77

OBJECT BACKGROUND CLASSIFICATION USING HOPFIELD TYPE NEURAL NETWORK

ASHISH GHOSH, NIKHIL R. PAL and SANKAR K. PAL

Electronics and Communication Sciences Unit
Indian Statistical Institute
203 B.T. Road, Calcutta 700 035, India

Object extraction algorithms with a Neural Network (NN) are described. The objective function to be minimized for object extraction from a scene is shown to be similar to the expression of energy of a nenral network. A modified version of Hopfield's Neural Network model is nsed here. The weights and input biases are given in such a way that the network self-organizes to form compact clusters. Both the discrete and the continuous dynamics of the network have been used for this purpose Performance of the proposed methods has been compared with that of the relaxation techniqne.

Keywords: Image segmentation, object extraction, self-organization, neural network.

1. INTRODUCTION

Image segmentation (object extraction) is one of the essential and important steps of computer vision. It can be done either by gray level thresholding or by pixel classification (modification).[1,2] The first approach, though computationally less expensive, is effective only for good quality images with bimodal histograms. On the other hand, pixel classification (or modification) techniques are computation intensive but usually more effective as they use the spatial distributions of gray levels. Therefore, an artificial neural network (ANN) system that can perform extremely fast parallel computation seems to be an attractive architecture for this type of application, as this will enable us to get the output in real time. The present paper attempts to develop algorithms for the extraction of compact objects from a noisy scene using the collective computational ability of an ANN.

Neural networks[3–9] (NNs) are designated by the network topology, connection strengths between pairs of neurons, node characteristics and the updating rules. The updating rules may be for updating the status of an individual neuron, or that of the connection strengths (weights). Normally an objective function, called the energy of the network, is defined which designates the overall status of the network. A minimum value of the energy function corresponds to a stable state of the network.

The present work uses the network as an associative memory[3,5] to extract objects from a scene. It has been established that the stable states of the network (local minima of its energy function) correspond to the partitioning of the scene into compact (meaningful) regions (say, neurons having ON status constitute the object regions and those having OFF status constitute the background region). So given an initial state, the status of the neurons are modified iteratively so as to attain a stable state, when the

International Journal of Pattern Recognition and Artificial Intelligence Vol 6 No 5 (1992) 989–1008

neurons with ON status constitute the object (background) regions. A few approaches to object extraction using ANNs can be found in Refs. 10–12. In addition to these, several authors[13,14] have used multi-layer perceptrons also for image segmentation. Blanz and Gish[13] used a three-layer perceptron trained with a set of similar images for pixel classification. On the other hand, Babaguchi *et al.*[14] used a similar concept for histogram thresholding. It is to be noted that all these perceptron-based techniques require a set of images for learning which may not always be available in real life situations.

The architecture and dynamics of the network used here are such that self-organization[5,6] of the neurons is enforced. It may be mentioned here that the term "self-organization" refers to the ability to find out the structure of the input data even when no explicit information is available about it. The proposed technique exploits the inherent structure of the input data to extract compact objects from a scene. The network architecture used is as follows. A neuron has been assigned corresponding to every pixel. Each neuron is connected to all of its neighbours. The network architecture can, therefore, be thought of as a modified version of Hopfield's network model[3,4] where the connection strengths to all the neurons outside its neighbourhood are zero. Depending upon the gray value of the pixel, the initial status of the corresponding neuron is fixed. In the present case we are using simply a linear transformed version of the gray value. The connection strength between a pair of neurons (weights) is assigned depending on the *a priori* information about the joint distribution of the corresponding gray level pair. The neurons operate in parallel and update their status (while in transition) taking the consensus of their neighbours.

The convergence of the network is checked with respect to every neuron. If the change in output status of every neuron is less than some preassigned small positive quantity (ε) the network is said to have converged. The input/output transfer function of a neuron can be any continuous monotonically increasing function of the instantaneous input to it. The exact nature of the function may depend upon the problem at hand. In the present work, the standard "S" function as used in fuzzy sets[15] is considered.

Since the weight assignment technique and the status updating procedure consider the effect of the neighbourhood, the object extraction procedure will not be sensitive to noise and will try to preserve connectivity among the pixels of the extracted regions. It is to be noted that the initial activation levels may affect the ultimate output to some extent.

The effectiveness of the proposed technique has been demonstrated on a synthetic binary image corrupted by noise and on two gray-tone real images. It has been found that the results are quite good and satisfactory. For compact objects, as expected, the proposed techniques are found to work better. Results have been compared with those of the relaxation technique.[1,16] An objective evaluation of the proposed methods has also been tried using the probability of correct classification.

2. DESCRIPTION OF NEURAL NETWORKS

2.1. General Description of Neural Networks

As mentioned earlier a neural network is characterized by the network topology, connection strength between pairs of neurons (weights), node characteristics and the status updating rules. Node characteristics mainly specify the primitive types of operations it can perform, like summing the weighted inputs coming to it and then amplifying the sum. The updating rules may be for weights and/or states of the processing elements (neurons). Normally an objective function is defined which represents the complete status of the network and its set of minima gives the stable states of the network. The status of any neuron of the network can be updated at random times independent of the other but in parallel. Since there are interactions among all the neurons the collective property inherently reduces the computational task. As our study will be concerned with networks similar to that of Hopfield's model,[3] a brief description of the same is given below. Discussion of the Hopfield model can be done under two headings, discrete and continuous, depending on the permissible values of the output status of a neuron during transition.

2.1.1. Discrete model

In this model bi-state neurons are used. Each neuron i, has two possible states ON and OFF, characterized by the output values (V_i) of -1 and $+1$. The input to a neuron comes from two sources, external input bias I_i and inputs from other neurons (to which it is connected). Thus the total input to a neuron i is

$$U_i = \sum_j W_{ij} \cdot V_j + I_i \tag{1}$$

where W_{ij} (the weight) is the synaptic interconnection strength from neuron j to neuron i. The connection strengths are assumed to be symmetric, i.e. $W_{ij} = W_{ji}$.

For this system the motion of the state of the NN in state space describes the computation that it performs. The present model describes it in terms of a stochastic evolution. Each neuron samples its input at any random time. It changes the value of its output or leaves it unaltered according to a threshold rule with the threshold θ_i. The rule is

$$V_i \to -1 \quad \text{if } U_i \le \theta_i \tag{2}$$

$$V_i \to +1 \quad \text{if } U_i > \theta_i \,. \tag{3}$$

The interrogation of each neuron is a stochastic process, taking place at a fixed mean rate. The times of interrogation of each neuron are independent of that of others thereby making the algorithm asynchronous.

It can be shown that any state change produced by the above threshold logic reduces the energy of the network defined by

$$E = -\sum_i \sum_j W_{ij} V_i V_j - \sum_i I_i V_i + \sum_i \theta_i V_i \,. \tag{4}$$

The change of energy ΔE due to a change of state of the neuron i by ΔV_i is

$$\Delta E = -\left[\sum_j W_{ij} V_j + I_i - \theta_i\right] \Delta V_i \,. \tag{5}$$

Now if ΔV_i is positive, then from Eq. (3) we see that the bracketed quantity in Eq. (5) is also positive, making ΔE negative. When ΔV_i is negative, the bracketed quantity becomes negative (from Eq. (2)) resulting again in making ΔE negative. Thus any change in E under the above algorithm is negative. Since E is bounded, the time evolution of the system is a motion in the state space that seeks out minima in E and stops at such points.

2.1.2. Continuous model

This model is based on continuous variables and responses. Let the output variable V_i for the ith neuron lie in the range $[-1, 1]$, and be continuous and a monotonically increasing function of the instantaneous input U_i to it. The input/output relation

$$V_i = g(U_i) \tag{6}$$

is sigmoidal with asymptotes -1 and $+1$. It is also necessary that the inverse of g exists, i.e. $g^{-1}(V)$ is defined.

A typical choice of the function g is

$$g(x) = 2\,\frac{1}{1 + e^{-(x-\theta)/\theta_0}} - 1 \,. \tag{7}$$

Here the parameter θ controls the shifting of the function g along the x-axis and θ_0 determines the steepness (sharpness) of the function. Positive values of θ shift $g(x)$ towards the positive x-axis and vice versa. Lower values of θ_0 make the function steeper while higher values make it less sharp. The value of $g(x)$ lies in $[-1, 1]$ with 0.0 at $x = \theta$.

Hopfield[3] has shown that

$$E = -\sum_i \sum_j W_{ij} V_i V_j - \sum_i V_i I_i + \sum_i \frac{1}{R_i} \int_0^{V_i} g^{-1}(V)\,dV \tag{8a}$$

is a Liapunov function for the above mentioned system and $dE/dt \leq 0$. Here R_i is the total input resistance of the amplifier realizing a neuron. The last term in Eq. (8a) is the energy loss term which becomes zero at the high gain limit, i.e. for higher value of g. Similar to the discrete case, the time evolution of the system will also lead to a minimum of E and stop there. The dynamics of the network are governed by the differential equations obtained from

$$-\frac{\partial E}{\partial V_i} = \frac{dU_i}{dt} . \tag{8b}$$

2.2. Description of the Network Used in the Present Study

A digital image can be viewed as a rectangular array of pixels $F_{P\times Q} = \{f(x, y) \in (1, 2, 3, \ldots, L)$, the set of gray levels$\}_{P\times Q}$. In this study one neuron is assigned to each of the $P \times Q$ pixels and each neuron is connected only to its neighbouring pixels. In other words, the (i, j)th neuron is connected to all neurons in its neighbourhood N^d (dth order neighbourhood). The selection of the neighbourhood system (N^d) is problem dependent. For a pixel (i, j) the 2nd order neighbourhood consists of the nearest four neighbours, whereas the nearest eight neighbours constitute the 3rd order neighbourhood. With a 3rd order (N^3) neighbourhood, the network topology takes the form depicted in Fig. 1. Thus we see that $W_{ij,kl} > 0$ if $(k, l) \in N^d_{ij}$, otherwise $W_{ij,kl} = 0$, where $W_{ij,kl}$ is the connection strength between the (i, j)th and the (k, l)th neuron. Obviously, if the size of the neighbourhood increases, the computation involved will also increase. Hence the uetwork can be thought to be a modified version of Hopfield's network[3,4] with the modification that the connection strength to all neurous outside the neighbourhood (N^d) is zero. However in the following discussion an N^3 neighbourhood will be assumed.

From Eq. (7) we notice that

$$x \xrightarrow{Lt} -\infty \qquad g(x) = -1$$

and

$$x \xrightarrow{Lt} +\infty \qquad g(x) = +1 .$$

In other words, as $x \rightarrow \pm\infty$, the function asymptotically approaches the limiting values of ± 1. In the present case, since the number of neighbours is eight, for normalized values of connection strengths the input value to a neuron lies in $[-8, 8]$. A polynomial function defined over a finite domain may also be used as an input/output transfer function. One such choice can be

$$
\begin{aligned}
g(x) &= -1 && x \leq a \\
&= 2^n\left(\frac{x-a}{c-a}\right)^n - 1 && a \leq x \leq b \\
&= 1 - 2^n\left(\frac{c-x}{c-a}\right)^n && b \leq x \leq c \\
&= 1 && x \geq c
\end{aligned}
\tag{9}
$$

for $n \geq 2$ and $b = (a + c)/2$. In this case $g(x)$ lies in $[-1, 1]$ with $g(x) = 0.0$ at $x = b$. The domain of x is $[a, c]$. The value of n controls the sharpness (steepness) of the function. The above function is nothing but a generalized version of the standard "S" function as used in fuzzy sets.[15]

In this case since the domain of x is $[-8, 8]$, i.e. $a = -8$, $c = 8$, the function g takes the form

$$
\begin{aligned}
g(x) &= -1 && x \leq -8 \\
&= \frac{1}{2^{3n}}(x+8)^n - 1 && -8 \leq x \leq 0 \\
&= 1 - \frac{1}{2^{3n}}(8-x)^n && 0 \leq x \leq 8 \\
&= 1\,. && x \geq 8
\end{aligned}
\tag{10}
$$

However, for quick convergence one can use the domain $[-1, 1]$, i.e.

$$
\begin{aligned}
g(x) &= -1 && x \leq -1 \\
&= (x+1)^n - 1 && -1 \leq x \leq 0 \\
&= 1 - (1-x)^n && 0 \leq x \leq 1 \\
&= 1\,. && x \geq 1
\end{aligned}
\tag{11}
$$

3. OBJECT EXTRACTION ALGORITHMS

3.1. Formulation of the Energy Function

In the present study we shall consider the image of an ideal scene corrupted by noise. By ideal scene we mean that the scene has one or more compact object region(s) with uniform characteristics. In fact any real image with some compact object(s) can be viewed like this. Given such an image, our objective here is to extract the object regions with the help of a neural network. Hence in the following, we shall try to formulate the energy function of the proposed network in such a manner that in the stable state of the network the object regions are clearly separated from the background.

Consider the energy function defined in Eq. (4), which has two parts. The first part is due to the local field or feedback and the second part corresponds to the input bias of the neurons. In terms of images, the first part can be viewed as the impact of the gray levels of the neighbouring pixels, whereas the second part can be attributed to the gray value of the pixel under consideration. The energy function can then be formulated in the following way.

Whatever be the input, without loss of generality, it can be assumed that the input values to the neurons lie in $[-1, 1]$. In an image, the gray value of a pixel is highly influenced by that of its neighbours. So, if a pixel belongs to an object region, the probability of its neighbours belonging to the same object region is very high. This suggests that if a pair of adjacent pixels is having similar values then the potential (energy) contributiou of this pair of pixels to the overall energy function should be much less. If the gray values of two adjacent pixels are V_i and V_j (from now onwards the two-dimensional indices of a neuron will be replaced by a oue-dimensional index), then one possible choice for the contribution of each of these pairs to the overall energy function can be

$$-W_{ij}V_iV_j \tag{12}$$

where W_{ij} is a non-negative constant for a particular i and j. This W_{ij} can be viewed as the connection strength between the ith and the jth neurons and V_i is the output status of the ith neuron. So the total energy contributed by all pixel pairs will be

$$\begin{aligned} -\sum_{i,j} W_{ij}V_iV_j &= -\sum_i \left(\sum_j (W_{ij}V_j)V_i \right) \\ &= -\sum_i h_iV_i \end{aligned} \tag{13}$$

where $h_i = \sum_j (W_{ij}V_j)$ is termed as the local field.

For every neuron i, there is an initial input bias I_i which can be taken to be proportional to the actual gray level of the corresponding pixel. This is quite logical, as even in a noisy environment, if the gray level of a pixel is high (low), the corresponding intensity value of the scene is expected to be high (low). The input bias value is taken in the range $[-1, 1]$. Under the above framework an ON(1) neurou corresponds to an object pixel whereas an OFF(-1) neuron represents a background pixel. So the threshold between object and background can logically be taken as 0. If any neuron has a very high positive bias (≈ 1) or a very high negative bias (≈ -1), then it is very likely that in the stable state it will be ON or OFF, respectively. So, the product I_iV_i should contribute less towards the total energy value, and the second part of the energy expression may be written as

$$- \sum_i I_i V_i \; . \tag{14}$$

Then the expression of energy for the object extraction problem takes the form

$$E = - \sum_i \sum_j W_{ij} V_i V_j - \sum_i V_i I_i \; . \tag{15}$$

Thus we see that expression (15) is to be minimized for separating the object regions from the background. The minimization of expression (15) requires the knowledge of W_{ij}, which will be addressed in the next section.

3.2. Assignment of Weights

We have already mentioned that if a particular pixel belongs to the object region then its neighbouring pixels are also expected to belong to the same region. In other words, the probability of the joint occurrence of that gray value pair is expected to be high. This suggests the use of the probability of co-occurrence of the gray values as the connection strength. However, if the image is highly corrupted by noise then the joint probability may not convey the desired information, and hence it is not desirable to use that as the connection weight. Under these circumstances, when the image is expected to be highly corrupted by noise, one can use $W_{ij} = 1$ or 0. In the following section, the estimation of the connection weight when the image is with moderate noise will be discussed.

Let n_{kl} be the number of occurrences of the gray level k (l) followed by l (k) in a particular fashion in the whole image. Then the weight (W_{ij}) between the ith and the jth neurons can be taken as

$$W_{ij} = \frac{n_{kl}}{\sum n_{mn}} = p_{kl} \tag{16}$$

where k = input gray level of the ith neuron

l = input gray level of the jth neuron

p_{kl} = the probability of the joint occurrence of the gray levels k followed by l.

Note that $\sum W_{ij} = \sum p_{kl} = 1$. Now, instead of taking $W_{ij} = p_{kl}$ one can also choose

$$W_{ij} = \frac{n_{kl}}{\max\{n_{mn}\}}$$

$$= \frac{n_{kl}}{\sum n_{mn}} \cdot \frac{\sum n_{mn}}{\max\{n_{mn}\}}$$

$$= C \cdot \frac{n_{kl}}{\sum n_{mn}} \tag{17}$$

where C is a positive constant for a given image.

The two criteria (Eqs. (16) and (17)) are basically the same, the only difference is in scaling. But the latter is preferred, since it strongly favours the most freqnent gray value pairs which is intuitively desirable.

3.3. Realization with Neural Network

3.3.1. Discrete model

The expression in Eq. (15) is equivalent to the expression in Eq. (4) with $\theta = 0$. Hence the network structure described in Sec. 2.2 can be used to obtain the minima of the expression in Eq. (15). To put it in other words, the network (Fig. 1) can solve the image segmentation (object extraction) problem. Here the status updating rule is the same as in Eqs. (2) and (3), with $\theta = 0$.

3.3.2. Continuous model

The expressions in Eqs. (8a) and (15) differ by the term

$$\sum_i \frac{1}{R_i} \int_0^{V_i} g^{-1}(V)dV$$

only. But for high gain[4] of "g" (Eq. (8a)), the above term vanishes. So the two expressions (Eqs. (8a) and (15)) become identical. Hence under the continuous framework the object extraction problem can be solved by using a neural network whose energy function takes the form

$$E = -\sum_i \sum_j W_{ij} V_i V_j - \sum_i V_i I_i + \sum_i \frac{1}{R_i} \int_0^{V_i} g^{-1}(V)dV \,. \qquad (18)$$

The search process can be activated by solving U_i ($\forall i$) from the following differential equations

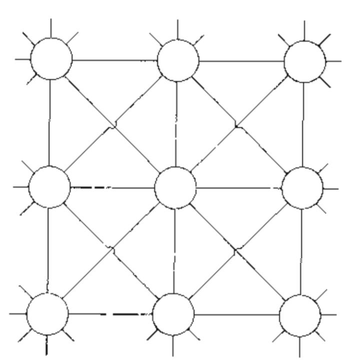


Fig. 1. Topology of the neural network.

$$-\frac{\partial E}{\partial V_i} = \left(\sum_j W_{ij} V_j + I_i - U_i \right) = \frac{dU_i}{dt} . \tag{19}$$

Once the U_i values are known, Eq. (6) can be applied to get the output status V_i.

4. COMPUTER SIMULATION AND RESULTS

To check the validity and effectiveness of the proposed technique, computer simulation has been done using some synthetic images (Figs. 3(a), 4(a), 5(a)) and some real images (Figs. 6(a), 7(a), 8(a)). The synthetic input images have been generated by adding $N(0, \sigma_i^2)$ noise to every pixel of the binary image depicted in Fig. 2. The value of σ_i is gradually increased so as to have SNR = 1.0, where SNR is defined as

$$\text{SNR} = \frac{\text{range of gray levels}}{\sigma_i} .$$

The different noisy versions of Fig. 2 and the corresponding extracted objects with both discrete and continuous dynamics of the network are given in Figs. 3–5. One of the real images used is a "NOISY TANK" (Fig. 6(a)). The outputs obtained by the discrete and the continuous models are depicted in Figs. 6(b)–(e). For these images the weight W_{ij} is assigned as one (1), if j is a neighbour of i and zero (0), otherwise.

The algorithm is also tested on an uncorrupted image of "BIPLANE" (Figs. 7(a) and 8(a)) so as to establish the effectiveness of weight assignment depending on the joint distribution of gray levels (Eq. (17)). The extracted outputs for this image with discrete and continuous models (with N^2) are shown in Figs. 7(b) and (c), respectively. On the other hand, Figs. 7(d) and (e) represent the same outputs with N^3. For the sake of comparison, we have also implemented the algorithm with $W_{ij} = 0$ or 1 and the corresponding outputs are shown in Figs. 8(b)–(e).

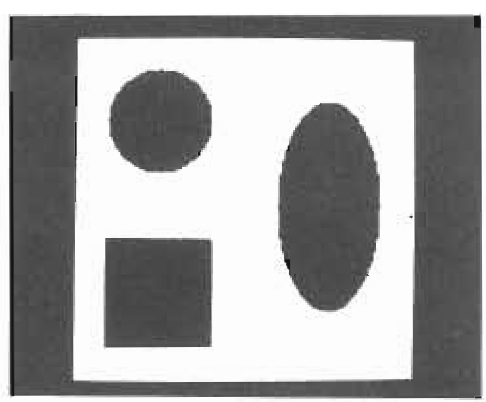


Fig. 2. Original synthetic image.

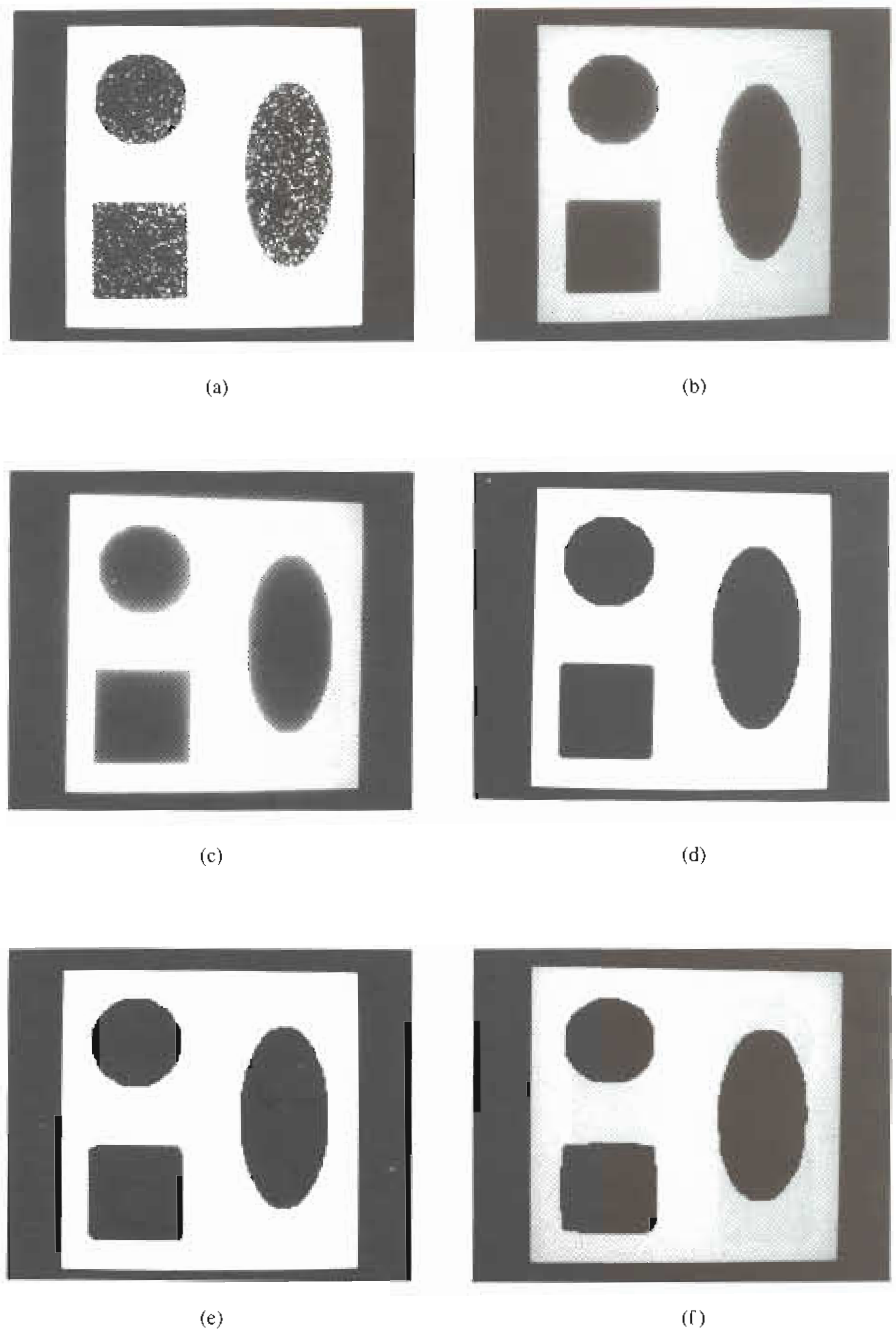


Fig. 3. (a) Noisy version of Fig. 2 with SNR = 4.8 (Input). (b) Output by discrete model with N^2. (c) Output by continuous model with N^2. (d) Output by discrete model with N^3. (e) Output by continuous model with N^3. (f) Output by relaxation method with N^3.

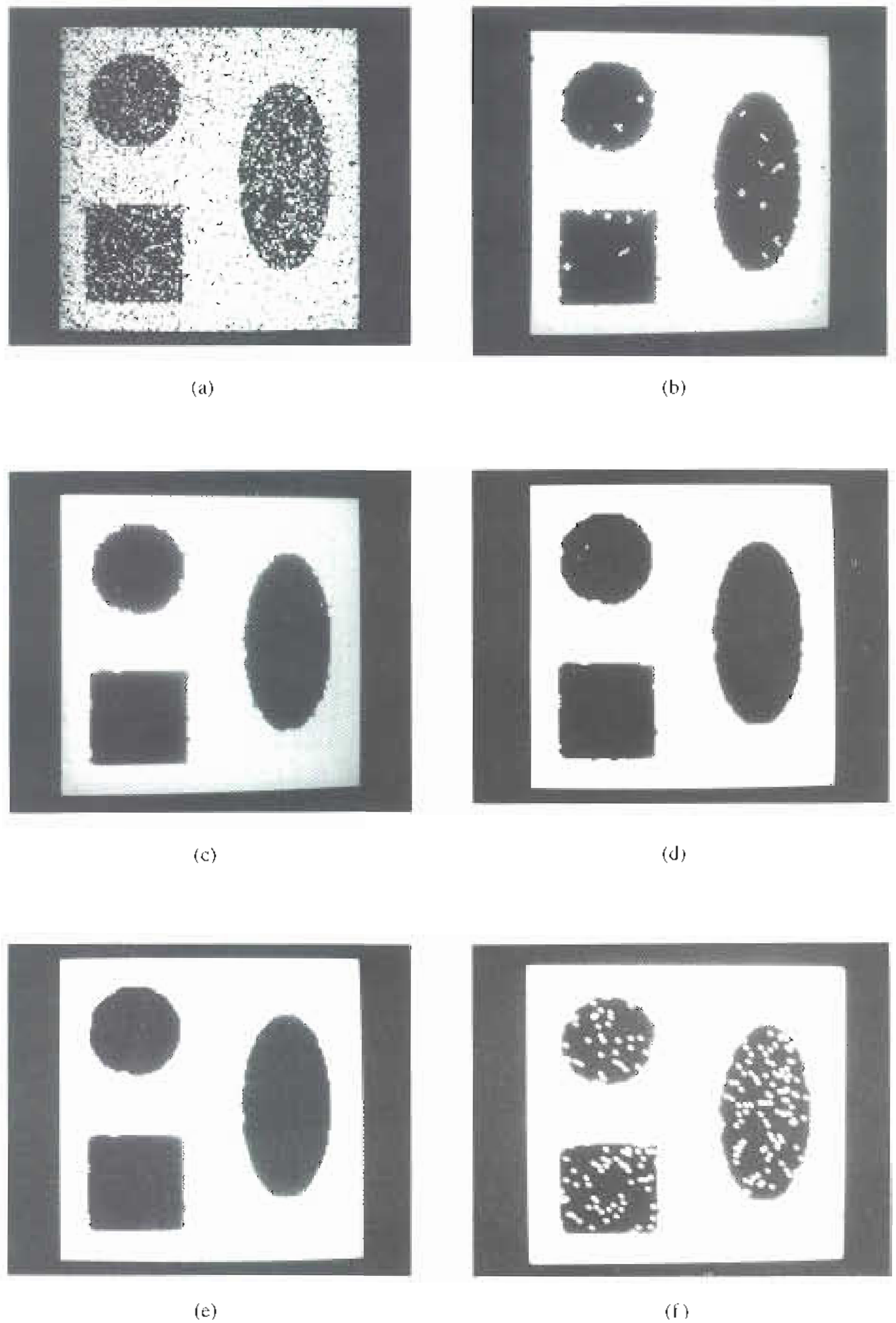


Fig. 4. (a) Noisy version of Fig. 2 with SNR = 1.6 (Input). (b) Output by discrete model with N^2 (c) Output by continuous model with N^2. (d) Output by discrete model with N^3 (e) Output by continuous model with N^3. (f) Output by relaxation method with N^3.

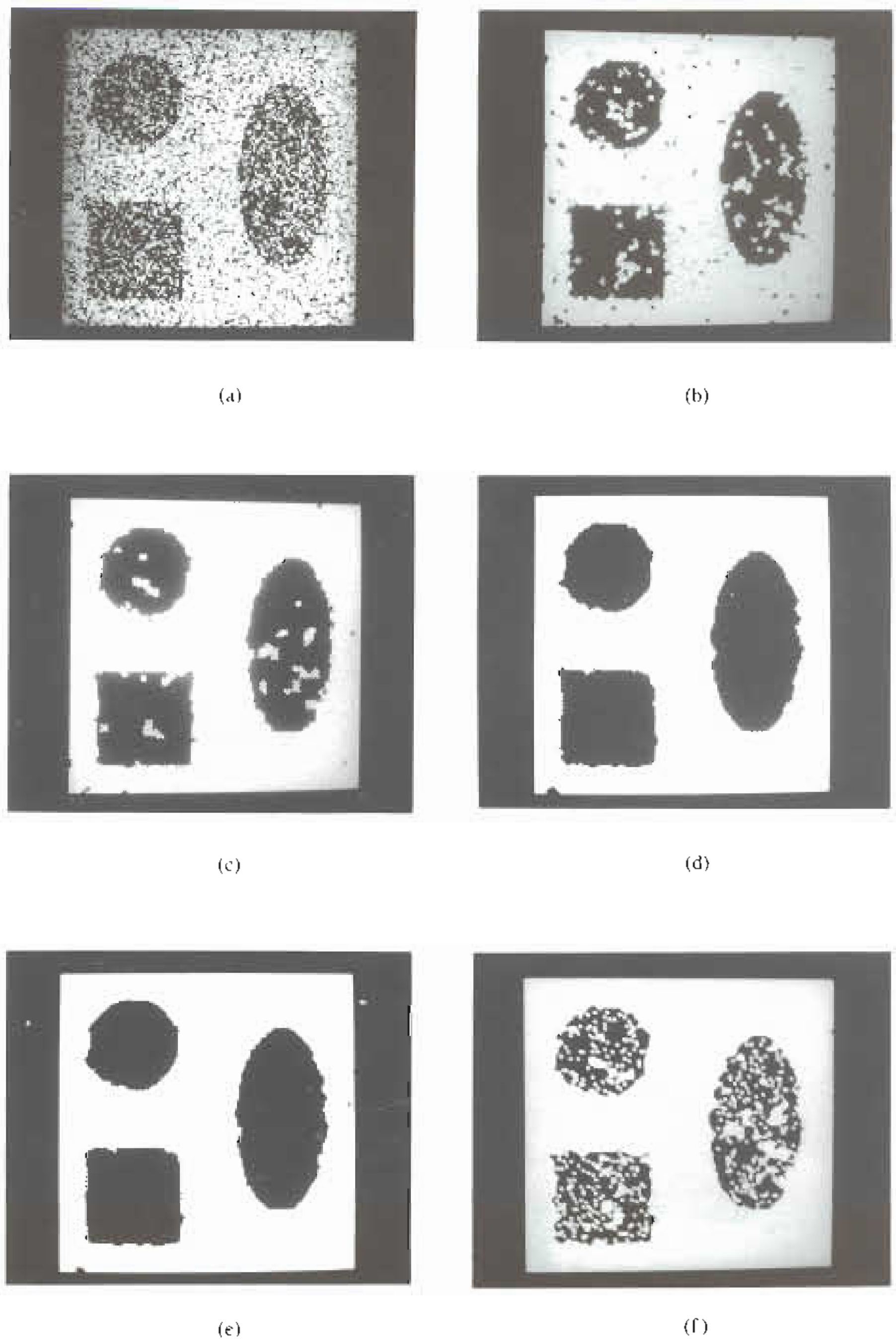
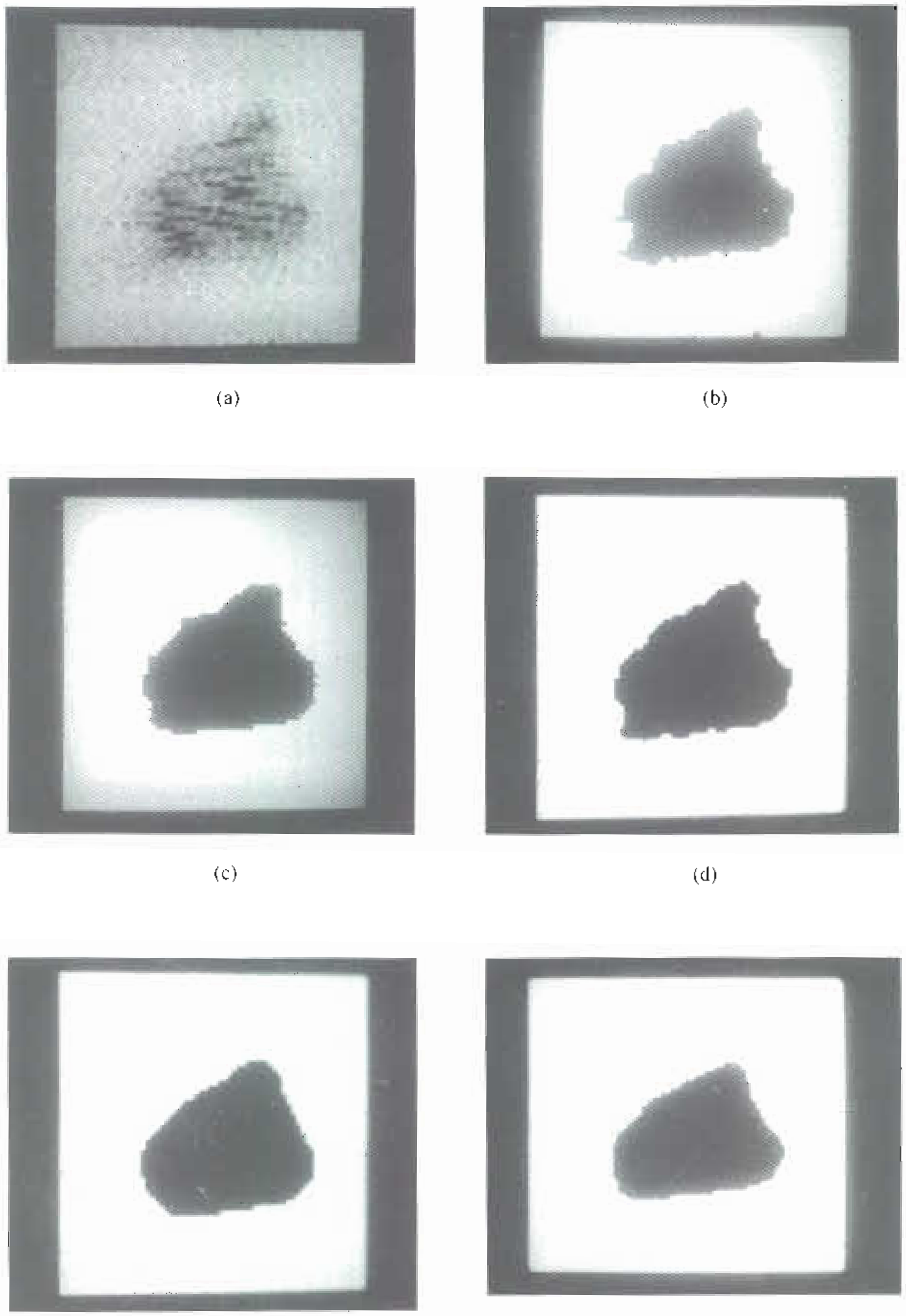


Fig. 5 (a) Noisy version of Fig. 2 with SNR = 1.0 (Input). (b) Output by discrete model with N^2. (c) Output by continuous model with N^2. (d) Output by discrete model with N^3. (e) Output by continuous model with N^3. (f) Output by relaxation method with N^3.

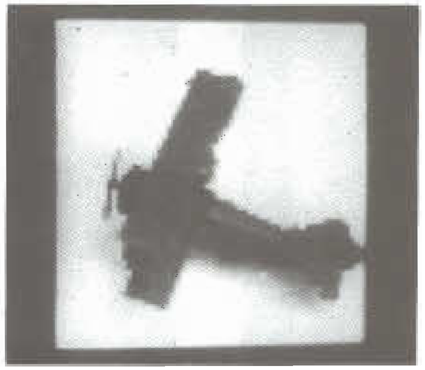


(a) (b)

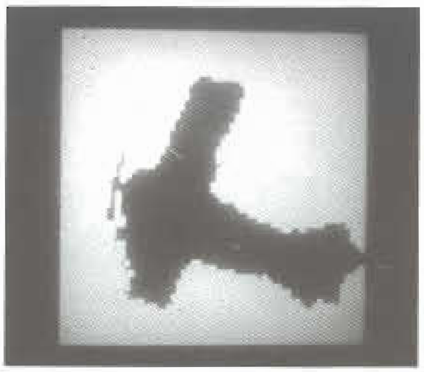
(c) (d)

(e) (f)

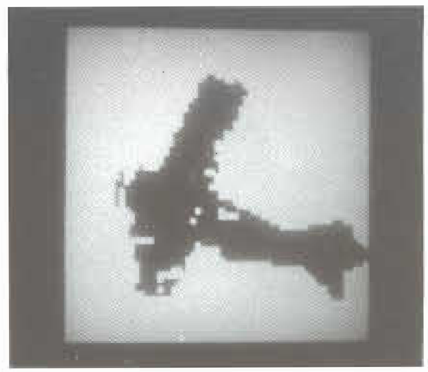
Fig. 6. (a) "NOISY TANK" image (Input). (b) Output by discrete model with N^2. (c) Output by continuous model with N^2. (d) Output by discrete model with N^3. (e) Output by continuous model with N^3. (f) Output by relaxation method with N^3.



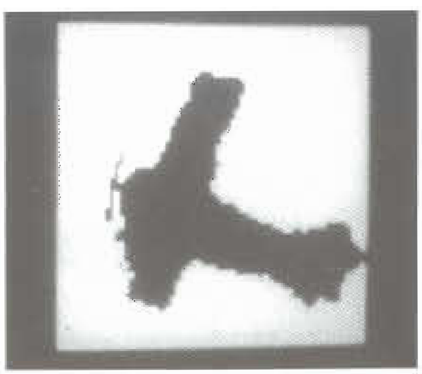
(a)



(b)



(c)



(d)

(e)

Fig. 7. (a) "BIPLANE" image (Input). (b) Output by discrete model with N^2 (W_{ij} by Eq. (17)). (c) Output by continuous model with N^2 (W_{ij} by Eq. (17)). (d) Output by discrete model with N^3 (W_{ij} by Eq. (17)). (e) Output by continuous model with N^3 (W_{ij} by Eq. (17))

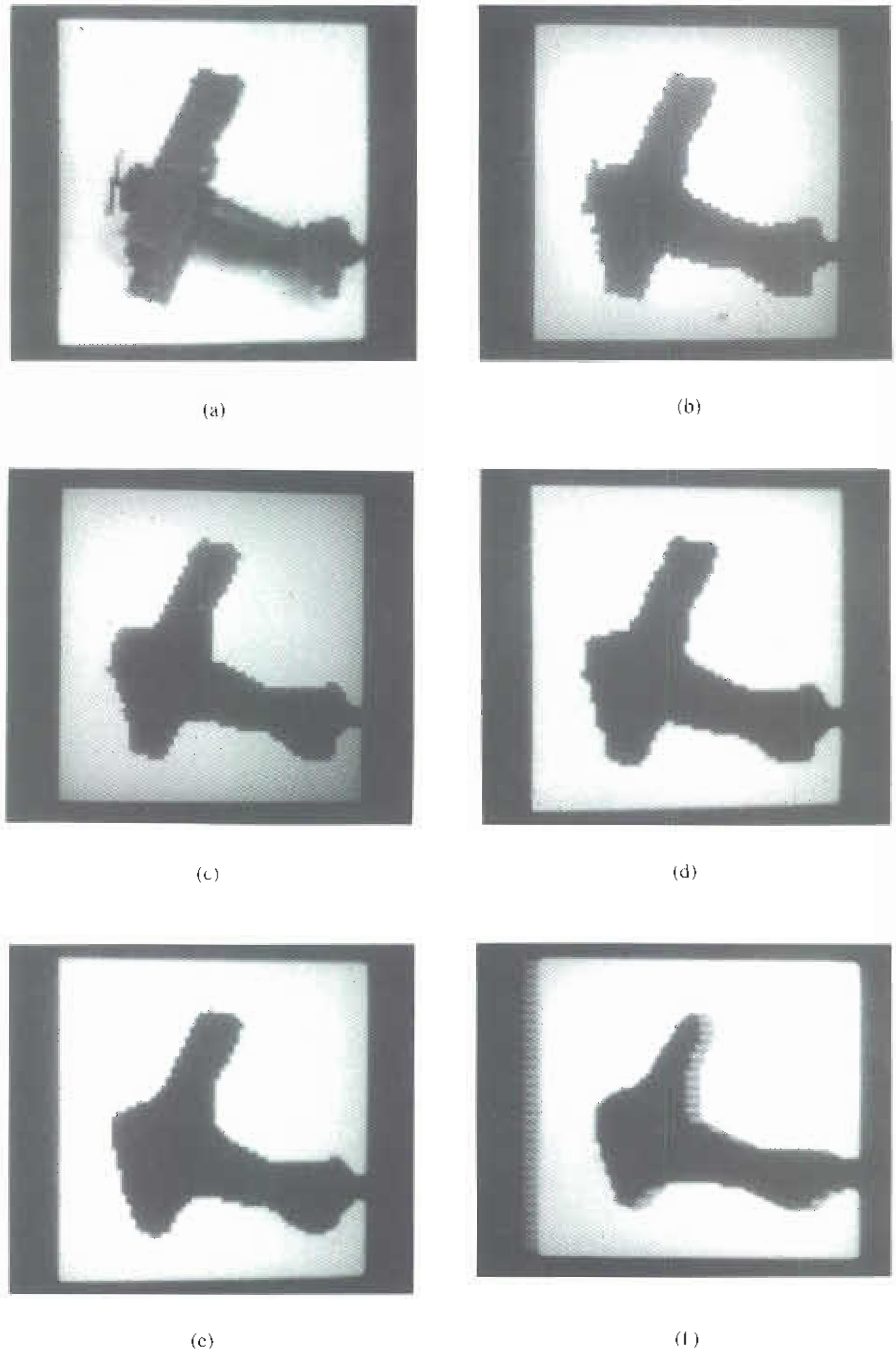


Fig. 8. (a) "BIPLANE" image (Input). (b) Output by discrete model with N^2. (c) Output by continuous model with N^2. (d) Output by discrete model with N^3. (e) Output by continuous model with N^3. (f) Output by relaxation method with N^3.

Examining the results (Figs. 3–6) it can be said that the outputs obtained by the continuous dynamics are better than those obtained by the discrete versions. It is seen that for high values of SNR, the results obtained by the two approaches are very close and for all practical purposes they cannot be distinguished. Since the discrete dynamics take much less time to stabilize, for higher values of SNR, the discrete model is preferred. It is also noticed from the results that with decrease of SNR, there is a slight deterioration in the quality of outputs.

From the results of the "BIPLANE" image (Fig. 7) it is seen that the weight assignment procedure using Eq. (17) is good enough to extract the exact shape of the object including the propeller of the "BIPLANE".

Moreover, it is clear from the results that N^2 connectivity is more noise sensitive than N^3 connectivity. But for higher values of SNR (i.e. for less corrupted images) the outputs are comparable. Thus for less corrupted images, N^2 connectivity with discrete dynamics can be used thereby saving both space and time.

To establish the superiority of the proposed method over the existing iterative algorithms a result-wise comparison is made with the relaxation technique. For the computation of the compatibility function in the relaxation method the following formula is used:

$$C(i, j; h, k) = \frac{p(a_i \varepsilon C_j,\ a_h \varepsilon C_k)}{p(a_i \varepsilon C_j) \cdot p(a_h \varepsilon C_k)}$$

where $C(i, j; h, k)$ gives a measure of the compatibility that the element a_i belongs to the class C_j and a_h to C_k; $p(a_i \varepsilon C_j)$ is the *a priori* probability of $a_i \varepsilon C_j$. The results obtained with N^3 connectivity are also included in Figs. 3–8. From the results it is evident that the objects extracted by the NN-based methods preserve the shapes and the approximate outlines of the actual inputs in a better way than those by the relaxation algorithm.

Objective evaluation of the results has been attempted using the percentage of correct classification of pixels. The numerical results are depicted in Table 1. From the

Table 1.

Synthetic Image with	Connectivity	Percentage of correct classification		
		Neural Network Algorithms		Relaxation
		Discrete	Continuous	
$\sigma = 05$	N^2	99.93	99.71	99.43
$\sigma = 15$	N^2	98.69	99.23	97.89
$\sigma = 24$	N^2	92.25	96.47	90.78
$\sigma = 05$	N^3	99.90	99.71	98.58
$\sigma = 15$	N^3	99.28	99.40	97.55
$\sigma = 24$	N^3	98.54	98.68	92.80

table it is found that for all different types of connectivity and noise levels the percentage of pixels correctly classified by the proposed algorithms is more than that by the relaxation technique. This also establishes the superiority of the proposed neural network-based algorithms.

It is to be noted from Eq. (8b) that in order to get the input to all neurons of the network (of size $N_1 \times N_2$) at an instant $(t + \Delta t)$ one has to solve $N_1 \times N_2$ differential equations with given initial values at time t. For this the Euler method is used here, i.e. we iterated (from Eq. (19))

$$U_i(t + \Delta t) = U_i(t) + \Delta t\left(\sum_j W_{ij} V_j(t) + I_i - U_i(t)\right) \tag{20}$$

until convergence. Numerical solutions for these differential equations require a stopping criterion which can be taken as

$$|U_i(t + \Delta t) - U_i(t)| < \varepsilon, \qquad \text{for all } i \tag{21}$$

where ε is a preassigned small positive quantity. The present simulation srudy uses $\Delta t \approx 10^{-5}$ and $\varepsilon \approx 10^{-6}$.

The network is assumed to attain a stable state if for every neuron i, $|V_i(t) - V_i(t + \Delta t)| < \varepsilon^1$, where ε^1 is another preassigned small positive quantity. The network may require a sufficiently large time ro achieve the exact value of -1 (OFF) or $+1$ (ON). Obviously, if the precision level is increased (value of ε^1 is decreased) the network takes more time to converge. Quick convergence can be obtained by increasing Δt or ε^1.

In this context it may be mentioned that for the present study the input bias of a neuron is taken as

$$I = 2\frac{i}{L} - 1 \tag{22}$$

where i is the gray value of the corresponding pixel and L is the maximum gray value. The input/output transfer function used is a second order polynomial (Eq. (11) with $n = 2$).

5. DISCUSSION AND CONCLUSIONS

The present work demonstrates an application of a neural network in the object extraction problem. A modified version of Hopfield's model is used as the neural network architecture. A single neuron is assigned to every pixel. The energy function is designed in such a manner that in a stable state of the network, neurons corresponding to compact homogeneous regions will be in ON (OFF) state while the others will be OFF (ON). Both the discrete and the continuous dynamics are studied for the purpose of object extraction.

The proposed technique has been implemented and tested on a set of compact highly noise corrupted images, and a gray tone uncorrupted image. The results obtained are quite satisfactory. The object pixels are found to self-organize to construct compact regions. The status updating procedure of a neuron gets consensus from its neighbours. Thus the system is not noise sensitive. In the case of uncorrupted (or less corrupted) images, the weights are to be assigned using the *a priori* knowledge of the co-occurrence of the gray levels instead of giving 0 or 1. Otherwise, the results may become blurred.

A comparative study is made with the iterative relaxation technique. An attempt on objective evaluation has also been done using the percentage of correct classification of pixels. This also conforms to the observation stated earlier.

REFERENCES

1. A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, Academic Press, New York, 1982.
2. S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images", *IEEE Trans. Pattern Anal. Mach. Intell.* **6** (1984) 721–741.
3. J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two state neurons", *Proc. Nat. Acad. Sci., USA* **81** (1984) 3088–3092.
4. J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems", *Biol. Cybern.* **52** (1985) 141–152.
5. T. Kohonen, *Self-Organization and Associative Memory*, 3rd edn., Springer-Verlag, Berlin, 1989.
6. T. Kohenen, "Self organised formation of topologically correct feature maps", *Biol. Cybern.* **43** (1982) 59–69.
7. D. E. Rumelhart, J. McClelland and PDP research group, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol. 1, MIT Press, Cambridge, MA, 1986.
8. L. O. Chua and L. Yang, "Cellular neural network: theory", *IEEE Trans. Circuits and Systems* **35**, 10 (1988) 1257–1272.
9. R. P. Lippmann, "An introduction to computing with neural nets", *IEEE ASSP Mag.* **61** (1987) 3–22.
10. A. Ghosh and S. K. Pal, "Self-organization of neural network and object extraction", in *Proc. Workshop on Signal Processing, Communications and Networking*, Bangalore, India, Jul. 1990, pp. 241–246.
11. A. Ghosh, N. R. Pal and S. K. Pal, "Object extraction using a self-organizing neural network", *Proc. Int. Symp. on Intelligent Robotics*, Bangalore, India, Jan. 1991, pp. 686–697.
12. G. L. Bilbro, M. White and W. Synder, "Image segmentation with neurocomputers", in *Neural Computers*, eds. R. Eckmiller and C. V. D. Malsburg, Springer-Verlag, 1988.
13. W. E. Blanz and S. L. Gish, "A connectionist classifier architecture applied to image segmentation", in *Proc. 10th Int. Conf. on Pattern Recognition*, Atlanric City, NJ, 1990, pp. 272–277.
14. N. Babaguchi *et al.*, "Connectionist model binarization", in *Proc. 10th Int. Conf. on Pattern Recognition*, Atlantic City, NJ, 1990, pp. 51–56.
15. S. K. Pal and D. Dutta Majumder, *Fuzzy Mathematical Approach to Pattern Recognition*, John Wiley (Halsted Press), New York, 1986.

16. A. Rosenfeld, R. A. Humel and S. W. Zucker, "Scene labelling by relaxation operations", *IEEE Trans. Syst. Man Cybern.* **6** (1976) 420–433.
17. J. Bruck and J. Sanz, "A study of neural networks", *Int. J. Intell. Syst.* **3** (1988) 59–75.

Received 15 February 1991; revised 26 June 1991.

Ashish Ghosh received the B.E. degree in electronics and telecommunication from Jadavpur University, Calcutta in 1987, and the M.Tech. degree in computer science from the Indian Statistical Institute, Calcutta in 1989. At present he is working as a Senior Research Fellow in the Indian Statistical Institute towards his Ph.D. degree. He received the Young Scientist Award in Computer Science from the Indian Science Congress Association in 1992. His research interests include neural networks, image processing, fuzzy sets and systems, and pattern recognition.

Nikhil R. Pal obtained the B.Sc.(Hons.) in physics and the M.B.M. (Operations Research) in 1979 and 1982, respectively, from the University of Calcutta. He received the M.Tech. and Ph.D. in computer science from the Indian Statistical Institute, Calcutta in 1984 and 1991, respectively. He was with Hindusthan Motors Ltd., W.B. from 1984 to 1985 and with Dunlop India Ltd., W.B. from 1985 to 1987. At present he is associated with the Electronics and Communication Sciences Unit of the Indian Statistical Institute and is currently visiting the University of West Florida, Pensacola. He is also a guest lecturer at the University of Calcutta. His research interests include image processing, pattern recognition, artificial intelligence, fuzzy sets and systems, uncertainty measures, and neural networks. He is a member of IEEE.

Sankar K. Pal is a Professor in the Electronics and Communication Sciences Unit, Indian Statistical Institute, Calcutta. He obtained his M.Tech. and Ph.D. degrees in radiophysics and electronics from Calcutta University, in 1974 and 1979, respectively, and his Ph.D./DIC in electrical engineering from Imperial College, London in 1982. He worked at Imperial College during 1979–83 (as a Commonwealth Scholar and an MRC Post-Doctoral Fellow), at the University of California, Berkeley and the University of Maryland, College Park, during 1986–87 (as a Fulbright Fellow), and at the NASA Johnson Space Center, Houston, Texas during 1990–1992 (as an NRC-NASA Senior Research Fellow). He served as a Professor-in-Charge of the Physical and Earth Sciences Division at the Indian Statistical Institute, during 1988–90. He was also a guest teacher in computer science, Calcutta University during 1983–86.

His research interests include pattern recognition, image processing, neural nets, and fuzzy sets and systems. He is co-author of the books *Fuzzy Mathematical Approach to Pattern Recognition* (Wiley, Halsted Press, N.Y., 1986) and *Fuzzy Models for Pattern Recognition: Methods that Search for Structures in Data* (IEEE Press, N.Y., 1992). He is an Associate Editor of the *International Journal of Approximate Reasoning* and *IEEE Transactions on Fuzzy Systems*, and a reviewer of *Mathematical Reviews* (American Mathematical Society). He is a Senior Member of IEEE, and a Life Fellow of IETE, New Delhi.

Dr. Pal received the 1990 Shanti Swarup Bhatnagar Prize (which is the highest and most coveted award for a scientist in India) in Engineering Sciences for his contribution in the field of pattern recognition and image processing.