Novel Algorithm for Translation from Image Content to Semantic Form

Janusz Rygał[1], Jakub Romanowski[1], Rafał Scherer[1], and Sohrab Ferdowsi[2]

[1] Institute of Computational Intelligence, Częstochowa University of Technology
al. Armii Krajowej 36, 42-200 Częstochowa, Poland
{janusz.rygal,jakub.romanowski,rafal.scherer}@iisi.pcz.pl
http://iisi.pcz.pl

[2] University of Geneva, Computer Science Department,
7 Route de Drize, Geneva, Switzerland
http://sip.unige.ch

Abstract. In this paper we present a new algorithm for translating visual information into a semantic form. In our approach we try to combine these two separate areas of computer since into one process. The main goal is to achieve very good performance at searching for similar images. In this paper we explain in details the design of the translation algorithm which is only one part of the whole process, but the most important one. This module is some kind of interface between information in the form of digital image and the information represented by lexems. We will also concisely demonstrate the structure of the whole SIA (Semantic Image Analysis) project.

Keywords: semantic translation, image transformation, semantic image analysis, CBIR.

1 Introduction

Nowadays we are surrounded by the immensity of information. Because of computers and power of digital processing we are able to analyze more and more information in a shorter period of time. Humans are already very good at searching and analyzing information in a textual form, everyone of us is familiar with using popular Internet search engines. In the area of digital image processing we are still far away from the quality and effectiveness of text retrieval. Translation of the information in the form of digital image into text is one of the most important parts of the SIA project [25]. To enable semantic analysis [25] on the data representing the image we have to convert it into text and this conversion has to be a deterministic and stable process [19]. Before we were ready to establish our translation algorithm we had had to prepare the whole preprocessing module, which is responsible for filtering out some unimportant information and noises from the image and fetching the most important one [3][13][14][21][24][27]. Fetched data are translated by our translation formula and saved in the database in the form of the vector of lexems. Our approach is designed to be applied on

L. Rutkowski et al. (Eds.): ICAISC 2014, Part I, LNAI 8467, pp. 783–792, 2014.

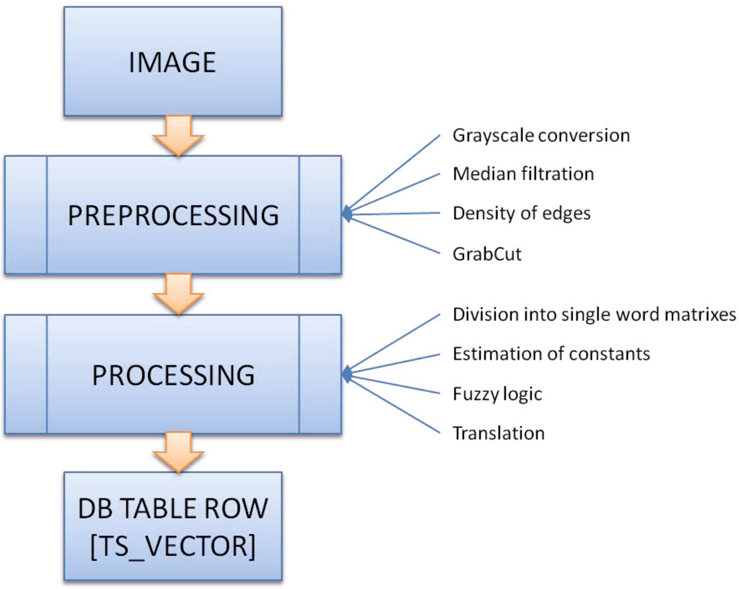

Fig. 1. Simplified structure of the conversion process used by the proposed approach

every kind of images but it could be specialized to process only specific classes of image objects [4] or to images with specific features [1][9][10][26] which may be classified by multiple methods [5][6][7][8][16][17][20][22].

2 Image Processing Algorithm

As we see in Fig. 1, the whole process of the image conversion into the semantic form consists in two general parts – preprocessing and processing. Results of our research concerning each preprocessing step have been already described in [24][27]. In this paper we are focused on our latest research about the translation algorithms. This part of the process is also divided into stages what is described in following subsections.

2.1 Division into Single Word Matrixes

The output from the preprocessing part is the grayscale image with marked important parts, which were selected as significant areas of the image [2][15]. Relying on the information enclosed in that image, we need first to select the ROI (Region Of Interest), which is a rectangle containing the significant areas of the image and then restrict that rectangle by searching for a shape which is the best pattern for areas representing the important parts of an image. The selected shape has to be divided into areas, which we called SWM (Single Word

Matrix). The question was how large these matrixes have to be. In our approach we assumed that the number of rows and columns of the SWM will be equal to 5% of the height and width of the selected ROI. Thanks to this operation we can keep our process robust for images containing the same objects in a different scale.

2.2 Estimation of Constants

In the result, each SWM has to be represented by a single word, so we had to find a method to convert a matrix of pixels into one word. What was very important, this algorithm had to be stable, that means that conversion of the matrix representing the same part of the image has to be always transformed into the same word. Because of that we employed markers, which represent the constant properties of that area. In the current implementation we used the following markers:

1. Percentage amount of edges contained by the matrix,
2. Average value of the intensity of pixels contained by the matrix.

At the beginning of the research we decided that this part has to be generic, which means, that adding new markers has to be feasible. That is why in the next planed phase of our project we can combine diverse markers to achieve the best selectivity of similar images.

2.3 Fuzzy Logic

We used fuzzy logic mechanisms by conversion the values of single SWM represented by array of markers. To each marker we assigned some number of levels, to this levels we round every marker value to reduce diversity and the number of possible combinations.

2.4 Translation

To translate the values of markers into single word, we used an English dictionary. Each array of markers, which represents the single SWM is converted by the SHA-1 algorithm into a hash code. Then the algorithm searches in the dictionary if there exists already a word to which this hash code was exactly assigned, if yes then the SWM becomes this word, if not, then to this value of hash code is assigned the next free word from the dictionary. The whole image is represented by the row of single word matrixes separated by the space characters.

3 Structure of the Conversion Mechanism

In this paragraph we present each step of the conversion. Description of each step is connected with the input and output information, so the order of presentation

Fig. 2. Original test image

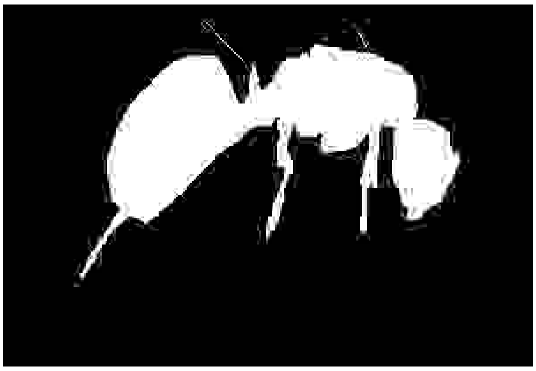

Fig. 3. Image representing the result of the preprocessing phase

is very important. The original test image is shown in Fig. 2. As already mentioned, in this paper we are focused on the processing part of the approach. The resulting image at the output of the preprocessing part is presented in Fig. 3. Now we present the steps of the algorithm:

1. In the first step we have to divide the selected ROI into SWMs. The number of rows and columns is equal to 5% of the height and width of the selected ROI. Fig. 4 presents the image with marked ROI divided into single word matrixes.
2. In the next part of the conversion we have to filter out these matrixes, which do not contain any pixels which were marked as the significant and important information. In our approach it is performed by finding the maximum value of intensity of pixels included into matrix, when this maximum is equal 255, then the whole matrix is selected as important one. Fig. 5 shows only these matrixes, which were marked as containing significant information and were selected for further computation.

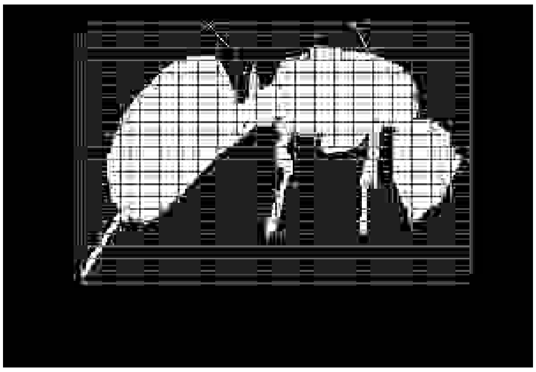

Fig. 4. Image representing the ROI divided into single word matrixes

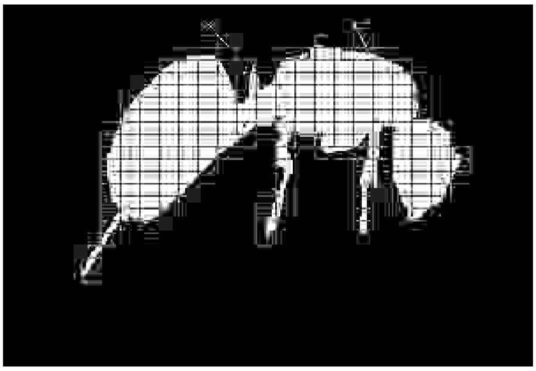

Fig. 5. Image representing selection of important single word matrixes

3. Having selected list of SWMs we are ready to compute constant markers for every matrix. Our algorithm was designed in such a way that adding new markers is an easy and fast action, but in this situation we need to compute values for every image in the database. In the current implementation we used the percentage value of edges in the matrix and average intensity of pixels. Thus, for one SWM these values can take values as follows:
   - Percentage value of edges: 22
   - Average value of intensity: 120
4. When we have already estimated values of constant markers for every SWM, we have to compute the hashcode, which is going to be the representation of these values. For this computation we use a SHA1 algorithm, e.g.:
   SHA1(22,120) = 231151646136144130502039643226195140178156113132386
5. In the next step we need to find a word which will be the representation of our hash code, if our hash code already exists in the database, then we will take the word which is assigned to this hash code, otherwise we will take the first one without assigned a word.

At the end of the process, each SWM will have assigned a word. All the words will be combined into a single text, separated by the space characters. Such prepared text will be the textual representation of the image. But it is not the last step of conversion, during adding the new image with the textual representation to the database we call a trigger which converts the text into lexems. The name of this form of data in PostgreSql database system is TS_VECTOR. Below we present an example of the structure with this form of information:

TS_VECTOR = airfar 127 aisl 289 alarmist 346 albedo 304 alert 201,217

We see words followed by certain numbers. These words are not ordinary words, but lexems, which are normalized words to make different variants of the same word look similarly. Numbers following the lexems show the position of the lexem occurrence.

4 Structure of the Research Environment

For our research we have developed the next version of the software which is responsible for the whole SIA [25] process. In this paper we are focused on the processing part, so the following information concerns only this phase. Our software called CBIR 4.0 is divided into server and database layers. CBIR 4.0 software was written using C# .NET 4.0 and PostgreSQL 9.0 technologies. In addition, for the purposes of image processing we use Emgu.CV-2.3.0 library. Because the end result of the conversion mechanism is a database item, it is worth to describe in details the database structure and functionalities. In Fig. 6 we presented the main part of the database structure of the conversion system. As we see in Fig. 6, the most important tables in the database are table "DICTIONARY" and table "IMAGE". Table "DICTIONARY" contains 109.583 rows, which represent single words in English language (obtained from the SIL Organisation, available at `http://www.sil.org`). Each item with filled field "HASHCODE" is already assigned to specific values of constant markers. Data from this table are strongly used and updated during the conversion of images into semantic form. Table "IMAGE" contains of course already transformed images, the most important field of the table is "TSV_BODY", which contains data representing digital image in the form of TS_VECTOR, which is the sorted list of lexems. Database table "IMAGE" together with the trigger, the trigger function and the GIN index create the independent fast text engine construction (PostgreSQL 9.1.2 Documentation, available at `http://www.postgresql.org/`). By each update or insert on the table "IMAGE" the trigger is started, which calls the trigger function, while the main task of this function is to convert data into TS_VECTOR and save them in silent mode inside table in the field "TSV_BODY".

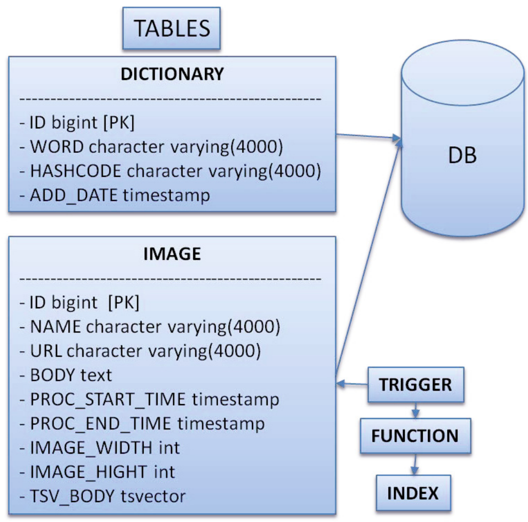

Fig. 6. The most important parts of the database structure

5 Experimental Results

For tests of our algorithm and system we chose a set of images representing different types of objects. Several of them represent different objects but of the same type [12]. For example there are multiple images representing ants in various scale, on diverse background and with other coexisting objects. By performing these tests we wanted to check following properties of our approach: determinism, performance and structure of converted data. Presented statistics and properties describe our test on 100 of test images from Caltech 101 images data base. Total number of words generated in dictionary is 109.583 with number of distinct words used for conversion the first 100 images in amount of 5.536. During conversion of first 100 images there were used 28.835 words. Average size of processed images is 232 pixels height and 294 pixels width and disk size of table containing 100 converted images in number 1264 kB. Average time used for the whole process of conversion is 2686 ms.

It is worth to mention that we also tested that the same image converted twice gives the same results. With this test we proved the deterministic behavior of our algorithm. More than 90% of time used for conversion is the preprocessing phase, the GrabCut algorithm is the the slowest part of the process.

6 Conclusions

With this paper we proved that the idea of conversion of images from pixels into words is not only an idea but the real way to combine two important branches of computer science. Because of usage of fast text search engine we obtained the possibility of very fast and effective analysis of data representing images. Because of advanced database structure and functions we have also possibility of applying fuzzy logic on the level of single table row, representing single words. It is possible to establish the level of similarity between two words. By extending the function we can achieve a connection between similarities of two words to similarity of constant markers representing the single word matrix what is an exact area of the image. This property gives us the possibility of effective applying fast text search engine analytic functions. The proposed algorithm of translation fulfilled all requirements, which we specified at the beginning of our research. The new approach is stable, deterministic, fast and easy to extend, these properties allow us to look in the future of the SIA with promising expectations. As already mentioned this is one of the last parts of our broad SIA project, but also the most important one. In the next step of our research we will develop new search and rank functions. Also we noticed the opportunity to expand our algorithms of SWMs classification by using neural networks [11][18][23] but it is one of additional functionality of the SIA. Our novel algorithm of translation is an important milestone on our way to establish fast and effective way to find similar images to the one given on the input. Our alfa tests already provided satisfying results and we hope that in a short period of time we will be ready to publish results of the whole SIA mechanism.

Acknowledgements. The project was funded by the National Center for Science under decision number DEC-2011/01/D/ST6/06957.

References

1. Bazarganigilani, M.: Optimized image feature selection using pairwise classifiers. Journal of Artificial Intelligence and Soft Computing Research 1(2), 147–153 (2011)
2. Blake, A., Rother, C., Brown, M., Perez, P., Torr, P.: Interactive image segmentation using an adaptive GMMRF model. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 428–441. Springer, Heidelberg (2004)
3. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. 8(6), 679–698 (1986)
4. Chang, Y., Wang, Y., Chen, C., Ricanek, K.: Improved image-based automatic gender classification by feature selection. Journal of Artificial Intelligence and Soft Computing Research 1(3), 241–253 (2011)
5. Cpalka, K., Rutkowski, L.: Flexible takagi-sugeno fuzzy systems. In: Proceedings of the 2005 IEEE International Joint Conference on Neural Networks, IJCNN 2005, vol. 3, pp. 1764–1769 (July 2005)
6. Cpalka, K., Rutkowski, L.: A new method for designing and reduction of neuro-fuzzy systems. In: 2006 IEEE International Conference on Fuzzy Systems, pp. 1851–1857 (2006)
7. Cpalka, K.: A new method for design and reduction of neuro-fuzzy classification systems. IEEE Transactions on Neural Networks 20(4), 701–714 (2009)
8. Cpalka, K., Rutkowski, L.: Flexible takagi sugeno neuro-fuzzy structures for non-linear approximation. WSEAS Transactions on Systems 4(9), 1450–1458 (2005)

9. Drozda, P., Górecki, P., Sopyla, K., Artiemjew, P.: Visual words sequence alignment for image classification. In: ICCI*CC, pp. 397–402. IEEE (2013)
10. 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 (LNAI), vol. 7894, pp. 532–539. Springer, Heidelberg (2013)
11. Duda, P., Jaworski, M., Pietruczuk, L., Scherer, R., Korytkowski, M., Gabryel, M.: On the application of fourier series density estimation for image classification based on feature description. In: Proceedings of the 8th International Conference on Knowledge, Information and Creativity Support Systems, Krakow, Poland, November 7-9, pp. 81–91 (2013)
12. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples. In: IEEE Proc. CVPR Workshop on Generative-Model Based Vision (2004)
13. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 3rd edn. Prentice-Hall, Inc., Upper Saddle River (2006)
14. Hocenski, Z., Vasilic, S., Hocenski, V.: Improved canny edge detector in ceramic tiles defect detection. In: IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics, pp. 3328–3331 (November 2006)
15. Mai, F., Hung, Y.S., Zhong, H., Sze, W.F.: A hierarchical approach for fast and robust ellipse extraction. Pattern Recogn. 41(8), 2512–2524 (2008)
16. Nowicki, R.: Rough-neuro-fuzzy system with micog defuzzification. In: 2006 IEEE International Conference on Fuzzy Systems, pp. 1958–1965 (2006)
17. Nowicki, R.: On classification with missing data using rough-neuro-fuzzy systems. International Journal of Applied Mathematics and Computer Science 20(1), 55–67 (2010)
18. Nowicki, R., Rutkowski, L.: Soft techniques for bayesian classification. In: Neural Networks and Soft Computing, pp. 537–544. Springer (2003)
19. Zhang, N., Watanabe, T.: Text-transformed image classification based on data compression. In: Gaol, F.L. (ed.) Recent Progress in DEIT, Vol. 1. LNEE, vol. 156, pp. 365–370. Springer, Heidelberg (2013)
20. Przybył, A., Cpałka, K.: A new method to construct of interpretable models of dynamic systems. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part II. LNCS (LNAI), vol. 7268, pp. 697–705. Springer, Heidelberg (2012)
21. Rui-ling, D., Qing-xiang, L., Yu-he, L.: Summary of image edge detection. Optical Technique 3, 415–419 (2005)
22. Rutkowski, L., Cpalka, K.: Neuro-fuzzy systems derived from quasi-triangular norms. In: Proceedings of the 2004 IEEE International Conference on Fuzzy Systems, vol. 2, pp. 1031–1036 (July 2004)
23. Rutkowski, L.: Non-parametric learning algorithms in time-varying environments. Signal Processing 18(2), 129–137 (1989)
24. Rygał, J., Najgebauer, P., Romanowski, J., Scherer, R.: Extraction of objects from images using density of edges as basis for grabCut algorithm. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2013, Part I. LNCS (LNAI), vol. 7894, pp. 613–623. Springer, Heidelberg (2013)

25. Rygał, J., Najgebauer, P., Nowak, T., Romanowski, J., Gabryel, M., Scherer, R.: Properties and structure of fast text search engine in context of semantic image analysis. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part I. LNCS (LNAI), vol. 7267, pp. 592–599. Springer, Heidelberg (2012)
26. Sopyła, K., Drozda, P., Górecki, P.: SVM with CUDA accelerated kernels for big sparse problems. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part I. LNCS (LNAI), vol. 7267, pp. 439–447. Springer, Heidelberg (2012)
27. Tabolt, J.F., Xu, X.: Implementing grabcut (2006)