for the query image. In an effort to reduce descriptor size we added a region of interest tool in the web endpoint. This way users can select only the part of the image that they want to search thus effectively reducing image size.

3.3 Implementation

Back end The back end of the online phase of the system is powered by Python⁴ and MATLAB. We use the Flask⁵ micro-framework for Python in order to speed up development. Also, with speed in mind, all of the MATLAB functions required for query image processing are precompiled and permanently added to the MATLAB work-path.

Front end The front end is constructed with the Bootstrap⁶ 3 framework. It features content based, text based and mixed retrieval. On this endpoint a user can upload the query image via drag and drop or regular upload from a file browser. Additionally a region of interest tool is added in order to effectively reduce input size and increase effectiveness. Research has shown that other than improving efficiency, cropping the query image to only the region of interest generates better retrieval results [13].

⁴ <https://www.python.org/>

⁵ <http://flask.pocoo.org/>

⁶ <http://getbootstrap.com/>

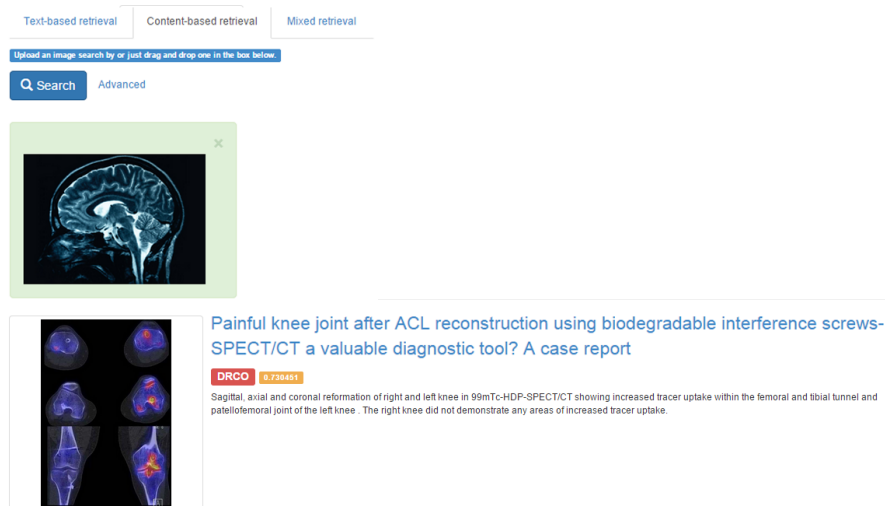


Fig. 2. Web client interface

Figure 2 illustrates the front end of our CBIR. In the upper left corner there is a region for uploading a query image. Below the query segment there is a vertical list with resulting images accompanied by relevant medical articles altogether.

4 Experiments and results

4.1 Dataset

Our system performs retrieval over the ImageCLEFmed 2013 data-set. This data-set comprises a large subset (over 300 000 images from over 45 000 different biomedical research articles) of the PubMed Central (R) repository hosted by the U.S. National library of Medicine. From this subset over 5000 images are extracted and manually annotated with applicable categories. This dataset has been used in various tasks for modality classification and image retrieval as a part of the ImageCLEF⁷ challenge.

4.2 Experimental setup

For our experiments we used a HP Z800 Workstation with a 4 Core Intel Xeon processor E5620 that runs on 2.40 GHz with 12MB cache. Our workstation is also equipped with 24 GBytes 1066 MHz RAM memory and a hard disk drive of 1 TB. All of the experiments were run on a 64 bit Debian 3.16.3-2 operating system, codename Jessie, running MATLAB⁸ 2014 R8. The computationally

⁷ <http://www.imageclef.org/>

⁸ <http://www.mathworks.com/products/matlab/>

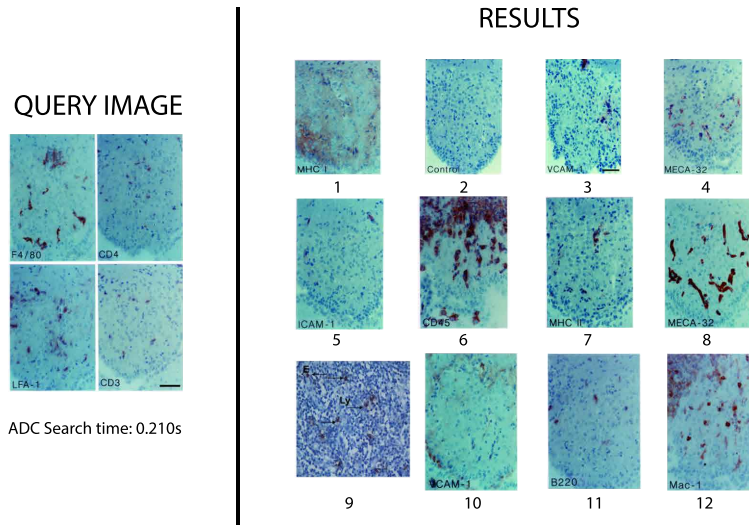


Fig. 3. Query results

demanding functions were optimized on the lowest possible level by using Inria’s YAEL⁹ library for MATLAB. As mentioned above in the text, feature extraction was performed using Van de Sande’s ColorDescriptor software.

4.3 Results and Discussion

In terms of memory consumption, results show that a structure which sizes up in Gigabytes can be effectively represented in a structure of just few tens of Megabytes. Table 2 shows a comparison between the combined set size before and after the product quantization. The first entry illustrates the size of the descriptors in the training set with PCA dimension reduction performed while the latter two entries show the suitable representations for the entire data-set after product quantization. This extreme shrinkage is due to substituting the vector representation of the images with a suitable code. Table 1 shows an example image descriptor file before and after the PCA dimension reduction has been performed. The PCA function is implemented in the Yael library for MATLAB. This part of the process is crucial because it speeds up the learning and search operations and reduces memory consumption from descriptor storage thus improving system scalability multiple times.

The first two entries in Table 2 relates to the size of the image training set before and after PCA dimension reduction accordingly. There is a four fold dimension difference between the PCA reduced and the former, original size dataset.

⁹ <https://gforge.inria.fr/projects/yael/>

Table 1. Descriptor file dimensions and size

File name	Dimensions	Dimension (after PCA)	Size	Size (after PCA)
image1.siftgeo	16250x384	16250x192	5500 Kb	1370 Kb
image2.siftgeo	9250x384	9250x192	3700 Kb	820 Kb

Table 2. Data structures that contain complete a dataset representation

File	Size (Mb)
Training-image-set.fvecs	143 812
Training-image-set-[PCA].fvecs	34 816
PQ-structure.mat	70
ADC-structure.fvecs	11

Table 3 shows the time it takes for the ADC algorithm to learn the structure and encode all the entries of the image archive over which the retrieval is performed. These functions are only executed once, after the fisher vectors for all the image in the archive have been generated. In the current data set there are 306538 images, hence 306538 fisher vectors. The algorithm takes 0.000393 seconds per image while learning and 0.000513 seconds while encoding the images. Having in mind that these function are executed only once, the time of execution is acceptable in even larger applications.

Table 3. ADC structure generation duration

Function	Execution time
ADC Learn	120.600 s
ADC Encode	157.300 s

5 Conclusion and Future work

In this paper we explained a content based image retrieval system which applies dimensionality reduction and product quantization methods in order to get a better trade-off ratio between search time and memory consumption. This better trade-off means greater scalability for the system. According to our results we can conclude that Product Quantization of Fisher vectors in content based image retrieval significantly improves the scalability and timing factors. The current state our system does not support multi-image queries. This is an important feature because sometimes medical conditions can not be expressed through a single image. The implementation of this feature is a near future goal and in the meantime we will continue to work on improving the system and report even faster search times.

Table 4. ADC search for 1 vector in a database of 306538

Function	Execution time
ADC Search	0.320 s

6 Acknowledgments

We would like to acknowledge the support of the European Commission through the project MAESTRA - Learning from Massive, Incompletely annotated, and Structured Data (Grant number ICT-2013-612944).

References

1. Dye, C., Reeder, J.C., Terry, R.F.: Research for universal health coverage. World Health Organization (2013)
2. Hwang, K.H., Lee, H., Choi, D.: Medical image retrieval: past and present. *Health-care informatics research* **18**(1) (2012) 3–9
3. Long, F., Zhang, H., Feng, D.: Fundamentals of content-based image retrieval
4. van de Sande, K.E.A., Gevers, T., Snoek, C.G.M.: Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **32**(9) (2010) 1582–1596
5. Bishop, C.M.: *Pattern Recognition and Machine Learning (Information Science and Statistics)*. 1 edn. Springer (2007)
6. Jégou, H., Perronnin, F., Douze, M., Sánchez, J., Pérez, P., Schmid, C.: Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2011)
7. Douze, M., Ramisa, A., Schmid, C.: Combining attributes and Fisher vectors for efficient image retrieval. In: *CVPR 2011 - IEEE Conference on Computer Vision & Pattern Recognition*, Colorado Springs, United States, IEEE (June 2011) 745–752
8. Sewell, M., Sewell, M.: The fisher kernel: A brief review (2011)
9. Yair Weiss, Antonio Torralba, R.F.: Spectral hashing. In Weiss, Y., Schölkopf, B., Platt, J., eds.: *In NIPS*. (2008)
10. Weiss, Y., Torralba, A., Fergus, R.: Spectral hashing. In: *Advances in neural information processing systems*. (2009) 1753–1760
11. Jegou, H., Douze, M., Schmid, C.: Product quantization for nearest neighbor search. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **33**(1) (2011) 117–128
12. Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. (jun 2010)
13. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge (2014)

..