IPL at CLEF 2013
Medical Retrieval Task

Spyridon Stathopoulos, Ismini Lourentzou,
Antonia Kyriakopoulou, and Theodore Kalamboukis

Information Processing Laboratory,
Department of Informatics,
Athens University of Economics and Business,
76 Patission Str, 104.34, Athens, Greece
spstath@gmail.com,lourentzouismini@hotmail.com,
tonia@aueb.gr,tzk@aueb.gr

http://ipl.cs.aueb.gr/index_eng.html

Abstract. This article presents an experimental evaluation on using a refined approach to the Latent Semantic Analysis (LSA) for efficiently searching very large image databases. It also describes IPL's participation to the image CLEF ad-hoc textual and visual retrieval as well as modality classification for the Medical Task in 2013. We report on our approaches and methods and present the results of our extensive experiments applying early data fusion with LSA on several low-level visual and textual features.

Key words: LSA, LSI, CBIR, Data Fusion

1 Introduction

Over the years latent semantic analysis has been applied with success in text retrieval providing successful results in several applications [1]. However, due to the challenges of LSA in terms of computational and memory requirements in cases of image retrieval, only small datasets have been tested. In our approach, our aim is the visual representation of an image with a feature vector of a moderate size, (m), and the use of a by-pass solution to the singular value decomposition which overcomes all its deficiencies and makes the method attractive for content-based image retrieval [2]. In this way instead of performing SVD to the feature-by-document matrix C, $(m \times n)$ we solve the eigenproblem of the feature-correlation matrix CC^T, $(m \times m)$.

Concerning the stability of the eigensolution for the matrix CC^T, the method may be unstable for two reasons: first, the conditioning number of the matrix is much higher, and second, perturbations introduced while forming the normal matrix (CC^T) may change its rank. In such cases, the normal matrix will be more sensitive to perturbations in the data than the data matrix (C).

However the numerical stability of an eigenproblem is ensured when the eigenvalues are well separated [3]. During preliminary experiments and previous work [2][4], we have observed that the eigenvalues of CC^T follow a power law distribution. This ensures that the largest eigenvalues are well separated. It was also indicated that a value of k (k largest eigenvalues) between 50 and 100 gives optimal results. Furthermore, matrix C is stored in integer form for both visual and textual data. Thus, no rounding is introduced in the computation of CC^T matrix. To further reduce the size of the CC^T matrix, we have applied a variance based feature selection. Thus, the largest eigenproblem that was required to be solved for this years' challenge was that of a CC^T[1400 x 1400] matrix.

In order to overcome the increased memory demands for the computation of the correlation matrix CC^T, matrix C is split into a number of blocks, such that each block can be accommodated into the memory. Subsequently, the eigenproblem is solved and the k largest eigenvalues, S_k, with their corresponding eigenvectors, U_k, are selected. The original feature vectors are then projected into the k-th dimensional space, using the transformation, $y_k = U_k^T y$, on the original vector representation of an image y. The same projection is also applied for a query image q_k with $q_k = U_k^T q$ and the similarity with an image $score(q_k, y_k)$, is calculated by the cosine function.

The proposed method seems to greatly improve the final database size, query response time and memory requirements. It is also shown that the efficiency of this method still holds in cases of large databases in cases such as the PubMED Database with 306.000 figures, used in this year's medical task [5]. It should be noted that by using the traditional solution of SVD for this database, the ad-hoc visual retrieval task would be impossible with our computer resources. This approach can also exploit the dimensionality reduction and enable the early data fusion of different low-level visual features, without increasing the cost in memory, disk space and response time of a retrieval system.

2 Visual Retrieval

2.1 Image Representation

For the low-level representation of each image, a set of localized image descriptors was extracted. In order to address the variations in resolution between images, first, a size normalization was performed by re-scaling each image to a fixed size of 256 x 256 pixels using bi-linear interpolation. Next, each image was split into 3 x 3 fixed sized blocks and a local visual descriptor was extracted from each block. The image's final feature vector was constructed by concatenating each local vector. i.e if for an image, we extract a gray color histogram in 128 colors per block, for a total of 9 blocks, the resulting feature vector will be of size 9 x 128 = 1152. This process is depicted in Figure 1.

In our experiments, the vector representation was based on seven types of low-level visual features:

1. Gray color histogram (CH) extracted in 128 gray scales.

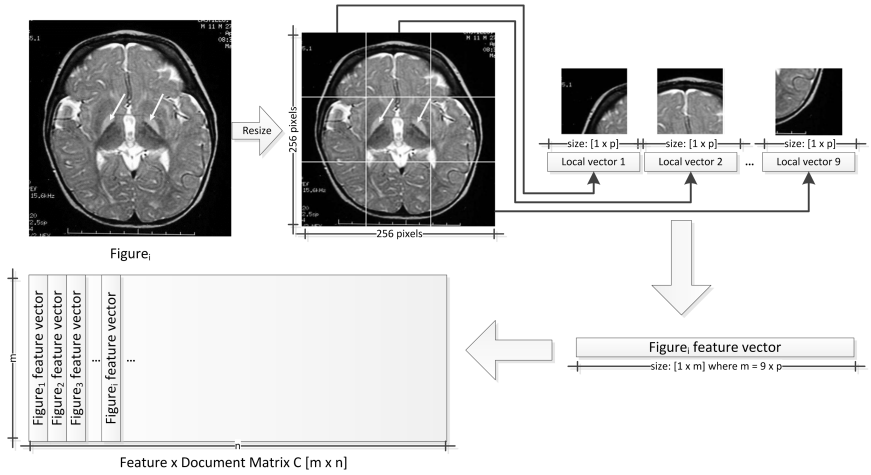

Fig. 1. Feature extraction process

2. Color layout (CL).
3. Scalable color (SC).
4. Edge histogram (EH).
5. CEDD.
6. FCTH.
7. Color Correlogram (CORR).

All the features were extracted using the Java library Caliph&Emir of the Lire CBIR system [6]. Finally, by using feature selection, the vector for each descriptor was reduced into a fraction of its original size. Table 1 lists the visual descriptors along with their corresponding vector size per block and per image before and after feature selection.

Table 1. Visual descriptors and their corresponding vector size before and after feature selection.

Visual descriptor	Size per block	Size per image	Final size per image
Gray Color Histogram (CH)	128	1152	100
Color Layout (CL)	120	1080	300
Scalable Color (SC)	64	576	200
Edge Histogram (EH)	80	720	300
CEDD	144	1296	400
FCTH	192	1728	300
Color Correlogram (CORR)	1024	9216	600

2.2 Early Fusion

In order to improve retrieval efficiency, different low level visual descriptors were combined with early fusion. Image descriptor vectors were concatenated to create a single fused vector thus, creating the feature-by-document matrix C previously mentioned. The query feature vectors were fused with the same order and the resulting vector was projected as described in 1. For example, the early fusion of CH, CEDD and FCTH features will form the matrix $C = [CH; CEDD; FCTH;]$ in matlab notation of size 306.000 x 800.

The solution of the $CC^TU = US^2$ problem was done for a CC^T of size 800 x 800. The eigs function of Matlab was used for this solution. For the visual retrieval task, several data fusion combinations of different descriptors were tested. These combinations are presented in Table 2 along with the task's corresponding run ID.

Table 2. Visual runs with combined image descriptors.

Run ID	Combined descriptors
IPL13_visual_r1	CORR,CL,CEDD
IPL13_visual_r2	CORR, CL, CH
IPL13_visual_r3	CORR, CL, CEDD, CH
IPL13_visual_r4	CORR, CEDD, FCTH, CH

2.3 Modality Class Vectors

To further improve the retrieval efficiency, modality class vectors were constructed by using a subset of the classifiers used for the modality classification task. For this method, a combination of low level feature vectors was extracted from each image as described above. In addition, each image was classified into one of the 31 modalities and a very sparse vector CV of size [31x1] was created by setting the value '1' at the index that corresponds to the predicted modality. The rest of the vector's elements are set to '0'. Finally, this vector was early fused with the rest of the visual vectors, i.e for the previous example, the matrix $C = [CH; CEDD; FCTH; CV;]$ was formed. Table 3 presents the runs using this method with their corresponding run ID, visual descriptors and classifier used.

2.4 Results

For the AMIA's medical task [5] we have submitted a total of eight visual runs using different combinations of low level feature descriptors with early fusion and class filtering. In Table 4, we list the runs ids with their corresponding results. Finally, we attempted to test how our retrieval method responds in image datasets of different sizes and context. Thus, in this year's ImageCLEF [7], we

Table 3. Visual runs with class filtering and the classifier used.

Run ID	Combined descriptors	Classifier
IPL13_visual_r5	CORR,CL,CEDD,CH	Centroids CEDD
IPL13_visual_r6	CORR,CL,CEDD,CH	Centroids SVD CEDD
IPL13_visual_r7	CORR,CL,CEDD,CH	Improved Centroids SVD CEDD
IPL13_visual_r8	CORR,CL,CEDD,CH	Centroids CEDD & CH

have also participated to the ImageCLEF's Personal photo retrieval sub-task [8] using the feature combinations listed in Table 2. Since no form of classification data were provided for this task, class filtering methods were not tested. In Table 5, we list the corresponding results obtained from the Average user ground truth set. The results for other types of users (non-expert, expert etc), were similar.

Table 4. IPL's performance results from medical task's visual retrieval.

Run ID	MAP	GM-MAP	bpref	p10	p30
IPL13_visual_r1	0.0083	0.0002	0.0176	0.0314	0.0257
IPL13_visual_r2	0.0071	0.0001	0.0162	0.0257	0.0257
IPL13_visual_r3	0.0087	0.0003	0.0173	0.0286	0.0257
IPL13_visual_r4	0.0081	0.0002	0.0182	0.0400	0.0305
IPL13_visual_r5	0.0085	0.0003	0.0178	0.0314	0.0257
IPL13_visual_r6	0.0119	0.0003	0.0229	0.0371	0.0286
IPL13_visual_r7	0.0079	0.0003	0.0175	0.0257	0.0267
IPL13_visual_r8	0.0086	0.0003	0.0173	0.0286	0.0257

Table 5. IPL's performance results from photo retrieval task for Average user.

Run ID	MAP	p5	p10	p20	p30	p100
IPL13_visual_r1	0.1118	0.6594	0.5152	0.4125	0.3725	0.3077
IPL13_visual_r2	0.1082	0.6303	0.4955	0.3899	0.3499	0.2910
IPL13_visual_r3	0.0771	0.5769	0.4141	0.3138	0.2741	0.2226
IPL13_visual_r4	0.1162	0.6627	0.5152	0.4173	0.3713	0.3126

3 Textual-based Ad-hoc Image Retrieval

In our approach, images were represented as structured units, consisting of several fields. We used Lucene's API in order to index and store each image as a set of fields alongside with boosting the fields of each image when submitting a query. This technique helped in experimenting with different weights and combinations of these fields.

For every image in the given database we stored five features: Title, Caption, Abstract, MeSH and Textual References. MeSH terms related with each article provide extra information for the contained figures. MeSH terms were downloaded from the Pubmed ID of the article. Finally, we extracted every sentence inside the article that refers to an image, and used this set of sentences as a consistent field named Textual References.

3.1 Experiments and Results

Details on the ad-hoc textual image retrieval task are given in [5]. Our experiments were based on previous participations [4] of the Information Processing Laboratory.

To achieve even higher MAP values than our 2012 runs, we carried out several experiments with different boosting factors, using as a train set the qrels from ImageClef 2012.

Motivated to achieve better results, we experimented in field selection, which revealed that the use of the Title along with Caption provides a strong combination. Moreover, a heuristic was applied to find the best boosting factors per field. Experiments with the best MAP values for the CLEF-2012 database are presented in Table 6, where T, C, A, M and TR are the boosting factors for Title, Caption, Abstract, MeshTerms, Textual References respectively. In addition, TC is a joint combination of Title and Caption in one field. The use of this field without any boosting factor was placed second in this year's ad-hoc textual retrieval task.

Table 6. Experimental results in ImageClef 2012 queries.

Run	Fields weight	MAP	GM-MAP	bpref	p10	p30
r1	T=0.65 A=0.57 C=3.50 M=0.57	0.2051	0.0763	0.2071	0.3227	0.2061
r2	T=0.625 A=0.57 C=3.50 M=0.5	0.2051	0.0762	0.2071	0.3227	0.2076
r3	T=0.625 A=0.555 C=3.50 M=0.555	0.2050	0.0757	0.2061	0.3227	0.2045
r4	T=0.1 A=0.113 C=0.335 M=0.1	0.2016	0.0765	0.1991	0.2955	0.2091
r5	T=1 A=1 C=6 M=0.2	0.2021	0.0729	0.2003	0.3182	0.2076
r6	TC (no boosting factor)	0.2177	0.0848	0.2322	0.3500	0.2045
r7	T=0.3 A=0.79 C=3.50 M=0.73 TR=0.11	0.2106	0.0797	0.2047	0.3227	0.2182
r8	TC=0.26 A=0.02	0.2215	0.0824	0.2397	0.3273	0.2136

These runs were our submissions to the textual ad-hoc image-based retrieval task. In r4 and r5 (Table 6) we have kept the boosting factors from our former participation at ImageClef 2012. In Table 7 we present the final results of these eight submissions of the IPL Group.

Table 7. IPL's performance results from textual retrieval.

Run ID	MAP	GM-MAP	bpref	p10	p30
IPL13_textual_r1	0.2355	0.0583	0.2307	0.2771	0.2095
IPL13_textual_r2	0.2350	0.0583	0.229	0.2771	0.2105
IPL13_textual_r3	0.2354	0.0604	0.2294	0.2771	0.2124
IPL13_textual_r4	0.2400	0.0607	0.2373	0.2857	0.2143
IPL13_textual_r5	0.2266	0.0431	0.2285	0.2743	0.2086
IPL13_textual_r6	0.2542	0.0422	0.2479	0.3314	0.2333
IPL13_textual_r7	0.2355	0.0579	0.2358	0.2800	0.2171
IPL13_textual_r8	0.2355	0.0579	0.2358	0.2800	0.2171

4 Modality Classification

4.1 Experiments Settings

All our experiments were run using various combinations of the seven types of low-level visual features presented in Section 2.1 and of the textual data described in Section 3, with and without early data fusion and/or LSA applied. We employed the SVM^{light} [9][10] implementation of Support Vector Machines (SVMs) and Transductive Support Vector Machines (TSVMs) to perform multi-class classification, using a one-against-all voting scheme. It should be noted here that with the term multi-class we refer to problems in which any instance is assigned exactly one class label. In our experiments, following the one-against-all method, k binary SVM/TSVM classifiers (where k is the number of classes) were trained to separate one class from the rest. The classifiers were then combined by comparing their decision values on a test data instance and labeling it according to the classifier with the highest decision value. No parameter tuning was performed. A binary classifier was constructed for each dataset, a linear kernel was used and the weight C of the slack variables was set to default.

4.2 Results

Details on the modality classification task are given in [5]. In Table 8, we present the results of the above experiments for the various types of classification, i.e. for textual, visual, and mixed classification. As a measure of classification performance we used accuracy.

As expected, mixed classification, on both visual and textual features, yielded the best performance in all cases compared to visual or textual only classification, scoring a 71.42% accuracy when applying SVMs on a combination of textual features (Title and Caption in one field with no boosting factor (TC)) with CORR,CL,CEDD, and CH visual descriptors. LSA was tested for different values of the k largest eigenvalues (50, 100, 150, 200). The best results were accomplished for $k = 200$, for some descriptors it was almost equal to SVMs applied on the whole dataset.

Textual classification with SVMs succeeds a 65.29% accuracy score. When LSA is applied on the dataset for $k = 150$, it gives competitive results compared to the original vectors. It should be reminded that the original vectors have ≈ 147.000 features.

For classification on visual features only, the CEDD descriptor with SVMs has the best performance against the other descriptors with 61.19% accuracy score. When more than one low level feature descriptors are combined with early fusion into one fused vector, SVMs perform better in all cases. LSA was tested for different values of the k largest eigenvalues (50, 100, 150). The best results were accomplished for $k = 150$ and it should be noted that they highly approximate those of SVM when applied on the original feature vectors.

Table 8. Classification performance on visual, textual and mixed data.

Classification Type	**Features**	**SVM**	**LSA, SVM**
Textual	TC	65.29%	64.60%
Visual	CORR	48.53%	46.44%
	CL	47.95%	45.39%
	CEDD	61.19%	60.81%
	FCTH	59.60%	55.58%
	CH	41.67%	41.32%
	CORR,CEDD,FCTH	62.94%	61.08%
	CORR,CEDD,CH	62.94%	60.42%
	CORR,CL,CEDD	61.74%	61.31%
	CORR,CL,CEDD,CH	63.67%	61.85%
Mixed	TC,CORR	68.36%	65.33%
	TC,CL	67.43%	62.47%
	TC,CEDD	69.25%	65.37%
	TC,FCTH	69.13%	66.22%
	TC,CH	66.62%	66.11%
	TC,CORR,CEDD,FCTH	70.95%	66.46%
	TC,CORR,CEDD,CH	70.29%	67.19%
	TC,CORR,CL,CEDD	71.11%	64.52%
	TC,CORR,CL,CEDD,CH	71.42%	65.10%

The runs that were submitted to the modality classification task were based on the experiments presented above, but other methods were also tested. A de-

scription of the runs is given in Table 9. It should be noted that the textual data used in the runs contained only terms with document frequency larger than 1000. In this case, the dimensionality of the textual dataset is dramatically reduced to $\approx$ 10.000 features, drastically less than the $\approx$ 147.000 features of the textual dataset used in the former experiments. Also, apart from using TSVMs for classification, we also experimented on using class-centroid-based classification [11]. This method had the advantage of short training and testing time due to its computational efficiency. However, the accuracy of the centroid-based classifiers was inferior. We speculate that the centroids found during construction were far from perfect locations.

Table 9. IPL's performance results from modality classification.

Run_id	Classification Type	Description	Accuracy score
IPL13_mod_cl_mixed_r1	Mixed	1. Early fusion: on CEDD, CH, and FCTH descriptors and textual data. 2. LSA applied on the fused vectors with k=50 3. Classify with class centroids.	9.56%
IPL13_mod_cl_mixed_r2	Mixed	1. Early fusion: on CEDD, CH, and FCTH descriptors and textual data. 2. Classify using TSVMs.	61.03%
IPL13_mod_cl_mixed_r3	Mixed	1. Early fusion: on CEDD descriptor and textual data. 2. Classify using TSVMs.	58.98%
IPL13_mod_cl_visual_r1	Visual	1. LSA applied on a combination of CEDD, CH, and FCTH descriptors with k=50 2. Classify with class centroids.	6.19%
IPL13_mod_cl_visual_r2	Visual	1. Classify using TSVMs on CEDD descriptor	52.05%
IPL13_mod_cl_textual_r1	Textual	1. LSA applied on textual data with k=50 2. Classify with class centroids.	9.02%

4.3 Conclusions

We have presented an approach to LSA for CBIR replacing the SVD analysis of the feature matrix C $(m \times n)$ by the solution of the eigenproblem for the matrix CC^T $(m \times m)$. The method overcomes the high cost of SVD in terms of memory and computing time. More work on stability issues is currently underway.

Moreover, some cases of the usage of modality class vectors in early fusion techniques, have shown that can further improve retrieval results. Additional

work in this direction is also in progress, by systematically testing more advanced classifiers and different low-level features.

Also, the inclusion of textual information, extracted from the meta-data provided, is also investigated. Specifically, the number of the extracted textual terms ($\approx 147,000$) is far greater in comparison to the size of visual features. Hence, increased memory requirements and complexity is introduced. This problem is open for future research on several solutions like term selection or the use of an ontology in order to extract semantic keywords that strongly define a document.

For the modality classification task, mixed classification, on both visual and textual features, yielded the best performance in all cases compared to visual or textual only classification. The application of SVMs for image classification had a positive impact, verifying previous findings.

References

1. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. JASIS **41**(6) (1990) 391–407
2. Stathopoulos, S., Kalamboukis, T.: An svdbypass latent semantic analysis for image retrieval. In Greenspan, H., Muller, H., Syeda-Mahmood, T., eds.: Medical Content-Based Retrieval for Clinical Decision Support. Volume 7723 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2013) 122–132
3. Golub, G.H., van Loan, C.F.: Matrix computations (3. ed.). Johns Hopkins University Press (1996)
4. Stathopoulos, S., Sakiotis, N., Kalamboukis, T.: Ipl at clef 2012 medical retrieval task. In Forner, P., Karlgren, J., Womser-Hacker, C., eds.: CLEF (Online Working Notes/Labs/Workshop). (2012)
5. Seco de Herrera, A.G., Kalpathy-Cramer, J., Fushman, D., Antani, S., Müller, H.: Overview of the imageclef 2013 medical tasks. In: Working notes of CLEF 2013, Valencia, Spain. (2013)
6. Lux, M., Chatzichristofis, S.A.: Lire: lucene image retrieval: an extensible java cbir library. In: ACM Multimedia. (2008) 1085–1088
7. Caputo, B., Müller, H., Thomee, B., Villegas, M., Paredes, R., Zellhofer, D., Goeau, H., Joly, A., Bonnet, P., Martinez Gomez, J., Garcia Varea, I., Cazorla, M.: Imageclef 2013: the vision, the data and the open challenges. In: Proceedings of CLEF 2013, Valencia, Spain, Springer LNCS (2013)
8. Zellhöfer, D.: Overview of the imageclef 2013 personal photo retrieval subtask. In: Working notes of CLEF 2013, Valencia, Spain. (2013)
9. Joachims, T.: Learning to Classify Text Using Support Vector Machines. Dissertation. Kluwer (2002)
10. Joachims, T.: Transductive inference for text classification using support vector machines. In: 16th International Conference on Machine Learning, San Francisco: Morgan Kaufmann (1999) 200–209
11. Han, E.H.S., Karypis, G.: Centroid-based document classification: Analysis & experimental results. In: 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD). (2000) 424–431