



# A framework and baseline results for the CLEF medical automatic annotation task

Mark O. Güld\*, Christian Thies, Benedikt Fischer, Thomas M. Deserno

Department of Medical Informatics, RWTH Aachen University, Pauwelsstraße 30, 52057 Aachen, Germany

## ARTICLE INFO

### Article history:

Available online 6 June 2008

### Keywords:

Medical image retrieval  
Content-based image retrieval system  
Classifier combination  
Image processing  
Image database

## ABSTRACT

The medical automatic annotation task issued by the cross language evaluation forum (CLEF) aims at a fair comparison of state-of-the art algorithms for medical content-based image retrieval (CBIR). The contribution of this work is twofold: at first, a logical decomposition of the CBIR task is presented, and key elements to support the relevant steps are identified: (i) implementation of algorithms for feature extraction, feature comparison, and classifier combination, (ii) visualization of extracted features and retrieval results, (iii) generic evaluation of retrieval algorithms, and (iv) optimization of the parameters for the retrieval algorithms and their combination. Data structures and tools to address these key elements are integrated into an existing framework for image retrieval in medical applications (IRMA). Secondly, baseline results for the CLEF annotation tasks 2005–2007 are provided applying the IRMA framework, where global features and corresponding distance measures are combined within a nearest neighbor approach. Using identical classifier parameters and combination weights for each year shows that the task difficulty decreases over the years. The declining rank of the baseline submission also indicates the overall advances in CBIR concepts. Furthermore, a rough comparison between participants who submitted in only one of the years becomes possible.

© 2008 Elsevier B.V. All rights reserved.

## 1. Introduction

In 2005, a medical automatic annotation task (MAAT) was introduced by the cross language evaluation forum (CLEF) as part of its annual retrieval challenges. It requires the non-interactive classification of images into categories based on a multi-axial, hierarchical code (Deselaers et al., 2008). To face the challenge of optimizing algorithms and parameters, a storage concept is required not only to hold the images (CLEF MAAT 2007 consists of 12,000 images), but also to organize the experiments. Existing approaches like the GNU image finding tool (GIFT<sup>1</sup>), or medGIFT (Hidki et al., in press) as a specialized version for the medical domain, provide such a framework for the development and application of retrieval algorithms. In particular, GIFT integrates user-implemented content descriptors, works on files, and stores extracted features in inverted files for fast retrieval. However, the algorithm model is rather strict, as the feature extraction works isolated per image, and the system does not focus on development issues like common pre-processing steps, access to intermediate features, or modularization.

The image retrieval in medical applications (IRMA) project has two main goals (Lehmann et al., 2004): on the system side, a framework is implemented which supports the development and execution of retrieval algorithms. It provides a database for storing feature data as well as definitions of algorithms, a runtime environment for their execution, and user interfaces for accessing the system functionality, e.g., interactive retrieval.<sup>2</sup> On the retrieval side, it aims to implement, evaluate and verify a multi-step approach for the abstraction process for image content. This process models image content on different levels and therefore uses a wide range of features: global features for the coarse classification of the images, local (per-pixel) features as a basis for segmentation, and a hierarchical data structure for modelling the relationship between image regions and object identification (structural features for scene description).

### 1.1. IRMA database model

The IRMA framework stores content descriptions inside a generic feature container (Güld et al., 2007), along with type information. A central database keeps track of their storage location and their generation history. The system supports both interactive and non-interactive retrieval algorithms. Algorithms are integrated as *methods*, which are implemented as functions in the C++ programming language and transform a tuple of input features into

\* Corresponding author. Fax: +49 241803388793.

E-mail addresses: [mgueld@mi.rwth-aachen.de](mailto:mgueld@mi.rwth-aachen.de) (M.O. Güld), [cthies@mi.rwth-aachen.de](mailto:cthies@mi.rwth-aachen.de) (C. Thies), [bfischer@mi.rwth-aachen.de](mailto:bfischer@mi.rwth-aachen.de) (B. Fischer), [deserno@ieee.org](mailto:deserno@ieee.org) (T.M. Deserno).

URL: <http://irma-project.org> (M.O. Güld).

<sup>1</sup> <http://www.gnu.org/software/gift>.

<sup>2</sup> The system is still in development and therefore not publicly available.

a tuple of output features. In addition to processing a set of single images independently from each other, IRMA also models transformations which compute features from a set of features, e.g., a principal component analysis (PCA). The method interfaces use the feature type specifications to enable compatibility checks when interconnecting methods. *Networks* are used to combine method instances into more complex retrieval algorithms by defining the propagation of features between the methods. *Experiments* are used to store a partial parameterization of networks by assigning parameters to some of the network sources. Thus, a network can be initialized for a specific task, leaving only certain sources unassigned. Such experiments can be used for retrieval, hiding internals from the user and requiring him only to specify relevant information at query time, e.g., the set of images to be searched and the query image.

### 1.2. Runtime environment

At runtime, a central scheduling service determines which methods inside the parameterized network have a complete tuple of input features and are therefore ready to run. The central service dispatches pending method calls to a set of daemons running on other computers inside the local area network. The daemon dynamically loads the method function, runs it for the given input features, stores the output and reports back to the scheduler. Feature storage is handled transparently to the method. Using the generation history of each feature inside the database, already completed method calls are identified and skipped by the scheduler. The daemons run as background processes on stock-house PC hardware (GNU/Linux) in a workstation pool of roughly a dozen machines. Early experiments showed results (Güld et al., 2007) that were mostly limited by the central database and the central file server hosting the image data.

### 1.3. User interfaces

All user interfaces are running on a web-server and are therefore accessed via a web browser. A PHP class library for commonly used widgets and application steps (user accounting, preferences, consistent look & feel) was built using the Smarty template engine.<sup>3</sup> Current interactive query functionality is based on the query by example (QBE) paradigm (Smeulders et al., 2000).

Through these components, the IRMA framework can be used to develop, run, and deploy a variety of image processing algorithms. Therefore, CLEF MAAT is regarded as a use-case for the platform. While the storage requirements are addressed by the framework, the method concept does not impose any constraints on the design and implementation of the CBIR algorithms to be employed for CLEF MAAT. To address the problems of parameter optimization, common data structures must be developed to use algorithm-independent tools for the evaluation and inspection of retrieval results. Beside common algorithm steps (like image pre-processing), the combination of CBIR algorithms as a whole must be supported efficiently. This includes both the developer's side and the user's side during the implementation and the application of the algorithms.

## 2. Methods

By identifying the elementary steps of the CLEF MAAT (and CBIR in general), a logical layer above the layer of the framework components is introduced. The main challenge with algorithms for image analysis and categorization is their parameterization for

optimal results. Furthermore, an early result observed in many experiments, e.g., the ImageCLEF2004med retrieval challenge (Thies et al., 2005), is the improvement of results if a combination of classifiers is used. Fig. 1 shows a decomposition of CLEF MAAT into logical processing steps. The input level consists of the images, the ground truth, i.e. content information for the images, and the parameters for the classification algorithms. On the computation level, the retrieval system performs the extraction of features, their comparison, and their combination. The iterative optimization of the parameters for the retrieval algorithm is the essential goal during the annotation task. For this purpose, the automated evaluation of classification results and the visual inspection of results is essential, which is modeled on the inspection layer.

### 2.1. Feature extraction, comparison, and combination

The concept of methods in the IRMA framework addresses the non-interactive steps of feature extraction, classification, and combination. However, the decomposition of algorithms for CLEF MAAT is not constrained by the framework. Data structures and appropriate interfaces must be defined between the logical layers described in Fig. 1 to support a generic (i.e. algorithm-independent) feedback loop into the query interface.

In IRMA, this is implemented via feature types, which standardize the data exchange by documenting the layout inside the generic feature container. For this purpose, several types are defined:

- vectors of floating point numbers, which suit most content descriptors,
- result matrices, which store distances or similarity scores for pairs of vectors, and
- query results, which are nearest neighbors, a list of feature identifiers and their distances or similarity scores for a certain query image.

All classifiers based on pairwise element comparisons generate a result matrix as their output feature. While still present as features in the database, result matrices themselves are stored outside the database due to their size, and are self-contained, i.e. they also contain the feature identifiers for the two sets (references and samples). The matrices can also be combined via a generic method, if they refer to the same sets of references and samples. The combination is performed as a post-processing step and yields a new result matrix which can be evaluated as well. For each sample, the matrix values are first converted into normalized distances. The resulting entry in the combination matrix is the weighted sum of the normalized distances from each single result matrix.

Since each method definition in the database contains information about the feature interface for this method, the creation of networks is possible without knowledge about method internals (Güld et al., 2007). For this purpose, the IRMA network builder was established as a visual environment above the source code level. The GUI for composing networks from methods works similar to the modelling process in the Cantata visual programming environment of the Khoros image processing system (Argiro et al., 2001). Implemented in C++ and using the Qt library, it reads all method definitions from the database and allows the user to add method instances, sources and sinks on a graphical work plane. For each method, its textual description is available, which is part of the method entity stored in the database. The user defines the propagation of generated features by connecting method outputs to method inputs. The network builder uses each method's feature interface definition to allow only connections of compatible feature types. Source and sink nodes are used to define the interface of the algorithm. Resulting networks can be exported or uploaded into the system.

<sup>3</sup> <http://smarty.net>.

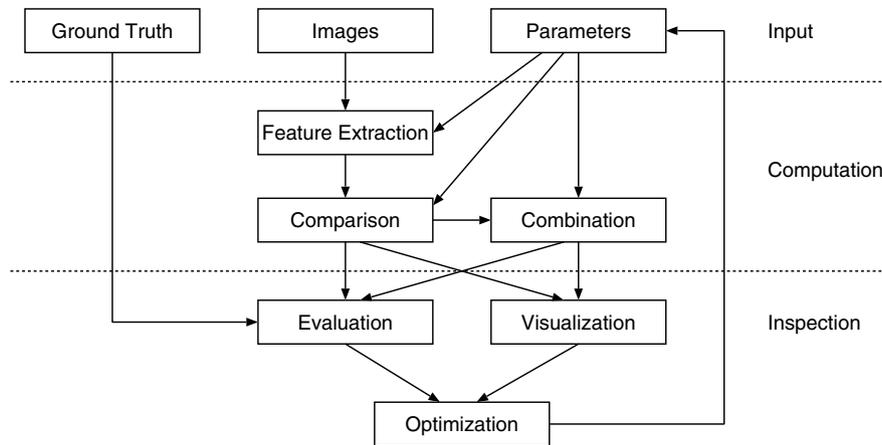


Fig. 1. Input data and application steps for image classification.

## 2.2. Visualization

During the development of retrieval algorithms, a visual verification of the implemented methods is required. The query interface only visualizes the final result of the algorithm. To display any feature stored inside the system, the IRMA feature browser is implemented in PHP as a web-based application, which directly accesses the feature storage of the database. It contains a type specific visualization function to display the contents of a feature container's content, and enables the navigation to preceding and succeeding features for the currently inspected feature by accessing the feature generation information.

This requires an efficient access to each feature stored inside the database, which relies on the information about its creation. In IRMA, the properties of method call are automatically stored along with each generated feature during the execution of method calls inside networks. The container for the feature values is generic and can hold numerical and textual data, feature IDs for reference purposes, as well as files.

## 2.3. Evaluation

The evaluation of the retrieval algorithms is based on result matrices. An evaluation tool is required that implements a nearest neighbor classifier by determining the closest reference images for each test image and maps the given category information onto these references, obtaining a decision for the unknown test image. The result matrices contain a flag that tells the evaluation tool to interpret the contained values as either distances or scores. The evaluation tool generates a graphical report which contains the confusion matrix and a list of the neighbor images for each sample image. By iteration over all samples in the result matrix, an error rate is obtained.

While the feature browser provides complete access to individual features with no particular application context, the visualization of the used features is also an important element for evaluating query results. In Deserno et al. (2007), this information is referred to as *relevance facts*, since the visualization of features helps to verify algorithm correctness and assess parameter settings. This step requires the simultaneous identification of generated features for a set of images or original features (e.g., when viewing the nearest neighbors for a query image), based on a common processing path. The identification of a particular intermediate feature is a costly operation, because it requires path searching in the feature generation information. To circumvent this, and to enable the simultaneous identification of semantically

identical features for a whole set of images, an additional data structure is used. Initial features, i.e. features which were imported and were not generated by a method, are assigned to a *root context*. In addition to the information about method calls for particular features, the calls are also logged with respect to the contexts of the input features, i.e. calling a method for features linked to a context results in the creation of new contexts (per method output). The generated features are then assigned to the new respective contexts in the database. A component implemented within the PHP web interface framework uses the context data structure from the database: it changes the iconic views (thumbnails) of the original images to the thumbnails of the features from the selected context. Thus, a selection like context *canny* (*scale* (ROOT, 256 × 256), 2.5, 0.5, 0.9) for a set of images can be processed efficiently on the context data structure. Searching the feature creation paths for each of the images is not necessary.

## 2.4. Optimization

Optimization fuses the information gathered from the automatic evaluation, feature visualization, and retrieval results into a set of parameters for the retrieval algorithms and their combination. In IRMA, the parameters are put into the processing chain at two points: source nodes of the network entities and the weights for the result matrices used by the combination tool. Command-line tools are used to insert features into the database and to run parameterized networks. Together with the evaluation tool and the tool for result matrix combination, these are integrated into a script, which implements a grid search and automatically perform runs using the different parameters. The error rates obtained from the evaluation tool are then used to find the best parameter set.

## 2.5. Obtaining CLEF MAAT baseline results

Since CLEF MAAT is a classification task, global features are employed, which capture properties of the whole image without performing a partitioning of the image or segmentation. Four content descriptors and their distance measures (or similarity measures, respectively) are implemented as single methods, which are afterwards integrated into networks. The content of a medical image is represented by texture features (TMM) and (CTM) proposed in Tamura et al. (1978) and Castelli et al. (1998), respectively. CTM was used during an separate experiment in 2006 and is not part of the baseline results. Down-scaled representations of the original images were computed to 32 × 32 and X × 32 pixels disregarding and according to the original aspect ratio, respectively. Since these

image icons maintain the spatial intensity information, variabilities which are commonly found in a medical imagery are modelled by the distance measure. These include radiation dose, global translation, and local deformation. In particular, the cross-correlation function (CCF) which is based on SHANNON, and the image distortion model (IDM) from [Keysers et al. \(2003\)](#) are used.

The single classifiers are combined within a parallel scheme, which performs a weighting of the normalized distances obtained from the single classifiers  $C_i$ , and applies the nearest neighbor-decision function  $C$  to the resulting distances:

$$d_{\text{combined}}(q, r) = \sum_i \lambda_i \cdot d'_i(q, r), \quad (1)$$

$$d'_i(q, r) = \frac{d_i(q, r)}{\sum_{r' \in R} d_i(q, r')} \quad (2)$$

where  $0 \leq \lambda_i \leq 1$ ,  $\sum_i \lambda_i = 1$  denotes the weight for the normalized distance  $d_i(q, r)$  obtained from classifier  $C_i$  for a sample  $q$  and a reference  $r$  from the set of reference images,  $R$ . Values  $0 \leq s_i(q, r) \leq 1$  obtained from similarity measures are transformed via  $d_i(q, r) = 1 - s_i(q, r)$ .

To address the evaluation scheme in 2007, the NN decision rule is modified: From the  $k$  neighbors, a *common code* is generated by setting differing parts (and their subparts) to *don't know*, e.g., two neighbors with codes 1121-120-434-700 and 1121-12f-466-700 result in a *common code* of 1121-12\*-4\*\*-700.

### 3. Results

#### 3.1. Network builder

For CLEF MAAT, the network builder was used to define networks for the extraction and comparison of four types of global features. These networks integrate ten methods, which were implemented by several programmers. The documentation (stored as part of the method entity in the database) enables the network composer to treat the methods as black boxes, i.e. independent from their source code, based solely on the type information from the method interface. While it is still a programming tool and is only used by the developers, the network builder hides technical details and allows to communicate relevant algorithm steps and their parameters between programmers and medical experts. The method for the down-scaling of the images as a pre-processing

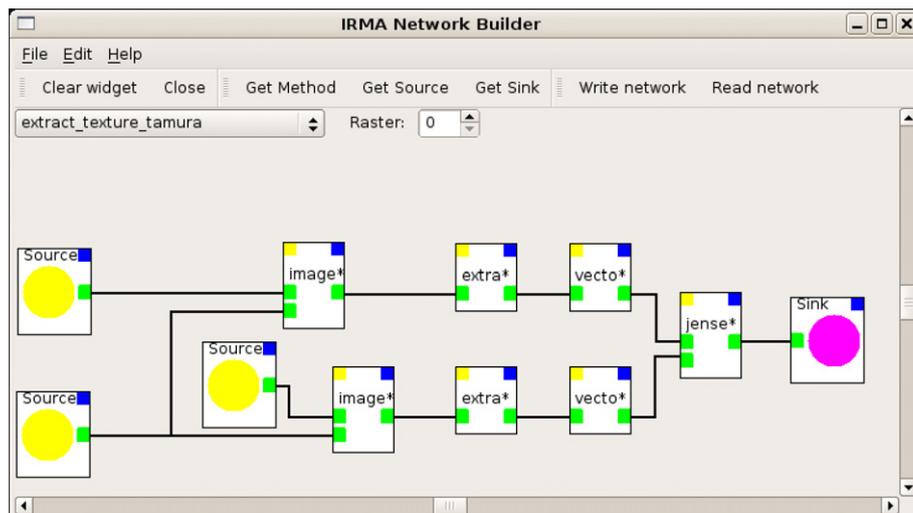
step was shared in the networks for the two texture-based features. For data exchange, definitions of feature types, methods, and networks, XML is used as a data format. Via system tools, the definitions are uploaded into the database can subsequently used within IRMA.

[Fig. 2](#) illustrates the building of a network for the extraction and comparison of texture features ([Lehmann et al., 2004](#)), which can be used for both the optimization phase (using the partitioning of training set vs. development set) and the submission phase (training set + development set vs. test set). First, the images are scaled to allow a roughly comparable scale for texture analysis. Afterwards, the extraction of the texture features is performed, resulting in a set of vectors. For better performance, these vectors are collected into a compact data structure. These steps are performed for both image sets. The propagation of the parameters into both scaling methods ensures the comparability of the features extracted for both sets. Finally, the two feature vector sets are propagated to the distance measure method, which yields a result matrix. This output is also marked as a relevant output of the network by propagating it to a sink node.

#### 3.2. Feature browser

The feature browser was used to track the generation of all required features for CLEF MAAT, primarily during the development of the methods for the extraction of the texture descriptions. The currently inspected feature is displayed in the upper right, along with its type and information about the method call. The input features for the method call, which produced the currently inspected feature, are displayed in the upper left. Features, which were derived from the current feature, are displayed at the bottom. Both preceding and succeeding features are selectable, which updates the application to inspect the respective feature, and thus follows the method calls along the computation chain. The browsing process starts with picking an imported image from the database. Alternately, the browser accepts the input of a feature ID, which points the browser to that feature.

[Fig. 3](#) exemplifies the use of the feature browser for the extraction of texture features proposed by [Tamura et al. \(1978\)](#), which are based on a re-scaled version of the original image. Predecessors of the re-scaled image are the original image and the scaling parameters, and two texture features were extracted. Here, the aspect ratio is disregarded.



**Fig. 2.** IRMA network builder. Method names are abbreviated: image\* stands for image scaling (first input is the image, second input is the parameter set), extra\* for texture feature extraction, vecto\* for the vector collector, jense\* for the distance measure (Jensen-Shannon divergence).



**Fig. 3.** IRMA feature browser. Non-image features are represented by icons, e.g., the scaling parameter in the upper left frame or the texture features (floating point vectors) in the bottom frame. In addition to the feature data in the center frame, the feature description from the database is displayed.

### 3.3. Feature visualization in the query interface

For CLEF MAAT, the implemented feature visualization component was integrated into the query interface, which allows switching the view from the query and result images to iconic representations of the features which were actually used during retrieval. The user chooses the context to be used from a pop-up window listing all available contexts. Due to the data structure introduced in Section 2.3, the identification of derived features does not increase the response time of the interface noticeably.

Using all 12,000 images from CLEF MAAT 2007, the single classifiers were deployed into the query interface for inspection. Fig. 4 shows the use of the visualization functionality in the query interface: a result is shown for image 17,761 from the test set, using CCF on down-scaled representations of the original images. The CCF, as similarity measure, yields values between  $-1$  and  $1$  and provides intensity normalisation. Beside the query image, none of the retrieved images is correct. In fact, this image also yields 12 incorrect neighbors for the combination of all four algorithms in the CLEF MAAT 2007 run. Via the alternate view of the  $16 \times 16$  pixel representations, the retrieval results become plausible, as the matched regional intensity distribution is visible. The interface is described in detail in Deserno et al. (2007).

### 3.4. Baseline results for CLEF MAAT

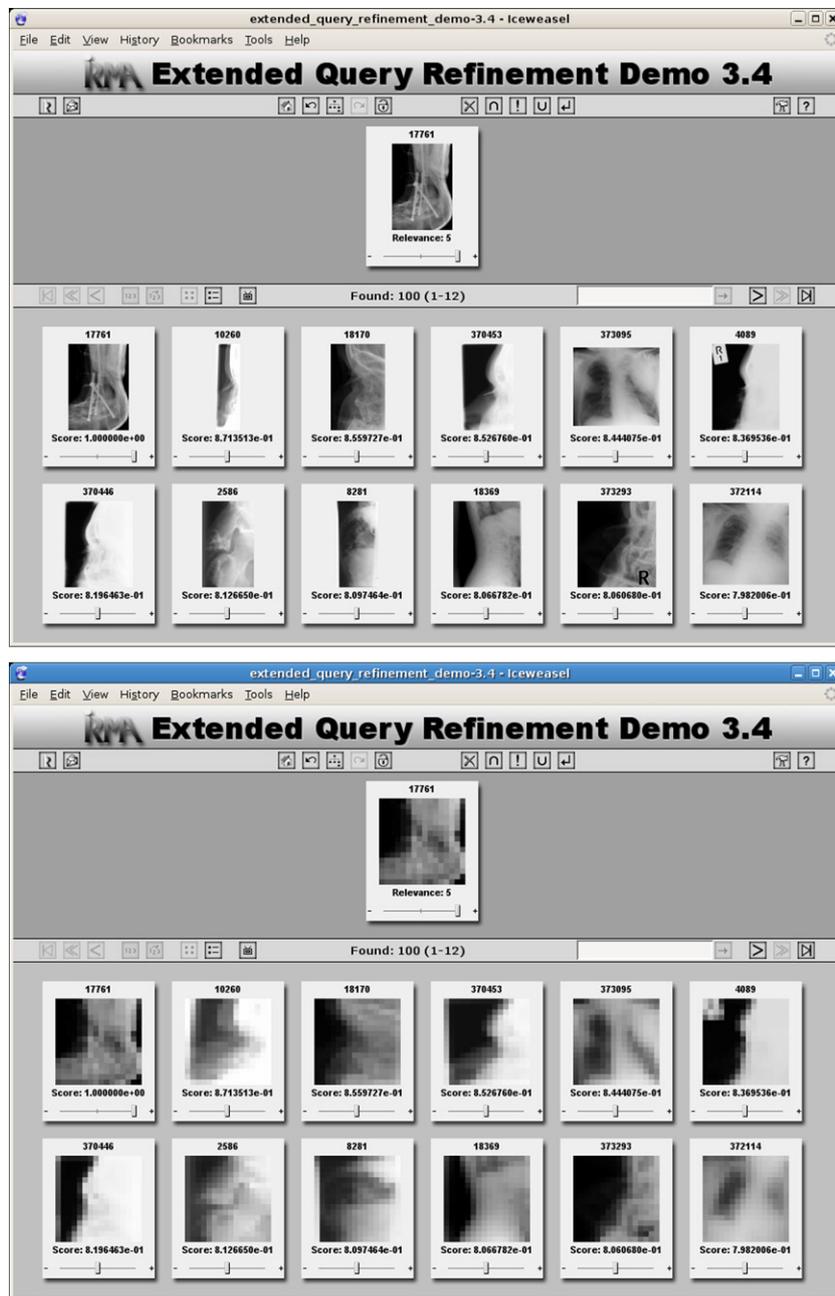
IRMA methods were implemented for the extraction and the comparison of four global features (Güld et al., 2006, 2007; Güld

and Deserno, in press). The methods which implement generic pre-processing steps like scaling were used in all networks which require this operation. All methods for classification generate a result matrix as their output feature. The combination of these features is done within a nearest neighbor scheme: a total distance between a sample image and a reference image is obtained by the weighted sum of the normalized distances from the single classifiers. The four content descriptors and their distance measures use the following parameters:

- *TTM*: texture histograms from down-scaled image ( $256 \times 256$ ), 384 bins, Jensen–Shannon divergence as a distance measure.
- *CTM*: texture features from down-scaled image ( $256 \times 256$ ), 43 values, Mahalanobis distance with diagonal covariance matrix  $\Sigma$ .
- *CCF*:  $32 \times 32$  icon,  $9 \times 9$  translation window.
- *IDM*:  $X \times 32$  icon, gradients,  $5 \times 5$  window,  $3 \times 3$  context.

The weighting coefficients were set empirically during CLEF MAAT 2005:  $\lambda_{IDM} = 0.42$ ,  $\lambda_{CCF} = 0.18$ , and  $\lambda_{TTM} = 0.4$ . CTM was used only in 2006 and its inclusion does not improve the results on the development set.

Table 1 lists the baseline results for the three tasks. Runs which were not submitted are displayed marked with asterisks, along with their hypothetical rank. In 2007, the evaluation was not done based on the error rate – the table contains the rank based on the modified evaluation scheme for the corresponding submission of full codes. The *common code* rule yields 80.47 when applied to



**Fig. 4.** Alternate viewing modes in the query interface: original images (top) and down-scaled images ( $16 \times 16$  pixels, bottom). For each retrieved image, the similarity score is given. The slider allowing the user to provide relevance feedback was not used for CLEF MAAT.

**Table 1**

Baseline error rates (ER) and ranks among submissions (Güld and Deserno, in press)

Year	References	Classes	$k = 1$		$k = 5$	
			ER (%)	Rank	ER (%)	Rank
2005	9000	57	13.3	2/42	14.8	* 7/42
2006	10,000	116	21.7	13/28	22.0	* 13/28
2007	11,000	116	20.0	17/68	18.0	18/68

the 5-NN results. Based on the tools for the combination and evaluation of result matrices, an automatic optimization of the weight parameters in the combination step was performed during CLEF MAAT 2006 (Güld et al., 2007), but did not improve the results significantly. Applying this rule resulted in little additional computa-

tion costs, because the result matrices of the four retrieval algorithms could be used, i.e. the expensive feature extraction and comparison steps for each of the four retrieval algorithms had to be done only once. However, it must be noted that the parameter optimization for the content descriptors, the classifiers and their combination was done in 2005 based on the error rate. Obtained error rates do not translate to similar performance w.r.t. the evaluation scheme in 2007, which penalizes the severity of classification errors. This becomes evident by an anomaly observed in the results: the results for the submissions in 2007 swap their ranks when the evaluation is based on the error rate.

In summary, the IRMA framework was successfully used to perform the experiments in CLEF MAAT. Since the challenges in 2006 and 2007 use the same class definitions, they are directly comparable. Judging from the baseline error rates obtained by the IRMA

framework, the task in 2007 is easier than 2006. This can be taken into account when comparing methods by groups who participated in only one of the past years. Comparing 2005 and 2006, the number of classes increased by 103%, while the error rate increased by 63% and 48% for 1-NN and 5-NN, respectively. This indicates that CLEF MAAT 2005 is easier than the tasks of 2006 and 2007. In general, the decreasing rank for the baseline results among the competition suggests advances in the field of content representation in medical images over the past years. The established framework will also be used for future CLEF MAAT challenges.

#### 4. Discussion

To support CLEF MAAT, logical application steps were established on top of the IRMA framework to allow the efficient evaluation, combination and optimization of CBIR algorithms based on nearest neighbor classifiers. The result matrix feature type ensures that any pair of element-based classifiers can be combined and evaluated with two generic programs. Via iterated runs, the optimization of parameters can also be performed automatically with little additional effort and computational costs. The runtime environment allows the distributed computation of CBIR algorithms, which is especially useful for feature extraction due to the low interdependency of the computation steps for sets of images.

The implemented user interfaces support multiple application steps: the network builder supports the user in defining retrieval and classification algorithms based on existing methods. Through the type information in each method's feature interface, input/output compatibility is ensured. By accessing the method's documentation, which is stored inside the database as part of the method entity, the composition of networks is manageable without any knowledge on the method's source code.

During the development of methods and networks, any computed feature can be easily inspected, and processing steps can be verified. Through the query interface, an effective mechanism is available to inspect the impact of parameters on the retrieval results. Here, the optional view of certain feature data within the query interface provides relevance facts to verify the algorithm. Since the feature browser and the query interface can be accessed over the Internet, they also support the collaboration between developers and medical experts over geographical distances.

#### Acknowledgement

This work is partly funded by the German Research Foundation (DFG), Grants Le 1108/4 and Le 1108/9. We also appreciate the peer reviewers' useful comments and suggestions.

#### References

- Argiro, D., Farrar, K., Kubica, S., 2001. Cantata: The visual programming environment for the Khoros system. In: Hamza, M.H. (Ed.), Proceedings of the IASTED International Conference on Visualization, Imaging and Image Processing. ACTA Press, Marbella, Spain, ISBN 0-88986-309-1, pp. 697–702.
- Castelli, V., Bergman, L.D., Kontoyiannis, I., Li, C.S., Robinson, J.T., Turek, J.J., 1998. Progressive search and retrieval in large image archives. *IBM Journal of Research and Development* 42 (2), 253–268.
- Deselaers, T., Müller, M., Deserno, T.M., 2008. Automatic Medical Image Annotation in ImageCLEF 2007: Overview, Results, and Discussion. *Pattern Recognition Letters*, Special Issue on Medical Image Annotation in ImageCLEF 2007, 29 (15), 1988–1995.
- Deserno, T.M., Güld, M.O., Plodowski, B., Spitzer, K., Wein, B.B., Schubert, H., Ney, H., Seidl, T., 2007. Extended query refinement for medical image retrieval. *Journal of Digital Imaging*, online-first DOI:10.1007/s10278-007-9037-4.
- Güld, M.O., Thies, C., Fischer, B., Lehmann, T.M., 2006. Content-based retrieval of medical images by combining global features. *Lecture Notes in Computer Science*, vol. 4022. Berlin, Germany, pp. 702–711.
- Güld, M.O., Thies, C., Fischer, B., Deserno, T.M., 2007. Baseline results for the ImageCLEF 2006 medical automatic annotation task. In: *Lecture Notes in Computer Science*, vol. 4730. Berlin, Germany, pp. 686–689.
- Güld, M.O., Thies, C., Fischer, B., Lehmann, T.M., 2007. A generic concept for the implementation of medical image retrieval systems. *International Journal of Medical Informatics* 76, 252–259.
- Güld, M.O., Deserno, T.M., in press. Baseline results for the ImageCLEF 2007 medical automatic annotation task using global features. In: *Lecture Notes in Computer Science*, Berlin, Germany.
- Hidki, A., Depeursinge, A., Iavindrasana, J., Pitkanen, M., Zhou, X., Müller, H., in press. The medGIFT project: Perspective of a medical doctor. *JAMIT Annual Meeting*, Tokyo, Japan.
- Keysers, D., Dahmen, J., Ney, H., Wein, B.B., Lehmann, T.M., 2003. A statistical framework for model-based image retrieval in medical applications. *Journal of Electronic Imaging* 12 (1), 59–68.
- Lehmann, T.M., Güld, M.O., Thies, C., Fischer, B., Spitzer, K., Keysers, D., Ney, H., Kohlen, M., Schubert, H., Wein, B.B., 2004. Content-based image retrieval in medical applications. *Methods of Information in Medicine* 43 (4), 354–361.
- Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., Jain, R., 2000. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (12), 1349–1380.
- Tamura, H., Mori, S., Yamawaki, T., 1978. Textural features corresponding to visual perception. *IEEE Transactions on Systems, Man and Cybernetics* 8 (6), 460–473.
- Thies, C., Güld, M.O., Fischer, B., Lehmann, T.M., 2005. Content-based queries on the casimage database with the IRMA framework: A field report. In: *Lecture Notes in Computer Science*, vol. 3491. Berlin, Germany, pp. 781–792.