ELSEVIER

Pattern Recognition Letters 20 (1999) 1553–1563

Pattern Recognition Letters

www.elsevier.nl/locate/patrec

Image segmentation based on multi-scan constraint satisfaction neural network ☆

Fatih Kurgöllüs [a,*], Bülent Sankur [b]

[a] *Intelligent System Group, Tubitak Marmara Research Center, Information Technologies Institute, P.O. Box 21, 41470 Gebze, Kocaeli, Turkey*

[b] *Department of Electric and Electronic Engineering, Boğaziçi University, 80815 Bebek, Istanbul, Turkey*

Received 25 January 1999; received in revised form 27 August 1999

Abstract

A novel image segmentation method based on a constraint satisfaction neural network (CSNN) is presented. The new method uses CSNN-based relaxation but with a modified scanning scheme of the image. The pixels are visited with more distant intervals and wider neighborhoods in the first level of the algorithm. The intervals between pixels and their neighborhoods are reduced in the following stages of the algorithm. This method contributes to the formation of more regular segments rapidly and consistently. A cluster validity index to determine the number of segments is also added to complete the proposed method into a fully automatic unsupervised segmentation scheme. The results are compared quantitatively by means of a novel segmentation evaluation criterion. The results are promising.

Keywords: Image segmentation; Constraint satisfaction; Neural network; Multi-resolution

1. Introduction

Image segmentation is the process of dividing the given image into regions homogenous with respect to certain features, and which hopefully correspond to real objects in the actual scene (Gonzalez and Woods, 1993). The segmentation process is perhaps the most important step in image analysis since its performance directly affects the performance of the subsequent processing steps in image analysis. Despite its importance, segmentation still remains an unsolved problem in the general sense as it lacks a general mathematical theory. The two main difficulties of the segmentation problem are its underconstrained nature (LaValle and Hutchinson, 1995) and the lack of definition of the "correct" segmentation (Horn, 1986). Thus determination of the correctness and of the consistency of the segmentation result of a given scene become feasible only in specific tasks, e.g., knowledge-based and ground-truthed.

Perhaps as a consequence of these shortcomings, a plethora of segmentation algorithms have been proposed in the literature. These algorithms range from simple ad hoc schemes to more

☆ Electronic Annexes available. See www.elsevier.nl/locate/patrec.

* Corresponding author. Tel.: +90-262-6411-2300 ext. 4753; fax: +90-262-6431-87.

E-mail address: kurugol@mam.gov.tr (F. Kurgöllüs)

0167-8655/99/$ - see front matter
PII: S0167-8655(99)00128-2

sophisticated ones using object and image models. The segmentation algorithms fall into three main categories, namely, measurement-space-based algorithms (e.g. multi-thresholding of histograms, clustering methods) (Yen et al., 1995; Yeng and Chen, 1993; Tsao et al., 1994; Bensaid et al., 1996), pixel-based algorithms (e.g., region growing, edge detection) (Adams and Bischof, 1994; Canny, 1986), and physic-based schemes (Rubin and Richards, 1984). There exists another class of algorithms, in fact common to all categories, that use relaxation tools. These two-tiered algorithms consist of an initial assignment of label probabilities to all the pixels and the subsequent refinement of the segment probabilities. The refinement process, enacted via relaxation, has benefits of smoothing the region boundaries, eliminating inconsistencies, removing small and irrelevant regions, etc. Finally the relaxation process gives an ultimate opportunity to impose additional constraints, such as edges, multi-scale constraints, which one was not able to take into account in the initial measurement-space-based labeling.

Segmentation tasks often invoke parallel processing to handle the massive computation load (Ghosh et al., 1993). Such speed-ups are especially desirable in real time segmentation tasks such as in object-based video coding or in industrial inspection. Neural networks, in this respect, have found extensive application both due to their learning ability in clustering and classification problems, and their massively parallel structure. For example, neural networks are employed as a clustering tool in (Verikas et al., 1997; Uchiyama and Arbib, 1994), as a label relaxation tool in (Lin et al., 1992), as a constraint energy minimization tool in (Raghu et al., 1997), as a multi-thresholding tool in (Vinod et al., 1992).

Constraint satisfaction neural network (CSNN) is one of the recent proposed neural network-based segmentation algorithm (Lin et al., 1992). This original algorithm assumes that the number of classes in known and that initially pixel label likelihoods somehow have been assigned. The CSNN modifies iteratively the label assignments to reach a final segmentation map. One shortcoming of the original version of the CSNN algorithm, proposed by Lin et al. (1992), is that it tends to fluctuate around boundaries due to contradicting evidences. Furthermore the resulting segmentation error does not decrease beyond a certain level, as shown in the ground-truthed cases.

In this paper, we propose a multi-resolution version of the CSNN algorithm that contributes both to an increased speed in convergence and to smoother boundaries where the label oscillations do not occur any more. In addition, the CSNN segmentation algorithm is enhanced via an automated cluster validity process to determine the number of segments. Finally the architecture is modified to accommodate multi-spectral images. A novel segmentation evaluation criterion based on combining classification error and relative ultimate measurement accuracy which uses turning angle functions of regions is also proposed.

2. The constraint satisfaction neural network segmentation

The neural network structure used in the original algorithm (Lin et al., 1992) is depicted in Fig. 1. CSNN consists of layers each of which corresponds to a different segmentation map. The number of layers, M, is given as a priori information. Each layer has size $N \times N$, which is the size of the given image, the total resulting in $N \times N \times M$ neurons. In fact neurons in the layers hold the probability that each pixel belongs to the segment represented by the corresponding layer. As shown in Fig. 2, each neuron has connections to a neighborhood of eight neurons in its own layer as well as to a similar neighborhood in all the other layers. Thus total $N \times N \times M \times 8/2$ connects exist in the topology. These connections denote spatial constraints on the segment label of each pixel. While in (Lin et al., 1992) only the constraint of neighborhoods were operative, in (Kurugöllü et al., 1996) edge constraints were explicitly forced in. Similarly in (Kurugöllü and Sankur, 1998) pyramidal constraints, i.e., constraints from adjacent resolution levels were brought into play. Connection weights represent evidence or counter-evidence for label consistency. They are updated in such a manner that a neuron contributes to excite the other neurons that rep-

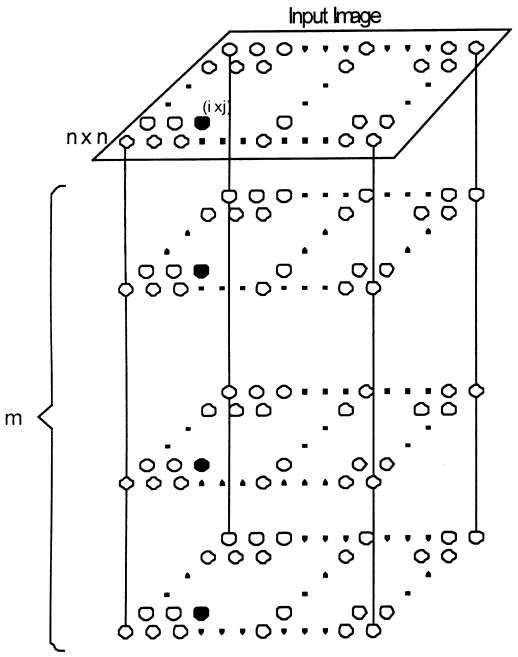

Fig. 1. Constraint satisfaction neural network topology. Each layer represents a segment. The (i,j)th neuron in each layer holds the probability that that pixel belongs to the segment represented by the layer.

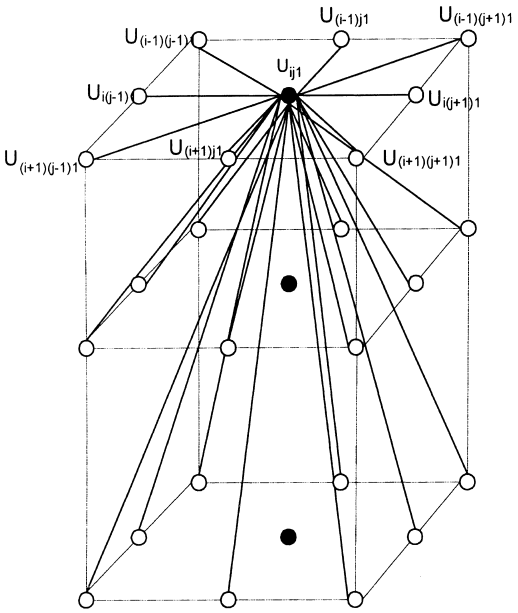

Fig. 2. Connections between a neuron and its neighbors. The weights of these connections are interpreted as constraints.

resent the same label, and inhibits the contradictory ones. The label probabilities grow or diminish in a winner-take-all style as a result of contention between segments. Each neuron in CSNN receives supports from its neighborhood as well as the feedback from its output. The update scheme is depicted in Fig. 3. The output of any neuron, O_{ijk}, depends on its previous state (feedback signal) and the neighbor contributions, H_{ijk}. Neighbor contributions come from the outputs of the neighbor neurons through connection weights that represent the spatial constraints. The contribution of the neighborhood is given as follows:

$$H_{ijk}^t = \sum_{l=1}^{m} \sum_{(q,r)\in N_{ij}} W_{ij,qr,k,l} O_{qrl}^t, \tag{1}$$

where the state of the (q,r)th neuron in layer l is reflected to the (i,j)th neuron in layer k via a weighting factor.

The nonlinear updating rule is as follows:

$$O_{ijk}^{t+1} = \frac{\mathrm{Pos}(O_{ijk}^t + \Delta O_{ijk}^t)}{\sum_{l=1}^{m} \mathrm{Pos}(O_{ijl}^t + \Delta O_{ijl}^t)}, \tag{2}$$

where

$$\Delta O_{ijk}^t = \begin{cases} \delta & \text{if } H_{ijk}^t = \max\left\{H_{ijl}^t\right\}, \\ -\delta & \text{otherwise}, \end{cases} \tag{3}$$

$$\mathrm{Pos}(x) = \begin{cases} x & \text{if } x \geqslant 0, \\ 0 & \text{otherwise}, \end{cases} \tag{4}$$

δ is a small positive number, typically chosen as 0.1 and it determines the step size between two

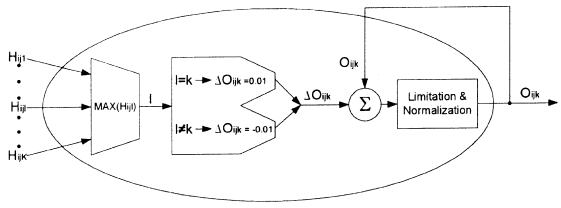

Fig. 3. Structure and update scheme of a neuron in CSNN. Input signals come from the outputs of neighbors as well as feedback from the neuron's output. The current output of the neuron is determined by means of a nonlinear function of the neighborhood contribution, H, and the feedback signal, O.

consecutive iteration epochs. The denominator term in Eq. (2) is the normalization factor which keeps the total output of the neuron residing in a specific location less than 1. The updating rule, given from (1) to (4), is based on two principles:

1. If H_{ijk} is the maximum one among all contributions at location (i,j), $H_{ij.}$, a positive contribution supports the U_{ijk}th neuron.
2. Otherwise, the output of U_{ijk}, O_{ijk}, is decremented.

The global satisfaction of the label assignment in the image domain gives rise to the final segmentation. After convergence of relaxation, the label with the largest accumulated weight is assigned to the corresponding pixel. The initial assignment of label probabilities can be effected via some algorithm, such as *K*-means, ISODATA, fuzzy *c*-means, or Kohonen's self-organizing map (SOM). In this study the algorithm is extended to RGB color features. The algorithm can in fact easily handle any multi-spectral image.

3. Multi-scan constraint satisfaction neural network

The novel method proposed, called Multi-Scan CSNN (MS-CSNN), aims to enhance the performance of CSNN on both the rate of convergence as well as the segmentation performance. The flow chart of the segmentation using MS-CSNN is depicted in Fig. 4.

The method consists of two stages. The first stage consists of a search for the most appropriate number of segments. To this effect the image is first analyzed with Kohonen's SOM algorithm. In each run of the algorithm a different number of segments is assumed, ranging from 2 to 6. Each resulting cluster map is evaluated using a cluster validity index proposed by Zhang and Modestino (1990). The resulting clustering map which yields the lowest validity index is called the "best sized cluster map" (size M), and it is passed on to the relaxation stage enacted by the "MS-CSNN" algorithm.

The second stage of the algorithm, called the MS-CSNN, updates via relaxation the segmentation labels obtained from the self-organizing map stage. The topology of the MS-CSNN algorithm is basically similar to that in Figs. 1 and 2. However, it differs in the scanning schedule of the pixels and their neighborhood sizes, which effectively emulate a multi-resolution approach. Notice that in this technique "scanning" means updating the label of specifically visited pixels via a CSNN using their neighborhood label information. Thus a different CSNN is run till convergence of each "resolution" stage. Each such CSNN has the same number of layers (M classes) as optimally determined with cluster validation. The CSNN structures differ however in the pixels they are handling at each stage and the corresponding neighborhood sizes (Fig. 5).

Another interpretation of the multi-scan approach is a pyramidal scheme, where the segmentation that takes place at the top level is propagated toward lower levels till the original lattice at the bottom. The higher levels are characterized by more sparse pixels and by larger neighborhoods. The lower levels have denser pixels to be visited but smaller neighborhoods as illustrated in Fig. 5. In the example of an image of 256×256, and five resolution stages, one updates at the top level, only every 16th pixel along rows and columns, each with its 11×11 neighborhood. Only the central visited pixels but not their neighborhoods are updated. Total connection in this level is $16 \times 16 \times M \times (11 \times 11 - 1)/2$. The third level will handle pixels spaced by 8×8, skipping the ones handled already in the fourth level. At the bottom level fully 128×128 pixel labels are updated, but each with 3×3 neighborhoods. The pixel labels that have been updated, let us say the 16×16 labels in the fourth level (Fig. 5(a)), are not changed any more in any of the

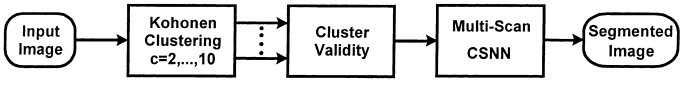

Fig. 4. Flow chart of the segmentation using multi-scan constraint satisfaction neural network.

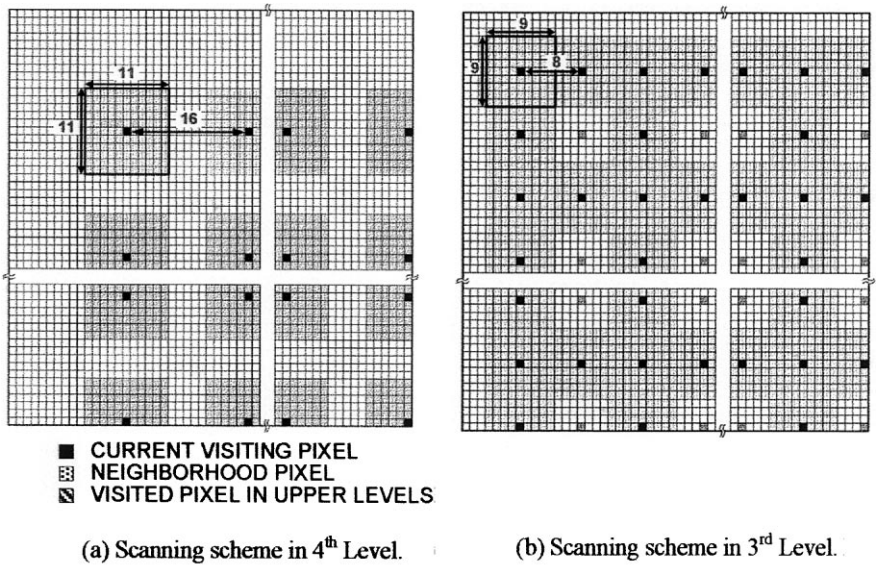

Fig. 5. Scanning scheme in the first two levels. The neighborhood of the scanned pixels is decreased in the lower levels as well as the scan step size.

lower levels, but they serve as neighborhood information for other pixels in the lower levels. This process consists of converging the labels of subsets pixels in each resolution level using a CSNN, and inheriting the fixated pixel labels from one level to the lower levels. In other words, the MS-CSNN begins the segmentation process from the top (fourth) level using the same dynamics as the CSNN algorithm. Whenever it converges to a solution, it begins another process in the subsequent lower level. This process continues until one reaches the bottom (0th) level.

We have found that five levels overall, including the original image (at level 0), were adequate. Levels from 4 to 0 are treated with a visiting schedule that gets sparser toward higher levels. The neighborhood sizes and the number of visiting pixels are shown in Table 1. The visiting schedule is run by subsampling the image at a rate 2^k at the kth level. More specifically, at the kth level, one advances by steps of length 2^k in the horizontal or vertical directions. Furthermore in the lower levels one does not visit the pixels already visited in one of the upper levels. Thus in the kth level one ends up visiting only

$$\left(\frac{N}{2^k}\times\frac{N}{2^k}\right)-\sum_{i=k+1}^{K}\left(\frac{N}{2^i}\times\frac{N}{2^i}\right)$$

pixels. The number of visited pixels and the neighborhood sizes with respect to levels are given in Table 1. Two stages of this scanning scheme are also illustrated in Fig. 5.

In summary large neighborhoods used in the upper levels give coarse but reliable segmentation, while at the lower levels one moves to a finer representation. In this manner, the noisy boundaries due to frequently changing pixel labels are avoided. This increased stability of segmentation

Table 1
Number of visited pixels and the neighborhoods sizes used in the levels

Level	Number of visited pixels	Neighborhood sizes
4	16×16	11×11
3	16×16	9×9
2	32×32	7×7
1	64×64	5×5
0	128×128	3×3
Total	256×256	

does not augment the computational cost, as the number of visited pixels, amounts, in the final analysis, to $N \times N$. Thus, the complexity of the proposed MS-CSNN method is the same as that of CSNN algorithm. But the performance and the convergence rate of the novel method are better than the CSNN algorithm as detailed in the following section.

4. Experimental results

The algorithm detailed in Section 3 has been implemented and run on a number of test images. The original CSNN algorithm is also used vis-à-vis MS-CSNN algorithm for comparison. Synthetic test images and some real-scene images are used in the experiments.

4.1. Generation of synthetic test images

For quantitative comparison, a test image generation procedure is proposed. The following considerations are taken into account in the test image generation:

1. The test image should contain variable number of objects.
2. The positions of the objects must be distributed randomly in the image.
3. Random occlusions must be allowed.
4. One must be able to control the distribution of object sizes.
5. There should be a stochastic process to "paint" the objects, that is texture or color rendering.

The steps of test image generation process are as follows:

Shape generation. In the proposed test image generation scheme, circular objects with random centers and radii are used to cover the plane completely. The center and radii of each circle are assigned by a uniformly distributed random number generator. The center can be anywhere in the $N \times N$ image plane while the values of radius lie between 5 and 71. In this random covering process arbitrary occlusions occur resulting in a rich variety of shapes as shown in Fig. 6. Finally the small objects that have area under 200 pixels are eliminated by forcing them into one of the neighboring objects.

Object coloring. Objects are painted by Gaussian distributed colors, that is RGB components. More specifically a color palette is created by clustering the pixels of some real world images. The number of clusters used, i.e. the palette size, varied from 2 to 6. One cluster center, i.e. RGB vector, is randomly chosen for each segment. In order to obtain a more natural appearance, each segment is painted by Gaussian distributed

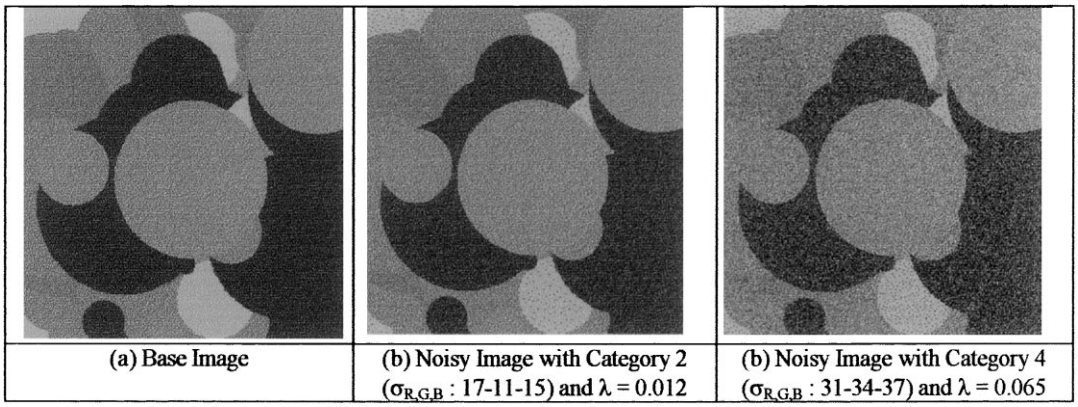

Fig. 6. Gaussian colored synthetic test images with 22 regions and five segments. (a) Base image with five randomly selected colors from palette. (b) Image with the same mean colors as in (a) but with colors spread $\sigma_R = 17$, $\sigma_G = 11$, $\sigma_B = 15$. (c) Image with the same mean colors as in (a) but with colors spread $\sigma_R = 31$, $\sigma_G = 34$, $\sigma_B = 37$. The associated λ parameters is given below the images. A color version of this figure is available as an Electronic Annex. See www.elsevier.nl/locate/patrec.

Table 2
The RGB spread categories used in test images

Category	Standard deviation interval of RGB
1	5–10
2	10–20
3	20–30
4	30–40

RGB values having as mean vector, μ, the chosen color of the palette and with randomly chosen variances, $(\sigma_R, \sigma_G, \sigma_B)$. The variance of each segment is chosen from a uniformly random number distributed between two limits. The four categories used in the experiments are listed in Table 2. Here, for example, category I means that $\sigma_R, \sigma_G, \sigma_B$ are uniformly randomly chosen in the interval [5,10]. Obviously the higher the variance the more difficult becomes the task of segmentation. The level of "difficulty" or confusion between segments can be quantified by the parameter λ given as follows:

$$\lambda = \frac{\text{Sum of within-segment variance}}{\text{Sum of between-segment variance}} = \frac{\sum_S \sigma_S^2}{\sum_S \sum_{S'} \|\mu_S - \mu_{S'}\|^2}, \quad (5)$$

where μ and σ are the resulting means color standard deviations in the segments, respectively.

In conclusion, a rich variety of ground-truthed test images can be generated by using different color palettes and variances. For instance, in Fig. 6, a base image painted with the colors of the palette and two other images painted with Gaussian RGB components are shown. "Difficulty Level" of each image is also given.

For this study, 200 different test images are generated in which the number of segments is varied from 2 to 6. For each segment number, ten different images are generated. These base test images are painted with colors randomly chosen from the palette and with variances randomly assigned in one of the categories.

4.2. Evaluation criteria

For quantitative comparison of segmentation results, a hybrid evaluation criterion is proposed. The proposed evaluation criterion (EC) consists of two parts: classification error measure (CE) and boundary fidelity measure (BF). CE, as proposed by Yasnoff et al. (1977), gives the correctly segmented pixel ratio between the ground truth and the segmented test image. It reflects the error of pixels that are assigned to the incorrect segments and the error for wrongly segmented pixels. CE formulation about segment k is given as follows:

$$\mathrm{CE}^{(k)} = 0.5 * \left[\frac{\sum_{i=1}^{N} C_{ik} - C_{kk}}{\sum_{i=1}^{N} C_{ik}} + \frac{\sum_{i=1}^{N} C_{ki} - C_{kk}}{\sum_{i=1}^{N} \sum_{j=1}^{N} C_{ij} - \sum_{i=1}^{N} C_{ik}} \right], \quad (6)$$

where C_{ij} denotes (i,j)th entity of the confusion matrix.

A measure of discrepancy between the segment shapes in the ground-truth and segmented images can be obtained from the respective turning angle functions. The turning angle function (TAF), was introduced by Arkin et al. (1991), as a geometric feature and a metric of shape characteristics of a polygonal object. Turning-angle is a function that represents the boundary characteristic of an object. It measures the angle of each polygon on the shape boundary with respect to the horizontal axis.

We adopted this scheme to arbitrary shapes by considering the turning angle at every pixel. The actual angles were computed using their three pixel neighborhoods.

Starting from a reference point on the boundary the function is constituted by successive differential angles of the boundary, with counter-clockwise turns counting as positive increments and clockwise turns as decrements. Since the perimeter of two shapes can differ in length, their turning angle functions are interpolated to the same length.

To measure discrepancy between the segment shapes in the ground-truth and segmented images, the turning-angle functions of the objects in ground truth and segmented test image, $\Theta_R(S)$ and $\Theta_S(S)$, are calculated. The $L1$ norm of the difference of these two functions called boundary fidelity (FD), is given by

$$\mathrm{BF} = \frac{\sum_{s=1}^{P} |\Theta_R(S) - \Theta_S(S)|}{P * 2\pi}, \tag{7}$$

where P is the perimeter of the object, i.e., the number of pixels on the boundary. Since the maximum difference between any two-pixel orientations is 2π and the vector length equals the perimeter of the object, this measure must take values between 0 and 1.

CE and BF have sometimes mutually exclusive properties. CE has good detection of classification error properly but poor shape distortion monitoring ability, while BF has good shape distortion monitoring but is a poor measure of classification capability. It is conjectured that the combination of these two criteria will satisfy both the classification and shape formation requirements. Such a combination can be achieved by using their Minkowski sum

$$M_p = (\alpha(\mathrm{CE})^p + \beta(\mathrm{BF})^p)^{1/p}, \tag{8}$$

where p is the order of the metric, α and β are the weights of the measures. This metric allows not only weighted contributions from the two criteria but also can adjust the penalty via the p power. For example, if p is equal to 1 with $\alpha = \beta = 1/2$ this metric becomes the arithmetic average. It becomes the root mean square average for $p = 2$. Finally, if p goes to infinity, then the M_p becomes a max operation, that is, it takes the maximum one between the two criteria.

The weights and p can vary according to the application. For instance, if one desires to give more importance to the detection of classification property of the algorithm, α must be greater than β. On the contrary, if the correctly formed shape is more important than classification then β must be greater than α. In this study, arithmetic average is used to combine two criteria. It must be noted that the smaller the value of M_p, the better the segmentation performance.

4.3. Quantitative results

Both algorithms CSNN and MS-CSNN are applied to test images and the segmentation results are evaluated using the combined metric given in (9). The performance of MS-CSNN algorithm has been found to be better than that of CSNN for 144 test images out of 200 images, that is in 72% of the tested cases. The detailed test scores are given in Tables 3 and 4. Table 3 shows the performance comparison as a function of the number of segments while Table 4 gives the scores as a function of the spread of the color. In each table the second and third columns indicate the number of cases when the M_p score in (9) was found superior with respect to the other method. The fourth column shows the percentage of cases when MS-CSNN scored higher that CSNN. Fig. 7 depicts the histogram of M_p for both the methods. One can notice that the mean value of M_p for MS-CSNN is definitely lower than that of CSNN.

4.4. Visual comparison of results

The segmentation results of test image given in Fig. 6 are shown in Fig. 8. Fig. 9 depicts detail of segmentation by zooming the central part of the images in Fig. 8. Table 5 shows the M_p values

Table 3
Performance of MS-CSNN according to the number of segments

Number of segments	MS-CSNN score	CSNN score	Percentage MS-CSNN is better than CSNN
2	37	3	92%
3	29	11	72%
4	23	17	57%
5	27	13	67%
6	28	12	70%
Total	144	56	72%

Table 4
Performance of MS-CSNN according to the color spread

Category	MS-CSNN score	CSNN score	Percentage MS-CSNN is better than CSNN
1	38	12	76%
2	36	14	72%
3	37	13	74%
4	33	17	66%
Total	144	56	72%

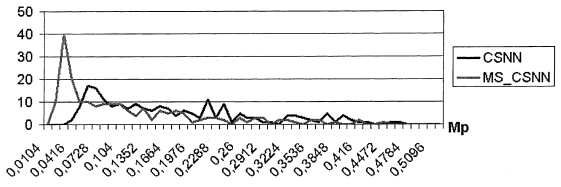

Fig. 7. Histograms of M_p for MS-CSNN and CSNN. Mean value of MS-CSNN is lower than that of CSNN.

obtained for test image given in Fig. 6. As expected, the performance improvement is especially prominent in the boundaries. In Fig. 9, this improvement is displayed in more detail.

Although the experiments are carried on color images, the method can be easily extended to any multi-spectral image. For example, in Fig. 10, the segmentation results are given with those of CSNN and MS-CSNN algorithms on a LANDSAT image involving the 3rd, 4th and 5th bands. The pixel feature vectors in this case are of the three gray values in the respective bands. It can be seen that the MS-CSNN result is more detailed and accurate as compared to the CSNN result.

The second improvement realized from the MS-CSNN algorithm was a higher convergence rate. In Fig. 11, we plotted the logarithm of convergence error against recursion steps for the Landsat

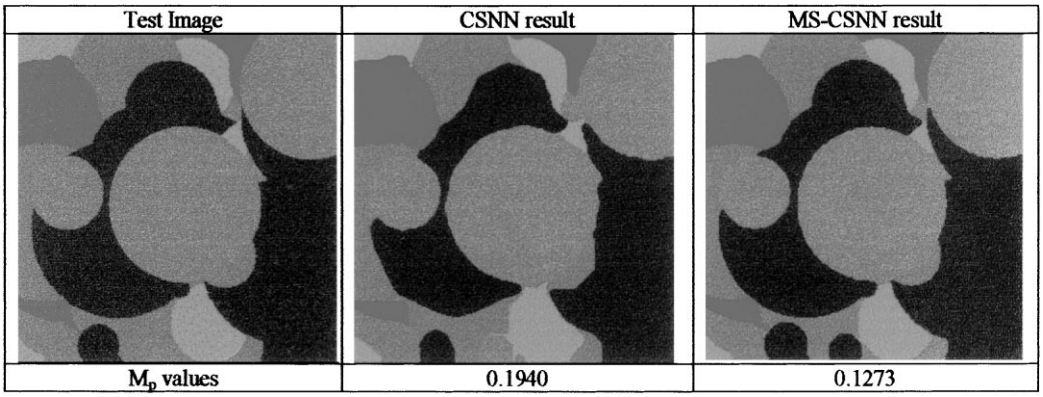

Fig. 8. The segmentation result of CSNN and MS-CSNN methods in the test image given in Fig. 6. A color version of this figure is available as an Electronic Annex. See www.elsevier.nl/locate/patrec.

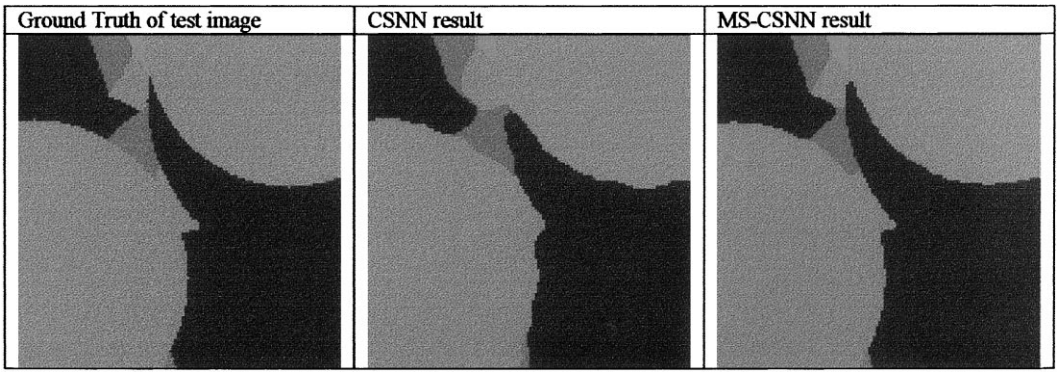

Fig. 9. Detail of segmentation in Fig. 8. Notice that MS-CSNN result in the boundaries much closer to the ground truth. A color version of this figure is available as an Electronic Annex. See www.elsevier.nl/locate/patrec.

Table 5
M_p values of CSNN and MS-CSNN in the test images

	CE	RUMA	M_p
CSNN	0.1208	0.2672	0.1940
MS-CSNN	0.0977	0.1638	0.1273

image. The convergence error is calculated as total square difference of neuron output values between two successive iteration steps. The error in the case of CSNN segmentation falls off gradually but does not diminish beyond a level of "10" even after 100 iterations. The MS-CSNN segmentation has a very different nature: At every resolution stage it converges more rapidly, but it has saccadic upswings at instants when not yet updated pixels are considered from level to level. Note that the asymptotic error always becomes lower that than of the CSNN segmentation.

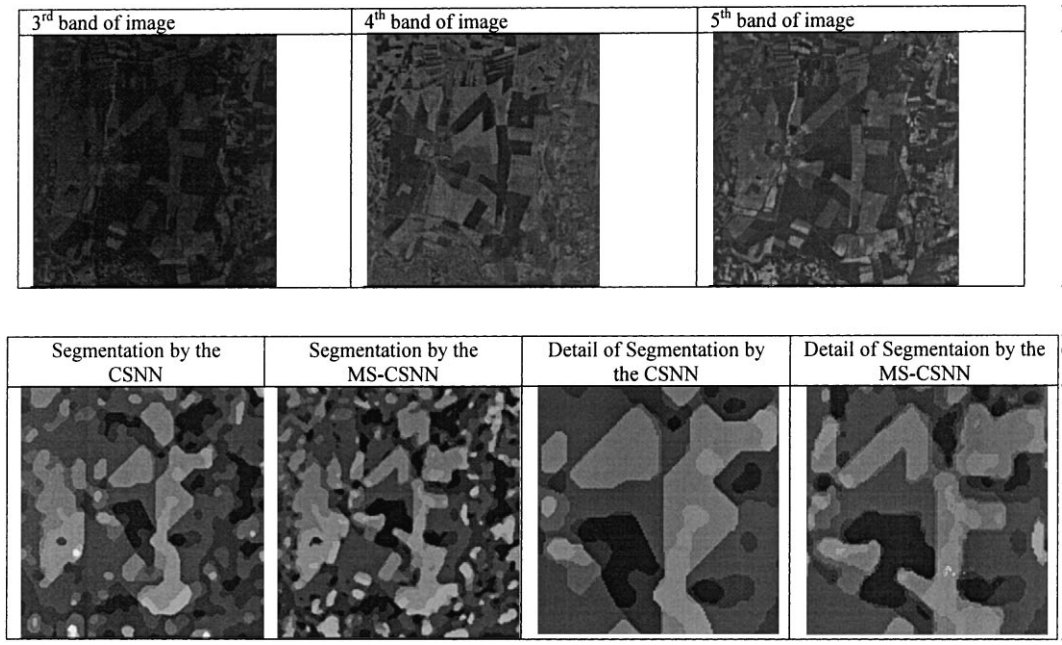

Fig. 10. Segmentation results with a Landsat image with seven segments which is found by modified Zhang–Modestino cluster validity criterion.

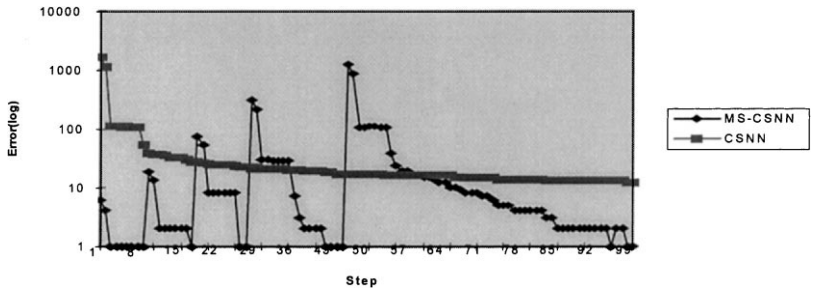

Fig. 11. Comparison of the convergence rates of CSNN and MS-CSNN. Notice how the error jumps at steps when new pixels whose labels are not yet updated via CSNN are first handled.

5. Conclusion

In this study, an improvement of the segmentation method based on the CSNN has been proposed. This method uses the basic CSNN structure for segmentation, but the scanning scheme of the pixels is modified to achieve a multi-resolution character, where pixels are scanned with sparse intervals but with large neighborhoods in the first level of the algorithm. The intervals between pixels are reduced as well as the neighborhood size toward lower layers. It appears that the main bodies of the segments are constructed in the high levels of the pyramid, while only fine details are dealt with in the lower levels. Thus the time spent to work on details is reduced and the convergence rate of the algorithm is improved. There is a risk, however, of error propagation in noisy images when in the first layers of the pyramid pixels are wrongly classified. This can be remedied by changing the visitation schedule and starting with more reliable pixels.

The other innovation of the proposed method is using the cluster validity to determine automatically the number of segments in the given image. The number of clusters in the given image is found by using modified Zhang–Modestino cluster validity index.

For further reading, see (Bezdek and Pal, 1991; Pal and Pal, 1993; Rosenfeld and Kak, 1982).

References

Adams, R., Bischof, L., 1994. Seeded region growing. IEEE Trans. PAMI 16, 641–647.

Arkin, E.M., Chew, L.P., Huttenlocker, D.P., Kedem, K., Mitchell, J.S.B., 1991. An efficient computable metric for comparing polygonal shapes. IEEE Trans. PAMI 13, 209–215.

Bensaid, A.M., Hall, L.O., Bezdek, J.C., Clarke, L.P., 1996. Partially supervised clustering for image segmentation. Pattern Recognition 29, 859–871.

Bezdek, J.C., Pal, S.K., 1991. Fuzzy Models for Pattern Recognition. IEEE Press, New York, USA.

Canny, J., 1986. A computational approach to edge detection. IEEE Trans. PAMI 8, 679–698.

Ghosh, A., Pal, N.R., Pal, S.K., 1993. Self-organization for object extraction using a multilayer neural network and fuzziness measure. IEEE Trans. Fuzzy Systems 1, 54–68.

Gonzalez, R.C., Woods, R.E., 1993. Digital Image Processing. Addision-Wesley, Reading, MA, USA.

Horn, B.K.P, 1986. Robot Vision. MIT Press, Cambridge, MA, USA.

Kurugöllü, F., Sankur, B., 1998. Color cell image segmentation using pyramidal constraint satisfaction neural network. In: IAPR Workshop on Machine Vision Applications–MVA'98, 17–19 November 1998, Makuri, Chiba, Japan.

Kurugöllü, F., Birecik, S., Sezgin, M., Sankur, B., 1996. Image segmentation based on boundary constraint neural network. In: Proc. Third Internat. Workshop on Image and Signal Processing, IWISP'96, Manchester, UK, pp. 353–256.

LaValle, S.M., Hutchinson, S.A., 1995. A framework for constructing probability distributions on the space of image segmentations. Computer Vision and Image Understanding 61, 203–230.

Lin, W.C., Tsao, E.C., Chen, C.T., 1992. Constraint satisfaction neural networks for image segmentation. Pattern Recognition 25, 679–693.

Pal, N.R., Pal, S.K., 1993. A review of image segmentation techniques. Pattern Recognition 26, 1277–1294.

Raghu, P.P., Poongodi, R., Yegnanarayana, B., 1997. Unsupervised texture classification using vector quantization and deterministic relaxation neural network. IEEE Trans. Image Processing 6, 1376–1387.

Rosenfeld, A., Kak, A.C., 1982. Digital Picture Processing. Academic Press, New York, USA.

Rubin, J.M., Richards, W.A., 1984. Color vision and image intensities: representing material changes. AI Memo 764, MIT Artificial Intelligence Laboratory, Cambridge, MA, USA.

Tsao, E.C.K., Bezdek, J.C., Pal, N.R., 1994. Fuzzy Kohonen clustering networks. Pattern Recognition 27, 757–764.

Uchiyama, M., Arbib, M.A., 1994. Color image segmentation using competitive learning. IEEE Trans. PAMI 16, 1197–1206.

Verikas, A., Malmqvist, K., Bergman, L., 1997. Colour image segmentation by modular neural network. Pattern Recognition Letters 18, 173–185.

Vinod, V.V., Chaudhury, S., Mukherjee, J., Ghose, S., 1992. A connectionist approach for gray level image segmentation. In: Proc. 11th Internat. Conf. on Pattern Recognition (IAPR'11), pp. 489–492.

Yasnoff, W.A., Mui, J.K., Bacus, J.W., 1977. Error measures for scene segmentation. Pattern Recognition 9, 217–231.

Yen, J.C., Chang, F.J., Chang, S., 1995. A new criterion for automatic multilevel thresholding. IEEE Trans. Image Processing 4, 370–378.

Yeng, P.Y., Chen, L.H., 1993. Random sampling thresholding: a new approach to multilevel thresholding. Signal Processing 34, 311–322.

Zhang, J., Modestino, J.W., 1990. Model-fitting approach to cluster validation with application to stochastic model-based image segmentation. IEEE Trans. PAMI 12, 1009–1017.