Query-by-Example Image Retrieval in Microsoft SQL Server

Paweł Staszewski[1], Piotr Woldan[1], Marcin Korytkowski[1], Rafał Scherer[1(✉)], and Lipo Wang[2]

[1] Institute of Computational Intelligence, Częstochowa University of Technology, Al. Armii Krajowej 36, 42-200 Częstochowa, Poland
rafal.scherer@iisi.pcz.pl

[2] School of Electrical and Electronic Engineering, Nanyang Technological University, Block S1, Nanyang Avenue, Singapore 639798, Singapore
http://iisi.pcz.pl

Abstract. In this paper we present a system intended for content-based image retrieval tightly integrated with a relational database management system. Users can send query images over the appropriate web service channel or construct database queries locally. The presented framework analyses the query image based on descriptors which are generated by the bag-of-features algorithm and local interest points. The system returns the sequence of similar images with a similarity level to the query image. The software was implemented in .NET technology and Microsoft SQL Server 2012. The modular construction allows to customize the system functionality to client needs but it is especially dedicated to business applications. Important advantage of the presented approach is the support by SOA (Service-Oriented Architecture), which allows to use the system in a remote way. It is possible to build software which uses functions of the presented system by communicating over the web service API with the WCF technology.

Keywords: Content-based image retrieval · Relational databases · Bag-of-features · Query by image · WCF · Microsoft SQL Server

1 Introduction

Content-based image retrieval (CBIR) is part of a broader computer vision area. Thanks to CBIR-related methods [1,5,6,13,14,20,21,23,27,33] we are able to search for similar images and classify them [35,41]. To compare images we have to extract some form of visual features, e.g. color [10,16,30], textures [3,8,12,37], shape [11,15,39] or edges [43]. Other choice can be local invariant features [24–26,28,36] with the most popular detectors and descriptors SURF [2], SIFT [24] and ORB [34]. To find similar images to a query image, we need to compare all feature descriptors of all images usually by some distance measures. Such comparison is enormously time consuming and there is ongoing worldwide research

L. Rutkowski et al. (Eds.): ICAISC 2016, Part II, LNAI 9693, pp. 746–754, 2016.
DOI: 10.1007/978-3-319-39384-1_66

to speed up the process. Yet, the current state of the art in the case of high-dimensional computer vision applications is not fully satisfactory. The literature presents countless methods, e.g. [32] and they are mostly based on some form of approximate search. Generally, when the amount of data is increasing, in a consequence the compunctions are more complex. Moreover, the process of loading images from storage requires more time.

Recently, the bag-of-features (BoF) approach [9,22,29,31,36,40,42] has gained in popularity. In the BoF method, clustered vectors of image features are collected and sorted by the count of occurrence (histograms). All individual descriptors or approximations of sets of descriptors presented in the histogram form must be compared. The information contained in descriptors allows for finding a similar image to the query image. Such calculations are computationally expensive. Moreover, the BoF approach requires to redesign the classifiers when new visual classes are added to the system.

All these aforementioned methods require a large amount of data and computing power to provide an appropriate efficiency. Despite applying some optimization methods to these algorithms, the data loading process is time-consuming. In the case of storing the data in the database, when a table contains n records, the similarity search requires $O(log_2 n)$ comparisons. Image comparison procedure can take less time when some sorting mechanisms are applied in a database management system. Noteworthy solutions are proposed by different database products [4,19,38]. A system designed for the image classification task based on fuzzy logic was presented in [18] and on BoF with a MS SQL Server database was presented in [17]. The system structure allowed to modify crucial components without resulting in interferences with the other modules. Thus, authors of the current paper modify the system to adapt it to image retrieval task, i.e. to detect similar images to the query image which was presented on the system input.

MS SQL Server offers the FileTable mechanisms thanks to the SQL Server FILESTREAM technology to store large files a filesystem. Modifying the content of objects stored in a FileTable can be performed by adding, or removing data from directories linked to this table and the changes are visible in the table automatically.

The paper is organizes as follows. In Sect. 2 the proposed approach for system architecture and functionality was presented. Section 3 contains examples of the system response to query images.

2 Description of the Proposed System

The system described in this paper allows to search similar images to the query image which was provided by a user or a client program. Users are able to interact with our system by executing a stored procedure. There is also a possibility of calling the methods from a WCF service in a remote way. This operation can be performed in a client software. When the user interacts with the system locally, the query images can by copied to a special directory called `Test`, which is the integral part of the database FileTable structure. As a consequence, the appropriate trigger is executed and adequate testing stored procedure is called. When client software connects to our system remotely, it is necessary to transfer the query image as stream over the network. The authors provided API mechanisms to perform this kind of interaction.

2.1 Architecture of the System

The main target of the system are business applications that need a fast CBIR functionality. It encapsulates computer vision algorithms and other mechanisms, thus the user do not have to know how to implement them. MS SQL Server 2012 provides the UDT mechanism (User Defined Types) which was used for crucial elements such as image keypoints, dictionaries, or descriptors. All UDT types were programmed with custom serialization mechanisms. These types are stored in assemblies included in the database which is linked to our system. The software was based on .NET platform. Moreover, the additional advantage is the use of the Filestream technology which is included in MS SQL Server. As a consequence, reading high resolution images is much faster than with using classical methods. The aforementioned technology provides the interaction with image database, based on the content of appropriate folders (linked to FileTable objects), designed to storing images. Placing new images in these folders fires the adequate trigger. It gives the advantage of automatic initialization of the corresponding database objects without additional operations. Users have to indicate a query image to compare. As a result, the system returns the sequence of images similar to the content of the query image. The process of extending the set of indexed images in the database boils down to copying images to FileTable directories. Then, the dictionary and descriptors are be generated automatically after inserting the number of words for dictionary in an appropriate stored procedure. Figure 1 presents the architecture which was divided into four layers. In the first layer, the user selects a query image for transferring to the system over the remote WCF channel or by copying to the `Test` folder locally. After processing the query image, user obtains the response as the sequence of similar images (sorted in descending order from the most similar image). The second layer is an interface which allows to perform queries to the system database. The list of similar images consists of file paths from a database and similarity levels assigned to appropriate files. The third layer acts as the physical MS SQL Server database. This is the place of storing the information about the images and their descriptors. The table with descriptors is indexed to speed up generating response. At this level it is also possible to execute a stored procedure which contributes to running the bag-of-features algorithm and indicating similar images over the WCF endpoint. The last layer contains the WCF service functionality. Methods shared by the web service module run the main algorithms, generate keypoints and descriptors basing on the dictionary. Having the dictionary, it is possible to perform the similarity calculation procedure. The response collected from the system contains a sorted list which is transferred to the second layer. The list stores top_n most similar images, which can be accessed from the first layer.

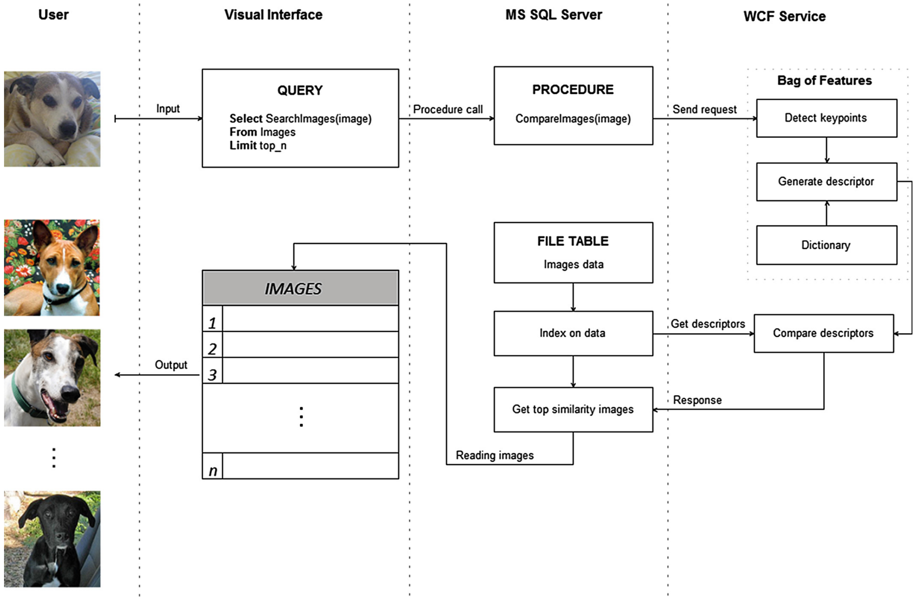

Fig. 1. System architecture.

2.2 System Functionality

The system was divided into modules, which are dedicated for specific functions. These modules include communication interfaces with other modules. The layered software implementation allows to modify some modules, without interfering with the other architecture parts of the system.

The domain model layer is a fundamental module for business logic of the system and was created with the Database First approach. Figure 2 presents the database diagram. Considering the integration of the applied mechanisms from .NET platform, Microsoft SQL Server 2012 was chosen. The database structure was designed based on the bag-of-features algorithm. Keypoints, dictionaries and descriptors were stored in the database as UDT (User Defined Types), for which serialization mechanisms were implemented. System functionality is mainly based on the bag-of-features algorithm [29]. This algorithm was chosen by the authors of this paper because of the relatively high effectiveness and fast operation. Keypoints are calculated using the SIFT method, nevertheless the system can use other visual feature generation techniques. The local features calculated for images are stored in the database along with the dictionary structures and descriptors generated basing on these dictionaries. This approach entails the requirement of only one generation of crucial data structures for the system. The `Images_FT` table was designed with the FileTable technology and contains images which are necessary for the training process. As a consequence, the entire content of this table influences cluster calculation and effectiveness of similarity detection.

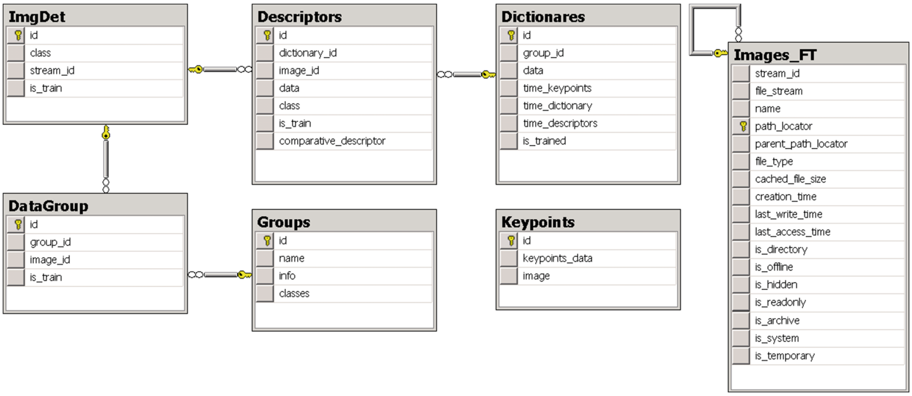

Fig. 2. Database diagram.

Query by image operation relies on the initial dictionary loading with appropriate identification number from the `Dictionaries` table. This operation is crucial for calculating descriptors for the adequate dictionary. The next procedure compares the query image descriptor with the other descriptors stored in the database. Vectors $\boldsymbol{x} = \{x_1, x_2, \cdots, x_n\}$ are generated for images from the database, and $\boldsymbol{y} = \{y_1, y_2, \cdots, y_n\}$ is calculated for the query image. The next procedure is responsible for comparing descriptors by the Euclidean distance. As a result, we determine the similarity factors for all comparisons sorted in descending order.

Our software has the functionality of classifying the query image, basing on the support vector machine (SVM) classifiers trained on descriptor collection. Information about the class membership can be used with the similarity results. The SVM classifiers are stored in the database after the process of training with the collection of descriptors from `Images_FT` table.

In an attempt to provide remote interaction with the system, we implemented SOA layer (Service Oriented Architectures) in .NET technology. To achieve this essential aim, WCF (Windows Communication Foundation) web service was programmed. In this case client software can execute procedures remotely. The system architecture also provides the distributed processing system, when a database server is situated in a different physical location. Hence, we implemented remotely executed WCF methods from stored procedures.

3 Numerical Experiments

In this section we present the results of example experiments performed to validate the correctness of the system. We used images taken from the PASCAL Visual Object Classes (VOC) dataset [7]. We queried the database with images and the returned images are shown with the distances to the query descriptor.

The first part of the tests was performed for query images which were not included in the database. When an image is presented on the system input, the response vector $R = S(x, y1), S(x, y2), ..., S(x, yN)$ obviously did not include similarity values being equal zero. It contained k similar images from an appropriate class of the query image. Figure 3 presents the example with several returned images. The next experiments consisted in showing images which had the exact representation in the database (in `Images_FT` table), i.e. they were included in the dictionary building process. In this case, the response vector obviously included the output with m values equal zero, when m indicates the amount of identical images contained in the database. If the request was configured for including the k similar images, when $k > m$, then response vector should comprise $k > m$ values greater than zero. Figure 4 shows an example of querying the database with an image that existed in the database.

Fig. 3. Querying test performed for an image which is not included in the database. The distance to the query image is given for each returned image.

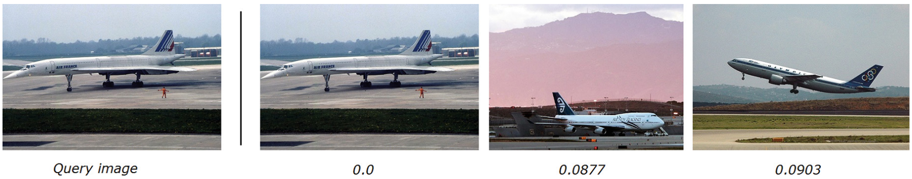

Fig. 4. Querying test performed for an image which was included in the database. The distance to the query image is given for each returned image.

4 Conclusions

We developed a system dedicated to image retrieval by providing an integrated environment for image analysis in a relational database management system environment. Nowadays RDBMS are used for collecting very large amount of data, thus it is crucial to integrate them with content-based visual querying methods. In the proposed system computations concerning visual similarity are encapsulated in the business logic of our system, users are only required to have knowledge about communication interfaces included in the proposed software. Applying database indexing methods affects positively speeding up the image retrieval.

Moreover, our system is integrated with .NET platform. The authors chose the WCF technology for providing the remote interaction with the system. MS SQL Sever allows to attach assemblies implemented in .NET to the database dedicated for image analysis. As a consequence, users can interact with the system locally by SQL commands, which execute remote procedures. It is an important advantage of the system. The system retrieves images in near real-time.

References

1. Akhtar, Z., Rattani, A., Foresti, G.L.: Temporal analysis of adaptive face recognition. J. Artif. Intell. Soft Comput. Res. **4**(4), 243–255 (2014)
2. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (SURF). Comput. Vis. Image Underst. **110**(3), 346–359 (2008)
3. Chang, T., Kuo, C.C.: Texture analysis and classification with tree-structured wavelet transform. IEEE Trans. Image Process. **2**(4), 429–441 (1993)
4. Chaudhuri, S., Narasayya, V.R.: An efficient, cost-driven index selection tool for Microsoft SQL server. VLDB **97**, 146–155 (1997)
5. Chu, J.L., Krzyzak, A.: The recognition of partially occluded objects with support vector machines and convolutional neural networks and deep belief networks. J. Artif. Intell. Soft Comput. Res. **4**(1), 5–19 (2014)
6. Drozda, P., Sopyła, K., Górecki, P.: Online crowdsource system supporting ground truth datasets creation. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2013, Part I. LNCS, vol. 7894, pp. 532–539. Springer, Heidelberg (2013)
7. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (VOC) challenge. Int. J. Comput. Vis. **88**(2), 303–338 (2010)
8. Francos, J., Meiri, A., Porat, B.: A unified texture model based on a 2-D Wold-like decomposition. IEEE Trans. Signal Process. **41**(8), 2665–2678 (1993)
9. Grauman, K., Darrell, T.: Efficient image matching with distributions of local invariant features. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 2, pp. 627–634, June 2005
10. Huang, J., Kumar, S., Mitra, M., Zhu, W.J., Zabih, R.: Image indexing using color correlograms. In: 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Proceedings, pp. 762–768, June 1997
11. Jagadish, H.V.: A retrieval technique for similar shapes. SIGMOD Rec. **20**(2), 208–217 (1991)
12. Jain, A.K., Farrokhnia, F.: Unsupervised texture segmentation using gabor filters. Pattern Recogn. **24**(12), 1167–1186 (1991)
13. Kanimozhi, T., Latha, K.: An integrated approach to region based image retrieval using firefly algorithm and support vector machine. Neurocomputing **151**, 1099–1111 (2015)
14. Karakasis, E., Amanatiadis, A., Gasteratos, A., Chatzichristofis, S.: Image moment invariants as local features for content based image retrieval using the bag-of-visual-words model. Pattern Recogn. Lett. **55**, 22–27 (2015)
15. Kauppinen, H., Seppanen, T., Pietikainen, M.: An experimental comparison of autoregressive and Fourier-based descriptors in 2D shape classification. IEEE Trans. Pattern Anal. Mach. Intell. **17**(2), 201–207 (1995)

16. Kiranyaz, S., Birinci, M., Gabbouj, M.: Perceptual color descriptor based on spatial distribution: a top-down approach. Image Vis. Comput. **28**(8), 1309–1326 (2010)
17. Korytkowski, M., Scherer, R., Staszewski, P., Woldan, P.: Bag-of-features image indexing and classification in Microsoft SQL server relational database. In: 2015 IEEE 2nd International Conference on Cybernetics (CYBCONF), pp. 478–482 (2015)
18. Korytkowski, M., Rutkowski, L., Scherer, R.: Fast image classification by boosting fuzzy classifiers. Inf. Sci. **327**, 175–182 (2016)
19. Larson, P., Clinciu, C., Hanson, E.N., Oks, A., Price, S.L., Rangarajan, S., Surna, A., Zhou, Q.: SQL server column store indexes. In: Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data, pp. 1177–1184. ACM (2011)
20. Lin, C.H., Chen, H.Y., Wu, Y.S.: Study of image retrieval and classification based on adaptive features using genetic algorithm feature selection. Expert Syst. Appl. **41**(15), 6611–6621 (2014)
21. Liu, G.H., Yang, J.Y.: Content-based image retrieval using color difference histogram. Pattern Recogn. **46**(1), 188–198 (2013)
22. Liu, J.: Image retrieval based on bag-of-words model (2013). arXiv preprint arXiv:1304.5168
23. Liu, S., Bai, X.: Discriminative features for image classification and retrieval. Pattern Recogn. Lett. **33**(6), 744–751 (2012)
24. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
25. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. Image Vis. Comput. **22**(10), 761–767 (2004). British Machine Vision Computing 2002
26. Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. Int. J. Comput. Vis. **60**(1), 63–86 (2004)
27. Murata, M., Ito, S., Tokuhisa, M., Ma, Q.: Order estimation of Japanese paragraphs by supervised machine learning and various textual features. J. Artif. Intell. Soft Comput. Res. **5**(4), 247–255 (2015)
28. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2006, vol. 2, pp. 2161–2168. IEEE, Computer Society, Washington, DC (2006)
29. O'Hara, S., Draper, B.A.: Introduction to the bag of features paradigm for image classification and retrieval (2011). arXiv preprint arXiv:1101.3354
30. Pass, G., Zabih, R.: Histogram refinement for content-based image retrieval. In: Proceedings of the 3rd IEEE Workshop on Applications of Computer Vision, WACV 1996, pp. 96–102, December 1996
31. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2007, pp. 1–8, June 2007
32. Rafiei, D., Mendelzon, A.O.: Efficient retrieval of similar shapes. VLDB J. **11**(1), 17–27 (2002)
33. Rashedi, E., Nezamabadi-pour, H., Saryazdi, S.: A simultaneous feature adaptation and feature selection method for content-based image retrieval systems. Knowl. Based Syst. **39**, 85–94 (2013)
34. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to sift or surf. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 2564–2571, November 2011

35. Shrivastava, N., Tyagi, V.: Content based image retrieval based on relative locations of multiple regions of interest using selective regions matching. Inf. Sci. **259**, 212–224 (2014)
36. Sivic, J., Zisserman, A.: Video google: a text retrieval approach to object matching in videos. In: Proceedings of the 2003 Ninth IEEE International Conference on Computer Vision, vol. 2, pp. 1470–1477, October 2003
37. Śmietański, J., Tadeusiewicz, R., Łuczyńska, E.: Texture analysis in perfusion images of prostate cancer–a case study. Int. J. Appl. Math. Comput. Sci. **20**(1), 149–156 (2010)
38. Srinivasan, J., De Fazio, S., Nori, A., Das, S., Freiwald, C., Banerjee, J.: Index with entries that store the key of a row and all non-key values of the row. US Patent 6,128,610, 3 October 2000
39. Veltkamp, R.C., Hagedoorn, M.: State of the art in shape matching. In: Lew, M.S. (ed.) Principles of Visual Information Retrieval, pp. 87–119. Springer, London (2001)
40. Voloshynovskiy, S., Diephuis, M., Kostadinov, D., Farhadzadeh, F., Holotyak, T.: On accuracy, robustness, and security of bag-of-word search systems. In: IS&T/SPIE Electronic Imaging, International Society for Optics and Photonics, p. 902807 (2014)
41. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 1794–1801, June 2009
42. Zhang, J., Marszalek, M., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: a comprehensive study. In: 2006 Conference on Computer Vision and Pattern Recognition Workshopp, CVPRW 2006, p. 13, June 2006
43. Zitnick, C.L., Dollár, P.: Edge boxes: locating object proposals from edges. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part V. LNCS, vol. 8693, pp. 391–405. Springer, Heidelberg (2014)