Local Adaptive Receptive Field Self-organizing Map for Image Segmentation

Aluizio R. F. Araújo[1], Diogo C. Costa[2]

Universidade Federal de Pernambuco
Centro de Informática - CIn
Departamento de Sistemas da Computação
Av. Professor Luís Freire, s/n
Cidade Universitária
50740-540, Recife, PE, Brazil

e-mail: { aluizioa[1], dcc[2], }@cin.ufpe.br

Keywords: Self-organizing Map, Grow When Required, Radial Basis Function, Image segmentation, Border segmentation, Image processing

Abstract— A new self-organizing map with variable topology is introduced for image segmentation. The proposed network, called Local Adaptive Receptive Field Self-organizing Map (LARFSOM-RBF), is a two-stage network capable of both color and border segment images. The color segmentation stage is responsibility of LARFSOM which is characterized by adaptive number of nodes, fast convergence and variable topology. For border segmentation RBF nodes are included to determine the border pixels using previously learned information of LARFSOM. LARFSOM-RBF was tested to segment images with different degrees of complexity showing promising results.

1 Introduction

Pattern Recognition and Computer Vision tasks are usually divided in smaller problems or steps [5][13][3]. Figure 1 shows the five steps widely adopted to create an intelligent or automatic image recognition system which is seen as a computer vision problem.

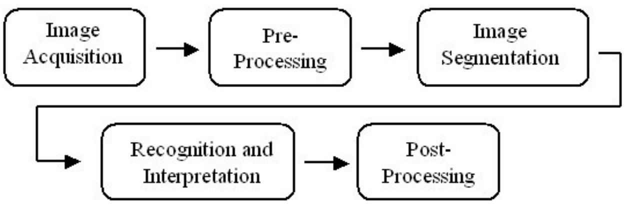


Figure 1 – Automated Image Recognition Processing Steps.

Data acquisition is the step in which the images generated by an imaging sensor are obtained. The pre-processing step aims to remove noise and disturbance. Image segmentation separates the objects or elements present in the data from the background. The recognition and interpretation step analyzes each segmented object to identify them. Finally, the post-processing yields a report or other kind of diagnostic about the acquired image. The present paper is focused on the segmentation step.

Segmentation is the process of dividing or separating (segmenting) an image in objects or elements that are coherent under some criteria [3]. This division process should occur until the desired information is correctly separated. The hardest goal in automatic segmentation is the determination of the moment to stop segmenting.

The segmentation techniques are often based on two images features [5][2]: discontinuity and similarity. Discontinuities are present in image regions where there are abrupt color changes. The most common techniques used for discontinuity detection are point detection, line detection and border detection. The latter is the most used one; Sobel and Prewitt Filters are broadly employed techniques. Similarities are found in image regions where there are not color changes or there are very slightly color changes. Common techniques for similarity segmentation in image processing are: thresholding and region growth.

A color value is defined as threshold and all image pixels above this value become white, and below become black. There are different ways to do thresholding, including intelligent algorithms [7][2]. There are also different ways to apply region growth segmentation [13]. The simplest one is to go trough all image pixels checking their color difference, then, different color regions are considered as object frontiers. The definition of the threshold and similarity values is not a trivial task.

Intelligent tools have been proposed [3][2], specially using neural networks, to determine threshold and

Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM 2007)
Published by the Neuroinformatics Group, Bielefeld University, Germany, ISBN 978-3-00-022473-7

similarity values, because of their capabilities of adapting themselves to environmental changes.

Many artificial neural network approaches have been presented that segment images directly from pixel similarity or discontinuity. In [3][2] we find two surveys treating image processing with neural networks and color image segmentation, respectively. In [3] more than 200 applications of neural networks in image processing are listed and a novel two-dimensional taxonomy for image processing algorithms is presented. The [2] summarizes many color image segmentation techniques such as histogram thresholding, characteristic feature clustering, edge detection, region-based methods, fuzzy techniques, neural networks approaches and others.

Recently, self-organizing map based techniques have been used. Color segmentation is successfully performed by SOM [9] related networks in the following works [15][4][11]. In [11] a two-stage strategy is used, first a fixed-size two-dimensional feature map (SOM) captures the dominant colors of an image in an unsupervised way, and then a second stage combines a variable-sized one-dimensional feature map and color merging to control the number of color clusters that is used for segmentation. The model in [4] is based on an unsupervised and supervised neural network approach. The unsupervised step is a SOM network to perform color reduction and then a simulated annealing seeks the optimal clusters from SOM prototypes. In the supervised step segmentation involves color learning and pixel classification in which a procedure of hierarchical prototype learning (HPL) is used to generate different sizes of color prototypes from the sample of object colors. The image pixels are classified by the matching of color prototypes. Edge detection based on SOM networks was also successful applied. In [14] aerial low contrast images were properly edge segmented using booth a self-organizing map (SOM) and a grayscale edge detector.

A color segmentation algorithm should be adaptive with respect to the number of remaining colors/objects. Fixed color algorithms often produce poor color segmentation results. SmART [15] is characterized by variable number of nodes which grows as new prototypes are needed. The new nodes are connected to two others under a triangle shape neighborhood. Instead of using the topological map to implement lateral plasticity control as SOM does, the topological relations between nodes work as an adaptive learning inhibitory function upon the prototype vectors.

Some color quantization implementations in neural network architecture have been proposed [12][1][3]. In fact, color quantization and color segmentation are based in the same process of reducing the image colors. The main difference is that color quantization usually results in a pre-defined final number of colors and the larger is the number of final color the better is the resultant image. However, in color segmentation each final color in the resultant image will represent an object, therefore only few colors are desired or else too many objects or subparts of objects will be detected.

The proposed algorithm tries to overcome the limitation of mentioned models [12][1][3][15][4][14][11]. LARFSOM is self-adaptive with respect to the number of final colors, a suitable feature to color segmentation, as opposed to fixed color quantization models [12][1][3] and even to the color segmentation model [11] with initial fixed size map. The proposed model does not need supervised training steps, as required by [4], then incremental learning is viable. Differently from SmART [15], LARFSOM-RBF does not have a restriction to a triangular shape neighborhood for new nodes, which limit nodes to at most four neighbors. LARFSOM-RBF topology freely grows and modifies itself during training. Therefore nodes may have as many neighbor nodes as necessary yielding an n-dimensional map. Finally, LARFSOM-RBF performs booth color and border segmentation in a fully unsupervised manner. Furthermore, in models such as [14], images need to be converted to grayscales before border segmentation, LARFSOM-RBF does not need it, then the resulting processing is faster.

The proposed model presents a new node insertion strategy based on similarity between an input pattern and the existing prototypes. The similarity is determined through an activation value threshold calculated with respect to a local neighbourhood receptive field value. As new nodes are added to the network, the topology is modified by creation and deletion of edges and nodes are free to have as many neighbours as necessary. In the border detections stage, RBF units detect object bundaries employing the local receptive field information extracted from LARFSOM trained nodes.

The remaining of this paper is organized as follows. Section 2 describes LARFSOM, as well as the two-stage LARFSOM-RBF network used for border segmentation. Section 3 shows the obtained results and discuss them. Finally, Section 4 brings the conclusion and future work.

2 The Model Description

The Local Adaptive Receptive Field Self-organizing Map (LARFSOM) takes advantage of nice characteristics of SOM [9] and Grow When Required (GWR) [10] networks. From SOM the competitive-learning and clustering capabilities are preserved as well as the topological distribution of learned data among the map neighbor nodes. As GWR the LARFSOM grows only when new nodes are required, based on an activation threshold, and the topology is also variable. However, LARFSOM is simpler than GWR in the sense that the

Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM 2007)
Published by the Neuroinformatics Group, Bielefeld University, Germany, ISBN 978-3-00-022473-7
All contributions to WSOM 2007 are available online at: http://biecoll.ub.uni-bielefeld.de

node winning counter is simpler calculated and only the best matching unit is trained.

The proposed segmentation technique is based on a two stage neural network. Firstly, LARFSOM is used for color quantization, reducing the amount of data to be processed and creating a color representation of the image. The acquired knowledge is then used for color segmentation. Secondly, Radial Basis Function [6] nodes are used to identify all edge pixels in the image, resulting in border segmentation. We call the whole model as LARFSOM-RBF.

2.1 Color Segmentation Algorithm

Color quantization [12][1][8] is a method to reduce the number of colors present in an image considering a minimal visual distortion. Color quantization has two steps: (i) autonomous selection of the most representative colors from all colors present in the original image to form the color palette; and (ii) mapping of each color in the original image to the nearest color in the palette. The final image will only have the selected colors and should be as similar as possible to the original one.

Color quantization process using LARFSOM is triggered by 3D-input vectors: values of red, blue and green to compose the possible colors for every pixel of an image, according to the RGB standard. As each pixel is an input to LARFSOM, then, the weight vectors are also 3D. The RGB standard values vary from 0 to 255. They are normalized (0 to 1) before given as input to the network.

LARFSOM has 10 steps: (1) Parameter initialization; (2) Selection of input pattern (pixel); (3) Best matching unit (BMU) search; (4) Connection insertion between two best units; (5) BMU local receptive field calculation; (6) BMU activity calculation based on receptive field; (7) Possible insertion of a new node; Else, update BMU weights; (8) Check stop criterion; (9) Build color palette; (10) Build color segmented image. These steps are detailed as follows.

Step 1: Parameter initialization: final learning rate (ρ_f), learning rate modulator (ε), activity threshold (a_T), number of wins of node i ($d_i = 0$), maximum number of wins of each node (d_m), the time iteration ($t = 0$), the minimum error ($e_{\min}$), and the initial number of connected nodes ($N = 2$), whose weights are copied from the RGB values of two randomly chosen image pixels.

Step 2: Present a randomly chosen image pixel $\boldsymbol{\xi} = [r\ g\ b]^T$ to the network as input data.

Step 3: Calculate the Euclidian distance between the sample $\boldsymbol{\xi}$ and the weight vectors ($\mathbf{w}_i$ s) as follows:

$$d(\boldsymbol{\xi}, \mathbf{w}_i) = \|\boldsymbol{\xi} - \mathbf{w}_i\|^2 \tag{1}$$

$$\|\boldsymbol{\xi} - \mathbf{w}_i\|^2 = (r - w_{ir})^2 (g - w_{ig})^2 + (b - w_{ib})^2 \tag{2}$$

Calculate the shortest distance between the input and all weight vectors to find the best matching unit (BMU):

$$d(\mathbf{w}_{s_1}, \boldsymbol{\xi}) \le d(\mathbf{w}_{s_2}, \boldsymbol{\xi}) \le d(\mathbf{w}_i, \boldsymbol{\xi}), \quad \forall i \in N \tag{3}$$

where N is the all-node set.

Increment the wins counter of BMU: $d_{s_1} = d_{s_1} + 1$

Step 4: Insert a new connection between s_1 and s_2 if it does not exist.

Step 5: Calculate the receptive field of s_1:

$$r_{s_1} = \sqrt{(w_{s_1r} - w_{s_2r})^2 + (w_{s_1g} - w_{s_2g})^2 + (w_{s_1b} - w_{s_2b})^2} \tag{4}$$

Step 6: Calculate the activity of s_1:

$$a_{s_1} = \frac{\exp(-\|\boldsymbol{\xi} - \mathbf{w}_{s_1}\|)}{r_{s_1}} \tag{5}$$

Step 7: Insert a new node if BMU activation is bellow a threshold (a_T), else update the BMU weight vector:

If $\quad a_{s_1} < a_T$

- Add a new node with weight vector $\mathbf{w}_n = \boldsymbol{\xi}$
- Update the number of nodes $N = N + 1$
- Remove the connection between $\mathbf{s}_1$ and $\mathbf{s}_2$
- Calculate the distances $d(w_n, w_{s_1}), d(w_n, w_{s_2}), d(w_{s_1}, w_{s_2})$
- Insert connections between nodes with the two smallest distances.

Else $\quad \Delta \mathbf{w}_{s_1} = \rho \times (\boldsymbol{\xi} - \mathbf{w}_{s_1}) \quad (6)$

$$\text{where } \rho = \begin{cases} \varepsilon \times \rho_f^{(d_i/d_m)}, & d_i \le d_m \\ \varepsilon \times \rho_f, & d_i > d_m \end{cases}$$

Step 8: Update the number of iterations $t = t + 1$ and return to step 2 unless if the stopping criterion is reached:

$$e = \frac{1}{N} \sum_{i=0}^{N-1} \|\mathbf{w}_{icurrent} - \mathbf{w}_{iformer}\|^2 \le e_{\min} = 10^{-4} \tag{7}$$

Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM 2007)
Published by the Neuroinformatics Group, Bielefeld University, Germany, ISBN 978-3-00-022473-7
All contributions to WSOM 2007 are available online at: http://biecoll.ub.uni-bielefeld.de

Step 9: After the training process, assign each color represented by a weight vector to a color in the palette.

Step 10: Replace each original pixel color in the image by its closest one in the color palette.

The number of nodes after the training process defines the number of palette colors. To reconstruct an image with the palette color, an interpolation method is proposed to generate more colors based on the palette ones. Then, a small network map, with just a few nodes is suitable for color segmentation of color full images reducing significantly the training time. If color interpolation is used Step 10 would be substituted Step 11.

Step 11: Replace each original pixel color in the image by a mean of the three first BMUs of the current input. Then, the interpolation for the red component of the image is:

$$R_j = \frac{\sum_{i\in\{s_1,s_2,s_3\}} m_{ir}\, d(\mathbf{w}_{s_i}-\xi)}{\sum_{i\in\{s_1,s_2,s_3\}} d(\mathbf{w}_{s_i}-\xi)} \quad (8)$$

where $R_j G_j B_j$ is the new color for the actual image pixel and m_{ir} is the color closest to the stimulus in the palette. The number of winners was found heuristically through tests with different winner numbers. Three nodes were satisfactorily fast and precise for interpolation.

2.2 Border Segmentation Algorithm

An edge or border is the limit between two regions sufficiently homogeneous in which discontinuity between them can be located just by analyzing the color changes. Edge detection is a technique based upon the detection of local discontinuities, often corresponding to the boundaries of objects in the image. The various options of edge detection of an image [5][13] aims to reduce the amount of data to be processed and filters out irrelevant information, preserving the main structural properties of an image. Edge detection is often the major step of image segmentation in computer vision systems. Figure 2 illustrates the two-stage procedure of the proposed model.

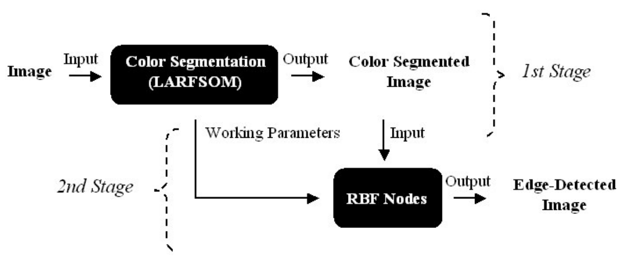


Figure 2 - Proposed two-stage edge detection algorithm.

The proposed edge detection algorithm is based on a two stage neural network. In the first stage, a particular image is color segmented by the LARFSOM. In the second stage, two RBF [6] nodes are responsible for identifying all pixels forming edges. The RBF nodes are not trained, i.e., their parameters are extracted from the LARFSOM weight vectors. Despite the absence of an explicit training stage, the RBF layer has its parameters adapted through the learning of LARFSOM, then, for different cluster formations, the RBF layer parameters are also updated.

After the color quantization stage, every pixel, n, of the quantized image is analyzed and classified according to the RBF nodes which are defined by complementary levels of activation (U_1, U_2). The current pixel, n, is an edge pixel if $U_1 < U_2$.

$$U_1 = \exp\left(-\frac{\left\|M_{past} - M_{fut}\right\|^2}{2r_n^{\,2}}\right) \quad (9)$$

$$U_2 = 1 - U_1 \quad (10)$$

where, r_n is the radius of the RBF function defined by Eq.11 as function of the output of LARFSOM whereas M_{fut} and M_{past} are the future and past pixel color averages, respectively, calculated by (13) and (14). To find vertical edges the pixels are consider from left to right, line by line. In this case the M_{fut} and M_{past} would be the right and left neighbor pixels value average, respectively. To seek for horizontal edges pixels are taken from top to bottom, column by column. When both edges are desired, the two previous procedures are executed.

The radius is calculates as follows.

$$r_n = \alpha\left\|C - C_{far}\right\|^2 \quad (11)$$

where $C = BMU(M_{fut})$ (12)

and $0 < \alpha < 1$, C is the quantized color, represented by the BMU LARFSOM, of the future pixel color average, and C_{far} is the farthest node of LARFSOM from the BMU node. The future pixel color average, M_{fut}, is used instead of the current pixel value to make the activation function more discriminating for local sharp contrasts, usually noise that should be removed. The radius, r_n, controls the color difference degree to identify an edge. The past and future edges are determined as follows:

$$M_{fut} = \left[\bar{R}(n), \bar{G}(n), \bar{B}(n)\right]^T \quad (13)$$

$$M_{past} = \left[\bar{\bar{R}}(n), \bar{\bar{G}}(n), \bar{\bar{B}}(n)\right]^T \quad (14)$$

Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM 2007)
Published by the Neuroinformatics Group, Bielefeld University, Germany, ISBN 978-3-00-022473-7
All contributions to WSOM 2007 are available online at: http://biecoll.ub.uni-bielefeld.de

where $\overline{R}(n)$ is the average of the red values of the future pixels and $\overline{\overline{R}}(n)$ is the average of the red values of the past pixels described by (15) and (16), respectively.

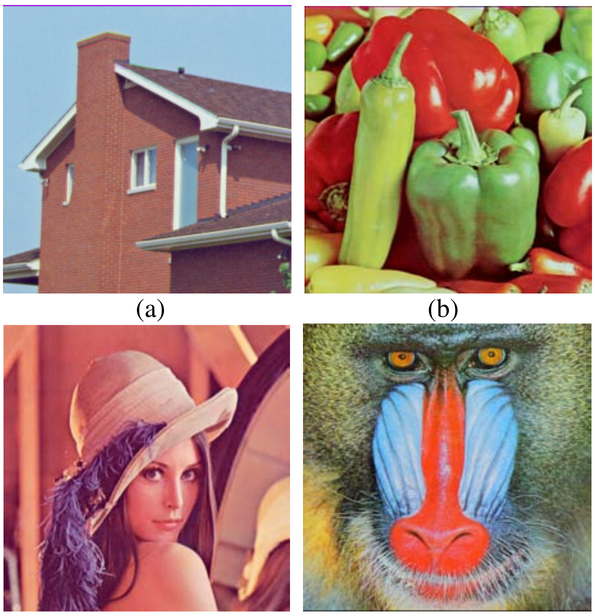
$$\overline{R}(n) = \rho R(n) + (1-\rho)\overline{R}(n+1) \qquad (15)$$

$$\overline{\overline{R}}(n) = \rho R(n-1) + (1-\rho)\overline{\overline{R}}(n-2) \qquad (16)$$

where $0 < \rho < 1$; $R(n)$ is the red value of the current pixel; ρ, defines the influence of future and past pixels upon the average. The green, $\overline{G}(n)$, $\overline{\overline{G}}(n)$, and blue, $\overline{B}(n)$, $\overline{\overline{B}}(n)$, averages are calculated as in Eq.15 and Eq.16.

3 Results and Discussions

This section presents the results of the proposed segmentation technique tested by means of four real world images[1]: house, pepper, Lena and baboon are shown in Figure 3 (a)(b)(c)(d), respectively.



(a) (b) (c) (d)

Figure 3 – The original images used to test the proposed algorithm: (a) house, (b) pepper, (c) Lena and (d) baboon.

Color segmentation complexity is different for the four images. The house (256x256 pixels) is the simplest one having 33,925 colors and few dominant and contrasting colors. Pepper (512x512) has 183,525 colors and more dominating colors, however the objects border are defined by color contrasts. Lena (512x512) has 148,279 colors and few dominant and similar colors then, the object segmentation is harder than the previous cases. Baboon (512x512) has 230,427 colors and plenty of soft and abrupt color changes, it is the hardest color segmentation.

The peak signal-to-noise ratio ($PSNR$) [1] is given by:

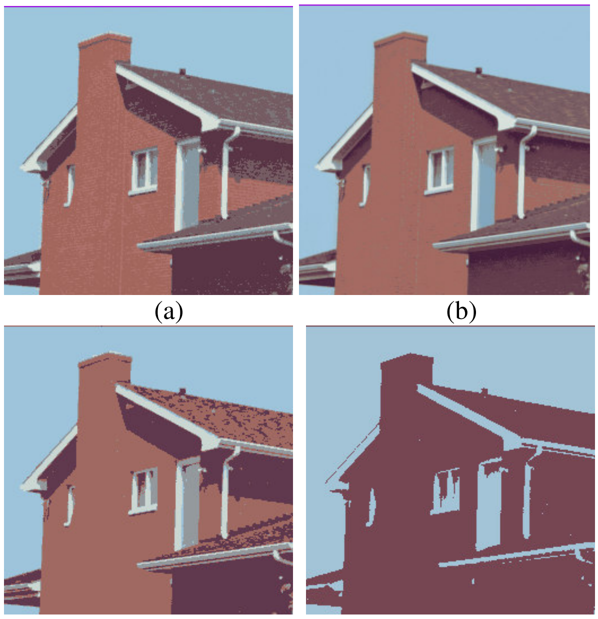
$$PSNR = 10\log\left(\frac{3\times 255^2}{MSE}\right) \qquad (17)$$

$$MSE = \frac{\left(\sum_{j=0}^{N_t-1}(X_j - X'_j)^2\right)}{N_t} \qquad (18)$$

where X_j and X'_j are the pixel values of the original and quantized image, and N_t is the total number of pixels. A higher PSNR value indicates a better quality image, usually above 30 is considered a good quality level [1]. The image with higher number of colors will have higher PSNR due to their similarity to the original images, however this does not ensure better segmentation.

3.1 Color Segmentation Results

LARFSOM parameters (ρ_f =0.05, $\varepsilon = 0.3$, d_m=100) were empirically chosen. Three activity threshold values (a_T) were used, 2.65, 1.65, and 1.0, also the color interpolation was illustrated for the 2.65 resultant image.



(a) (b) (c) (d)

Figure 4 – LARFSOM color segmentation of house image.

A combination of many parameters was tested with the images in Figure 3. The chosen parameters showed to achieve good results for all four images. Figure 4 to Figure 7 shows the results of the color segmentation performed by LARFSOM. Tables 1 to 4 provide further details such as threshold value, the use of the interpolation (step 11), and the final network parameters.

[1]House, Lena, pepper and baboon images were collected at the USC-SIPI Image Database: http://sipi.usc.edu/database/.

Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM 2007)
Published by the Neuroinformatics Group, Bielefeld University, Germany, ISBN 978-3-00-022473-7

Table 1 – Parameters for color segmentation of house image.

Image	a_T	Interpolation	Iterations	PSNR	Nodes	Colors
(a)	2.65	no	977	29.49	12	12
(b)	2.65	yes	977	31.67	12	3109
(c)	1.65	no	157	25.29	5	5
(d)	1.0	no	32	19.45	2	2

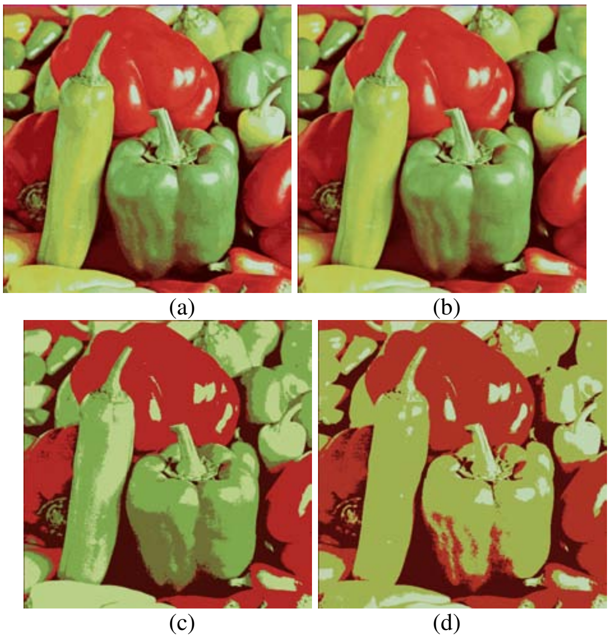


Figure 5 - LARFSOM color segmentation of pepper image.

Table 2 – Parameters for color segmentation of pepper image.

Image	a_T	Interpolation	Iterations	PSNR	Nodes	Colors
(a)	2.65	no	9018	28.12	26	26
(b)	2.65	yes	9018	30.60	26	12140
(c)	1.65	no	236	21.26	5	5
(d)	1.0	no	222	20.10	4	4

LARFSOM did not need many iterations to capture the image color distribution. Images having 262,144 pixels, for a_T =2.65, needed less than 10,000 iterations to learn. Despite the fast convergence, LARFSOM can determine the most significant colors (Figure 4 to Figure 7).

The self-adaptive number of nodes also appears as an interesting feature. Two nodes (Table 1) were enough to distinguish the house from the background. Pepper was easily segmented due to its very abrupt color contrast with four or five nodes. Also, four nodes segmented most of the objects of Lena image whereas the poorest segmentation of baboon was reached with four nodes. The PSNR values were also satisfactory due to the intense color reduction during the color segmentation.

Figure 6 - LARFSOM color segmentation of Lena image.

Table 3 - Parameters for color segmentation of Lena image.

Image	a_T	Interpolation	Iterations	PSNR	Nodes	Colors
(a)	2.65	no	434	26.94	9	9
(b)	2.65	yes	434	28.86	9	5266
(c)	1.65	no	117	23.57	4	4
(d)	1.0	no	73	21.64	3	3

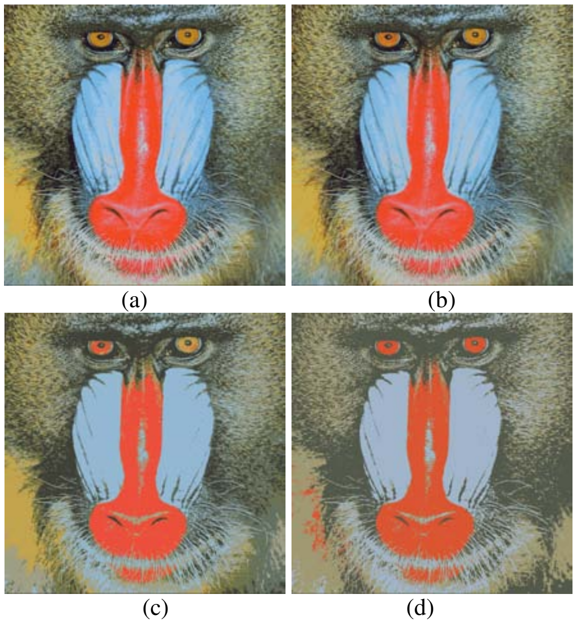


Figure 7 - LARFSOM color segmentation of baboon image.

Table 4 - Parameters for color segmentation of baboon image.

Image	a_T	Interpolation	Iterations	PSNR	Nodes	Colors
(a)	2.65	no	2453	25.98	26	26
(b)	2.65	yes	2453	28.64	26	23404
(c)	1.65	yes	364	21.47	7	7
(d)	1.0	no	224	19.39	4	4

Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM 2007)
Published by the Neuroinformatics Group, Bielefeld University, Germany, ISBN 978-3-00-022473-7
All contributions to WSOM 2007 are available online at: http://biecoll.ub.uni-bielefeld.de

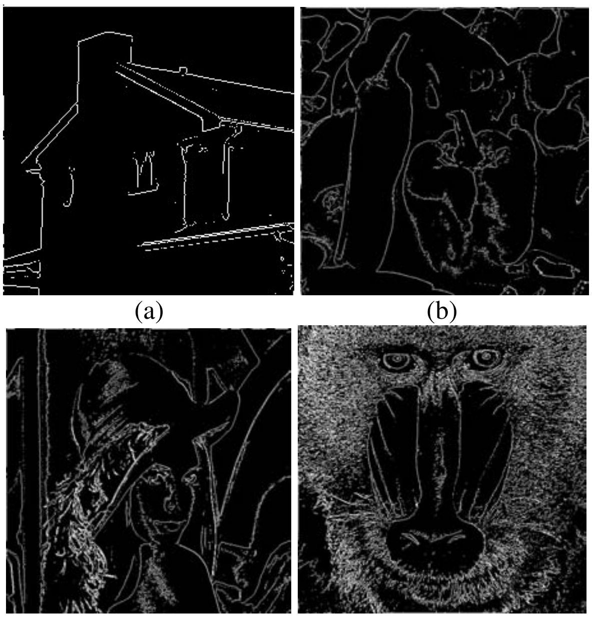
The interpolation of the palette colors generated many final colors making it harder the color segmentation. Even though, the interpolation capability may be very useful for precise image reconstruction.

3.2 Border Segmentation Results

The edge detection must neglect all color information and preserve the relevant structural information, i.e., only significant color changes should remain in the image. The edge detector was applied to the original images (Figure 3), after the color segmentation stage, to produce the results shown in Figure 8. The chosen parameters for color segmentation were the same as before, without color interpolation and a_T=1.65. The empirical border segmentation parameters were ρ =0.80, and α=0.66. Once more, different parametric combinations were tested to find a good parameter combination.

White pixels are borders whereas black pixels are not. In house and pepper images the border segmentation was very accurate. In Lena due to its soft color changes, edge detection is harder, but even so the most relevant edges were determined. Border segmentation of baboon is easy because it is formed of very contrasting colors.

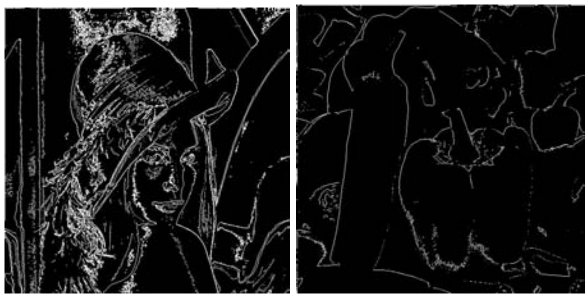
The RBF units can determine if a particular pixel is an edge. Filters like Sobel give different intensities of gray values to each pixel. Therefore a further processing is necessary to threshold the Sobel filtered image and chose the gray intensities to be defined as borders.



(a) (b) (c) (d)

Figure 8 – Results of applying two-stage LARFSOM-RBF border segmentation to images of Figure 3.

The parameters ρ and α allow fine-tuning for the edge detection criterion. The greater α the more selective the algorithm is; i.e., the growth of α causes sensitivity only to high contrasts. The parameter ρ controls the border width criterion; high values of ρ causes detection of thick edges. For instance, for α = 0.50 the number edges detected in (Figure 9 (a)) are higher than those in Figure 8(c). For ρ = 0.90, only wider edges were preserved in pepper image Figure 9 (b) as opposed to Figure 8(b).



(a) (b)

Figure 9 - Segmentation of (a) Lena image with ρ=0.80 and α=0.50 and (b) of pepper image with ρ=0.90 and α=0.66.

3.3 Further Comparisons

The most significant feature of LARFSOM is its fast convergence with suitable PSNR values. Comparing LARFSOM with SOM and SmART, one may notice these features. Table 5 shows[2] that LARFSOM needed 614 iterations and 0.01 seconds to be trained for Lena image leading to 12 nodes. SOM needed 21.232 iterations to converge with 12 nodes for the same image and its PSNR was significant lower. SOM parameters were initial learning rate of 0.02 and initial neighbourhood influence of 0.5 and the same convergence criteria as LARFSOM was used.

Table 5 – Compares SOM and LARFSOM performance for Lena image.

Network	Nodes	Iterations	PSNR	CPU Time (seconds)
SOM	12	21.323	25.56	0.32
LARFSOM	12	614	27.85	0.01

SmArt achieved a MSE rate for Lena image of 562.29 (PSNR of 25.40) with 12 nodes and learning rate of 0.1 as shown in [15], while the LARFSOM PSNR was 27.85.

4 Conclusion and Future Work

The usage of intelligent algorithms for image segmentation is a rich research field nowadays. Different systems were proposed in the literature, some of them presenting good results, however, often, they do not have fast, accurate and adaptive training steps. In this paper a

[2] The computer used for tests was a Pentium 4 with 2.66GHz and 512MB of RAM.

Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM 2007)
Published by the Neuroinformatics Group, Bielefeld University, Germany, ISBN 978-3-00-022473-7
All contributions to WSOM 2007 are available online at: http://biecoll.ub.uni-bielefeld.de

robust and fast Local Adaptive Receptive Field Self-organizing Map (LARFSOM), was presented to achieve these goals. Moreover the LARFSOM-RBF network features booth color and border segmentation making it very suitable for image processing systems.

The major contribution of LARFSOM-RBF among other recent and successful image segmentation models, like the SmART[15], for example, is the ability to booth color and border segment images fully based on unsupervised learning, while other approaches just perform on type of segmentation or need supervised steps.

The segmentation technique is based on a two-stage neural network. For color segmentation the clustering characteristics with growing number of nodes of LARFSOM is used. For border segmentation RBF nodes uses previously learned information of LARFSOM to determine the border pixels, therefore an LARFSOM - RBF two-stage network is proposed for the whole segmentation process.

Four different images with higher segmentation complexities were tested and successfully color segmented. The achieved results showed that LARFSOM is a very fast learner; just few training iterations were enough for the network to make an appropriate understanding of the images color distribution. It was shown in most of the time less than 1% of the image pixels needed to be randomly presented to LARFSOM in training step, for a fine color distribution understanding. The adaptive number of nodes was also a major feature of LARFSOM, once the number of objects in the images are prior unknown.

The edge detection algorithm showed to be effective in segmenting the four images. The fine-tuning parameters made it possible to adjust the edge detection selectivity as desired. The automatic adjustment of edge detection fine tuning parameters is of future interest. The idea is to incorporate these parameters as network input for self and adaptive optimum calibration.

References

[1] CH. Chang, P. Xu, R. Xiao, R. S. T. Srikanthan, “New adaptive color quantization method based on self-organizing maps“, *IEEE Transactions on Neural Networks*, Vol.16, No.1: pp.237-249, 2005.

[2] H.D. Cheng, X.H. Jiang, Y. Sun, J.L. Wang, "Color image segmentation: advances and prospects", *Pattern Recognition,* Vol.34, No.12, pp.2259-2281, 2001.

[3] D. deRidder, H. Handels, “Image Processing with Neural Networks–A Review”, *Pattern Recognition*, Vol.35, pp.2279–2301, 2002.

[4] G. Dong, M. Xie, "Color Clustering and Learning for Image Segmentation Based on Neural Networks", *IEEE Transactions on Neural Networks*, Vol.16, No.4, 2005.

[5] R.C. Gonzalez, R.E. Woods, *Digital Image Processing*, Edgard Blücher, 2000.

[6] D.S. Broomhead and D. Lowe, “Multivariable functional interpolation and adaptative networks”, Complex Systems, Vol. 11, pp. 321–355, 1988.

[7] C.V. Jawahar, P.K. Biswas and K. Ray, “Investigations On Fuzzy Thresholding Based On Fuzzy Clustering”, *Pattern Recognition*, Vol.30, No.10, 1997.

[8] K. Kanjanawanishkula, B. Uyyanonvara, "Novel fast color reduction algorithm for time-constrained applications", *Journal of Visual Communication & Image Representation*. Vol.16, pp.311–332, 2005.

[9] T. Kohonen, *Self-organizing Maps. 2nd edition*, Springer-Verlag. 1997.

[10] S. Marsland, J. Shapiro, U. Nehmzow, “A self-organizing network that grows when required”. *Neural Networks*, Vol.15 pp.1041-1058, 2002.

[11] S.H. Ong, N.C. Yeo, K.H. Lee, Y.V. Venkatesh, D.M. Cao, “Segmentation of color images using a two-stage self-organizing network”, *Image and Vision Computing*, Vol.20, pp.279–289, 2002.

[12] N. Papamarkos, A. E. Atsalakis, C. P. Strouthopoulos, “Adaptive color reduction”, *IEEE Transactions On Systems Man And Cybernetics Part B-Cybernetics* Vol.32, No.1, pp.44-56, 2002.

[13] J.R. Parker, *Algorithms for Image Processing and Computer Vision*, John Wiley and Sons, 1997.

[14] P.J. Toivanen, J. Ansamaki, J.P.S. Parkkinen, J. Mielikainen, "Edge detection in multispectral images using the self-organizing map", *Pattern Recognition Letters*, Vol.24, pp.2987–2994, 2003.

[15] N.C. Yeo, K.H. Lee, Y.V. Venkatesh, S.H. Ong, "Colour image segmentation using the self-organizing map and adaptive resonance theory", *Image and Vision Computing*, Vol.23, pp.1060–1079, 2005.

Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM 2007)
Published by the Neuroinformatics Group, Bielefeld University, Germany, ISBN 978-3-00-022473-7
All contributions to WSOM 2007 are available online at: http://biecoll.ub.uni-bielefeld.de