Modified neocognitron for improved 2-D pattern recognition

C.N.S.Ganesh Murthy
Y.V. Venkatesh

Indexing terms: Neocognitron, 2-D pattern recognition, Training

Abstract: Some modifications to an existing neural network, the neocognitron, are proposed in order to overcome some of its limitations and to achieve an improved recognition of patterns (for instance, characters). Motivation for the present work arose from the results of extensive simulation experiments on the neocognitron. Inhibition during training is dispensed with, including it only during the testing phase of the neocognitron. Even during testing, inhibition is totally discarded in the initial layer because it leads, otherwise, to some undesirable results. However, inhibition, which is feature-based, is incorporated in the later stages. The number of network parameters which are to be set manually during training is reduced. The training is made simple without involving multiple training patterns of the same nature. A new layer has been introduced after the C-layer (of the neocognitron) to scale down the network size. Finally, the response of the S-cell has been simplified, and the blurring operation between the S- and the C-layers has been changed. The new architecture, which is robust with respect to small variations in the value of the network parameters, and the associated training are believed to be simpler and more efficient than those of the neocognitron.

1 Introduction

In visual pattern recognition, the objects to be recognised are subjected to various forms of transformation, involving shift, scale and rotation. The standard techniques found in the literature on pattern recognition come nowhere close to the human ability to perform transformation-invariant recognition. Obviously inspired by the remarkable power of the human vision system, attempts are being made to design an artificial neural vision machine for imitating some of its aspects. A basic problem in this context is to synthesise a network of these artificial neurons, in order to endow it with pattern-recognition capabilities.

The human visual system seems to have a hierarchical structure (see Hubel and Wiesel [1]), in which simple features are first extracted in the early layers from a stimulus pattern, then these are integrated, in the higher layers, into more complicated versions. In this hierarchy, a cell in a higher stage is likely to receive signals from a wider area of the retina, and, perhaps, as a consequence, is more insensitive to the position of the stimulus.

Many techniques have been proposed to solve the problem of character recognition. See, for instance, [2] for a review of these techniques. It is now acknowledged that neural techniques offer advantages in terms of speed, fault tolerance and adaptation. Many models have been suggested in the literature on the application of neural networks. These models differ in complexity and capabilities, and can be classified as models with and without self-organisation and feedforward networks consisting of layers of neurons. However, in this report, we consider only those which have been proposed for recognition of 2-D patterns.

The multi-layer perceptron (MLP) [3,4] is a well known architecture in feed-forward networks. However the MLP treats a 2-D pattern as a stacked 1-D vector, and hence, two similar patterns shifted by even one pixel are regarded as two different patterns. For the same reason, in an MLP, invariance to scale and distortion cannot be achieved in a straightforward way: the patterns are to be pre-processed to extract invariant features before using MLP as the classifier. Hence, a stand alone MLP without preprocessing cannot be used for recognition.

2 Review of related work

Fukushima has proposed neural networks [5–9] for the recognition of 2-D patterns which might have been subjected to certain types of deformation. His architecture consists of a cascade connection of pairs of cell layers, called S (simple) and C (complex) layers. Each layer consists of many planes; and each plane is a 2-D array of cells. The output of an S-layer is the input to the succeeding C-layer. The input to the first S-layer is from the photo-receptor, and the output of the last C-layer is the result of the recognition operation. In the last C-layer, each cell corresponds to a specific pattern (see Fig. 1).

The neocognitron is a hierarchical multi-layered neural network capable, according to its creator, of deformation-invariant visual pattern recognition. Even if the input pattern is deformed in shape, only one cell, corresponding to the category of the input pattern, is acti-

vated on the highest stage of the network. Other cells respond to other categories. Fukushima has reported that recognition of one of the ten numerals, with small shifts within the 16 by 16 image, has been successful.

Fig.1 Schematic structure of neocognitron. Inhibitory neurons are not shown

The Cognitron [5], the earlier model of Fukushima, uses unsupervised learning, and does not have the ability to correctly recognise position-shifted or shape-distorted patterns. In the neocognitron architecture [6], a technique dealing with deformations and shifts in position is addressed, but scaling is not taken care of. Because of unsupervised learning used in [6], the number of planes in the U_{s1} and the U_{s2} layer can only be figured out based on experimentation and heuristics. Among the 24 planes in the U_{s1} layer (see Fig. 12 in [6]) four of them respond to the vertical line (different positions). Also, because of the type of blurring and reduction of size from U_{s1} to U_{c1} (16x16 plane in U_{s1} corresponds to 10x10 in U_{c1}), patterns presented at some particular shifted positions will not be recognised properly, as demonstrated experimentally in [10,11]. To overcome these defects of [6], the recent paper of Fukushima [8] deals with an application of the neocognitron, and ostensibly attempts to make the network tolerant to deformations, shift and scaling. In [8], Fukushima uses shifted and scaled versions of the exemplars, to train the neocognitron to achieve shift and scale invariance.

A recent paper by Li and Wu [12] discusses the problem of rotation invariant-pattern recognition using the neocognitron. Li and Wu make a reference to the Fourier-Mellin transform approach of Casasent [13] for invariant recognition of 2-D patterns. It is well known that Fourier transform techniques entail high precision arithmetic, and are not suitable for computer implementation. While the multi-layered structure of Li and Wu is the same as the neocognitron, the input to their second layer is a set of rotated versions of the object, as though each rotated version is a separate object. The functions of the other layers are similar to those of the corresponding layers in the neocognitron.

Menon and Heinemann [14] have found that the neocognitron does not perform satisfactorily when it has to discriminate between three (somewhat larger) objects with larger shifts in a 128 by 128 image. A similar but independent conclusion, that the neocognitron does not possess a shift invariance property, has also been reported in [10,11]. It is found by Menon and Heine-

mann [14] that shift invariance can be obtained only by creating a model, which simply responds to the total energy in the image. However, this is not a satisfactory result as far as neural nets are concerned. It may be noted here that, Barnard and Casasent [15] have also analysed why the neocognitron fails to be an intrinsically shift-invariant pattern recogniser.

We now examine some aspects of processing carried out by the neocognitron, which is supposed to recognise patterns after an appropriate training phase, even in the presence of shifts, scale changes, and distortions. In this network, the C-cells are meant to make the network robust against distortion. The input and the output of a C-cell are related as follows:

$$C_L[p, j, k] = \Theta \left[\sum_r \sum_s D[r, s] U_{S_L}[p, j+r, k+s] \right] \quad (1)$$

where

$$\Theta(x) = x / (\sigma_L + x) \text{ if } x \geq 0$$

and

$$\Theta(x) = 0 \text{ if } x < 0$$

The parameter σ_L is a positive constant, which determines the degree of saturation of the output, and the weights $D[r, s]$ are chosen so as to decrease monotonically with respect to the distance from $(r, s) = (0, 0)$. Further, in the neocognitron, the feature extraction is carried out by S-cells, whereas tolerance to positional shift is accomplished by the C-cells. The output of a typical S-cell is given by:

$$S_L[i, j, k] = r_L \omega \left[\frac{1 + \sum_p \sum_r \sum_s A_L[i, p, r, s] C_{L-1}[p, j+r, k+s]}{1 + r_L * B_L[i] \frac{V_{L-1}(j, k)}{(1 + r_L)}} - 1 \right] \quad (2)$$

where $\omega(x) = \max(x, 0)$.

The parameter r_L controls the selectivity of the S-cell to an input pattern. The value of the inhibitory input V_{L-1} to the S-cell is given by

$$V_{L-1}[j, k] = \sqrt{\sum_i \sum_r \sum_s g[r, s] C_{L-1}^2[i, j+r, k+s]}$$

where the weights $g[r, s]$ are chosen so as to decrease monotonically with respect to the distance from $(r, s) = (0, 0)$.

The reinforcement strategy is as follows [2]:

$$A_L[i, p, r, s] = q_L g[r, s] C_{L-1}[p, r, s]$$

$$B_L[p] = q_L V_{L-1}[j, k]$$

where (j, k) denotes the coordinates of the representative S-cell, and q_L is a positive constant which determines the amount of reinforcement.

In order to provide the motivation for the present paper, the neocognitron (Fig. 1) has been simulated for verifying its ability to recognise two-dimensional patterns [10,11]. We present below an analysis of these results.

3 Analysis of experimental results obtained from the neocognitron

We recall that the scale invariance of Fukushima [5-8] actually requires training the network with a number of scaled versions of the same pattern, and then grouping the outputs. In order to show that the neocognitron

fails if the shifted and scaled versions of the patterns are not used to train the network, the training set is chosen to contain no shifted and scaled versions of the exemplars. This does not mean that using the shifted and scaled versions of the patterns to train the network will enhance the performance of the network. The specifications of the neural network actually simulated are given in Table 1.

Table 1: Specifications of the simulated network (Fukushima)

Input layer	1 plane 9 x 9 Layer 1	Layer 2	Layer 3
S-layer	4 planes (9x9)	8 planes (7x7)	4 planes (5x5)
Inhibitory input to the S-layer	1 plane (9x9)	1 plane (7x7)	1 plane (5x5)
Receptive area for each S-cell from each preceding C-plane	(3x3)	(3x3)	(3x3)
C-layer	4 planes (7x7)	8 planes(5x5)	4 planes (1x1)
Receptive area for each C-cell from the corresponding S-plane	(3x3)	(3x3)	(3x3)

The training patterns used for layers 1, 2 and 3 are shown in Fig. 2. For testing the pattern recognition capabilities of the simulated neocognitron, specific patterns were presented at the photoreceptor. These patterns and the corresponding recognition results are presented in Figs. 3-6.

On the basis of these experimental results, the following conclusions can be drawn:

- There is no robust way of choosing the fixed C-layer weights. It appears to be impossible to obtain a theoretical upper bound on the shift and scaling for the correct recognition of the learnt patterns.
- The weights $D[r, s]$ in the C-layers decrease monotonically with respect to the distance from $(r, s) = (0, 0)$. This results in the blurring of features, which, in turn, leads to ambiguity in the recognition of patterns.
- The global features extracted in the intermediate layers are a combination of averaged versions of the primitives which are input to the first layer. As a result, 'credit assignment' cannot be done uniquely.

Fig.2 Patterns used to train the various layers of the network (Fukushima)
Top row: Patterns used to train the four planes of layer 1
Second and third rows: Patterns used to train the eight planes of layer 2
Bottom row: Patterns used to train the four planes of layer 3

Fig.3 Patterns used to test the performance of the network (Fukushima) and the corresponding results for V
Top two patterns are correctly recognised, bottom 2 patterns are not correctly recognised

Fig. 4 Patterns used to test the performance of the network (Fukushima) and the corresponding results for H
Top two patterns are correctly recognised, bottom 2 patterns are not correctly recognised

Fig.5 Patterns used to test the performance of the network (Fukushima) and the corresponding results for Z
Top two patterns are correctly recognised, bottom 2 patterns are not correctly recognised

Fig.6 Patterns used to test the performance of the network (Fukushima) and the corresponding results for T
Top two patterns are correctly recognised; bottom 2 patterns are not correctly recognised

4 Modifications to the structure of the neocognitron

We now deal with modifications to be incorporated in the structure of the neocognitron [5–8], in order to achieve improved recognition capabilities. The inhibition is completely discarded in the training, but incorporated during the testing phase in the second and the third layers only. Patterns used for training the network are shown in Figs. 7–15. Note that, in the first layer, inhibition is not considered for the following reason: assume that the S-cell, in the initial (feature-extracting) layer, is designated to respond to a horizontal line. The presence of inhibition in this layer causes the S-cell to give a smaller output when a vertical line also exists along with the horizontal line in its field of view (see, for instance, Fig. 8f which is a part of character ‘T’), than when the horizontal line alone is present. Note that, in the initial layers, the presence of one primitive should not affect the response of the S-cell designated to respond to another primitive.

Fig.7 Patterns used to train the first layer of the proposed network

Fig.8 Patterns used to train the second layer of the proposed network (pattern set 1)

Fig.9 Patterns used to train the second layer of the proposed network (pattern set 2)

The inhibition of a cell is feature-based. For example, consider pattern Fig. 13d. This is the pattern ‘I’ which contains ‘T’ (Fig. 12d) as a sub-pattern, and which is close to the pattern ‘1’ (Fig. 14a). When ‘I’ is fed as input to the network, the inhibition to ‘T’ should come from the fact that the primitives in Fig. 8e and Fig. 9b

present in ‘I’ are absent in ‘T’, and the inhibition to ‘1’ should be as a result of the fact that primitives in Fig. 9a and Fig. 10b (in ‘I’) and Fig. 4n (in ‘1’) are different.

Fig.10 Patterns used to train the second layer of the proposed network (pattern set 3)

Fig.11 Patterns used to train the second layer of the proposed network (pattern set 4)

Fig.12 Patterns used to train the third layer of the proposed network (pattern set 1)

The training strategy used here, after modifying the neocognitron architecture, is distinct from that of Fukushima [5–8], as explained below.

The input characteristic of an S-cell during the training phase is modified as follows:

$$S_L[i, j, k] = \Phi \left[\frac{\sum_p \sum_l \sum_m A_L[i, p, l, m] C_{L-1}[p, j+l, k+m]}{\|A_L\| \|C_{L-1}\|}, T, \alpha \right] \quad (3)$$

where the norm of the matrix A , is obtained from summing the squares of the elements of A_L for the indices p, l and m (i.e. excluding i the first index); and the norm of C_{L-1} is similarly obtained from summing the squares of the elements of C_{L-1} , but only for those values of indices (p, l and m) for which $A_L[i, p, l, m]$ is non-zero. The characteristic of Φ is shown in Fig. 16. This kind of response of an S-cell enables it to extract a certain feature of a prescribed pattern, in spite of the presence of other features.

Fig. 13 Patterns used to train the third layer of the proposed network (pattern set 2)

Fig. 14 Patterns used to train the third layer of the proposed network (pattern set 3)

Fig. 15 Patterns used to train the third layer of the proposed network (pattern set 4)

Fig. 16 Characteristics of Φ in the equation for $S_L[i, j, k]$ in Section 4

The input-output characteristic of a C-cell is prescribed as follows:

Find $max = \max S(j+l, k+m)$, $-1 \leq l, m \leq 1$

Let the coordinates of max be (u, v)

Calculate $d(l, m) = \sqrt{(u-l)^2 + (v-m)^2}$

The output of the C-cell is given by

$$C(j, k) = \left[\sum_{l, m} S(j+l, k+m) q^{d(l, m)} \right]$$

where $q \approx 0.1$.

The response of the C-cell is similar to the one used in the neocognitron [5–8]. However, $d(l, m)$ has a peak at (u, v) (where S has a maximum) instead of at $(0, 0)$. This kind of response gives the same effect as the one obtained by training with shifted primitives. Note that no shifted and scaled versions of the patterns are used for training.

The training algorithm is as follows: The central S-cell of the S-plane, which is to be trained, is selected as the representative. The receptive area of the S-cell at the input layer is calculated by backtracking. The pattern is presented within this receptive area at the input, and the weights are fixed in the following manner:

$$A_L[i, p, l, m] = C_{L-1}(p, l, m)$$

where i = index of the S-plane in module L ; p = index

of the C-plane in module $(L - 1)$; (l, m) = coordinates of the weight and C-plane arrays.

A new inhibition strategy, based on features, is employed during the testing phase, to enhance the performance of the network in classifying patterns which are close, like '1' and 'T' and 'E' and 'F'. To this end, the responses of neurons of second and third S-layers are modified during the testing phase. The modifications, for instance, for the second layer ($L = 2$) are as follows:

$$S_L[i, j, k] = \Phi \left(\frac{\sum_p \sum_l \sum_m A_L[i, p, l, m] \times C_{L-1}[p, j + l, k + m] - \lambda * ib2}{\|A_L\| \|C_{L-1}\|}, T, \alpha \right)$$

On the right hand side, the term involving $\lambda * ib2$ (with the latter parameter of the order 0.002) provides an inhibitory input, whenever there is a mismatch between the values of $C_{L-1}[p, j + l, k + m]$ and $A_L[i, p, l, m]$. This term will have larger magnitude if the input pattern contains a primitive not represented by the weights $A_L[i, p, l, m]$, or if the input pattern does not contain the primitive represented by the weights $A_L[i, p, l, m]$.

For the third layer, we further calculate the following quantities:

$$taef[z] = \max_{(j,k)} S[z, j, k]$$

$$feat[i][z] = \sum_l \sum_m A_3[i, z, l, m]$$

For each training pattern i of the third (final) layer, the weights $A_L[i, z, l, m]$ represent primitive z of the second layer. Hence, $feat[i][z]$, in a way, indicates whether the pattern i contains the primitive z of the second layer. Similarly, $taef[z]$ represents the presence of primitive z in the input test pattern. The dot normalised product,

$$asum[i] = \frac{\sum_z taef[z] feat[i][z]}{\|taef\| \|feat[i]\|}$$

is a measure of similarity between the vector of second layer primitives present in the training pattern i , and the vector of the second layer primitives present in the input test pattern. Let

$$max = \max_i asum[i]$$

The quantity $(max - asum[i])$ (which is minimum when $asum[i]$ is maximum and vice-versa) is used as an additional inhibitory input in calculating the response of the S-cell in the third layer:

$$S_L[i, j, k] = \Phi \left(\frac{\sum_p \sum_l \sum_m A_L[i, p, l, m] \times C_{L-1}[p, j + l, k + m] - \lambda * ib3}{\|A_L\| \|C_{L-1}\|}, T, \alpha \right) - (max - asum[i]) * 0.5$$

where λ is defined as before, and $ib3$ is of the order 0.02. As explained earlier, λ will be larger in magnitude if the input test pattern contains a primitive not repre-

sented by $A_L[i, p, l, m]$, or does not contain the primitive represented by $A_L[i, p, l, m]$. We should add here, that these changes have to be made only in the testing phase of the network. The responses of the S-cells are given by eqn. 3 during the training phase.

Two types of network arrangements are considered incorporating the above modifications. The input layer C_0 followed by $S_1 - C_1 - R_1 - S_2 - C_2 - S_3 - C_3$ in the first type and by $S_1 - C_1 - S_2 - C_2 - R_2 - S_3 - C_3$ in the second. As explained before, the S_i layers extract the features, and C_i layers do the blurring, making the network more tolerant. The R_i layers are introduced to scale down the size. In the first type, the scaling down is affected just after the C_1 layer, whereas in the second type, it happens after the C_2 layer. In both the cases, the scaling (which is non-overlapping) is by a factor of 1/4.

5 Comparison with the neocognitron

On the basis of extensive simulation studies, it has been found that the modifications proposed lead to a substantial improvement in the performance of the neocognitron for pattern recognition. Some advantages of the modified neocognitron are now given.

- The response of the S-cell is different from the one in [5-8], and is greatly simplified.
- Since only one set of training patterns is used, and no shifted and scaled versions are considered, it takes less time to train the net.
- The number of constants to be fixed during the training of the network is now reduced to six. These are the threshold constants (**Th1**, **Th2**, **Th3**) and the slopes of the nonlinearity ($\alpha_1, \alpha_2, \alpha_3$) at the output of the S-cell in the three layers. The α_i s are generally in the range 0.1 - 0.15, and the performance of the network is robust with respect to small variations in the value of α_i . The threshold values during the training phase were fixed as **Th1** = 0.7 and **Th2** = 0.88 (**Th3** does not figure during training), and during the testing the values were **Th1** = 0.7, **Th2** = 0.82 and **Th3** = 0.82. As in the case of the α_i s, small variations in the values of (**Th1**, **Th2**, **Th3**) do not affect the performance of the network significantly.
- Using the training pattern set which does not contain shifted and scaled versions of the patterns, the network is still tolerant to the scale changes in the patterns, to some extent. The tolerance to the scale changes is due to the presence of the C-layers. This can be explained in a heuristic way. Consider the pattern 'V' (Fig. 12a), made up of primitives shown in Fig. 7a, c and d (of the first layer) and primitives shown in Fig. 8a, b and Fig. 11c (of the second layer). When the pattern 'V' is presented to the network to train the last layer, the outputs of the corresponding planes of the second S-layer have a larger value at the points where the features occur, thus maintaining the positional relationship between the features. Now these points will be smeared by the following C-layer, and the weights, corresponding to the primitives (Fig. 8a, b and Fig. 11c) are given in Table 2. It is this smearing caused by the C-layer that is responsible for the network to be tolerant to some extent to changes in scale. Fig. 17 shows the positions of the significant values when the three matrices corresponding to the primitives (Fig. 8a, b and Fig. 11c) are superimposed. It can be seen that when a

small 'v' or a big 'V' is presented, its features fall within the space covered by the weights (as shown in Fig. 8), and the recognition is effected. Thus, in a heuristic way, it can be demonstrated that the network has a reasonable scale tolerance. However, for some patterns, the range of scale for successful recognition was found to be as high as 1 : 4 in our simulations.

Table 2: Weights (corresponding to primitives Fig. 8a, Fig. 8b and Fig. 11c) of the neuron for extracting the feature Fig. 12a

0.23	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.23
0.35	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.33
0.66	0.05	0.00	0.00	0.00	0.00	0.00	0.01	0.31
1.12	0.59	0.00	0.00	0.00	0.00	0.00	0.00	0.25
1.21	0.82	0.11	0.00	0.00	0.00	0.00	0.00	0.18
1.11	0.82	0.37	0.00	0.00	0.00	0.00	0.00	0.09
0.56	0.53	0.40	0.28	0.00	0.00	0.00	0.00	0.00
0.00	0.39	0.41	0.30	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.19	0.19	0.00	0.00	0.00	0.00	0.00
0.24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.24
0.34	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.36
0.32	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.67
0.25	0.00	0.00	0.00	0.00	0.00	0.00	0.59	1.12
0.18	0.00	0.00	0.00	0.00	0.00	0.11	0.83	1.21
0.08	0.00	0.00	0.00	0.00	0.00	0.38	0.83	1.11
0.00	0.00	0.00	0.00	0.01	0.29	0.42	0.54	0.57
0.00	0.00	0.00	0.00	0.18	0.31	0.42	0.40	0.00
0.00	0.00	0.00	0.00	0.18	0.20	0.20	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.05	0.05	0.00	0.00	0.00	0.00	0.00	0.06	0.06
0.05	0.15	0.14	0.00	0.00	0.00	0.15	0.16	0.06
0.05	0.21	0.21	0.19	0.00	0.20	0.22	0.22	0.06
0.00	0.20	0.35	0.82	0.89	0.82	0.35	0.21	0.00
0.00	0.00	0.37	1.14	1.24	1.14	0.37	0.00	0.00

Fig.17 Schematic depicting the scale tolerance of the network
Encircled regions: location of large values of weights due to the three primitives of Fig. 8a, Fig. 8b and Fig. 11c

However, there are limitations to the proposed modifications.

- As explained earlier, the parameters α_i and the threshold values $Th1$, $Th2$ and $Th3$ are to fixed manually, and this procedure is believed to be quite simple.
- If the number of patterns for training (and hence for recognition) is to be increased, then the network size also has to be increased. This is owing to the fact that more planes are to be employed to detect more primitives in the S2 layer, and is an inherent requirement of neocognitron-like architectures. However, the number of planes to be included depends on the number of primitives needed to represent all the patterns under consideration. A careful choice of primitives can be

made before hand, to represent as many patterns as possible. It should therefore be noted that the number of planes needed to detect the primitives does not necessarily increase with the number of patterns, but saturates after some stage.

- Another reason for the increase in the network size is that we may have to increase the field of view of S-cells, to enable them to differentiate patterns closely resembling each other. The reason for an increase in the field of view is given by considering, for example, primitives shown in Fig. 9a and c. After blurring, these primitives will have overlapping supports, and hence they appear to be similar. Therefore a field of view of 5×5 is insufficient to correctly distinguish between the two. In order to succeed in this case, we need to train the network with larger primitives, so that, even after blurring, the limbs of the primitives are of sufficient length to preserve their identity. This necessitates the choice of larger patterns and a greater field of view to accomodate the limbs. Note, however, that this increase is limited.

We give, in Table 3, the details of a typical simulated network. Other configurations (like $C_0 - S_1 - C_1 - S_2 - C_2 - R_2 - S_3 - C_3$) for recognising 16 patterns have also been tried out, which are omitted here in view of space constraints. See [11] for details.

Table 3: Network specifications for recognising 16 patterns for the configuration $C_0 - S_1 - C_1 - R_1 - S_2 - C_2 - S_3 - C_3$ where the network is scaled down in layer R_1

Input layer	1 plane 21 x 21		
	Layer 1	Layer 2	Layer 3
S-layer	4 planes (21x21)	24 planes (10x10)	16 planes (10x10)
Receptive area for each S-cell from each preceding C-plane	(3x3)	(7x7)	(8x8)
C-layer	4 planes (21x21)	24 planes (10x10)	16 planes (1x1)
Receptive area for each C-cell from the corresponding S(R)-plane	(3x3)	(3x3)	(10x10)

5.1 Discussions

We have extensively tested the performance of the modified neocognitron, using patterns of various sizes and shifts shown in Figs. 18–33. The specifications for designing the network to recognise sixteen patterns are given in Table 3. The training patterns for the first layer are given in Fig. 7, those for the second layer are shown in Figs. 8–11, and those for the third layer in Figs. 12–15. In Figs. 18–33, the outputs of the neurons in the last layer corresponding to each of the patterns are given. The character corresponding to the largest output is taken as the recognition result. If some of the patterns are close to the input test pattern, the corresponding outputs are seen to be significant. The results shown in Figs. 18–33 indicate the superiority in the performance, obtained as a result of the proposed modifications to the network.

Fig.18 Patterns used to test the network configuration $C_0 - S_1 - C_1 - R_1 - S_2 - C_2 - S_3 - C_3$ and the corresponding results (V)

Fig.22 Pattern used to test the network configuration $C_0 - S_1 - C_1 - R_1 - S_2 - C_2 - S_3 - C_3$ and the corresponding results (M)

Fig.19 Patterns used to test the network configuration $C_0 - S_1 - C_1 - R_1 - S_2 - C_2 - S_3 - C_3$ and the corresponding results (H)

Fig.23 Pattern used to test the network configuration $C_0 - S_1 - C_1 - R_1 - S_2 - C_2 - S_3 - C_3$ and the corresponding results (N)

Fig.20 Patterns used to test the network configuration $C_0 - S_1 - C_1 - R_1 - S_2 - C_2 - S_3 - C_3$ and the corresponding results (Z)

Fig.24 Patterns used to test the network configuration $C_0 - S_1 - C_1 - R_1 - S_2 - C_2 - S_3 - C_3$ and the corresponding results (W)

Fig.21 Patterns used to test the network configuration $C_0 - S_1 - C_1 - R_1 - S_2 - C_2 - S_3 - C_3$ and the corresponding results (T)

Fig.25 Patterns used to test the network configuration $C_0 - S_1 - C_1 - R_1 - S_2 - C_2 - S_3 - C_3$ and the corresponding results (I)

6 Conclusions

In order to provide motivation for the proposed modifications of the neocognitron [5–8], we have presented some experimental results obtained from its simulation.

It has been found that, for an improved recognition of two-dimensional patterns by the neocognitron, changes are needed. There should be no inhibition in the initial layer, thus enabling a more efficient extraction of the

Fig.26 Patterns used to test the network configuration $C_0 - S_1 - C_1 - R_1 - S_2 - C_2 - S_3 - C_3$ and the corresponding results (I)

Fig.30 Patterns used to test the network configuration $C_0 - S_1 - C_1 - R_1 - S_2 - C_2 - S_3 - C_3$ and the corresponding results (J)

Fig.27 Patterns used to test the network configuration $C_0 - S_1 - C_1 - R_1 - S_2 - C_2 - S_3 - C_3$ and the corresponding results (4)

Fig.31 Patterns used to test the network configuration $C_0 - S_1 - C_1 - R_1 - S_2 - C_2 - S_3 - C_3$ and the corresponding results (X)

Fig.28 Patterns used to test the network configuration $C_0 - S_1 - C_1 - R_1 - S_2 - C_2 - S_3 - C_3$ and the corresponding results (E)

Fig.32 Patterns used to test the network configuration $C_0 - S_1 - C_1 - R_1 - S_2 - C_2 - S_3 - C_3$ and the corresponding results (Y)

Fig.29 Patterns used to test the network configuration $C_0 - S_1 - C_1 - R_1 - S_2 - C_2 - S_3 - C_3$ and the corresponding results (F)

Fig.33 Patterns used to test the network configuration $C_0 - S_1 - C_1 - R_1 - S_2 - C_2 - S_3 - C_3$ and the corresponding results (K)

primitives the input pattern is made up of. Inhibition can be included in the later stages and should be feature-based to enable efficient classification. However, it should also be noted that the proposed architecture is still not intrinsically meant for recognition of patterns

subjected to rotation and/or occlusion. In fact, a different approach based on feature extraction (based, for instance, on location of corners and their connectivity) is needed. These results will be reported elsewhere [16,17].

7 References

- 1 HUBEL, D., and WIESEL, T.: 'Shape and arrangement of columns in the cat's striate cortex', *J. Physiology*, 1963, **165**, pp. 559–567
- 2 GOVINDAN, V.K., and SHIVAPRASAD, A.P.: 'Character recognition – a survey', *Pattern Recognit.*, 1990, **23**, (7), pp. 671–683
- 3 LIPPMAN, R.P.: 'An introduction to computing with neural networks', *IEEE ASSP Mag.* April 1987, pp. 4–22
- 4 LIPPMANN, R.P.: 'Pattern classification using neural networks', *IEEE Commun. Mag.* November 1989, pp. 47–64
- 5 FUKUSHIMA, K.: 'Cognitron: a self-organising multi-layered neural network model', *Biol. Cybern.*, 1975, **20**, pp. 121–136
- 6 FUKUSHIMA, K., and MIYAKE, S.: 'Neocognitron: a new algorithm for pattern recognition tolerant of deformations and shifts in position', *Pattern Recognit.*, 1982, **15**, (6), pp. 455–469
- 7 FUKUSHIMA, K.: 'Neocognitron, a self-organising neural network model for a mechanism of pattern recognition unaffected by shift in position', *Biol. Cybern.*, 1980, **36**, pp. 193–202
- 8 FUKUSHIMA, K.: 'Handwritten alphanumeric character recognition by the neocognitron', *IEEE Trans. Neural Netw.* 1991, May 1991, **2**, pp. 355–365
- 9 FUKUSHIMA, K.: 'Analysis of the process of visual pattern recognition by the neocognitron', *Neural Netw.*, 1989, **2**, pp. 413–420
- 10 VENKATESH, Y.V., and GANESH MURTHY, C.N.S.: 'Experimental investigations on the performance of the neocognitron for 2-D pattern recognition', International conference on *Information processing*, Singapore, December 1991
- 11 VENKATESH, Y.V., and GANESH MURTHY, C.N.S.: 'Modified neocognitron for improved 2-D pattern recognition', Technical Report, March 1994 and March 1995 (revised), Department of Electrical Engineering, Indian Institute of Science, Bangalore, India
- 12 LI, C., and WU, C.H.: 'Introducing rotation invariance into the neocognitron model for pattern recognition', *Pattern Recognit. Lett.* December 1993, **14**, pp. 985–995
- 13 CASASSENT, D., and PSALTIS, D.: 'Position, rotation and scale invariant optical correlation', *Appl. Opt.*, 1976, **15**, pp. 1795–1799
- 14 MENON, M.M., and HEINEMANN, K.G.: 'Classification of patterns using a self-organising neural network', *Neural Netw.*, 1988, **1**, pp. 201–215
- 15 BARNARD, E., and CASASSENT, D.: 'Shift invariance and the neocognitron', *Neural Netw.*, 1990, **3**, pp. 403–410
- 16 GANESH MURTHY, C.N.S., and VENKATESH, Y.V.: 'Character recognition using encoded patterns as inputs neural networks', National conference on *Neural networks and fuzzy systems*–NCNNFS 1995, Anna University, Madras, India, 16–18 March 1995, pp. 104–114
- 17 GANESH MURTHY, C.N.S., and VENKATESH, Y.V.: 'A new method for encoding patterns for classification by neural networks', *IEEE Trans. Neural Netw.* (submitted)