To the Graduate Council:

I am submitting herewith a thesis written by Basem F. Awad entitled "OBJECT RECOGNITION USING NEOCOGNITRON." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

Dragana Brzakovic, Major Professor

We have read this thesis
and recommend its acceptance:

Accepted for the Council:

Associate Vice Chancellor
and Dean of The Graduate School

# OBJECT RECOGNITION USING NEOCOGNITRON

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Basem F. Awad

May 1992

## ACKNOWLEDGMENTS

I would like to express my sincere appreciation to my major professor, Dr. Dragana Brzakovic, for her continuous support and help throughout my graduate study. Many thanks to Drs. J.M. Bailey and P.B. Crilly for serving on my thesis committee. Many thanks are due to Hal E. Beck for proofreading this thesis.

I also would like to thank the Center for Measurement and Control, and Instrumentation and Controls Division at ORNL for support, and in particular Mr. D. McDonald for his support.

Finally, I would like to thank my family and friends whose constant support and encouragement made it possible for me to reach this goal.

# ABSTRACT

The implementation of a neural network model called *Neocognitron* is presented in this thesis. The Neocognitron is a hierarchical, multiresolution, self-organizing structure (learning without a teacher) that recognizes objects on the basis of the geometrical similarity of their shapes. First, the basic structure of the Neocognitron was modified for the recognition of multiple inputs. Next, the Neocognitron was expanded into a modular design in order to acquire the ability to recognize a large number of classes. The modular design consists of a number of Neocognitron-type networks that are connected in parallel and that receive input from a single input layer. The modular version of the Neocognitron was trained with five different groups of alphanumeric characters.

# TABLE OF CONTENTS

# LIST OF FIGURES

| FIGURE | | PAGE |
|---|---|---|

| FIGURE | | PAGE |
|---|---|---|

# CHAPTER 1

# INTRODUCTION

Inspired by biological information processing systems, neural networks are, in essence, computer-based simulations which resemble, in various degrees, the structure of the biological nervous system. The theory and implementation of neural network models have been developing rapidly during the past two decades. A great deal of research is devoted to such networks, especially in the medical and military areas, e.g., analyzing speech in hearing aids for the profoundly deaf, reading X-ray pictures, and classifying radar signals.

One of the most serious difficulties in designing pattern recognition type machines has been the problem of handling the shift and size variations and distortion of an input pattern. Neural networks are more capable of handling such difficulties than conventional methods. Conventional methods include invariant pattern recognition techniques such as moment invariance [19] and polar-coordinates Fourier transform [2]. These methods have been successful in a limited range of applications. In the hopes that neural networks will perform better than conventional methods, they have become the target for invariant pattern recognition.

This thesis presents a neural network called Neocognitron [1, 6, 7, 9, 10, 11], which recognizes patterns on the basis of the geometrical similarities of their shapes. The design of this network was motivated by the hierarchical model of the visual nervous system suggested by Hubel and Wiesel [18]. According to the hierarchic model of Hubel and Wiesel, the neural network in the visual cortex has a hierarchic structure. In a similar manner, the Neocognitron has a hierarchical, multilayered structure. It consists of an input layer followed by cascade connections of a number of multiresolution layers. The input pattern is a two-dimensional binary image. The boundaries (edges) of the object are represented by the binary value "1" and the background is represented by the binary value "0". A second characteristic adapted from physiological systems is the network ability to self-organize. During the learning process, the network develops and updates its learning parameters without interference from the outside. The network performs pattern recognition as follows: local features of the input pattern are extracted by cells of the first layer, and these local features are gradually integrated into more global features at higher layers. Finally, each cell of the highest layer integrates all the extracted features of the input pattern and responds with a single output, signifying recognition of the input pattern

Presented here is a modification of the basic structure of the Neocognitron for multiple input recognition. Originally, the Neocognitron was designed to recognize a single pattern presented at its input layer. By modifying the firing mechanism of the Neocognitron, the network is shown to acquire the ability to recognize two patterns presented simultaneously at its input layer.

The Neocognitron has difficulties in terms of the number of training patterns it can handle. These difficulties involve:

- Tedious modifications of its simulation parameters.

- Increase of the network size which results in less efficient pattern recognition performance.

An alternative approach is presented here for learning a large number of training patterns. The approach involves an extension of the Neocognitron into a modular design. A number of Neocognitron-type networks are connected in parallel and they receive their input from one input layer. This approach avoids the tedious modification of the network parameters and the size increase of the Neocognitron.

This thesis is divided into two parts. The first part, Chapter 2, describes the structural and the mathematical concepts of the Neocognitron and the modification of the basic structure of the Neocognitron. Chapter 2, contains a numeral recognition example by the Neocognitron. The second part, Chapter 3, describes the modular design of the Neocognitron. An example of alphanumeric character classification by the modular design of the Neocognitron is also presented in Chapter 3.

# CHAPTER 2

# NEOCOGNITON

## 2.1 Neural Networks: an Overview

The field of neural networks has attracted the interest of many researchers in the areas of engineering, computer science, and mathematics. This is due to the fact that neural networks are capable of reacting and adapting to the changing environment. Generally, a neural network structure consists of a number of layers. First is the input layer, where the values of the input pattern are represented by a vector. Second is the middle "hidden" layer, which consists of "feature detectors". Sometimes there is more than one hidden layer. This layer is called hidden because it does not communicate with the outside environment, but it does communicate with the input and output layers. The last layer is the output layer. The output layer functions as a classifier of the different input patterns. Information is transmitted from one layer to another via variable or fixed connections called weights. A layer consists of one or more processing elements (cells). Each processing element receives either one or more connections as its input from the previous layer and one or more connections diverge from its output to the next layer. When we say a neural network trains or learns with a training set, basically we mean reinforcements or adjustments of the variable connections (weights) between layers. Neural networks have the ability of supervised or unsupervised learning. In supervised learning, a teacher is required. A teacher may be a set of data or an observer who evaluates the performance of the network. In other words, the network is forced to respond in a predetermined way. This may be done by comparing the output of the network with the desired output. In unsupervised learning, the network does not require a teacher or interference from the outside. Unsupervised learning is also called *self-organization*. The network uses certain learning criteria to self-organize and adjust its weights. Both in the supervised and unsupervised methods, learning is done by reinforcements of the variable connections. Many neural networks models have been developed, yet the Neocognitron is the largest and the most complicated neural network in terms of its structure and its learning mechanism. In order to give some insight on how the Neocognitron differs from other neural networks, we will compare it to a neural network model called counterpropagation. Counterpropagation network consists of two layers, a self-organizing map [21, 22] (called Kohonen layer) and Grossberg layer [14, 15]. The Kohonen layer is a feature-extracting layer and the Grossberg layer is a classifier. Feature extraction in the counterpropagation network uses a Euclidean distance between the input pattern and the Kohonen layer, while feature extraction in the Neocognitron network uses the correlation between the input pattern and the feature extraction layer. The counterpropagation network receives its input as a one-dimensional array; the Neocognitron receives its input as a two-dimensional vector. Classification in the counterpropagation network requires supervision, while classification in the Neocognitron doesn't.

One unique feature about the Neocognitron is its hierarchical structure. This means that the cells of each layer concern themselves with the processing of small regions of the input. At the last stage, all the features extracted by lower layers are combined and extracted by a single cell at the output layer to produce the final result.

## 2.2 Introduction To Neocognitron

This chapter presents a neural network model for object recognition called the *Neocognitron*. It is a hierarchical, multilayered neural network which resembles to a certain extent the mechanism of information processing in the mamalian visual system. The Neocognitron was a follow up work on an earlier model called Cognitron [5]. The Cognitron is also a hierarchical neural network for object recognition, but its response is affected by the position of the input pattern, that is, the same pattern presented at two different positions is recognized as two different patterns. In the Neocognitron, however, the response of the network is not affected (for some tolerance) by the position of the input pattern. The Neocognitron has the ability of supervised or unsupervised learning. In the unsupervised case, repeated presentation of the input pattern is sufficient to self-organize the network, and it is not necessary to give any information about the categories in which these patterns should be classified. In other words, the network uses the firing strength of the cells (maximum output cells) as the basis of adjusting the weights of the interconnected coefficients between layers. The network builds its own feature-extracting cells without external influence. In the supervised case, maximum output cells are selected by a teacher to have their connections reinforced. The Neocognitron recognizes input patterns according to the similarity in their shapes. As mentioned earlier, the input patterns are binary images. The boundary of the pattern is described by a binary value "1" and the background is described by a binary value "0".

## 2.3 Structure of the Network

Figure 2.1 shows the overall architecture of the network. It consists of three parts: the input layer, the learing units, and the recognition layer.

### 2.3.1 Input Layer

The input layer is a two dimensional array which serves as a buffer to the next layer because the images presented in the input layer are not preprocessed. The input patterns are binary images with one pixel width describing the boundary of the objects.

### 2.3.2 Learning Units

As mentioned earlier, the Neocognitron has a hierarchical structure. This means that it is a multiresolution structure and the size of the layers decreases as we approach the output layer. The network consists of a number of similarly struc-

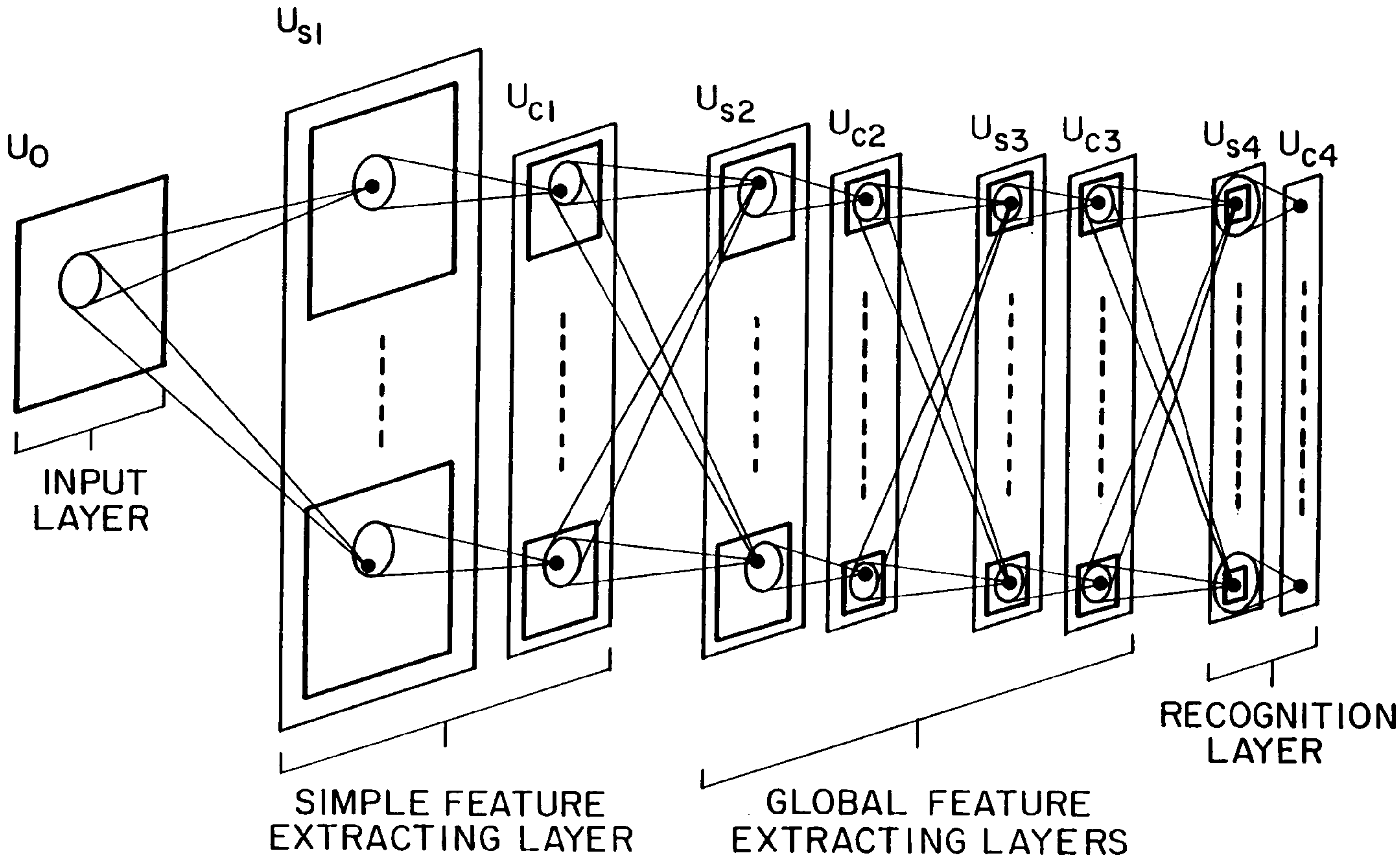



Figure 2.1: Structure of the Neocognitron. Four Us-Uc pair units are shown.

tured units. Each unit consists of two layers denoted by Us and Uc layers. In Figure 2.1, there are four units of Us and Uc layers. Each layer is subdivided into a number of two dimensional planes (cell-planes) called S-planes and C-planes, respectively. An S-plane contains a number of S-cells and a C-plane contains a number of C-cells. The number of cells depends on the size of the cell-plane. For example, an 8 x 8 cell-plane contains 64 cells. The S-cells feature extracting cells, and the C-cells serve as an input to the S-layer which comes after it. Each cell receives its input from a small region of the layer that precedes it. For example, an S-cell from the first Us layer receives its input from the C-cells enclosed by the circle in $U_0$. For the sake of simplicity, only one cell is shown in each cell plane. Connections converging to an S-cell are variable and modifiable. Connections converging to C-cells are fixed and they are ummodifiable. Figure 2.2 shows input connections to an S-cell from a number of C-cells. All the other S-cells in the same S-plane receive the same kind of connections. Each S-cell receives two types of variable connections: excitatory connections and inhibitory connections. Figure 2.3 shows the two types of variable connections converging to an S-cell. Excitatory variable connections are denoted by $a(\nu)$, and inhibitory variable connections are denoted by $b$.

$$U_{sl}(k_l, n) = r_l\varphi[\frac{1 + \sum_{k_{l-1}=1}^{K_{l-1}} \sum_{\nu \in S_l} a_l(k_{l-1}, \nu, k_l) U_{cl-1}(k_{l-1}, n+\nu)}{1 + \frac{r_l}{1+r_l} b_l(k_l) V_{cl-1}(n)} - 1], \quad (2.1)$$

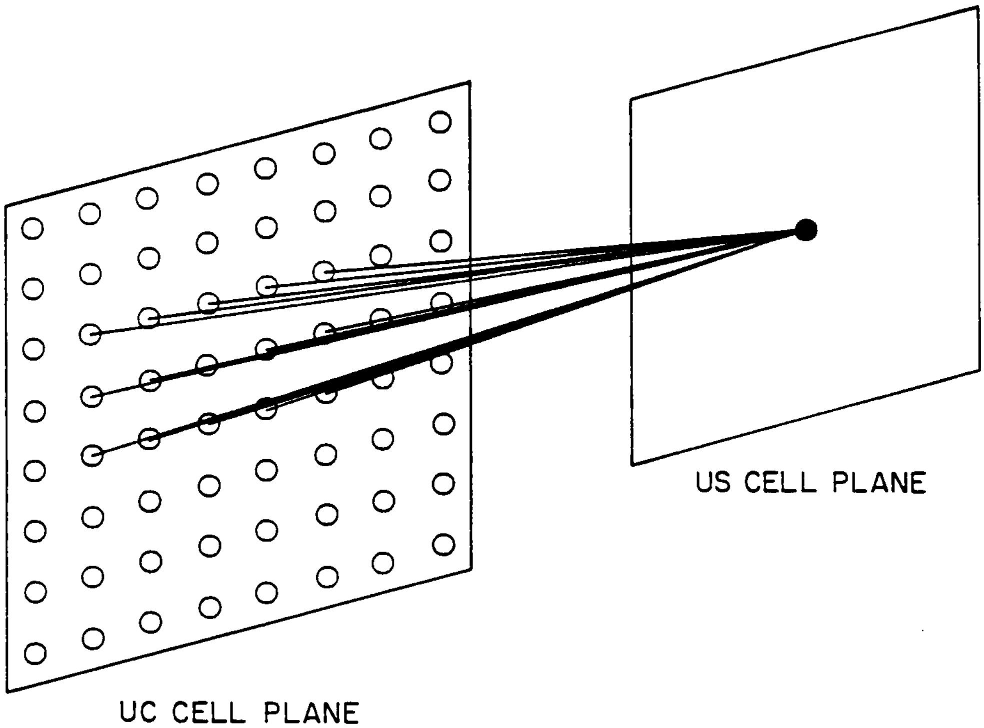
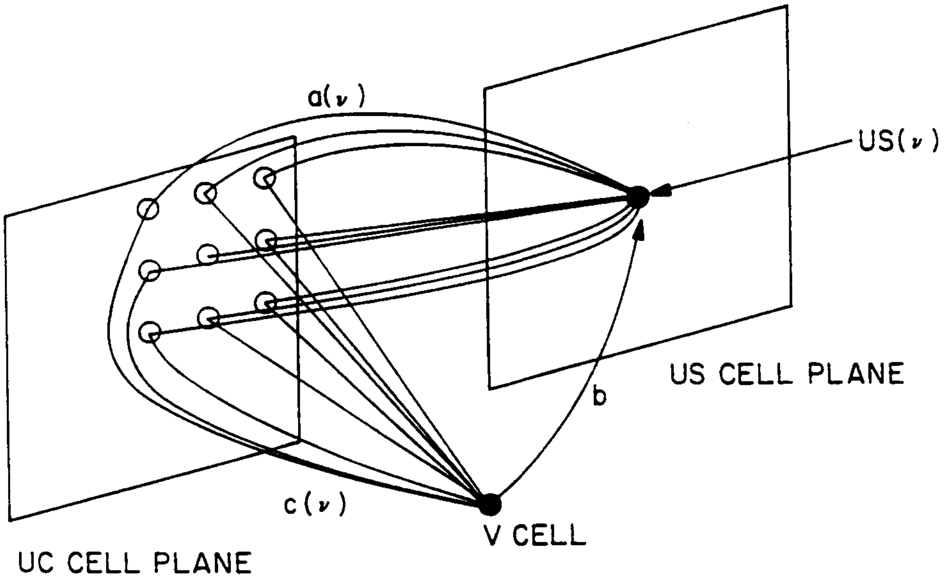



Figure 2.2: Connections converging to an S-cell in an S-plane from C-plane. Only one cell is shown in the S-plane.




$Us(\nu)$ A processing element in a Us cell plane.

$a(\nu)$ Excitory variable connections converging to an S-cell.

$c(\nu)$ Fixed connections converging to the inhibitory cell V.

$V$ CELL Inhibitory cell.

$b$ Inhibitory Variable connection converging to an S-cell.

Figure 2.3: Two types of variable connections converging to an S-cell from C-plane: excitory and inhibitory connections.

where

$$\varphi[x] = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}, \tag{2.2}$$

and the subscript $l$ indicates the unit number. In Figure 2.1, $l$ ranges from 1 - 5. In the case of $l = 1$ in Equation 2.1, $U_{cl-1}(k_{l-1}, n)$ stands for the input layer $U_0(n)$, where n = (nx,ny) is a two-dimensional coordinate indicating the location of a cell. Symbol $K_l$ denotes the S-plane number in an S-layer; $S_l$ is the connectable area of an S-cell. In Figure 2.1, the connectable area for a cell is the area enclosed by the circle. A two dimensional coordinate describing the position of the cells in $S_l$ is denoted by $\nu = (\nu x, \nu y)$; $r_l$ is the selectivity parameter and will be described later in the chapter.

$V_{cl}(n)$ is an inhibitory cell described by

$$V_{cl}(n) = \sqrt{\sum_{k_{l-1}=1}^{K_{l-1}} \sum_{\nu \in S_l} c_{l-1}(\nu) U_{cl}^2(k_{l-1}, n + \nu)}. \tag{2.3}$$

The term $c_{l-1}(\nu)$ is a two dimensional Gaussian mask which controls the weight adjustments of the feature extracting planes as well as the inhibitory parameter $b$. In Equation 2.1, $a_l(k_{l-1}, \nu, k_l)$ represents the excitatory variable connections and $b_l(k_l)$ is the inhibitory connection. If the relevant features are present in the input, the excitatory connections $a_l(k_{l-1}, \nu, k_l)$ will be larger than the inhibitory connection, which will make the output larger than zero and the S-cells in all S-planes fire. By the same token, if the relevant features are not present in the input, value $b_l(k_l)$ will be larger than the excitatory connections and the output will be less than zero, which will cause the cell to shut off.

Similarly, the output of a C-cell in the $k_l$ C-plane in the *l-th* module is computed by

$$U_{cl}(k_l, n) = \Psi[\frac{1 + \sum_{\nu \in D_l} d_l(\nu) U_{sl}(k_l, n + \nu)}{1 + V_{sl}(n)} - 1], \tag{2.4}$$

where

$$\Psi[x] = \varphi[x/(\alpha + x)], \tag{2.5}$$

and $V_{sl}(n)$ is an inhibitory cell described by

$$V_{sl}(n) = 1/K_l \sum_{k_l=1}^{K_k} \sum_{\nu \in D_l} d_l(\nu) U_{sl}(k_l, n + \nu). \tag{2.6}$$

The term $d_l(\nu)$ is also a two dimensional Gaussian mask; $D_l$ is the connectable area of a C-cell; $\alpha$ is a positive constant which determines the degree of saturation of the output.

The variable interconnections from C-cells to S-cells are reinforced according to the equation:

$$\Delta a_l(k_{l-1}, \nu, k_l) = q_l c_{l-1}(\nu) U_{cl-1}(k_{l-1}, n + \nu), \tag{2.7}$$

and

$$\Delta b_l(k_l) = q_l v_{cl-1}(n), \tag{2.8}$$

where $q_l$ is a positive constant prescribing the speed of reinforcement.

### 2.3.3 Recognition Layer

The recognition layer consists of a number of cell planes; each cell plane containing one cell. Each cell in this layer is connected to the layer preceding it in such way that it is looking at the whole pattern in the input layer. The output of each cell in the recognition layer is also computed using Equation 2.1.

## 2.4 Self-organization of the Network

As mentioned earlier, the Neocognitron has the ability of unsupervised learning. The network adjusts its learning parameters without any influence from the outside. In the learning mode, the network is repeatedly presented with stimulus patterns. Every time a pattern is presented to the network, the output of each S- cell in an S-plane within each S-layer is computed using Equation 2.1. In order to find a cell in a cell-plane that is most capable of extracting certain features of the input pattern, first subdivide all the S-planes into small regions. Each region contains a group of S-cells. The receptive fields of these cells are situated within a small area of the input pattern. Second, the maximum output cell from these regions is chosen by the network as a candidate for reinforcement. The network searches all the regions until all S-planes are checked. If more than one candidate appears from one cell plane, the cell that has the maximum output is chosen as a representative. Once the representatives are chosen from the S-planes, the cells are reinforced by updating their excitatory and inhibitory connection at the same rate using Equations (2.7) and (2.8), respectively. This process is repeated and the reinforcement of the excitatory and inhibitory connections progresses until the output of the cells in Us1 layer approaches a value of 1. Then, the output of the first layer is considered to be stabilized. The output of the Uc1 layer, which is directly driven by Us1 layer, is computed using Equation 2.4. The output of the Us2 layer is computed using Equation 2.1. The same process of choosing the representatives and reinforcing the variable connections in Us1 layer is repeated for the Us2 layer. In summary, the first layer is trained until the output is stable, then the next layer is trained until its output is also stable. The same process continues untill the last layer is reached.

### 2.5 Pattern Recognition by the Neocognitron

In order to help understand the mechanism by which the Neocognitron performs pattern recognition, an example of how the neocognitron recognizes patterns is shown in Figure 2.4. In this example, four layers are used, the input layer, Us1, Us2, and Us3. Generally, the first Us layer (Us1), which is directly connected to the input layer, extracts small features of the input pattern. These features are combined by cells in the second Us layer (Us2). A cell in the last layer (Us3) combines these features from the second layer and responds only to a particular pattern. It is assumed that the network has been trained with the numerals (0-4). After the learning process is finished, several cells in the first layer have extracted various features of simple orientation. This means that the

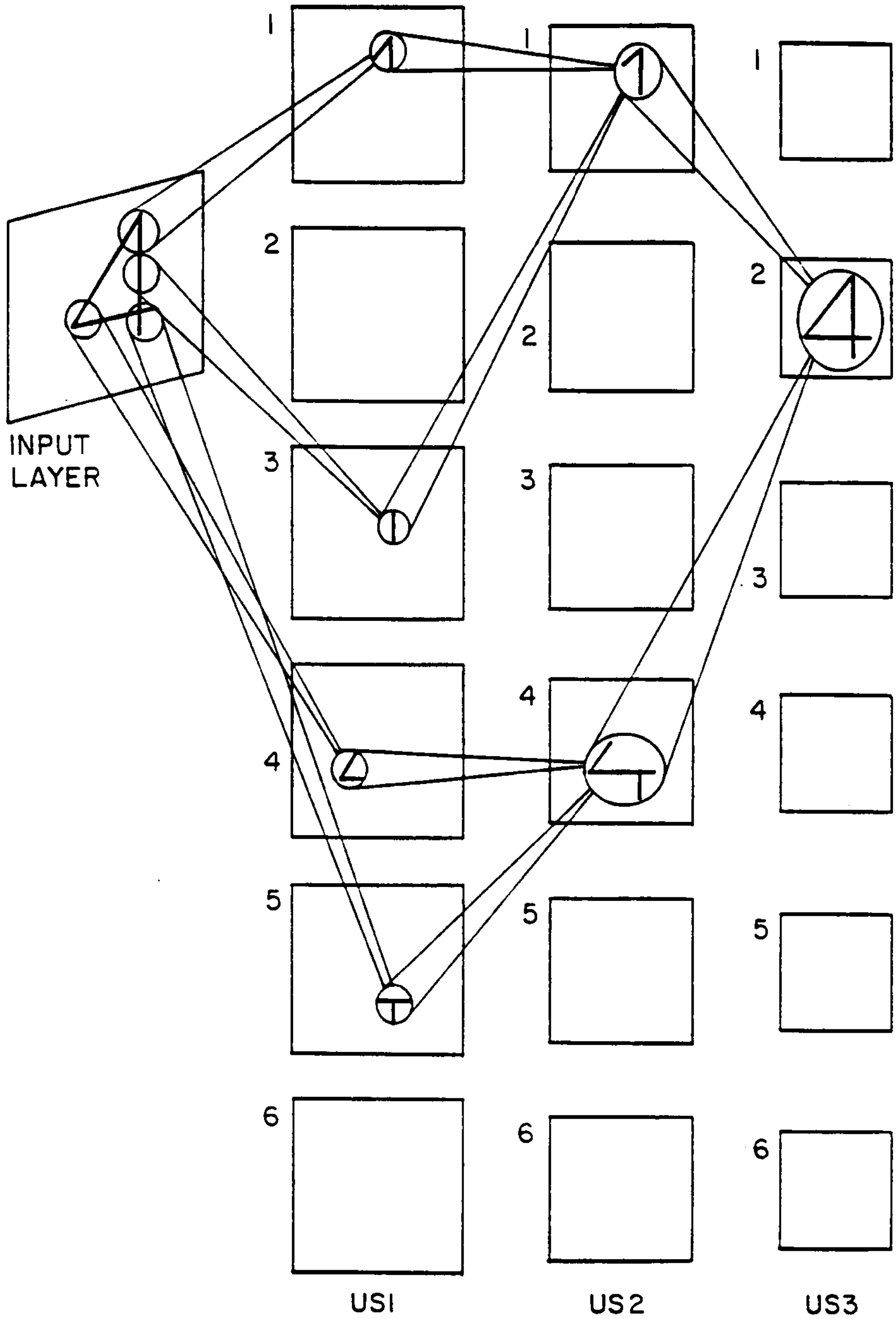



Figure 2.4: Pattern recognition by the Neocognitron. Each layer consists of six cell planes. Layer Us1 extracts small regions of the input pattern. Layer Us2 extracts a wider area of the input pattern. Layer Us3 extracts the whole pattern.

variable interconnections converging to these cells are reinforced using Equations (2.7) and (2.8) in such way as to extract features shown in cells 1, 3, 4, and 5. Similarly, the cells of Us2 have their variable interconnections enforced in such a way that when cells 1, 3, 4, and 5 are activated in Us1, cells 1 and 4 are activated. Cell 2 of the last layer (Us3) has its variable interconnections enforced in such away that when cells 1 and 4 are active in Us2, cell 2 is activated. When pattern "4" is presented at the input layer, cells 1, 3, 4, and 5 in Us1 fire because the relevant features are present in the input pattern, and cells 2 and 6 shut off because of the inhibitory connection $b_l(\nu)$ in Equation 2.2. At the second layer Us2, cells 1 and 4 fire because their variable interconnections are enforced in such a way as to fire every time these cells in layer Us1 fire. At the last layer, Us3, cell 2 fires because its variable interconnections are also enforced to fire every time cells number 1 and 4 respond from the second layer. In summary, the stimulus pattern is first observed within a narrow range by the cells in the first layer, Us1. In the next layer, these features are combined and extracted by certain cells. In essence, the second layer looks at a wider range of the input pattern. In the last layer, each cell looks at the whole input pattern.

## 2.6 Effect of the Parameter $r$

The parameter $r$ controls the degree of inhibition and the selectivity of the features to be extracted. Let $p(\nu)$ be the stimulus pattern; then $p(\nu) = U_0(n+\nu)$. With this notation and dropping all the subscripts in Equation 2.1, Equation

2.1 can be written:

$$U = r\varphi[\frac{1 + \sum_\nu a(\nu)p(\nu)}{1 + \frac{r}{1+r}bv} - 1]. \tag{2.9}$$

Let $s$ be defined by

$$s = \frac{\sum_\nu a(\nu)p(\nu)}{bv}. \tag{2.10}$$

Then, equation 2.9 reduces to

$$U = r\varphi[\frac{1+r}{r}s - 1], \tag{2.11}$$

provided that $a(\nu)$ and $b$ are sufficiently large. When an arbitrary pattern is presented at the input layer, if it satisfies $s > \frac{r}{r+1}$, we have $U > 0$ by Equation 2.11. Conversely, for a pattern which makes $s < \frac{r}{r+1}$, cell U does not respond.

Let

$n_0 =$ the number of pixels which are active in the training pattern but have zero value in the test pattern,

$n_1 =$ the number of pixels which are active in the test pattern but have zero value in the training pattern,

$N =$ the number of active pixels in the training pattern.

Using these notations, $s$ can be expressed by

$$s = \frac{1 - \frac{n_0}{N}}{\sqrt{1 + \frac{n_1 - n_0}{N}}}. \tag{2.12}$$

If the parameter $r$ is in the range $s > \frac{r}{1+r}$, i.e., in the range

$$\frac{1 - \frac{n_0}{N}}{\sqrt{1 + \frac{n_1 - n_0}{N}}} > \frac{r}{r+1}, \tag{2.13}$$

then, we need to choose a value for $r$ which satisfies this condition. Let the pattern shown in Figure 2.5(a) be the training pattern. After the S-cell has been trained with this pattern, we want to design a feature-extracting cell (an S-cell) which responds to the pattern shown in Figure 2.5(b) but not to the pattern shown in Figure 2.5(c). These patterns are labeled test pattern 1 and test pattern 2 in Figure 2.5. Since there are three active pixels in the training pattern, $N$ =3. Comparing test pattern 1 and test pattern 2 with the training pattern, we have for pattern 1, $(n_1, n_0) = (1, 0)$. Similarly, for pattern 2, $(n_1, n_0) = (1, 1)$. In order to find the acceptable range for the parameter $r$, we substitute the values for $n_1$ and $n_0$ in Equation 2.13. This yields a value for $r$ greater than 2.0 but less than 6.46. Any value for $r$ between those two values will make the S-cell respond to test pattern 1 but not to test pattern 2.

## 2.7 Numeral Recognition Example

As an illustration of how the network recognizes objects, the Neocognitron was trained with five numerals, 0-4. Similar experiments were conducted on the Neocognitron by Fukushima [6,7]. In this experiment, the network consists of three units. Each unit is subdivided into two layers. These layers are arranged in the following order: Usl-Ucl-Us2-Uc2-Us3-Uc3. Each layer consists of 24 cell

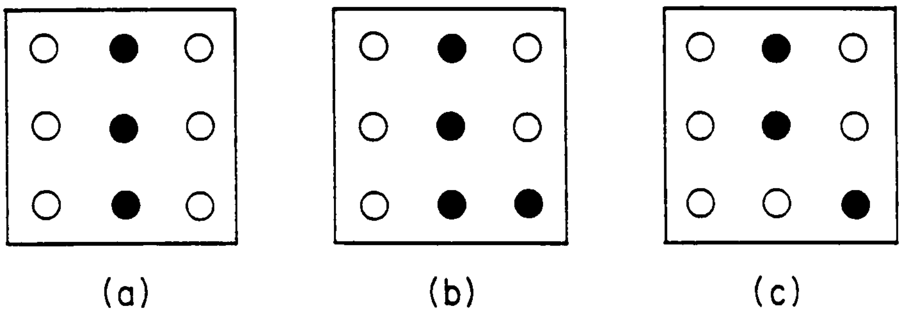



Figure 2.5: (a) Training pattern. (b) Test pattern 1. (c) Test pattern 2.

planes. Table 2.1 shows the size of the cell planes in each layer. Initially, the excitatory variable interconnections $a(\nu)$ described in Equation 2.1 in Section 2.3.2 are set at small positive values and they have no preferred orientations. The inhibitory values $b$ are set at zero. The patterns are presented repeatedly at the input layer. The S-cells which respond with the maximum output have their excitatory and inhibitory variable interconnections updated according to Equations (2.7) and (2.8), respectively. The input pattern is presented as a binary image (two-dimensional vector). The size of the input pattern (which is the same as the size of the input layer) is 16 x 16 pixels. Each pattern is presented 50 times to the input layer. After the learning process is complete, each pattern elicits a unique response at the output layer. Figure 2.6 shows the features extracted by the cells of the first Us layer. Figure 2.7 shows the firing response of the five numerals. Only the Uc layers are shown in the figure. In this figure, the top 24 squares correspond to the 24 cell planes in the second layer Uc1 of the first unit. The middle 24 squares correspond to the second layer Uc2 of the second unit. The bottom 24 squares correspond to the 24 cells of the recognition layer. When a pattern is presented at the input layer, the cells in layer Uc1 function as template matching. If the relevant features are present in the input pattern, specific cells fire and the other cells shut off. As shown in Figure 2.7, when pattern "2" is presented at the input layer, cells 12, 13, 15, 19, and 22 fire. The reason these cells fire is because the features extracted by cells 12, 13, 15, 19, and 22 in Figure 2.6 match the features presented in the "2" pattern. In layer Uc2, the cells fire if the relevant cells in the previous layer (Uc1)

Table 2.1: **Simulation parameters of the (0-4) numeral recognition example shown in Figure 2.7.**

| Layer | No. of exe. cells | No. of exc. input inter-connections per cell | Size of an S-column (No. of cells per S-column) | $r_l$ | $q_l$ |
|---|---|---|---|---|---|
| $U_0$ | 16 x 16 | | | | |
| $U_{s1}$ | 16 x 16 x 24 | 5 x 5 | 5 x 5 x 24 | 10.0 | 1.0 |
| $U_{c1}$ | 10 x 10 x 24 | 5 x 5 | | | |
| $U_{s2}$ | 8 x 8 x 24 | 5 x 5 x 24 | 5 x 5 x 24 | 3.0 | 16.0 |
| $U_{c2}$ | 6 x 6 x 24 | 5 x 5 | | | |
| $U_{s3}$ | 2 x 2 x 24 | 5 x 5 x 24 | 2 x 2 x 24 | 1.5 | 16.0 |
| $U_{c3}$ | 1 x 1 x 24 | 2 x 2 | | | |

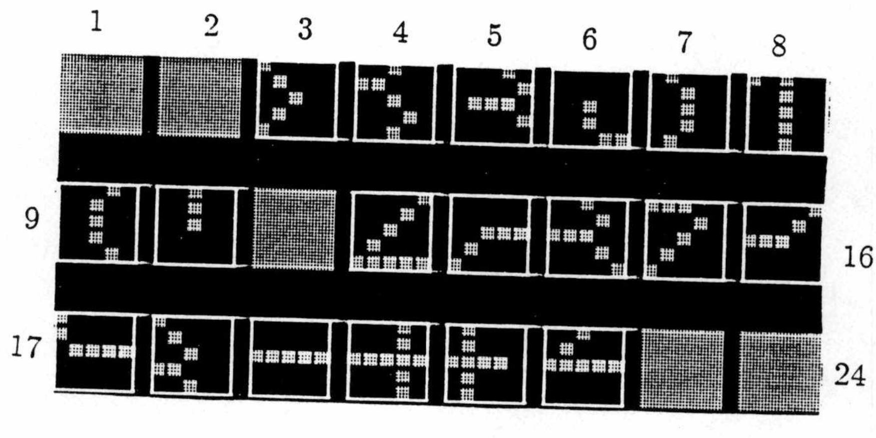



Figure 2.6: Features extracted by the first layer Us1.

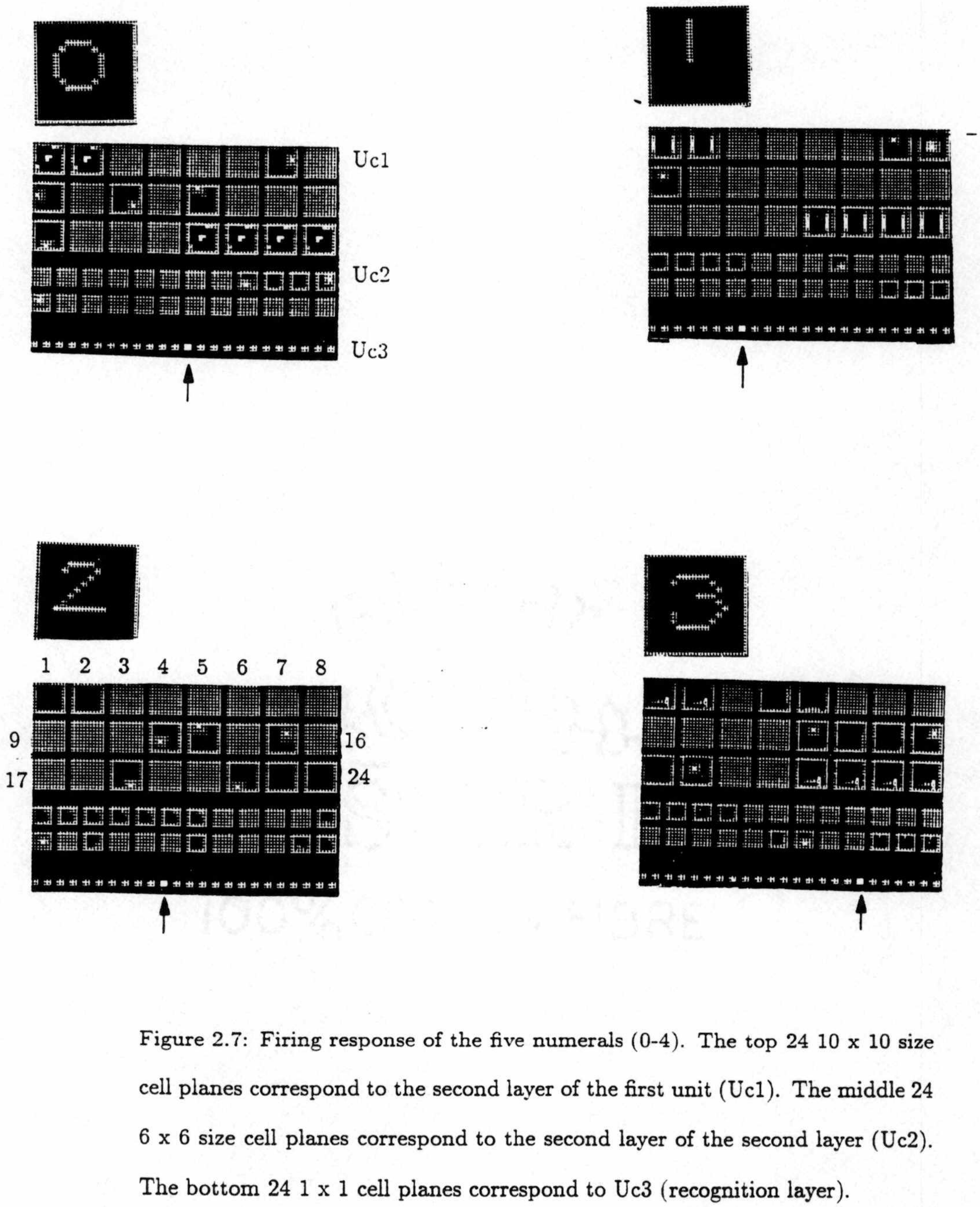



Figure 2.7: Firing response of the five numerals (0-4). The top 24 10 x 10 size cell planes correspond to the second layer of the first unit (Ucl). The middle 24 6 x 6 size cell planes correspond to the second layer of the second layer (Uc2). The bottom 24 1 x 1 cell planes correspond to Uc3 (recognition layer).

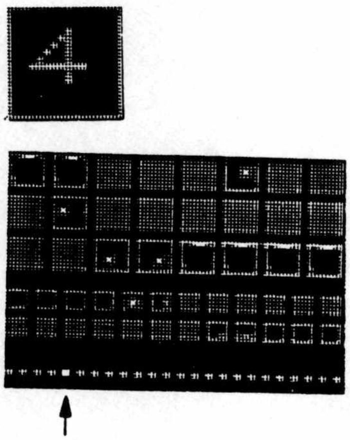


Figure 2.7: (Continued).

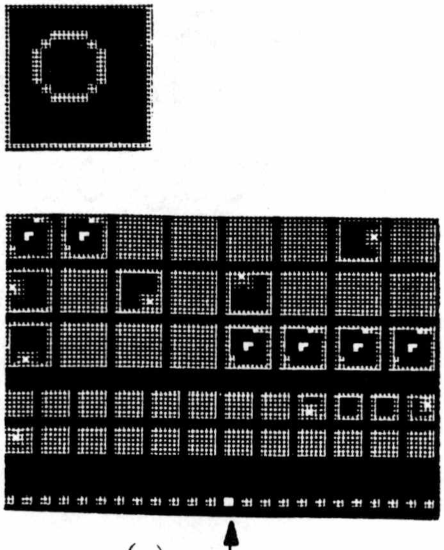
fire. In layer Uc3, only cell number 11 fired. The simulation parameters for this example are shown in Table 2.1. The parameter $r$ described by Equation 2.1 controls the intensity of inhibition. The larger the value of $r$, the more sensitive becomes the cell's response to its specific feature.

## 2.8 Sensitivity Analysis

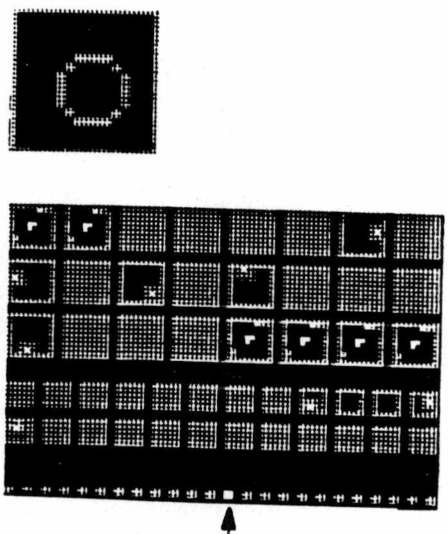
As mentioned earlier, the network response is invariant to shift, size, deformation, and noise. This invariance makes the network powerful in its performance and in its robustness. The following sections give a detailed study of the network sensitivity to shift, size, deformation, and noise.

### 2.8.1 Shift Invariance

The capability of the network to handle shift in the input patterns is because all the cells in the same cell plane have the same variable interconnections. This means that all the cells in a common cell plane extract the same features but at different positions. For instance, if the pattern is shifted by two or three pixels in the input layer, another cell in the same cell plane at that position will respond. Figure 2.8(a) shows the firing response of the "0" pattern that the network was trained with, and Figure 2.8(b) shows the firing response of the same pattern shifted in position. As shown in the last layer, the same cell responded for both patterns, which means both patterns were recognized as being the same.



(a)



(b)

Figure 2.8: Effect of shifting on the Neocognitron. (a) Training pattern; the cell indicated by an arrow in the last layer responded for the trained pattern. (b) Shifted pattern.

### 2.8.2 Size Invariance

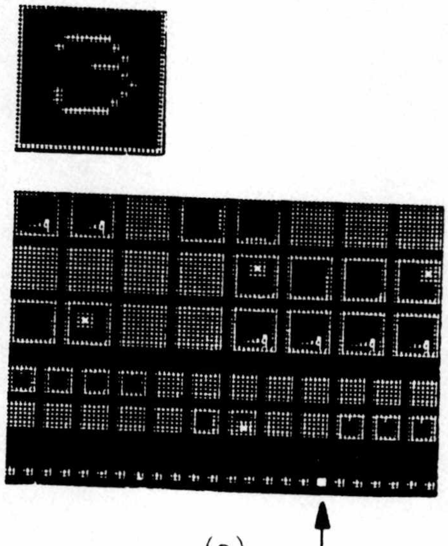
The network has the ability to recognize smaller or larger scales of an input pattern (as long as the geometrical shape is retained) within some tolerance. Figure 2.9(a) shows the size of the pattern and its firing response that the network was trained with, and Figure 2.9(b) shows a smaller size of the same pattern and its firing response. Both patterns were recognized as being the same, as shown in the last layer in Figure 2.9.

### 2.8.3 Deformation Invariance

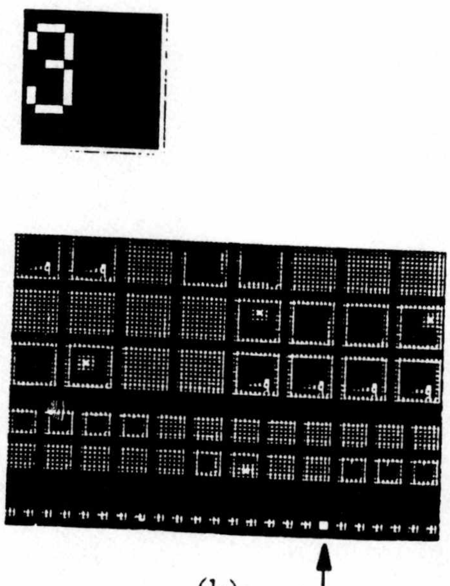
The parameter $r$ controls the similarity between the training pattern and the test pattern. This parameter allows the feature-extracting cells to judge how close both patterns should be in order to be recognized as being the same. In Figure 2.10(b), although some pixels are missing from the top part of the pattern, the network recognized it as being the same as the pattern shown in Figure 2.10(a).

### 2.8.4 Noise Invariance

The Neocognitron performance was tested for noisy patterns. Gaussian noise with zero mean and a standard deviation of 15 was added to the input pattern. As shown in Figure 2.11, the network recognized the noisy "2" pattern as being the same as the "2" pattern shown in Figure 2.11(b). Even though the pattern looks fuzzy as a result of noise, the pattern is recognized as being the same because local features are preserved in the noisy pattern.



(a)



(b)

Figure 2.9: Effect of size change on the Neocognitron. (a) Training pattern. (b) Smaller size pattern.

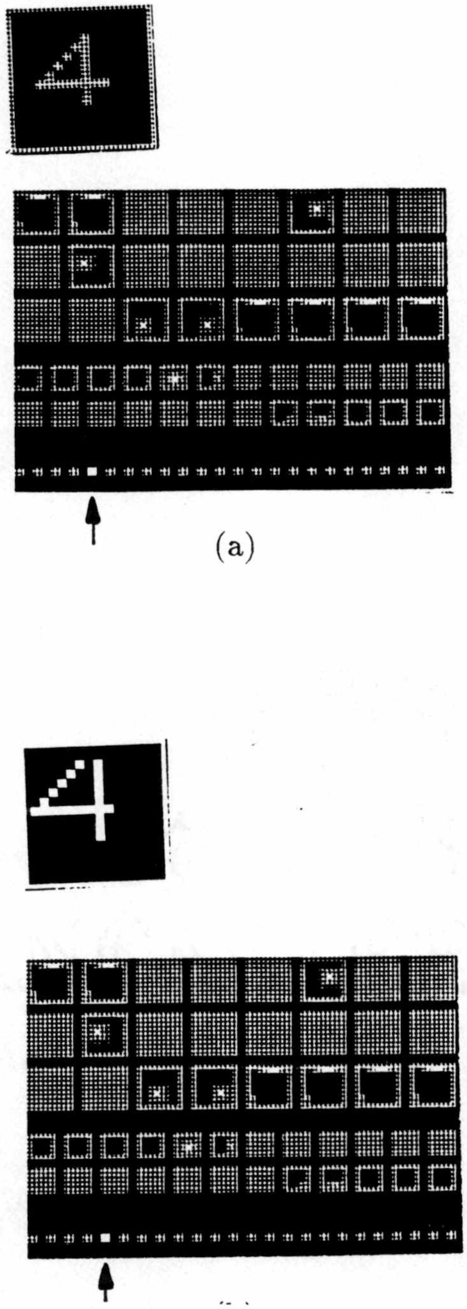
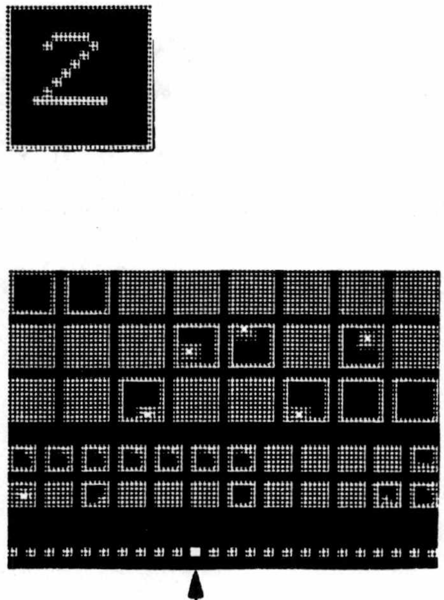
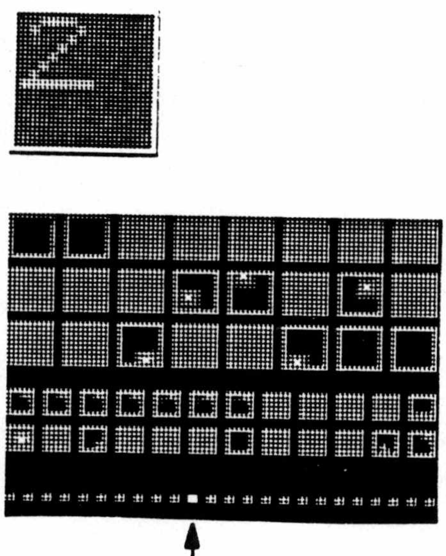



Figure 2.10: Effect of deformation on the Neocognitron. (a) Training pattern. (b) Deformed pattern.



(a)



(b)

Figure 2.11: Effect of noise on the Neocognitron. (a) Training pattern. (b) Noisy pattern.

## 2.9 Ten Numeral Recognition

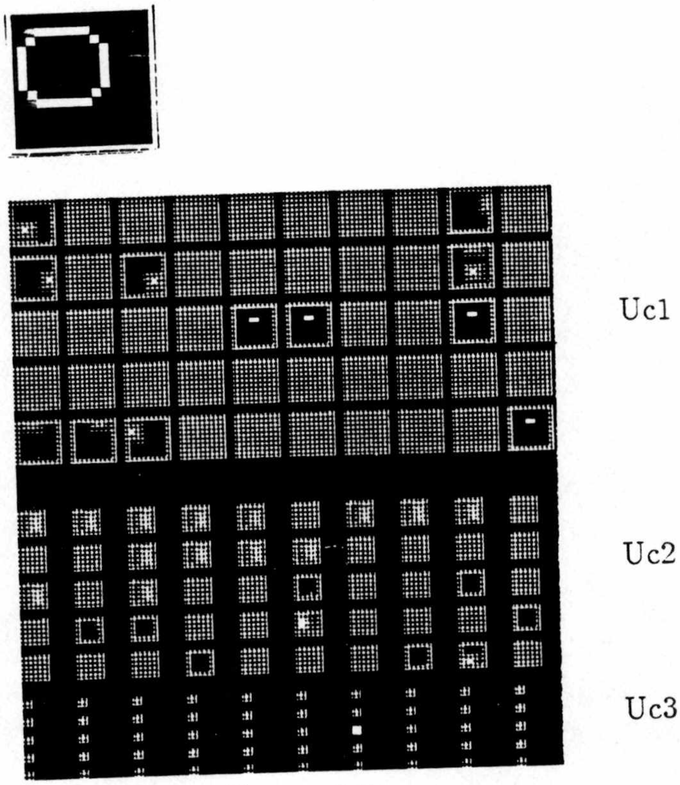
In this experiment, the number of training patterns was increased to ten numerals (0-9). Since the number of training patterns was increased, the size of the network was increased. In the previous example in Section 2.7, 24 cell planes were needed for feature extraction. This number was increased to 50 cell planes. The selectivity parameter $r$ was also increased from 10 to 20 for the first layer (Us1). Each pattern was presented 50 times to the network. Table 2.2 shows the simulation parameters for the expanded network. Figure 2.12 shows the firing response of the ten numerals (0-9). In the figure, the top 50 squares correspond to the feature extracting cell-planes of Uc1 layer. The middle 50 squares correspond to the 50 cell-planes of Uc2. The bottom 50 squares correspond to the 50 cells of the recognition layer (Uc3).

Increasing the number of training patterns to be trained by the Neocognitron was troublesome and time consuming. This is due to the following reasons: first, it is not an easy task to know how many more feature extracting cell planes needed to be added when the number of training patterns was increased. Not only that, but also increasing the number of cell planes requires the coefficients of the two- dimensional masks (c and d) described in Section 2.3.2 to be modified in order to satisfy the following two equations:

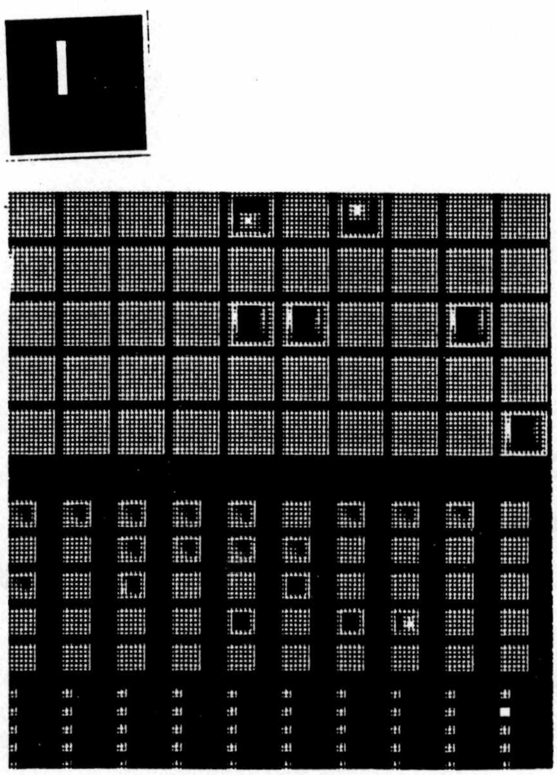
$$\sum_{k_{l-1}=1}^{K_{l-1}} \sum_{\nu \in S_l} c_{l-1}(\nu x, \nu y) = 1, \tag{2.14}$$

Table 2.2: Simulation parameters of the expanded network (0-9) shown in Figure 2.12.

| Layer | No. of exe. cells | No. of exc. input interconnections per cell | Size of an S-column (No. of cells per S-column) | $r_l$ | $q_l$ |
|---|---|---|---|---|---|
| $U_0$ | 16 x 16 | | | | |
| $U_{s1}$ | 16 x 16 x 50 | 5 x 5 | 5 x 5 x 50 | 20.0 | 1.0 |
| $U_{c1}$ | 10 x 10 x 50 | 5 x 5 | | | |
| $U_{s2}$ | 8 x 8 x 50 | 5 x 5 x 50 | 5 x 5 x 50 | 3.0 | 16.0 |
| $U_{c2}$ | 6 x 6 x 50 | 5 x 5 | | | |
| $U_{s3}$ | 2 x 2 x 50 | 5 x 5 x 50 | 2 x 2 x 50 | 1.5 | 16.0 |
| $U_{c3}$ | 1 x 1 x 50 | 2 x 2 | | | |

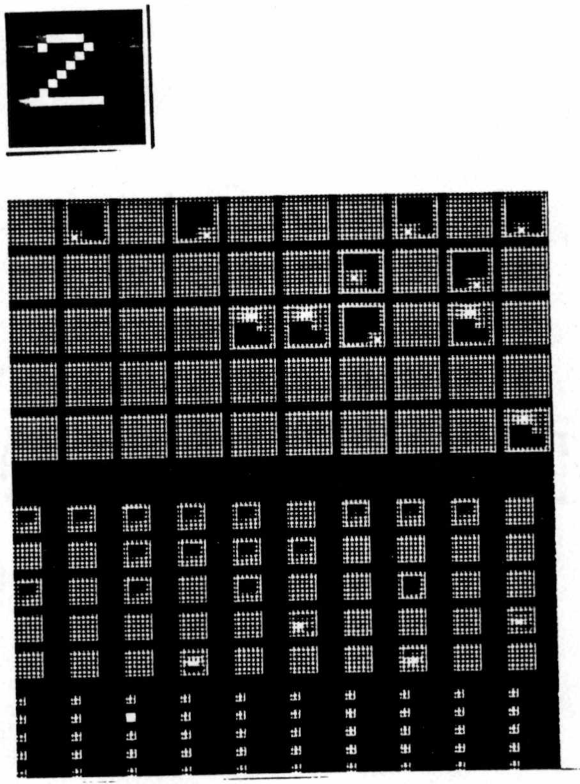



Numeral Zero

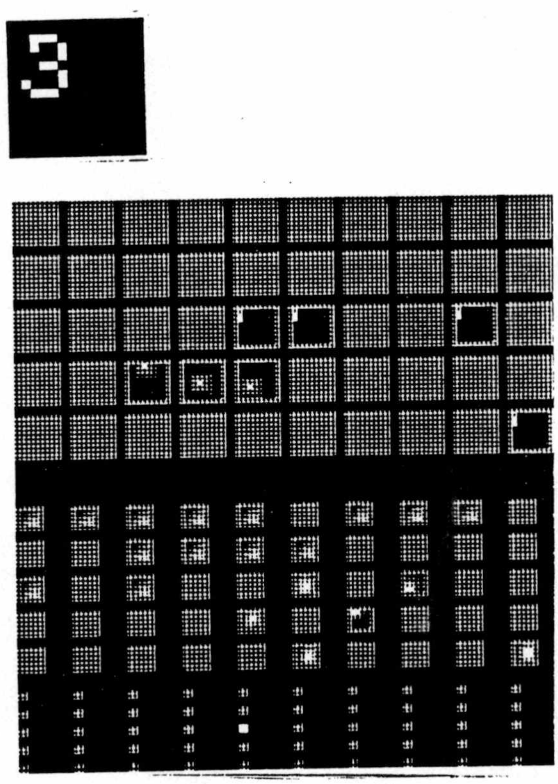


Numeral One

Figure 2.12: Firing response of the ten numerals (0-9).

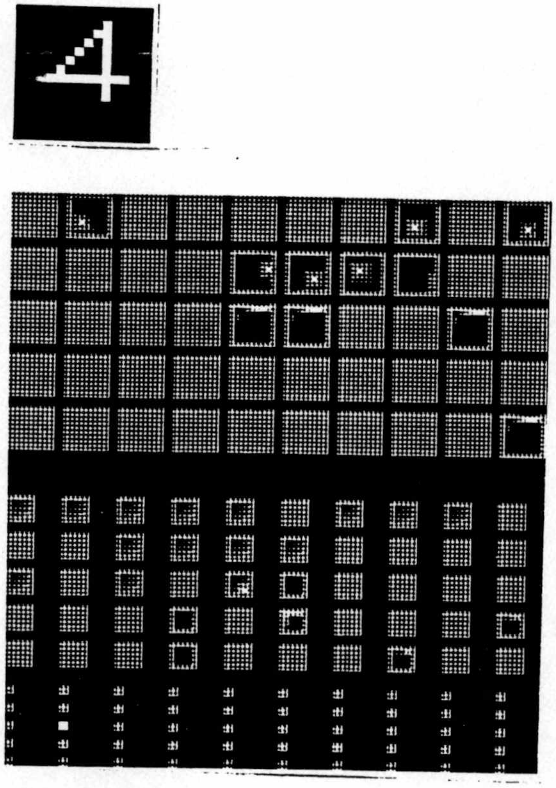


Numeral Two

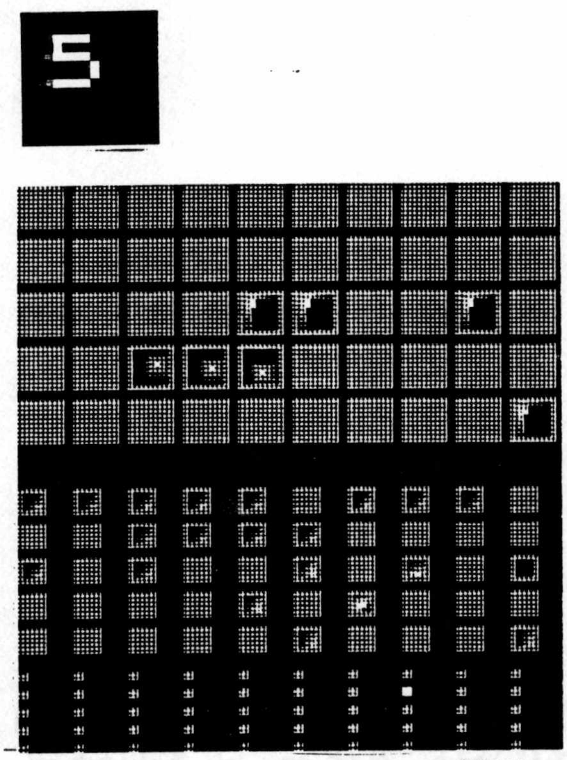


Numeral Three

Figure 2.12: (Continued).

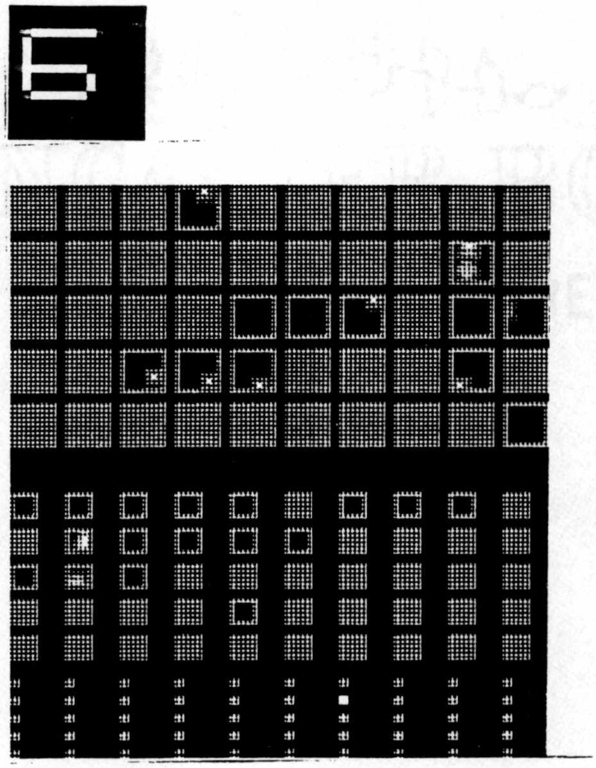


Numeral Four

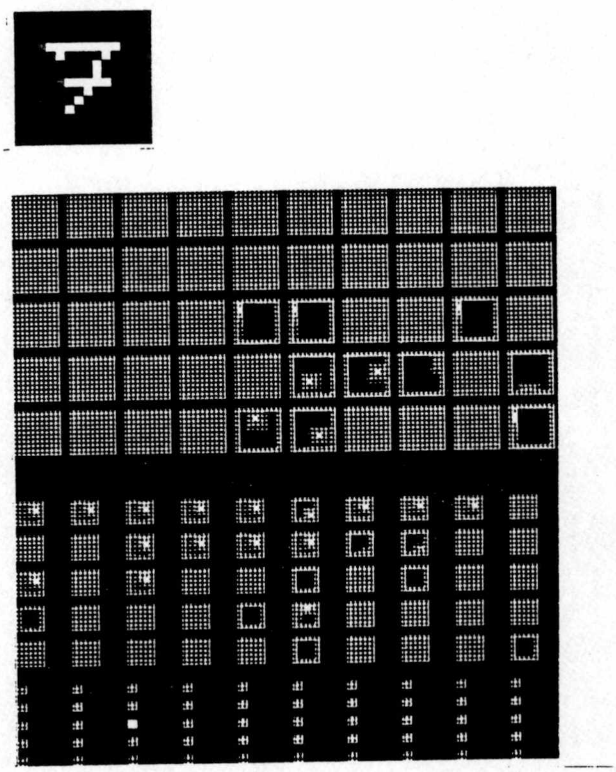


Numeral Five

Figure 2.12: (Continued).

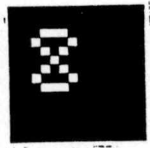


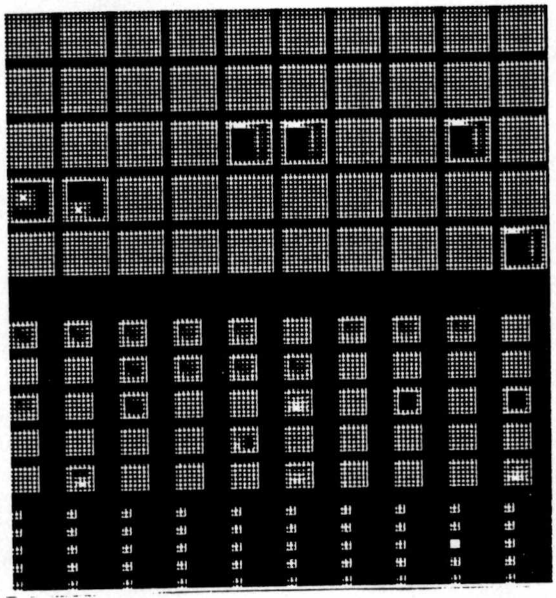
Numeral Six



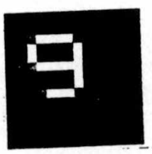
Numeral Seven

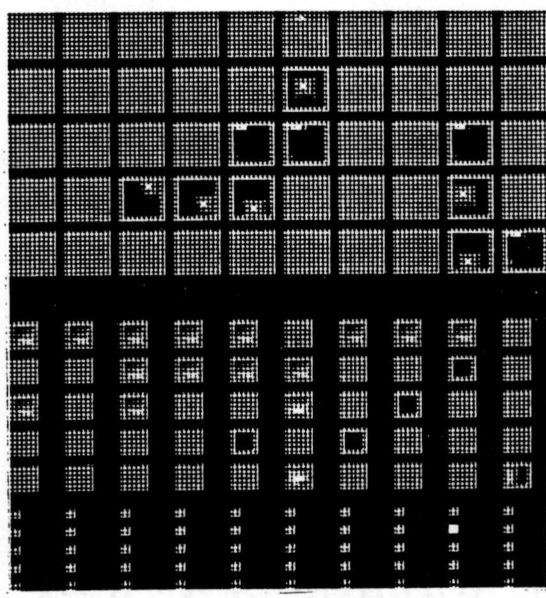
Figure 2.12: (Continued).





Numeral Eight





Numeral Nine

Figure 2.12: (Continued).

and

$$\sum_{k_{l-1}=1}^{K_{l-1}} \sum_{\nu \in D_l} d_{l-1}(\nu x, \nu y) = 1, \tag{2.15}$$

where $K_{l-1}$ is the number of cell planes in an S-layer, $S_l$, and $D_l$ are the connecting areas. In the previous example (numeral recognition of 0-4), the number of cell planes used was 24, so the coefficient values of these masks were chosen to satisfy the following summation:

$$\sum_{k_{l-1}=1}^{24} \sum_{\nu \in S_l} c_{l-1}(\nu x, \nu y) = 1, \tag{2.16}$$

and

$$\sum_{k_{l-1}=1}^{24} \sum_{\nu \in D_l} d_{l-1}(\nu x, \nu y) = 1, \tag{2.17}$$

where $D_l$ and $S_l$ are connecting areas of a 5 x 5 pixel. In this example, the number of cell planes is 50 with the same size connectable area; therefore, the coefficient values were modified to satisfy the following summation:

$$\sum_{k_{l-1}=1}^{50} \sum_{\nu \in S_l} c_{l-1}(\nu x, \nu y) = 1, \tag{2.18}$$

and

$$\sum_{k_{l-1}=1}^{50} \sum_{\nu \in D_l} d_{l-1}(\nu x, \nu y) = 1. \tag{2.19}$$

Second, the selectivity parameter $r$ in the expanded network was increased, which caused the network to be more vulnerable to size, deformation, and noise of the input pattern.

## 2.10 Multiple Digit Recognition

In the previous examples, only one pattern was presented to the receptive field of the network (i.e., at the input layer). In this experiment, after the network was trained with the five numerals (0-4), two patterns were presented simultaneously to the input layer. For example, the "0" digit and the "2" digit were presented at the input layer. The cells that were trained for the "0" digit and the "2" digit individually responded at the first layer but the intensity of the firing was stronger for one pattern than for the other, which made the network recognize only the pattern with the stronger firing intensity. The difference in the firing intensity is because one pattern is closer to the receptive field of the network than the other. By modifying the firing mechanism of the original Neocognitron, the network acquired the ability to recognize two input patterns presented to the input layer simultaneously. The modification of the firing mechanism is based on the fact that the cells which responded for each pattern individually in the first layer (Us1) are known from the learning process prior to the presentation of both patterns simultaneously to the input layer.

When both patterns are presented to the network, the cells for the pattern that was recognized were shut off. Then, the network was tested again for the other pattern. In other words, when two patterns are presented to the input layer, the network first recognizes the pattern that has the stronger firing intensity. By shutting off the cells which respond to the recognized pattern, the network then recognizes the other pattern. Figure 2.13(a) and (b) shows the cells that fired for the digit "0" and "2" . As shown in the figure, the cells that fired for the "0" digit in the first layer (top 24 squares) are 5, 6, and 11. Similarly, the cells that fired for the "2" digit are 1, 8, 9, 10, and 13. Figure 2.13(c) shows the firing response of both patterns presented at the input layer. As shown in the figure, the cells that were trained for both pattern individually responded at the first layer. Since the cells for the "0" pattern fired stronger than the cells for the "2" pattern in the first layer, the network recognized the "0" pattern first. As shown in the last layer (recognition layer) in both Figure 2.13(a) and (c), cell number 5 fired. In order for the network to recognize the "2" pattern, cells number 5, 6, and 11 were shut off in the first layer. By shutting off these cells, the network recognized the "2" pattern, as shown in Figure 2.13(d).

The recognition of multiple inputs presented simultaneously at the input layer was done by Fukushima [8]. His approach to the problem is different from the one described in this section. He used the Neocognitron model with a feedback connecting higher layers with lower layers. The purpose of the feedback connections is for selective attention to the patterns presented at the input layer. The forward connections carry information about the patterns from lower-

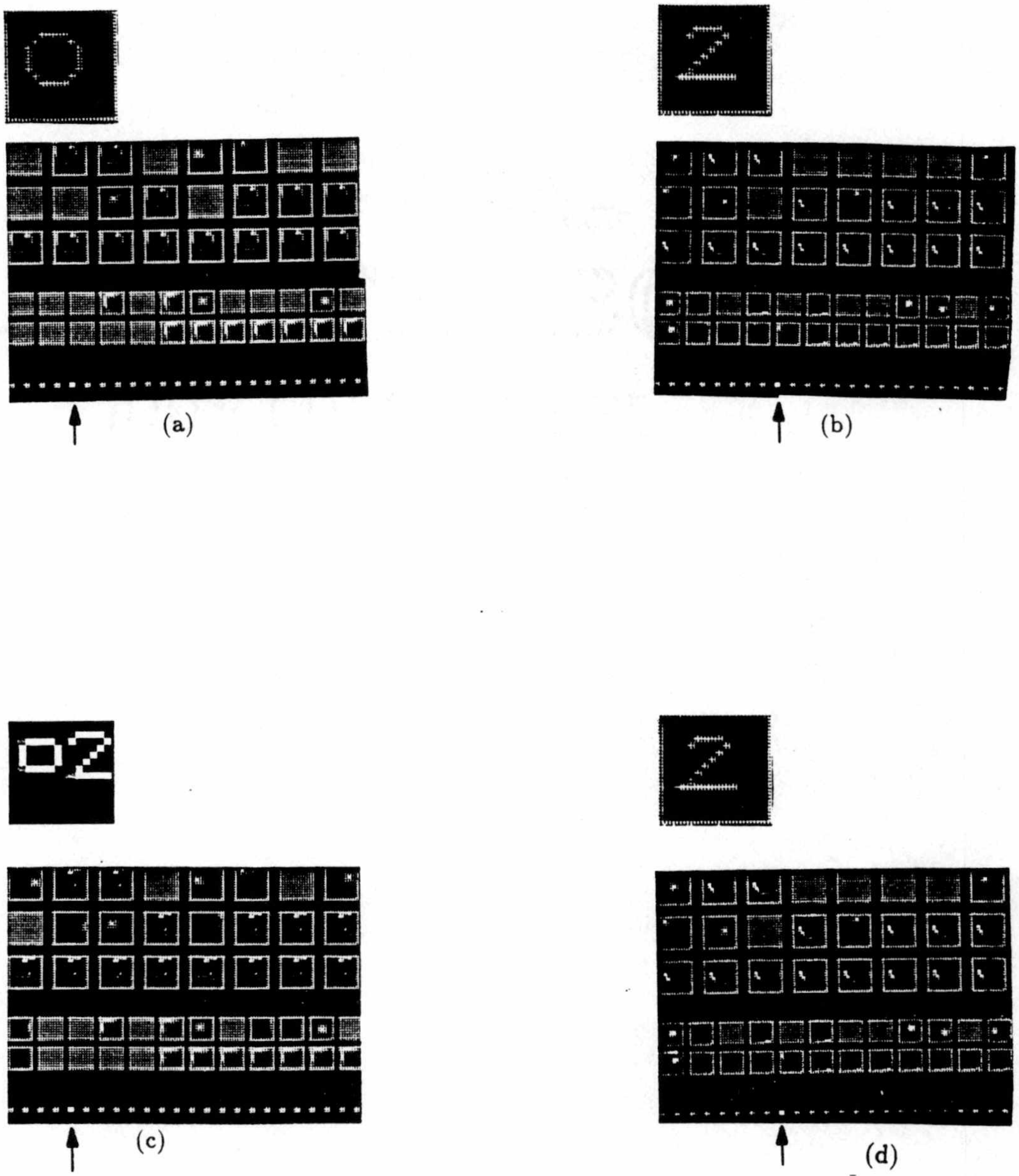


Figure 2.13: Multiple input recognition. (a) Firing response of "0" digit, (b) Firing response of "2" digit, (c) Firing response of both patterns presented at the input layer simultaneously. (d) Firing response of the "2" digit after the cells for "0" digit were shut off.

order cell layers to higher-order cell layers; while the backward connections carry information from higher-order cell layers to lower-order cell layers.

# CHAPTER 3

# MODULAR DESIGN OF THE NEOCOGNITRON

One of the main difficulties encountered with the Neocognitron is recognizing a large number of patterns. This difficulty can be explained by considering the numeral recognition example (0-4) and (0-9). When the network was first trained with five numerals and then later with 10 numerals, some questions such as how selective the parameter $r$ should be and how many feature extracting cell planes should be added to the network to handle the 10 patterns arose. Furthermore, certain masks like the (c-mask and d-mask) described in Equations (2.10) and (2.11) need to be modified every time new cell planes are added to the network. In the (0-4) numeral recognition example, the network had 24 feature extracting cell planes, but in the (0-9) numeral recognition example, 26 more feature extracting cell planes were added.

In order to answer the above questions, we need to experiment with the network during the learning process. This is time consuming and sometimes frustrating because it can take hours to train the network only to find out later that not enough cell planes were added or the proper modifications for the parameters were not carefully tuned. This chapter presents an alternative method for classifying a large number of classes by the Neocognitron.

### 3.1 Modular Structure

Figure 3.1 shows the structure of the proposed network. A number of Neocognitron-type networks are connected in parallel, and they all are connected to one input. Each network is trained separately with a number of training patterns. After each network has finished training, the modular design of the network can be tested by presenting the stimulus pattern at the input layer and by testing all Neocognitron-type networks simultaneously. In the modular design of the Neocognitron, modifification of the parameter $r$ or the c-mask and d-mask every time we increase the number of training patterns is unnecessary. If we have more training patterns, another Neocognitron-type network is added. However, it is necessary to take into consideration the possibility of misclassification and the availability of untrained cells at the output. This is due to the fact that there is a limit to the number of training patterns in the modular design of the Neocognitron. This limit is controlled by the individual Neocognitron structure. The number of training patterns in the modular design of the Neocognitron cannot exceed the number of cell planes of the last layer in the Neocognitron. For example, if the number of cells (since there is only one cell in each cell plane of the last layer) in the last layer is 24, the modular design of the Neocognitron cannot be trained with more than 24 patterns. This is because at the last layer (the recognition layer), only one cell is associated with each pattern. So, if there are more than 24 patterns, it is necessar to increase the number of cells at the output layer or confusion will result at the recognition

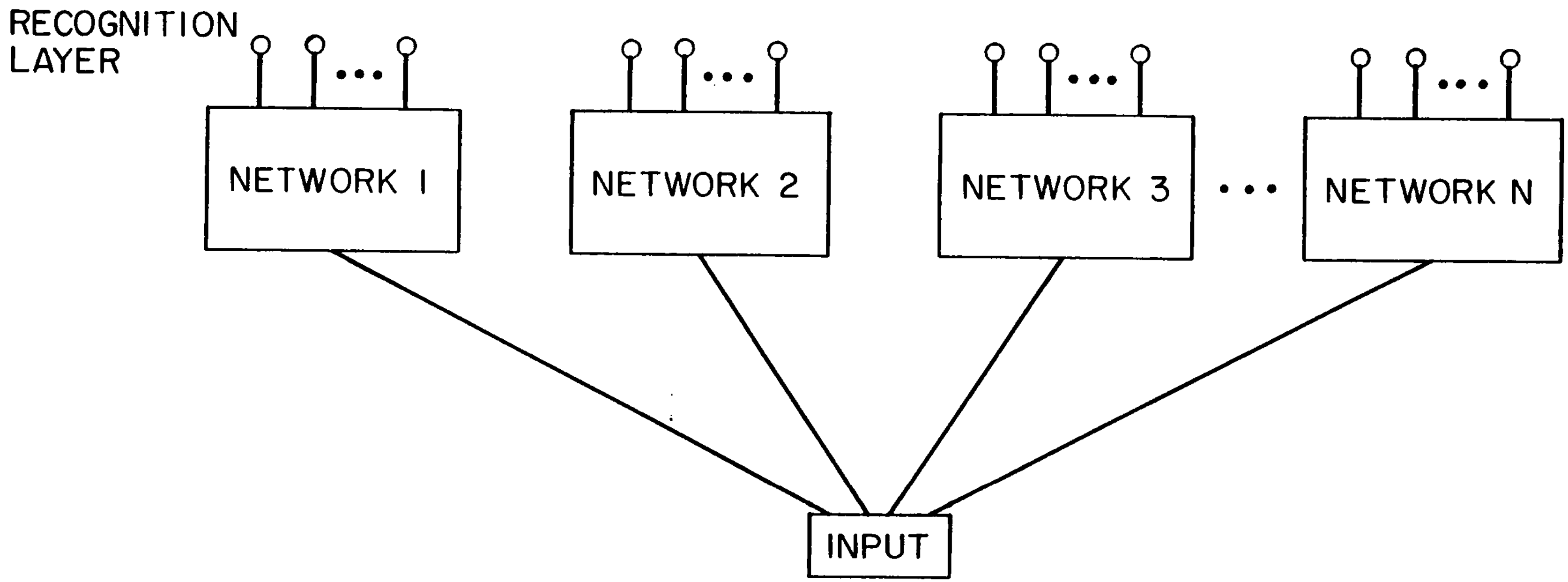



Figure 3.1: Structure of the modular design of the Neocognitron. A number of Neocognitron-type networks are connected in parallel.

layer.

### 3.1.1 One-Pattern Classifier Network

Since the Neocognitron has the ability of self-organization, the features for the training patterns are extracted by the internal learning mechanism of the network, and we have no control over what the network learns. If two patterns have geometrical similarity in their shapes or if one pattern is a subset of another pattern, a possibility of misclassification by the Neocognitron could occur. The one-pattern classifier subdivides a number of training patterns into small groups to be trained by the modular design of the Neocognitron and groups the patterns that have common features in one class. Figure 3.2 shows a schematic diagram of the one-pattern classifier.

### 3.1.2 An Example of Misclassification

This example demonstrates the possibility of misclassification by the Neocognitron. If the network was trained with the digit "1" and then tested for the letter "T", it might recognize the "T" pattern as "1". This is because the "1" pattern contains only vertical features, so the network will extract those vertical features during the learning process. Therefore, when the letter "T" is presented to the network, the cells that extract the vertical features will respond, which will cause the network to recognize the "T" pattern as "1" . Figure 3.3 shows an example of misclassification. In Figure 3.3(a), the network was trained with the pattern "1". The firing map shows that cells 8 and 10 from the first layer respond for

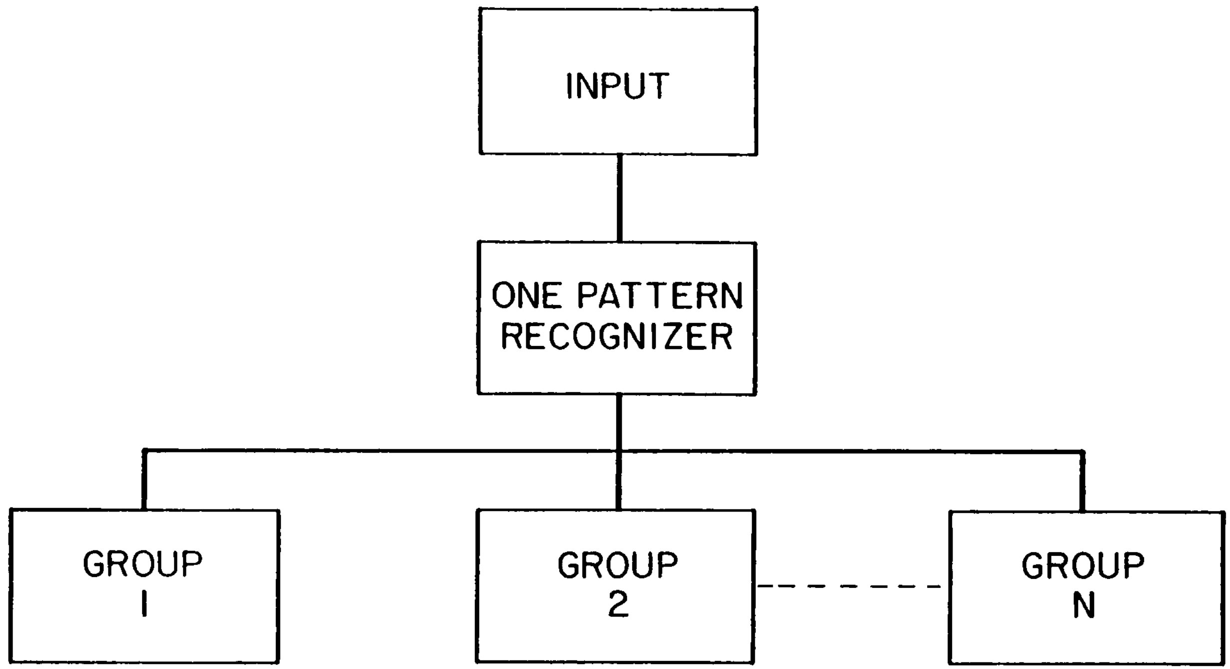



Figure 3.2: Schematic diagram of the one-pattern classifier. The one-pattern classifier subdivides the training patterns into small groups to be trained by the modular design of the Neocognitron.

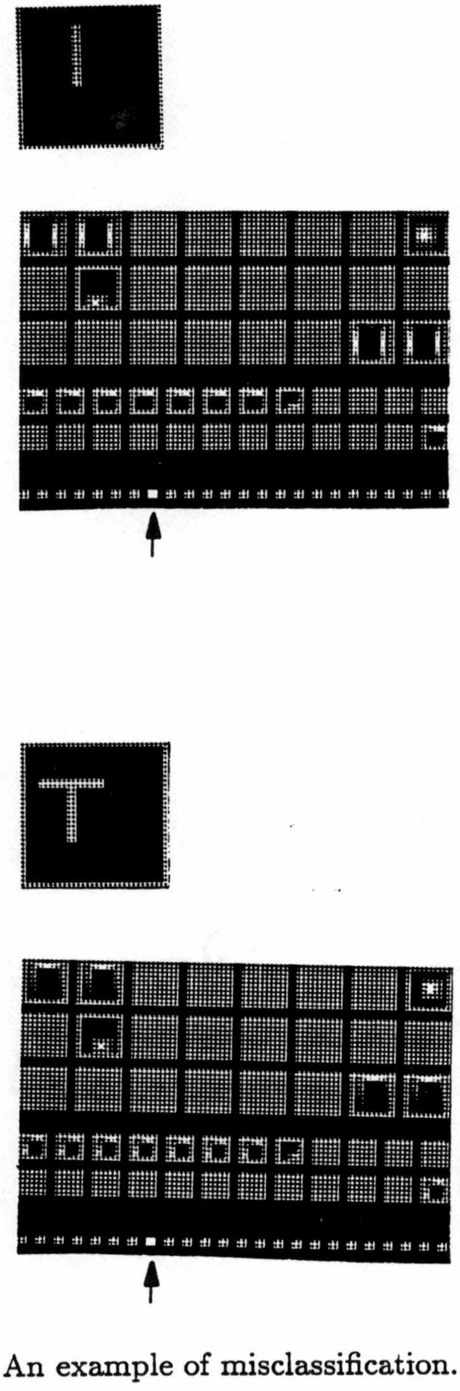


Figure 3.3: An example of misclassification. (a) Firing response of the training "1" pattern, (b) Firing response of the "T" pattern.

the "1" pattern. When the "T" pattern is presented at the input layer in Figure 3.3(b), cell numbers 8 and 10 also fire because there are vertical features in the "T" patterns. This causes the network to recognize the "T" pattern as the "1" pattern. This is shown in the last layer (Uc3) of Figure 3.3(a) and (b) where cell 8 fired for both patterns.

### 3.1.3 Algorithm

This section presents an algorithm for subdividing a large number of patterns into small groups. For K patterns, the following steps group the patterns that have geometrical similarities:

Step 1. Train a small network with the first pattern.

Step 2. Test the trained network with the K-1 patterns.

Step 3. Identify all patterns recognized the same as the first pattern.

Step 4. Group them in one set.

Step 5. Repeat steps 1-4 with the remaining patterns.

## 3.2 An Example of Classification

In this example, the modular design of the Neocognitron was trained with 24 alphanumeric patterns. Figure 3.4 shows the 24 patterns subdivided into five different groups according to the classification of the one-pattern classifier. As mentioned earlier, the one-pattern classifier is utilized to avoid the possibility of misclassification. It groups the patterns that have common features or the patterns that are subsets of others. For example, pattern "C" and pattern "G" are grouped in one set because the "C" pattern is a geometrical subset of the "G" pattern. If they are grouped in two different sets, there will be a possibility of misclassification. During the learning process, not only do they have to be in the same set, but also the "G" pattern must be trained before the "C" pattern. If the "C" pattern is trained first, the network will still recognize the "G" pattern as the "C" pattern, according to the learning mechanism of the network. However, the opposite is not true.

Another important point to be considered during the learning process is that once the first group is trained by the network, the cells of the last layer (recognition layer) that responded for that particular group must be shut off when training the next group. The reason for this is that each cell at the output must be associated with one pattern.

Once the proper grouping is done, each group is trained by one Neocognitron-type network. Since we have five groups, five Neocognitron-type structures are required. Each group is trained separately by the Neocognitron, and its learned parameters are saved in separate data files. After the learning process is complete, one cell at the output responds for each pattern. In the testing mode, when a pattern say from group 1 is to be tested by the modular design of the Neocognitron, it is presented at the input layer and tested by all five networks simultaneously. Figure 3.5 shows pattern "C" from group 1 tested by the networks. Since pattern "C" is from group 1, network 1 responds at the last layer. As shown in the figure, the other networks do not respond for the

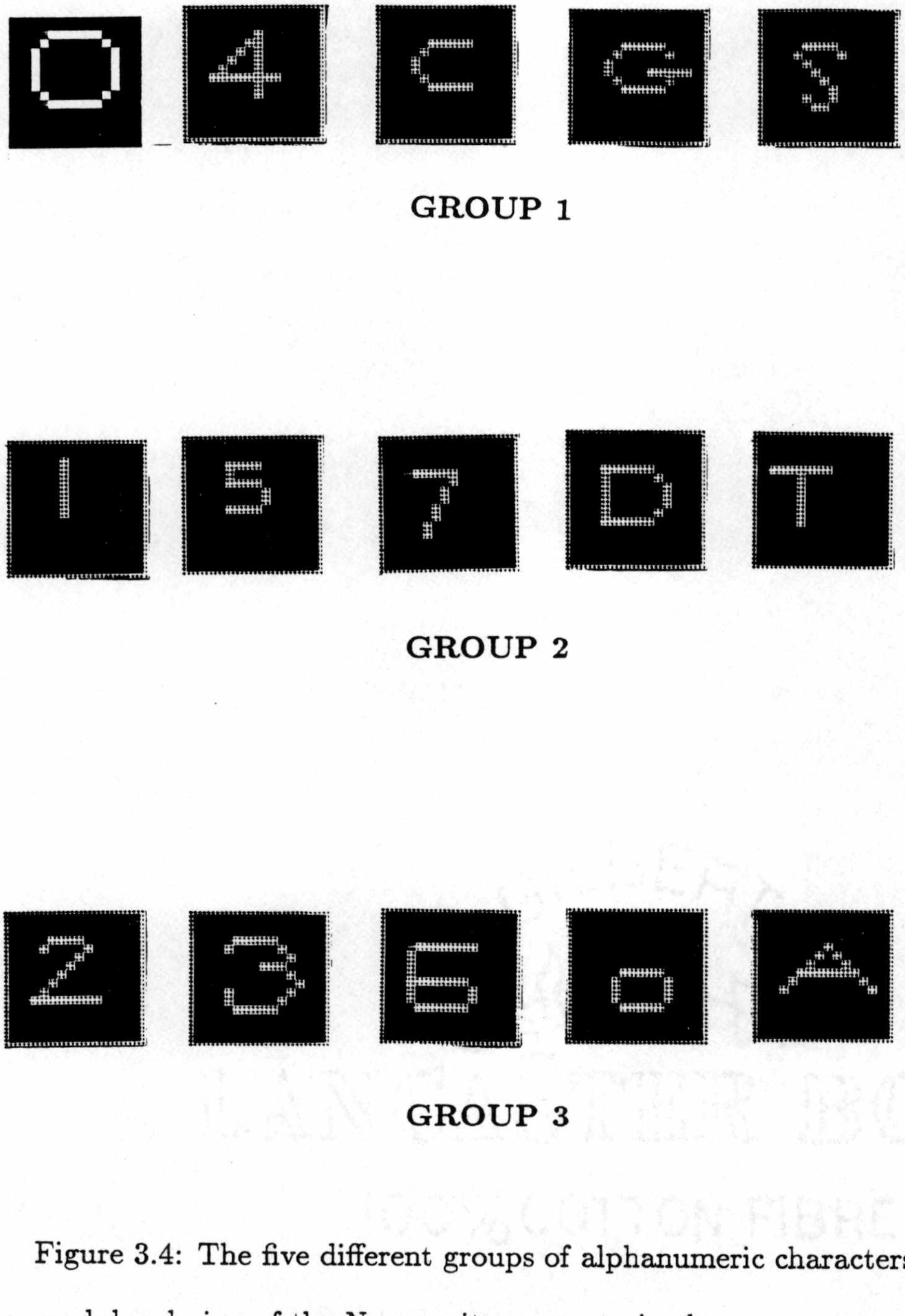
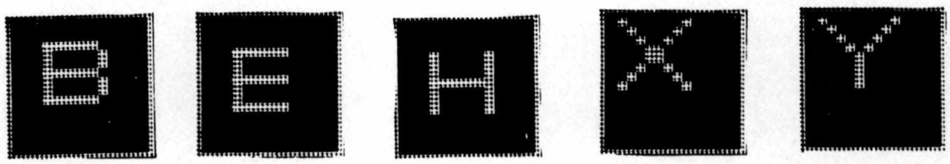
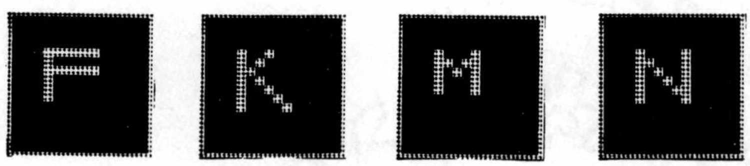



Figure 3.4: The five different groups of alphanumeric characters with which the modular design of the Neocognitron was trained.




**GROUP 4**




**GROUP 5**

Figure 3.4: (Continued).

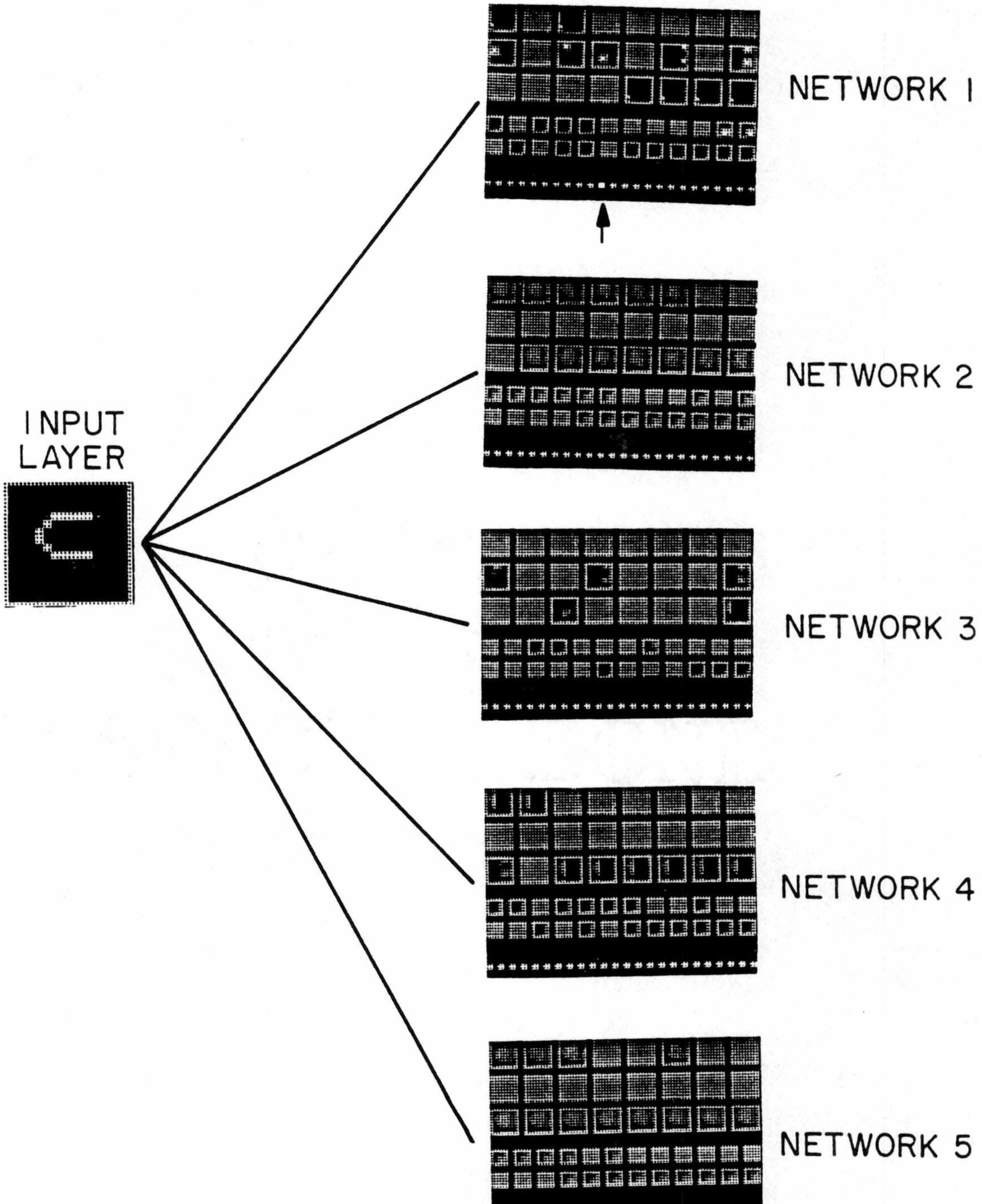



Figure 3.5: An example of classification; firing response of the modular design to the "C" pattern from group 1.

"C" pattern. Although some cells fire at the first layer of network 3, the firing intensity of these cells is not strong enough to activate the next layers.

# CHAPTER 4

# CONCLUSIONS

This study has presented a visual pattern recognition neural network called Neocognitron. This neural network can recognize patterns on the basis of the geometrical similarity of their shapes. Chapter 2, presented the structure and the learning algorithm of the Neocognitron. It also presented a numeral recognition example by the Neocognitron. The network is shift and size invariant because all cells in the same cell plane have the same spatial distribution and they extract the same features. The network is also capable of recognizing deformed and noisy patterns because of the selectivity parameter $r$. The selectivity parameter $r$ judges how close the trained pattern and the test pattern should be to be recognized as the same. How selective should the parameter $r$ be? This depends on the designer and the type of application. Increasing the number of layers in the network yields better performance in terms of shift invariance, but on the other hand, it slows the learning and testing process due to the extensive computation by the network. The number of layers in the network depends on the size of the input image. Increasing the number of training patterns would require further increase of the cell planes in the network. The position of the patterns in the input layer has little effect on the type of features extracted by the network during the learning process. The basic structure of the Neocognitron was modified for the recognition of multiple inputs. This study has proposed an alternative approach to increasing the number of training patterns. Chapter 3, presented the modular design of the Neocognitron and the algorithm for subdividing the number of patterns into small groups. This method avoids the addition of cell planes and the modification of the masks when we increase the number of training patterns. One important point which needs to be mentioned about the modular design of the Neocognitron is that the number of training patterns cannot exceed the total number of cell planes in the last layer.

This method has been tested with alphanumeric characters. The performance of the modular design of the Neocognitron was highly satisfactory because it can classify a large number of patterns and it is highly parallel and capable of quick recognition.

# BIBLIOGRAPHY

# BIBLIOGRAPHY


[1] Barnard, E., and Casasent, D. "Shift Invariance and the Neocognitron," *Neural Networks*, Vol. 3, pp. 403- 410, 1990.

[2] Casasent, D., and Psaltis, D. "Position, Rotation and Scale Invariant Optical Correlation," *Applied Optics*, 15, 1795-1799, 1976.

[3] Carpenter, G. A. *Pattern Recognition by Self-organizing Neural Networks*, MIT Press, 1991.

[4] Dayhoff, J. E. *Neural Network Architectures An Introduction*, Van Nostrand Reihold, 1990.

[5] Fukushima, K. "Cognitron: A Self-organizing Multilayered Neural Network,"*Biol. Cyber.*, 20(3/4), pp. 121-136, 1975.

[6] Fukushima, K. "Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position," *Biol. Cyber.*, 36(4), pp. 193-202, 1980.

[7] Fukushima, K., and Miyake, S. "Neocognitron: A New Algorithm for Pattern Recognition Tolerant of Deformations and Shifts in Position," *Biol. Cyber.*, 15(6), pp. 455-469, 1982.

[8] Fukushima, K. "A Neural Network Model for Selective Attention in Visual Pattern Recognition," *Biol. Cyber.*, 55(1), pp. 5-15, 1986.

[9] Fukushima, K., Miyake, S., & Ito, T. "Neocognitron: A Neural Network Model fo a Mechanism of Visual Pattern Recognition," *IEEE Transactions on Sys. Man Cyber.*, SMC-13(5), pp. 826-834, 1983.

[10] Fukushima, K. "Neocognitron: A Hierarchical Neural Network Capable of Visual Pattern Recognition," *Neural Networks,* Vol. 1, pp. 119-130, 1988.

[11] Fukushima, K. "Visual Feature Extraction by a Multilayerd Network of Analog Threshold Elements," *IEEE Trans. Sys. Sci. and Cyber.*, SSC-5, pp. 322- 333, 1969.

[12] Goos, G., and Hartmanis, J. *Neural Networks*, Springer-Verlag, 1990.

[13] Grossberg, S., and Kuperstein, M. *Neural Dynamics of Adaptive Sensory-Motor Control*, Pergamon Press, 1989.

[14] Grossberg, S. *Studies of Mind and Brain: Neural Principles of Learning, Perception, Development, Cognition, and Motor Control,* Reidel Press, Boston, 1982.

[15] Grossberg, S. "Embedding Fields: Underlying Philosophy, Mathematics, and Applications to Psychology, Physiology and Anatomy," *J. Cyber.*, 1(1), 28-50, 1971.

[16] Grossberg, S. *Neural Networks and Natural Intelligence*, The MIT Press, 1988.

[17] Hecht-Nilsen, R. *Neurocomputing,* Addison-Wesley, 1990.

[18] Hubel, D. H., and Wiesel, T. N. "Receptive Fields, Binocular Interaction and Functional Architecture in the Cat's Visual Cortex," *Journal de Physiologie*, 160(1), 106-154, 1962.

[19] Hu, M. "Visual Pattern Recognition by Moment Invariants," *IEEE Transactions on Information Theory*, IT-8, 179-187, 1962.

[20] Judd, J. S. *Neural Network Design and the Complexity of Learning*, The MIT Press, 1990.

[21] Kohonen, T. "A Simple Paradigm for the Self-organized Formation of Structured Feature Maps, "*Competition and Cooperation in Neural Nets*, (Lecture Notes in Biomathematics), 45, Springer-Verlag, 1982.

[22] Kohonen, T. " Self-organized Formation of Topologically Correct Feature Maps,"*Biol. Cyber.*, 43, 59-69, 1982.

[23] Mammone, R. J., and Zeevi, Y. *Neural Networks Theory and Applications*, Academic Press, Inc., 1991.

[24] Muller, B. *Neural Networks: An Introduction,* Springer-Verlag, 1990.

[25] Zeidenberg, M. *Neural Networks in Artificial Intelligence*, Ellis Horwood, Limited, 1990.

# VITA

Basem F. Awad was born in Nablus, Palestine on January 10, 1965. He graduated from High School in Amman, Jordan in 1983. He entered the University of Tennessee at Knoxville in March 1984 and received his Bachelor of Science degree in Electrical and Computer Engineering in August 1988. He enrolled as a graduate student and became a research assistant in September 1989. The research is being concentrated in the area of image processing and pattern recognition.