Image Indexing and Retrieval Using GSOM Algorithm

Marcin Gabryel[1(✉)], Rafał Grycuk[1], Marcin Korytkowski[1], and Taras Holotyak[2]

[1] Institute of Computational Intelligence, Częstochowa University of Technology
Al. Armii Krajowej 36, 42-200 Częstochowa, Poland
{marcin.gabryel,rafal.grycuk,marcin.korytkowski}@iisi.pcz.pl
http://iisi.pcz.pl

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

Abstract. Growing Self Organized Map (GSOM) algorithm is a well-known unsupervised clustering algorithm which a definite advantage is that both the map structure as well as the number of classes are automatically adjusted depending on the training data. We propose a new approach to apply it in the process of the image indexation and retrieval in a database. Unlike the classic bag-of-words (BoW) algorithm with k-means clustering, it is completely unnecessary to predetermine the number of classes (words). Thanks to that, the process of indexation can be fully automated. What is more, numerous modifications of the classic algorithm were added, and as a result, the retrieval process was considerably improved. Results of the experiments as well as comparison with BoW are presented at the end of the paper.

1 Introduction

Effective browsing and searching large image databases based on their content is one of the most important challenges of computer science. It is required in many various fields of life e.g. medicine, architecture, forensic, publishing, fashion, archives and many others. The aim can be a retrieval of a similar image. Retrieving mechanisms use image recognition methods. This is a sophisticated process which requires the use of algorithms from many different areas such as computational intelligence [21] in particularly fuzzy systems [5, 6, 13], rough neuro-fuzzy systems [18, 19], evolutionary algorithms [7, 12, 16] and mathematics [22] and image processing [3, 17].

One of the most popular and widely spread algorithm used for indexation and images retrieval is the bag-of-words model (BoW) [8, 14], known also as a bag of features. This algorithm is based on a concept of text search methods within collections of documents. Single words are stored in dictionaries with emphasis on appearing in various documents. BoW in a similar way creates dictionaries of characteristic features appearing in images. Additionally, classification process

L. Rutkowski et al. (Eds.): ICAISC 2015, Part I, LNAI 9119, pp. 706–714, 2015.
DOI: 10.1007/978-3-319-19324-3_63

enables during the search to determine what type of image class we are dealing with. In [14] we can find a detailed description of BoW and associated algorithms.

While working with the classic BoW algorithm we can easily notice its drawback as it is required to determine an initial number of classes for the k-means clustering algorithm. Afterwards, a classifier is used (in most cases the Support Vector Machine algorithm) which task is to provide an arbitrary class to which a searched image belongs to [8]. In such model there is no possibility of returning a list of similar images stored in a database. The method which we present in contradiction to BoW does not require the knowledge of the number of classes creating words in a dictionary. It allows to find the most similar images to the examined one and the additional classifier is no longer required. One of modifications proposed in the paper is called neuronal activity thresholding and it is used during creating images histograms. Once the threshold is applied, neuron which is activated the least in the whole class is eliminated. The research proved that this fact significantly improves the classification efficiency. Ultimate decision concerning a class to which a given image belongs, is taken through the majority voting.

The article is divided into several parts. In Section 2 we can find familiar algorithms such as Speeded Up Robust Features (SURF) and GSOM. Those are the ones which we use in our method. In the following section there is also a description of our idea of an indexing images and creating new databases algorithms with the use of GSOM algorithm. In the last section we present the results of experiments as well as the summary of our work.

2 Algorithms Used in the Proposed Approach

The proposed method of image indexing and retrieval implements several algorithms. In this section we present the GOSM algorithm which is used for clustering and for the reduction of initial interest point number and the SURF algorithm which task is to find and describe those points.

2.1 SURF

SURF (Speeded Up Robust Features) is a robust local feature detector, first presented in [2] by Herbert Bay in 2006. It is partly inspired by the SIFT descriptor [15]. SURF gives description of the image by selecting characteristic keypoints. This method combines selection of keypoints with calculating 64-element vector (a descriptor). In SURF, integral image and filter approximation of block Hessian determinant is applied. To detect interesting points, a special Hessian-matrix approximation is used. For features, orientation is based on information from circular region around the pixel. Then, a square region aligned to selected orientation is constructed and the SURF descriptor is extracted from it. It uses the sum of the Haar wavelet responses around an interest point. The local feature around the point is described by a 64-number vector.

2.2 GSOM

GSOM algorithm (Growing Self-Organizing Map) was invented in 1995 [11]. In fact it is a modification of SelfOrganizing Map (SOM) which additionally was equipped in the ability of expansion. The number of neurons is adjusted to data during learning. There are many papers where we can find a description of the GSOM algorithm [23] or the Growing Hierarchical Self-Organizing Map (GHSOM) [9, 20]. In this section we present our modified version of the original GHSOM algorithm [11].

The network we consider consists of $n \times m$ neurons N_j, where $j = 1, ..., N_c$, N_c - number of neurons, $N_c = n \cdot m$, n - number of rows and m - number of columns. Initially, we create a network made of four neurons. They are placed on every vertex of a square. Learning data are vectors $\mathbf{x}_i = (x_{i1}, x_{i2}, ...x_{iK})$, where $i = 1, ..., M$, M – number of vectors, K - vector dimension). Every neuron N_j has a weight vector $\mathbf{w}_j = (w_{j1}, ...w_{jK})$. In our case, in order to allow parallel algorithm operation we used a modified version of the SOM algorithm. Single iteration of learning consists of six steps:

1. Set the vector of changes $\mathbf{dx}_i$ so that $dx_{jk} = 0$, $j = 1, ..., K$, $k = 1, ..., N_c$.
2. Repeat for every $i = 1, ..., M$ steps 3-6.
3. Find such a neuron N_s that fulfils the inequality

$$\|\mathbf{w}_s - \mathbf{x}_i\| \leq \|\mathbf{w}_j - \mathbf{x}_i\|. \tag{1}$$

4. Modify changes vector $\mathbf{dx}_s$

$$\mathbf{dx}_s = \mathbf{dx}_s + (\mathbf{x}_s - \mathbf{x}_i). \tag{2}$$

5. Increase the counter of winnings

$$\tau_s = \tau_s + 1. \tag{3}$$

6. Modify weight of all neurons

$$\mathbf{w}_j = \mathbf{w}_j - \alpha \cdot \mathbf{dx}_j / t_j, \text{ for } t_j \neq 0. \tag{4}$$

Afterwards, the expansion algorithm is used. Initially, a neuron N_q is found for which inequality is as follows:

$$\tau_q \geq \tau_j. \tag{5}$$

If the number τ_q exceeds a fixed number τ_{max}, what happens is the expansion of a network in one of four directions. We assume that the neuron N_f is a neuron which value τ_f is the highest of all neighbouring neurons N_q. Between neurons N_q and N_f a new column (or row) is inserted. Weights of a new neuron $N_{q'}$ are determined as follows:

$$\mathbf{w}'_q = 0.5 \cdot (\mathbf{w}_q + \mathbf{w}_f). \tag{6}$$

The number of columns m (or rows n) as well as the general number of neurons N_c are increased. All counters of winnings are reset

$$\tau_j = 0, \ i = 1, ..., N_c. \tag{7}$$

After the predetermined number of generated iteration there are N_c groups with the centres of clusters which coordinates are $\mathbf{w}_j$.

3 Indexing and Retrieval Using GSOM Algorithm

The presented algorithm consists of several modules. Just as in the case of BoW, we initially create a dictionary of words. Based on them we obtain histogram, containing the presence of those words in each single image.

1. Consider images database I_i, where $i = 1, ..., L$, L number of images.
2. Find the characteristic points (for example with SURF algorithm) x_{ij}, $i = 1, ..., L$, $j = 1, .., K$, K - the dimension of the vector describing characteristic point (for SURF $K = 64$).
3. Group the points x_{ij} with the use of GSOM algorithm. Obtain group centres $\mathbf{w}_j$, $j = 1, ..., N_c$.
4. Create histograms $h_i = [h_{i1}, h_{i2}, ..., h_{iN}]$, where

$$h_{ij} = \sum_{j=1}^{N_c} \delta_{ji}, \tag{8}$$

$$\delta_{ji} = \begin{cases} 1 \text{ if } \|\mathbf{w}_j - \mathbf{x}_k\| \le \|\mathbf{w}_k - \mathbf{x}_k\| \text{ for } \mathbf{x}_k \in \mathbf{I}_i \\ 0 \text{ otherwise} \end{cases}. \tag{9}$$

   Variable δ_{ji} is an indicator if a neuron N_j is the closest vector (a winner) for any sample from an image $\mathbf{I}_i$.

A novel, active neurons thresholding method is used in the presented algorithm. This approach is responsible for calculating the activity of neurons for every class of image. After that, the element of histogram h_{ij}, which on account of too weak activity of neuron N_j in a given class did not reach the threshold θ, is reset. The algorithm consists of two steps:

1. For every class c calculate the activity of neurons α_{jc}

$$\alpha_{jc} = \sum_{j=1}^{N_c} \delta_{ji}, \ \mathbf{I}_i \subset c. \tag{10}$$

2. If inequality

$$\alpha_{jc} < \theta \tag{11}$$

   is satisfied, then

$$h_{ji} = 0, \text{ for } \mathbf{I}_i \subset c. \tag{12}$$

Our experiments clearly confirm that removing inactive neurons boosts the classification efficiency. In the next chapter we depicted detailed results.

Image search $\mathbf{I}_t$ in the database consists in preparation of a histogram $\mathbf{h}_t$ in accordance with the algorithm described above. Next, we look for the closest histograms calculating the L1 norm between them:

$$d_{fi} = \sum_{j=1}^{K} |h_{fi} - h_{ji}|, \ i = 1, ..., K. \tag{13}$$

Ultimate decision concerning a class of an image $\mathbf{I}_t$, is taken through the majority voting. An image $\mathbf{I}_t$ belongs to the class c if in this particular class is the highest number of sorted L/C images with regard to distance d_{fi}.

4 Experiments

We present two experiments to show the effectiveness of the algorithm. In the first one we present to what extent applying the neuron activation threshold in the histogram creation improves search and classification of images. The second experiment shows the efficiency of the new algorithm in comparison to the BoW algorithm implemented with the use of functions available in the OpenCV library [4]. The research is performed on the Caltech 101 image database (collected by L. Fei-Fei et al. [10]). Four categories of images were chosen: planes, cars, cats and motorcycles. For every category images were divided into the learning part (80% of the available number of images) and the testing part (20%).

Our algorithm was implemented in Java language with the use of parallel computing (Concurrent library) as well as JavaCV [1] library function. JavaCV is a library which adopts functions available in OpenCV for Java language needs. We used this function in order to generate characteristic points with the use of the SURF algorithm.

Tables 1 and 2 present effectiveness of images classification for various values of neuron activity threshold. We can easily observe that adding this modification considerably improved classification efficiency. Graphs of these dependencies are presented in Fig. 2 and 3.

The algorithm was compared with the BoW algorithm which was implemented with standard functions of OpenCV according to [8]. In this particular algorithm points are grouped by the k-means clustering. Next, a classifier is learned by the Support Vector Machine algorithm (SVM). The results of single groups as well as comparison of all classes are presented in Table 3. The classification performed with the use of our algorithm is more efficient than the BoW with k-means and the classifier.

Table 1. Influence of neuronal activity thresholding on the effectiveness of classification for $\tau_{max} = 1000$

τ_{max}	treshold of neuron activity	efficiency %
1000	0	85.59
1000	5	85.59
1000	7	86.44
1000	10	86.44
1000	12	87.29
1000	15	85.59
1000	20	83.90

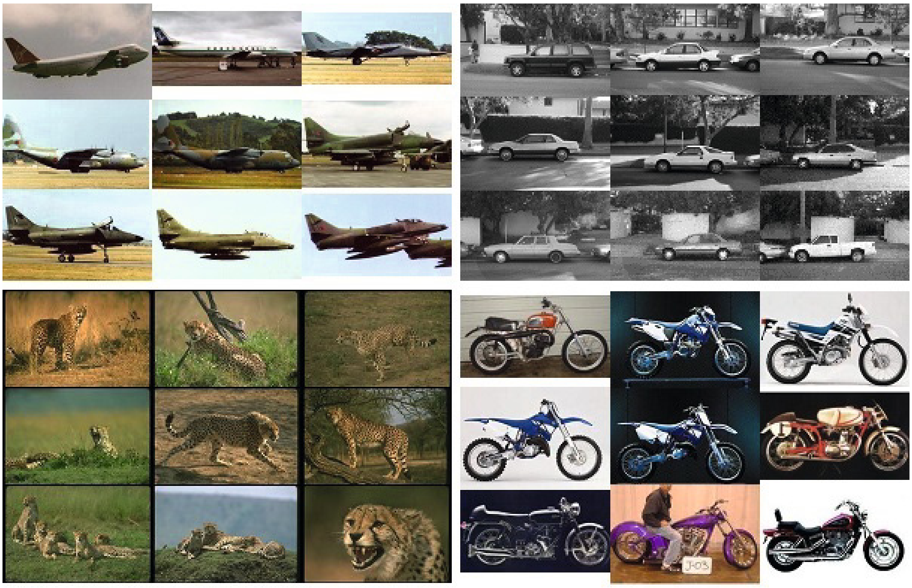

Fig. 1. Sample images from the Caltech 101 database for the four categories used in the experiments

Table 2. Influence of neuronal activity thresholding on the effectiveness of classification for $\tau_{max} = 2000$

τ_{max}	treshold of activity neurons	efficiency %
2000	0	86.44
2000	5	86.44
2000	10	86.44
2000	15	86.44
2000	20	86.44
2000	25	87.29
2000	30	87.29
2000	35	88.16
2000	40	86.44

Table 3. Comparison algorithm efficiencies for the test images

image type	classic BoW efficiency	our algorithm efficiency
plane	88.37%	93.18%
cars	82.61%	91.66%
wild cats	70.83%	84.00%
motorbike	91.66%	80.00%
all	84.21%	88.16%

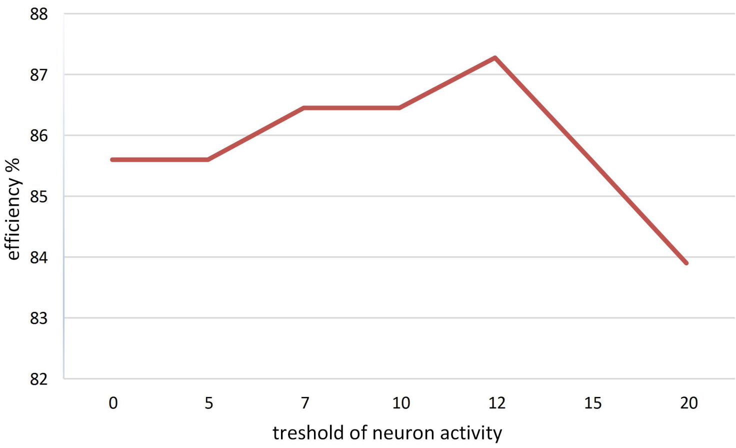

Fig. 2. Influence of neuronal activity thresholding on the effectiveness of classification for $\tau_{max} = 1000$

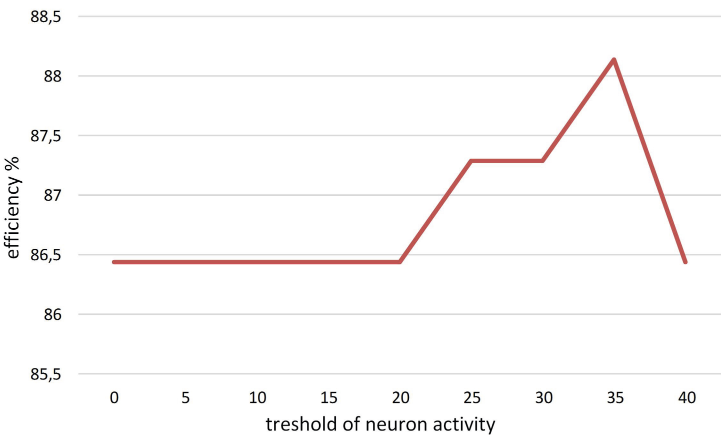

Fig. 3. Influence of neuronal activity thresholding on the effectiveness of classification for $\tau_{max} = 2000$

5 Final Remarks

As we have presented in the paper, content-based image classification with the GSOM algorithm, despite its clear and simple construction, has many advantages. What is worth mentioning, the lack of a classifier at the output does not cause any effectiveness loss. Moreover, the described thresholding method of inactive neurons contributed to increase in efficacy of the algorithm. In future, we would like to improve the effectiveness of the algorithm for greater number of classes. What is more, to modify inactive neuron handling in such a way that their number will be adjusted to every class separately. Our ultimate goal however is to implement this algorithm in a relational database (for example Microsoft SQL Server or MySQL) to expand its functionality with the possibility of content-based images search and retrieval.

Acknowledgements. This work was supported by the Polish National Science Centre (NCN) within project number DEC-2011/01/D/ST6/06957.

References

[1] Audet, S.: JavaCV (2014), http://bytedeco.org/ (Online; accessed December 1, 2014)

[2] Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded up robust features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006, Part I. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006), http://dx.doi.org/10.1007/11744023_32

[3] Biniaz, A., Abbasi, A.: Fast FCM with spatial neighborhood information for brain mr image segmentation. Journal of Artificial Intelligence and Soft Computing Research 3(1), 15–25 (2014)

[4] Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000)

[5] Chen, M., Ludwig, S.: Particle swarm optimization based fuzzy clustering approach to identify optimal number of clusters. Journal of Artificial Intelligence and Soft Computing Research 4(1), 43–56 (2014)

[6] 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)

[7] Cpalka, K.: On evolutionary designing and learning of flexible neuro-fuzzy structures for nonlinear classification. Nonlinear Analysis: Theory, Methods & Applications 71(12), e1659 – e1672 (2009), http://www.sciencedirect.com/science/article/pii/S0362546X09002831

[8] Csurka, G., Dance, C.R., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Workshop on Statistical Learning in Computer Vision, ECCV, pp. 1–22 (2004)

[9] Dittenbach, M., Merkl, D., Rauber, A.: The growing hierarchical self-organizing map. In: IEEE-INNS-ENNS International Joint Conference on Neural Networks, vol. 6, p. 6015. IEEE Computer Society (2000)

[10] Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In: Conference on Computer Vision and Pattern Recognition Workshop, CVPRW 2004, p. 178 (June 2004)

[11] Fritzke, B.: Growing grid – a self-organizing network with constant neighborhood range and adaptation strength. Neural Processing Letters 2(5), 9–13 (1995), http://dx.doi.org/10.1007/BF02332159

[12] Koshiyama, A.S., Vellasco, M.M.B.R., Tanscheit, R.: Gpfis-control: A genetic fuzzy system for control tasks. Journal of Artificial Intelligence and Soft Computing Research 4(3) (March 2015)

[13] Łapa, K., Przybył, A., Cpałka, K.: A new approach to designing interpretable models of dynamic systems. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2013, Part II. LNCS (LNAI), vol. 7895, pp. 523–534. Springer, Heidelberg (2013), http://dx.doi.org/10.1007/978-3-642-38610-7_48
[14] Liu, J.: Image retrieval based on bag-of-words model. CoRR abs/1304.5168 (2013), http://arxiv.org/abs/1304.5168
[15] Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
[16] Ludwig, S.: Repulsive self-adaptive acceleration particle swarm optimization approach. Journal of Artificial Intelligence and Soft Computing Research 4(3), 189–204 (2015)
[17] Najgebauer, P., Nowak, T., Romanowski, J., Rygał, J., Korytkowski, M., Scherer, R.: Novel method for parasite detection in microscopic samples. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2012, Part I. LNCS, vol. 7267, pp. 551–558. Springer, Heidelberg (2012)
[18] Nowicki, R.: Rough–neuro–fuzzy system with MICOG defuzzification. In: Proceedings of IEEE International Conference on Fuzzy Systems, IEEE World Congress on Computational Intelligence, Vancouver, BC, Canada, pp. 1958–1965 (July 2006)
[19] Nowicki, R.: Nonlinear modelling and classification based on the MICOG defuzzification. Journal of Nonlinear Analysis, Series A: Theory, Methods & Applications 7(12), e1033–e1047 (2009)
[20] Rauber, A., Merkl, D., Dittenbach, M.: The growing hierarchical self-organizing map: exploratory analysis of high-dimensional data. IEEE Transactions on Neural Networks 13(6), 1331–1341 (2002)
[21] Starczewski, J., Scherer, R., Korytkowski, M., Nowicki, R.: Modular type-2 neuro-fuzzy systems. In: Wyrzykowski, R., Dongarra, J., Karczewski, K., Wasniewski, J. (eds.) PPAM 2007. LNCS, vol. 4967, pp. 570–578. Springer, Heidelberg (2008)
[22] Woźniak, M., Kempa, W.M., Gabryel, M., Nowicki, R.K.: A finite-buffer queue with single vacation policy - analytical study with evolutionary positioning. International Journal of Applied Mathematics and Computer Science 24(4), 887–900 (2014)
[23] Zhu, G., Zhu, X.: The growing self-organizing map for clustering algorithms in programming codes. In: 2010 International Conference on Artificial Intelligence and Computational Intelligence (AICI), vol. 3, pp. 178–182 (October 2010)