

Content Based Image Retrieval Using a Bootstrapped SOM Network

Apostolos Georgakis* and Haibo Li

Digital Media Laboratory (DML)
Department of Applied Physics and Electronics,
Umeå University, SE-90187, Sweden
apostolos.georgakis@tfe.umu.se

Abstract. A modification of the well-known PicSOM retrieval system is presented. The algorithm is based on a variant of the self-organizing map algorithm that uses bootstrapping. In bootstrapping the feature space is randomly sampled and a series of subsets are created that are used during the training phase of the SOM algorithm. Afterwards, the resulting SOM networks are merged into one single network which is the final map of the training process. The experimental results have showed that the proposed system yields higher recall-precision rates over the PicSOM architecture.

1 Introduction

Image retrieval systems have become a necessity due to the mass availability of image information brought about by the spread of digital cameras and the Internet. The available systems today can be devised into two categories; the *keyword-based* systems and the *content-based* systems (CBIR). Keyword-based retrieval uses traditional database techniques to manage images. Using textual labels the images can be organized by topical or semantic hierarchies to facilitate easy navigation and browsing based on standard Boolean queries. Comprehensive surveys of early text-based image retrieval methods can be found in [1, 2]. Most text-based image retrieval systems require manual labeling which of course is a cumbersome and expensive task for large image databases. CBIR has been subjected to intensive research effort for more than two decade [3–5].

CBIR uses features related to the visual contents of an image such as color, shape, texture, and spatial layout to represent and index the images in the database (*training set*, \mathcal{I}_{tr}). Through this approach a typical image is described by a high-dimensional feature vector. The feature vectors corresponding to the images of the database form the feature space.

This paper provides a novel CBIR system which is based on the *self-organizing map* [6]. The proposed system is a variant of the *Picture SOM* (PicSOM) system which has been proposed by Laaksonen *et. al.* in [7]. The PicSOM system is a

* The work was supported by the Faculty of Science grand No. 541085100.

framework on which various algorithms and methods can be applied for content-based image retrieval. It relies on the so-called *Self-Organizing Map* (SOM).

In what follows, section 2 provides a brief description in feature extraction, section 3 covers the standard SOM algorithm while section 4 describes the presented variant. Finally, section 5 correspond to the experimental results related to the performance of the proposed system against the testbed system.

2 Feature extraction

Let \mathcal{I}_{tr} denote the image database. Let also $\mathbf{I}_i \in \mathbb{R}^{N_w}$, where N_w corresponds to the dimensionality of the feature vectors, denote the i th image in the database. In encoding each image into a numerical vector the following steps are taken:

- Each image is resized into a predetermined size.
- Each image is automatically segmented into a set of adjacent regions [8].
- Each region is encoded into a set of descriptors. The descriptors consist of property values which are defined over an entire region. The specific properties can be geometric, statistical or textural, or properties specified in some transform domain (*e.g.*, Fourier, Hough, Wavelet, Splines).
- All the descriptors corresponding to one image are packed together into on vector.

Since the PicSOM system corresponds to a framework for CBIR no particular preference is given on the descriptors employed. In this paper a wavelet transform is employed.

3 Self-Organizing Maps

The SOM are feed-forward, competitive artificial neural networks that were invented by T. Kohonen [6]. The neurons on the computational layer are fully connected to the input layer and are arranged on a low-dimensional lattice. Grids with low dimensionality have prominent visualization properties, and therefore, are employed on the visualization of high-dimensional data.

Let \mathcal{W} denote the set of reference vectors of the neurons, that is, $\mathcal{W} = \{\mathbf{w}_l(t) \in \mathbb{R}^{N_w}, l = 1, 2, \dots, L\}$, where the parameter t denotes discrete time and L is the number of neurons on the lattice. Due to its competitive nature, the SOM algorithm identifies the best-matching, winning reference vector $\mathbf{w}_s(t)$ (or winner for short), to a specific feature vector \mathbf{I}_j with respect to a certain distance metric. The index s of the winning reference vector is given by:

$$s = \arg \min_{l=1}^L \|\mathbf{I}_j - \mathbf{w}_l(t)\|, \quad (1)$$

where $\|\cdot\|$ denotes the Euclidean distance. The reference vector of the winner as well as the reference vectors of the neurons in its neighborhood are modified

toward \mathbf{I}_j using:

$$\mathbf{w}_i(t) = \begin{cases} \mathbf{w}_i(t) + a(t) [\mathbf{I}_j - \mathbf{w}_i(t)] & \forall i \in \mathcal{N}_s \\ \mathbf{w}_i(t) & \forall i \notin \mathcal{N}_s \end{cases} \quad (2)$$

where $a(t)$ is the learning rate and \mathcal{N}_s denotes the neighborhood of the winner and the transition between the time instants t and $t + 1$ is achieved whenever the entire feature space has been presented to the network.

3.1 Tree structured SOM

The sequential search of the winner neuron in both the training as well as the testing phase of the SOM algorithm imposes a severe speed bottleneck in any SOM manifestation that deals with either large data sets or high-dimensional spaces. Among the various SOM speed-up approaches that can be found in the literature prominent position has the so-called *tree structured SOM* (TS-SOM) [6]. In TS-SOM the features are represented in hierarchical 2D or 3D grids of neurons where each grid is a standard SOM. The tree topology reduces the complexity (both time and computation) for the identification of the winner neuron.

4 Bagging SOM

From Eq. (2) is evident that the standard SOM algorithm performs an approximation of the unknown pdf of the feature space. This approximation is evident in the updating of the reference vectors of the neurons comprising the lattice. A logical question that arise here is how to boost the performance of the approximation. In doing so one probable answer is the ensemble of predictors.

Recently a number of predictor combining have been proposed [9, 10]. Perhaps the simplest approach is to bag the predictors. This paper proposes a variant of the standard SOM algorithm which relies in *bagging*, that is, on an ensemble or combination of predictors. Bagging works by applying a learning algorithm on a number of bootstrap samples of the feature space. Each of these applications yields a clustering or classification. The resulting ensemble of classifiers is combined by taking a uniform linear combination of all the constructed classifiers.

Bootstrapping is a simple but also effective approach of estimating a statistic of the feature space. The method consists of creating a number of pseudo data subsets, $\mathcal{T}^i, i = 1, 2, \dots, D$, by sampling the set \mathcal{I}_{tr} with uniform probability with replacement of each sample.

Each instance of the standard SOM algorithm is then trained separately using one of the $\mathcal{T}^i, \forall i$ data subset. Afterwards the networks are “merged” in order to create a final network in a process which will be explained in the subsection 4.1. Due to the fact that the SOM networks are trained on modified instances of the feature space (the density functions of the subsets \mathcal{T}^i are expected to be

different), then, with high probability we expect to get slightly different resulted network topologies.

Petrakieva *et. al.* in [11] are outlining the problems that arise with the SOM algorithm when the reference vectors of the neurons are randomly initialized. Namely, the resulted network topologies, even in the case of using the same feature space, are not completely identical. Since the SOM algorithm performs a dimensionality reduction through the mapping of the feature space on the 2D lattice, the maps are free to twist in any direction which oftenly offers local minimizations.

4.1 Network merging

Let \mathcal{L}_{ij} denote the set of feature vectors that are assigned to the neuron \mathbf{w}_i^j . The first step which generally admittedly is the most difficult one is the combination of the outputs of the several SOM networks toward the formation of one final SOM map, that is to find the associations between the neurons $\mathbf{w}_i^j, \forall i, j$.

For example, let suppose that there are only three SOM networks and each is assigned to extract five clusters from the space \mathcal{I}_{tr} . The goal afterwards is to combine the networks in a way so that features to be placed in a same neuron of the final map if and only if they were assigned to a same neuron in both of the networks. This task is not trivial because there is no guarantee that the i th cluster in the first network corresponds to the i th cluster in the second network. So, the networks must be aligned before they can be merged.

In this paper, the neurons are aligned according to the assumption that neurons that are “similar” should be close to each other also in the \mathbb{R}^{N_w} . In reality the order is reversed; neurons that are close to each other in the \mathbb{R}^{N_w} should be similar. That is, lets $\mathbf{w}_{a_1}^{b_2}$ and $\mathbf{w}_{a_3}^{b_4}$ be the neurons whose sets $\mathcal{L}_{a_1}^{b_2}$ and $\mathcal{L}_{a_3}^{b_4}$ respectively contain more common features than any other neuron couple. Then, the reference vectors corresponding to these neurons will be, with higher probability, closer to each other than another possible neuron combination under the Euclidean distance.

In aligning the networks one should partition the LD neurons into L disjoint clusters with two constraints: a)each cluster will contain only D neurons, and b) each cluster will contain only one neuron from each of the D networks. The constrains simply state that each neuron of the output map is the union of just one neuron from each of the D constituent SOM networks. Under the above constrains is evident that a brute force approach to the global minimization problem has complexity $O(L^D)$ ¹ which obviously is unacceptable even for small values of the parameter D whereas the suboptimal solution described in subsection 4.2, which relies on dynamic programming, has complexity $O((D - 1)L^2)$.

¹ The reported computational complexity is due to the fact that we need to construct all the D -tuplets where each SOM network contributes with just one neuron per arrangement.

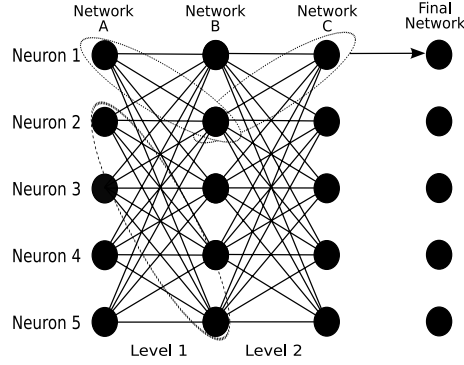


Fig. 1. The alignment of three networks and the subsequent merging into one final map.

4.2 Neuron alignment through dynamic programming

In ordering the neurons according to dynamic programming the proposed approach uses the paired distance of neuron pairs in the \mathbb{R}^{N_w} space. That is, if just two networks are to be merged then each neuron from the hypothetical network A should be matched with the neuron from the network B that would have been closer. Figure 1 depicts the above process. The first step is to merge networks A and B . In Fig. 1 it can be seen that the neuron \mathbf{w}_1^1 is closer to \mathbf{w}_2^2 and further more \mathbf{w}_2^2 is closer to \mathbf{w}_5^2 and so on. The average vector between each pair of neurons will be used afterwards in the second level to merge the third map (network C) into the previous two networks.

In merging the third map onto the previous two, one need to compute the distances between the average vectors from the previous level and the reference vectors from network C . In that case the pair $\{\mathbf{w}_1^1, \mathbf{w}_2^2\}$ is closer to \mathbf{w}_1^3 and therefore these three neurons are grouped together. After the last network has been merged with the previous two maps we need to “build” the final network (see Fig. 1). In doing so the reference vector of each neuron is the average vector of the neurons grouped together in the previous step.

The last step towards the formation of the final map is the creation of the set \mathcal{L}_{ifinal} which is the set of features assigned to the i th neuron of the final map. These sets are the unions of the sets corresponding to the clusters of neurons formed in the previous step. Let f_{ij} denote frequency of the j th image in the set \mathcal{L}_{ifinal} . If the frequency is close to the value D then more neurons in the constituent networks had that particular image assigned to them during the training phase. Therefore, the more frequent an image is the higher its importance to the particular neuron of the final map. That is, the images assigned to a particular neuron are ordered into descending order of frequency.

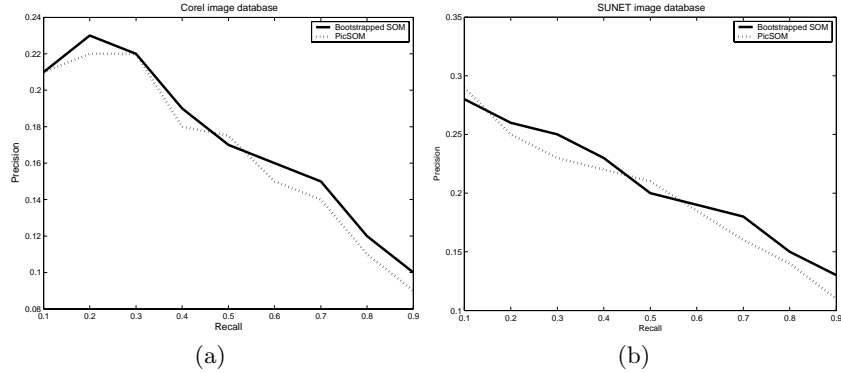


Fig. 2. The average recall-precision curves for the PicSOM and the proposed SOM variant for: (a) the Corel image database and (b) the SUNET image database.

5 Evaluating retrieval performance

The proposed bootstrapped SOM approach is evaluated against the PicSOM architecture with a set of experimental settings using two image databases. The first one corresponds to the Corel Gallery [12] and the second collection corresponds to the SUNET image database [13].

Aiming at assessing the retrieval performance of the proposed SOM variant against that of the basic SOM method used in PicSOM two retrieval systems are trained using the two image databases. Afterwards, the systems are queried using query-images randomly selected from the same datasets. The query-images undergo the same preprocessing steps as the images in the \mathcal{I}_{tr} .

For each image-based query, the system retrieves those training images that are represented by the best matching neuron of the final SOM map for both architectures. The retrieved images are by default ranked inside each neuron due to the process described in subsection 4.2. Finally, the retrieved images are labeled as either relevant or not to the query-images, with respect to the annotation category they bear. For each query, this classification leads to a different partition of the training set according to the retrieval volumes. The effectiveness of the proposed algorithm against the standard SOM is measured using the *precision* and the *recall* ratios [14, 15].

As the volume of retrieved images increases the above ratios are expected to change. The sequence of $(recall, precision)$ pairs obtained yields the so-called *recall-precision curve*. An average over all the curves corresponding to the same annotation categories that were obtained from the test set produces the average recall-precision curve [14].

Figures 2a and 2b depict the average recall-precision curves for the PicSOM architecture and the proposed SOM variant for all the annotation categories. It becomes evident that, in general, the bootstrapping of the feature space provides

superior performance over the standard SOM algorithm with respect to volume of retrieved images despite the isolated lags.

6 Conclusions

This paper has provided a variant of the well-know PicSOM architecture for content-based image retrieval. The proposed modification relies on bootstrapping. In bootstrapping, the feature space is randomly sampled and a series of subsets are created that are used during the training phase of the SOM algorithm. Afterwards, the resulted SOM networks are merged into one single network which is the final map of the training process. The experimental results have showed that the proposed system yields higher recall-precision rates over the PicSOM architecture.

References

1. Chang, S.K., Hsu, A.: Image information systems: where do we go from here? *IEEE Trans. on Knowledge and Data Eng.* **5** (1992) 431–442
2. Tamura, H., N.Yokoya: Image database systems: A survey. *Pattern Recognition* **1** (1984) 29–43
3. Bimbo, A.D.: *Visual Information Retrieval*. San Mateo, CA: Morgan Kaufmann (1999)
4. Long, F., Zhang, H.J., Feng, D.: Fundamentals of content-based image retrieval. In Feng, D., Siu, W.C., Zhang, H.J., eds.: *Multimedia Information Retrieval and Management - Technological Fundamentals and Applications*. Springer (2002)
5. Lew, M.S.: *Principles of Visual Information Retrieval*. Springer Verlag, Heidelberg, Germany (2000)
6. Kohonen, T.: *Self Organizing Maps*. 3rd edn. Springer Verlag, Heidelberg, Germany (2001)
7. Laaksonen, J., Koskela, M., Oja, E.: PicSOM - A Framework for Content-Based Image Database Retrieval using Self-Organizing Maps. In: *Proc. of 11th Scandinavian Conf. on Image Analysis (SCIA'99)*. (1999)
8. Pal, N.R., Pal, S.K.: A review on image segmentation techniques. *Pattern Recognition* **26** (1993) 1277–1294
9. Bakker, B., Heskes, T.: Clustering ensembles of neural network models. *Neural Networks* **12** (2003) 261–269
10. Breiman, L.: Using iterated bagging to debias regressions. *Machine Learning* **45** (2001) 261–277
11. Petrakieva, L., Fyfe, C.: Bagging and bumping self-organising maps. *Computing and Information Systems Journal* (2003)
12. Corel: 1.300.000 photo gallery. ("<http://www.corel.com>")
13. SUNET: Image database. ("<ftp://ftp.sunet.se/pub/pictures>")
14. Korfhage, R.R.: *Information Storage and Retrieval*. NY: J. Wiley (1997)
15. Sebastiani, F.: Machine learning in automated text categorization. *ACM Computing Surveys* **34** (2002) 1–47