Handwritten Alphanumeric Character Recognition by the Neocognitron

Kunihiko Fukushima and Nobuaki Wake

Abstract—**A neural network model of visual pattern recognition, called the neocognitron, was previously proposed by one of the present authors. It is capable of deformation-invariant visual pattern recognition. In this paper, we discuss a pattern-recognition system which works with the mechanism of the neocognitron. The system has been trained to recognize 35 handwritten alphanumeric characters. The ability to correctly recognize deformed characters depends highly on the choice of the training pattern set. This paper offers some techniques for selecting training patterns useful for deformation-invariant recognition of a large number of characters.**

I. Introduction

A neural network model for visual pattern recognition called the neocognitron has been previously proposed by one of the present authors [1]–[3]. It is capable of deformation-invariant visual pattern recognition.

The neocognitron is a hierarchical network consisting of several layers of neuronlike cells. There are forward connections between cells in adjoining layers. Some of these connections are variable, and can be modified by learning. The neocognitron can acquire the ability to recognize patterns by learning, and can be trained to recognize any set of patterns. Since it has a large power of generalization, presentation of only a few typical examples of deformed patterns (or features) is enough for the learning. It is not necessary to present all the deformed versions of the patterns which might appear in the future. After learning, it can recognize input patterns robustly, with little effect from deformation, changes in size, or shifts in position. In contrast to most conventional pattern recognition systems, it does not require any preprocessing such as normalizing the position, size, or deformation of input patterns.

The ability of the neocognitron has already been demonstrated in various experiments. As an example, the author and his group constructed a pattern recognition system with the mechanism of the neocognitron. The system was trained to recognize handwritten numerals, and it operates on several kinds of computers—from microcomputers to parallel computers [3]–[5].

Manuscript received June 18, 1990; revised December 20, 1990.

The authors are with the Department of Biophysical Engineering, Faculty of Engineering Science, Osaka University, Toyonaka, Osaka 560, Japan.

IEEE Log Number 9143027.

There were only ten numeric characters that were recognized by the above system. In order to show that the neocognitron works well when there are more categories of patterns, we have expanded the scale of the previous system and constructed a new one. The new system has been trained to recognize not only numeric characters but also alphanumeric characters, for a total of 35 characters (where the alphabetic character O and the numeral 0 are treated as the same pattern). Preliminary reports of the new system have already appeared [6], [7]. This paper discusses the system in detail.

We chose supervised learning to train this system. The ability to correctly recognize deformed characters depends highly on the choice of local features to be extracted in intermediate stages of the hierarchical network. These local features are determined by training patterns given to the network. This paper offers some techniques for selecting training patterns useful for deformation-invariant recognition of a large number of characters.

II. The Structure of the Network

A. An Outline of the Neocognitron

An outline of the neocognitron is given in this section, and a mathematical description of the network appears in the Appendix.

The hierarchical structure of the network is illustrated in Fig. 1, in which each rectangle represents a two-dimensional array of cells. The lowest stage of the network is the input layer, which consists of a two-dimensional array of receptor cells. Each succeeding stage has a layer consisting of cells called S cells followed by another layer of cells called C cells. Thus, in the whole network, layers of S cells and C cells are arranged alternately. S cells are feature-extracting cells. The C cells are inserted in the network to allow for positional errors in the features. The layer of C cells at the highest stage is the recognition layer, representing the final result of the pattern recognition by the neocognitron.

The notation U_{Sl} and U_{Cl} is used to denote the layers of S cells and C cells of the lth stage, respectively. The input layer is denoted by U_0.

Each layer of S cells or C cells is divided into subgroups, called cell planes, according to the feature to which they respond. The cells in each cell plane are arranged in a two-dimensional array. In Fig. 1, each quadrangle drawn with heavy lines represents a cell plane, and

1045-9227/91/0500-0355$01.00 © 1991 IEEE

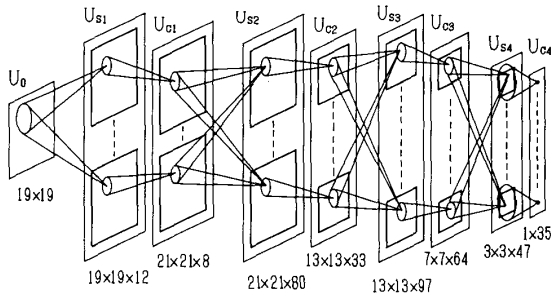


Fig. 1. Hierarchical network structure of the neocognitron. The numerals at the bottom of the figure show the total numbers of S and C cells in individual layers of the network.

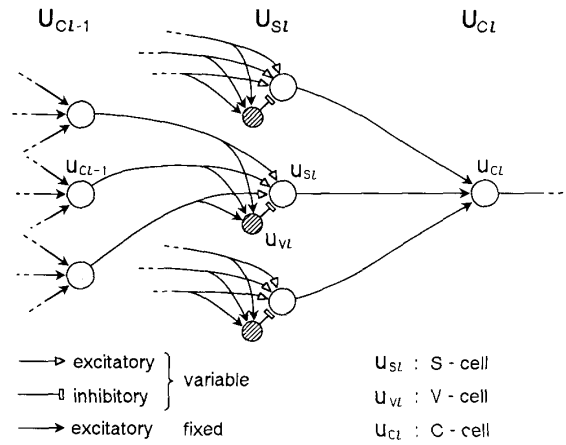


Fig. 2. Connections between cells in the network.

each vertically elongated quadrangle drawn with thin lines, in which cell planes are enclosed, represents a layer of S cells or C cells. Incidentally, each layer of S cells contains subsidiary inhibitory cells, called V cells, but they are not shown explicitly in Fig. 1.

The connections converging to the cells in a cell plane are homogeneous: all the cells in a cell plane receive input connections of the same spatial distribution, in which only the positions of the preceding cells shift in parallel with the position of the cells in the cell plane.

The density of cells in each layer is designed to decrease with the order of the stage, because the cells in higher stages usually have larger receptive fields and the neighboring cells receive similar signals.

Fig. 2 illustrates how the cells are connected in the network. Connections converging to feature-extracting S cells are variable and are reinforced by training (or learning). After finishing the training, S cells, with the aid of the subsidiary V cells, can extract features from the input pattern. In other words, an S cell is activated only when a particular feature is presented at a certain position in the input layer. The features which the S cells extract are determined by training patterns given to the input layer. Generally speaking, local features, such as a line at a particular orientation, are extracted in the lower stages. In the higher stages, features that are more global are extracted, such as a part of a training pattern.

C cells have the function of tolerating positional errors of the features extracted by the S cells. Connections from S cells to C cells are fixed and invariable. Each C cell receives signals from a group of S cells which extract the same feature, but from slightly different positions. The C cell is activated if at least one of these S cells is active. Even if the stimulus feature is shifted in position and another S cell is activated instead of the first one, the same C cell keeps responding. Hence, the C cell's response results in lower sensitivity to shifts in position of the input pattern.

In the entire network, with its alternate layers of S cells and C cells, the processes of feature extraction by the S cells and toleration of positional shift by the C cells are repeated. Tolerating positional errors a little at a time at each stage, rather than all in one step, is effective for deformation-invariant pattern recognition. During this process, local features that are extracted in a lower stage are gradually integrated into more global features. Finally, each C cell of the recognition layer at the highest stage integrates all the information of the input pattern and responds only to one specific pattern. In other words, only one C cell is activated, corresponding to the category of the input pattern. Other cells respond to the patterns of other categories.

B. Optimal Scale of the Network

The optimal scale of the neocognitron changes depending on the set of patterns to be recognized. Although it is difficult to show precisely how to choose the network scale parameters, we can offer the following guidelines.

If the complexity of the patterns is high, the sizes of spatial spread of connections between cells have to be small, and the total number of stages in the hierarchical network needs to be large. On the other hand, if the number of categories of the patterns is large, the number of cell planes in each stage has to be increased. The following is a more detailed discussion of this.

The size of A_l in (1) and (3) in the Appendix, which determines the spatial spread of the excitatory input connection of an S cell (and also of a V cell), corresponds to the size of features to be extracted by the S cell. In intermediate stages, it has to be determined depending on the density of local features contained in the patterns to be recognized. If the density of local features is high, the size of A_l has to be relatively small, but if the local features are sparse, the size can be large. The density of local features is highly correlated with the complexity of the input patterns to be recognized. The more complex the patterns, the smaller the size of A_l.

The optimal size of A_l is also affected by the degree of deformation which the neocognitron has to tolerate. Especially in lower stages, in which the size of A_l coincides with the size of local features, the size of A_l can be large

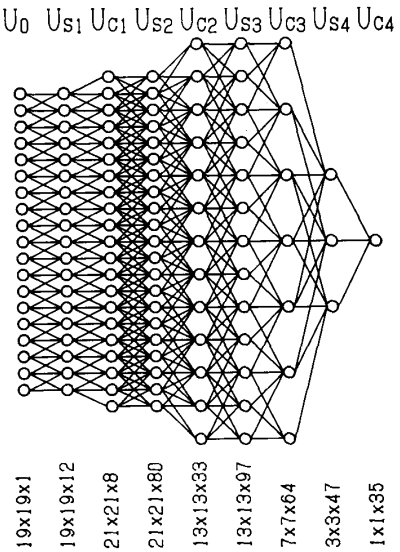


Fig. 3. One-dimensional cross section of interconnections between cells of different cell planes. Only one cell plane is drawn in each layer.

if the probable deformation of each feature is small, but it must be small when large deformation has to be tolerated.

The size of D_l in (4) in the Appendix, which determines the spatial spread of the fixed excitatory input connection of a C cell, also has a tendency similar to that of A_l. If the density of features in a pattern is large, the size of D_l has to be small. Otherwise, detecting the configuration of local features becomes ambiguous. If the density is sparse, the size of D_l can be large; consequently a larger deformation can be absorbed at one stage. This means that the network can tolerate larger deformation with a smaller number of stages.

Therefore, the total number of stages of the network is determined by the complexity of the patterns to be recognized. Complex patterns require small sizes of A_l and D_l. If the relative sizes of A_l and D_l with respect to the size of the cell planes (which is closely related to the size of the patterns to be recognized) are small, a large number of stages are necessary to integrate information from all parts of the input patterns to each cell at the highest stage.

On the other hand, the necessary number of cell planes in each stage of the network is determined by the number of categories of the patterns to be recognized. More specifically, it is equal to the number of different features to be extracted in each stage, and is determined by the training pattern set. The selection of training patterns and the necessary number of cell planes will be discussed in detail in the following sections.

The present system for alphanumeric character recognition is a four-stage (i.e., nine-layer) network. The number of stages in the network is the same as that of the system for numeric character recognition, because the complexity of each pattern does not differ significantly between numeric and alphanumeric characters. The number of S or C cells in each layer is indicated in Fig. 1. Fig. 3 shows how the cells of different cell planes are spatially interconnected. This figure, in which only one cell plane is drawn for each layer, illustrates a one-dimensional cross section of the connections between S and C cells. The sizes of A_l and D_l, as well as the sizes of cell planes, can also be read from Fig. 3.

The recognition layer U_{C4} has 35 cell planes, each of which contains only one C cell. These 35 C cells corresponds to 35 alphanumeric patterns: namely, 26 uppercase alphabetic characters and ten Arabic numerals, where the alphabetic character O and the numeral 0 are treated as the same pattern.

III. Training the Network

A. An Outline of the Training

The neocognitron can be trained either by supervised learning or by unsupervised learning. The present system for alphanumeric character recognition is trained by supervised learning (or learning with a teacher) [3].

The variable input connections of the S cells are reinforced by the training. Their initial values before training are all zero. Training is performed step by step from the lower stages to the higher stages. In other words, training of a higher stage is performed after completely finishing the training of the preceding stages.

All of the stages are trained with the same process. Let us consider a case of training a particular stage. From the layer of S cells in the stage, a "teacher" (or a trainer) first chooses a cell plane to be trained. He then presents a training pattern to the input layer U_0 and indicates which cell in the cell plane should be the "seed cell." The seed cell is pointed to by the position of its receptive field center.

The amount of reinforcement of each input connection to the seed cell is proportional to the intensity of the response of the cell from which the relevant connection is leading. The response of the latter cell is obtained (or can be calculated) by simply presenting the training pattern to the input layer, since the training of the preceeding stages has already been completed.

As a result of this learning principle, the variable input connections to the seed cell grow so as to work as a "template" which exactly matches the spatial distribution of the responses of the cells in the preceding layer. Thus, the seed cell acquires the ability to extract the feature of the stimulus which has been presented during the training period.

The seed cell works like a seed in crystal growth: all the other S cells in the cell plane follow the seed cell and have their input connections reinforced by having the same spatial distribution as those of the seed cell. Therefore, the homogeneity of connections within cell planes is always preserved, and all the cells in the cell plane come to extract the same feature at different locations.

This means that all the S cells in a cell plane share the same set of input connections. This is an advantageous characteristic for implementation of the system on a digital computer. The system requires only a small memory space to store the connections, and the computational overhead can be reduced.

B. A Strategy for Selecting Features

When recognizing a pattern consisting of line drawings by the neocognitron, the cells in a lower stage extract local features such as line segments in a particular orientation, intersections of lines, the corner of a bent line, the curvature of lines, and the end of lines. Among these features, line segments show peculiar characteristics different from others.

Suppose an intermediate layer of the neocognitron, say layer U_{S2}, contains cell planes that extract line segments, as well as other cell planes that extract corners, crossings, and so on. Let a pattern such as "L," which consists of a bent line, be presented to the input layer. Many cells will be activated in the cell plane which extracts horizontal line segments, as well as in the cell plane which extracts vertical line segments, as shown in Fig. 4(a). However, only one cell (or at most a few cells around it) will be activated in a cell plane which extract corners. The same is true for each end of the bent line. If the output signals from these cell planes are treated under the same criterion, the influence of the line segments becomes extremely strong because of the large number of activated cells. To make matters worse, the number of activated cells varies considerably with the change in size of the input pattern, depending on the length of the straight lines in the pattern, as shown in parts (a) and (b) of Fig. 4. This makes deformation-invariant recognition difficult.

Therefore, if size- and deformation-invariant pattern recognition of line drawings is desired, ignoring the line-extracting cells in the intermediate layers can be effective. Readers might feel that a pattern consisting of only a straight line, for example, cannot be recognized without detecting line segments. However this is not the case. In order to recognize a straight line, it is enough to detect the existence of both ends of the line and the absence of all other features between them. If the input pattern is not a straight line, some other features must be detected between the two ends of the line. Thus we can recognize input patterns without detecting line segments explicitly in the middle part of lines. Since cells that detect line segments do not exist in the layer, the number of activated cells does not vary much even if the size of the input pattern changes.

Therefore, layers U_{S2} and U_{S3} of the neocognitron are trained to extract local features other than line segments: these can be corners, curvature, intersections, ends of lines, etc. Training patterns which would create cell planes that extract line segments in intermediate layers are not given.

This does not mean, however, that line-extracting cells are completely eliminated from all stages of the neocognitron. On the contrary, the first layer, U_{S1}, of the present system consists of cell planes that extract line segments only, and no other features are extracted in this stage. All other features are extracted indirectly in the higher stages using the output of layer U_{S1}.

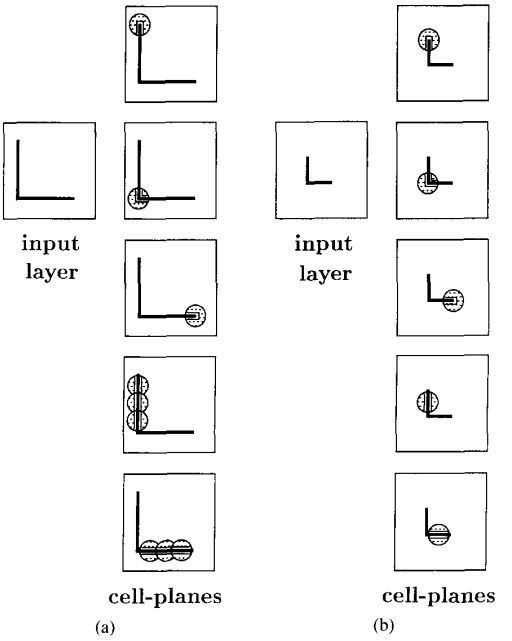


Fig. 4. Local features in an L-shaped pattern. Among these features, line segments show peculiar characteristics different from others.

C. Training Patterns

1) Training Patterns for Layer U_{S1}: Layer U_{S1} is trained to extract line components of different orientations. Fig. 5 shows the 12 training patterns used to train the 12 cell planes of layer U_{S1}. Each of the training patterns is presented to the network only once. The cell at the center of the cell plane to be trained is always appointed as the seed cell. As can be seen from Fig. 3, each cell of this layer has a receptive field that is 3×3 in size. Hence, only the central 3×3 area of each training pattern is effective during training, and only this central area is shown in Fig. 5.

Since the size of the receptive fields of S cells is as small as 3×3, it is difficult to extract all parts of a line by only one cell plane when the line has an inclination ratio of 1 : 2. Therefore, two cell planes are used to extract such a slanted line component, and the outputs from these two S cell planes are joined together and made to converge to a single C cell plane. Each bracket drawn to the left of the training patterns in Fig. 5 indicates how the outputs of the corresponding S cell planes are joined together.

The selectivity of the response of S cells in feature extraction can be controlled by the efficiency of the inhibitory inputs to the S cells [8]. The efficiency of the inhi-

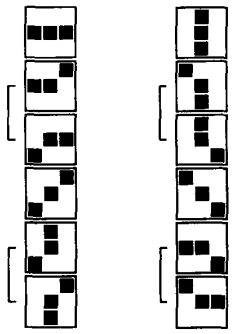


Fig. 5. Training patterns used to train the 12 cell planes of layer U_{S1}. Each bracket shows that the outputs of the corresponding S cell planes are joined together and converge to a single C cell plane at U_{C1}. Only the central 3 × 3 area of each training pattern is shown, because the outside of this area is not effective for the training of layer U_{S1}.

bition is determined by the positive parameter $r_l(k)$ in (1) in the Appendix. A larger $r_l(k)$ corresponds to a smaller tolerance of noise and deformation of the feature.

For layer U_{S1}, the parameter $r_1(k)$ has been determined in such a way that the orientation selectivity of S cells does not become too large and that the S cells respond not only to optimally oriented lines, but also to lines of slightly different orientations. For example, S cells trained with the training pattern shown at the top of the left column of Fig. 5 respond to a small degree to the second and third patterns of the same column, and also to the fifth and sixth patterns in the right column. More specifically, $r_1(k) = 1.7$ is chosen.

2) Training Patterns for Layer U_{S2}: Fig. 6 shows the training patterns used to train the 80 cell planes of layer U_{S2}. Only the central 9 × 9 areas of the training patterns are shown here, because the S cells of this layer have receptive fields 9 × 9 in size. Incidentally, the size of the receptive field of a cell can be read easily from Fig. 3. Again, the cell at the center of the cell plane to be trained is appointed as the seed cell.

As can be seen in Fig. 6, each training pattern generally consists of a part of an alphanumeric pattern which is supposed to appear during the process of pattern recognition. In other words, typical examples of deformed patterns are presented to the network as training patterns.

Sometimes, a single cell plane is trained with more than one training pattern. This is effective in increasing the S cell's ability to extract deformed features. A group of patterns arranged in a horizontal line in Fig. 6 represents such a set of training patterns. When superimposing a set of training patterns, the adjustment of their relative positions is important. If they are not properly aligned, poor results will be obtained. The value of $c_l(\mathbf{v})$ in (6) in the Appendix is determined to be large at the center and gradually decreases in the periphery. This means that the excitatory variable connections to S cells are reinforced more strongly at the center than at its periphery. Therefore, the positions of the training patterns presented to the same seed cell are adjusted in such a way that they overlap at the center as precisely as possible, and that the offset occurs only in the peripheral parts.

In handwritten character recognition, considerably different shapes have to be treated as the same feature. If the differences in shape among such features are too large to be extracted by a single cell plane,[1] they are extracted separately by several S cell planes in parallel, and the outputs from these S cell planes are joined together at the input of a C cell plane. As with layer U_{S1} shown in Fig. 5, the brackets in Fig. 6 indicate how the cell planes are joined.

As an example of this, Λ-shaped features are contained in several patterns, such as "A," "M," "N," "W," and "4," but these features have shapes that differ considerably with respect to one another. The deformation among these features is too large to be detected by a single cell plane. Hence four cell planes are used separately to extract these deformed shapes. Lines 13 to 16 in the third column of Fig. 6 shows the training patterns used to train these cell planes. They are a sharp angle, a wide angle, an angle slanted to the right, and an angle slanted to the left. The outputs from these S cell planes are joined together at the input of a C cell plane.

Another important factor which determines the ability of feature extraction is deciding which part of the training pattern should come at the receptive field center of the seed cell. For example, the corners of the L-shaped training patterns in the top line in the second column of Fig. 6 are placed not at the center but to the upper right of the center. This kind of offset is useful in detecting the absence of line components in the lower left part of the receptive field and consequently increases the ability to discriminate L-shaped patterns from, say, cross-shaped patterns.

There are some features which are difficult to detect directly in one step in layer U_{S2}. An example of such a features is the central part of "X," where many lines cross each other. Because of a large excitatory effect from many lines, the absence of only one of them is difficult to detect by a single S cell. Therefore, the cell might erroneously be activated if even one of the lines is missing. In the present system, several features are extracted around such a crossing point separately by a number of cell planes in layer U_{S2}, and these features are integrated into one at the succeeding stage, U_{S3}. If we try to detect the difference with only layer U_{S2} by increasing the value of the parameter $r_2(k)$, this simply results in a decreased ability to extract deformed features.

The present system is designed so that we can adjust the selectivity of the response of each cell plane separately, by changing the value of the parameter $r_l(k)$. However, in most cases we did not feel such a need, and each

[1]If a single seed cell is forced to be trained with patterns that differ too much in shape from one another, the cell stops responding to any pattern. This can be easily understood from the mathematical analysis given in [8]. Parameter λ in equation (17) of that paper becomes too small, and the right side of the equation becomes larger than 1.0, which is the maximum possible value of s.

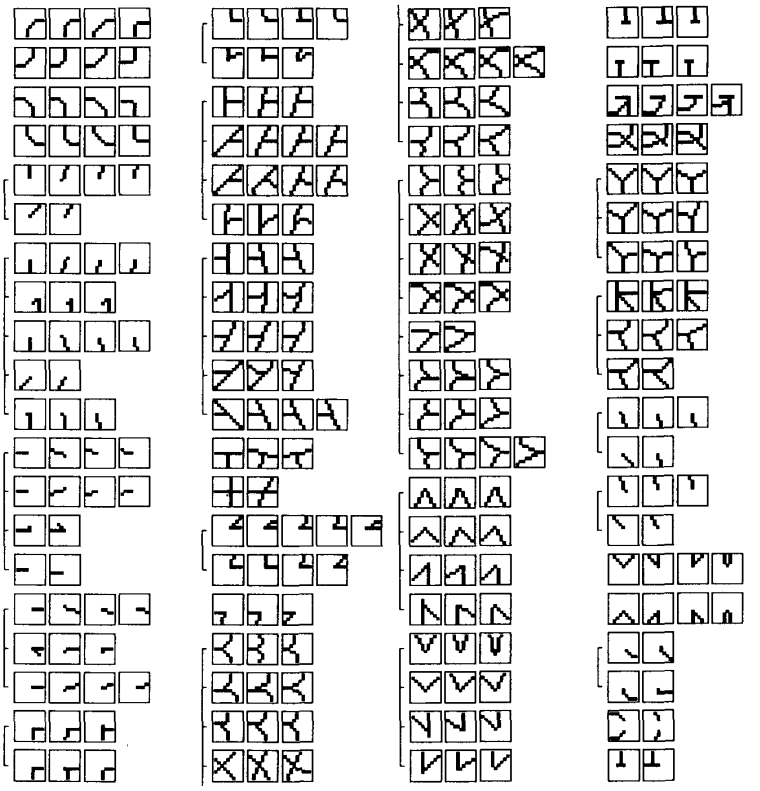


Fig. 6. Training patterns used to train the 80 cell planes of layer U_{S2}.

parameter is actually adjusted to the same value for most of the cell planes, except for a few in the same layer. To be more exact, we chose $r_2(k) = 3.8$ for the cell planes trained by the training patterns at the bottom of the second column, the first and second and the seventh through ninth training patterns in the third column of Fig. 6. We chose $r_2(k) = 4.0$ for all the other cell planes.

Generally speaking, good selection of training patterns is most important for layer U_{S2} among all the layers. If the training patterns for layer U_{S2} are properly selected, the network usually acquires a considerably high ability of pattern recognition, even though the selection of the training patterns for other layers is not complete.

3) Training Patterns for Layer U_{S3}: Layer U_{S3} extracts global features by combining local features extracted in the preceding layer, U_{S2}. After training, each cell plane in U_{S3} come to receive input connections from several cell planes extracting different features.

Fig. 7 shows the training patterns used to train the 97 cell planes of layer U_{S3}. Since the receptive fields of S cells of this layer are larger in size than the input layer U_0, the cell at the center of a cell plane cannot always be appointed as the seed cell. Therefore, the position of the seed cell (that is, the receptive-field center of the seed cell) is marked by a cross in each training pattern in Fig. 7. Incidentally, $r_3(k) = 1.5$ is chosen for this layer.

In order to prevent confusion between characters that resemble each other, it is important to choose features which emphasize the difference between them. To discriminate "E" and "F," for example, we use features which are contained in only one of them, as shown in the first and second lines in the third column of Fig. 7.

Generally speaking, we tried to select features to be extracted in layer U_{S3} in such a way that the same number of cell planes is always activated in layer U_{S3}, independent of the categories of the input patterns.

4) Training Patterns for Layer U_{S4}: Fig. 8 shows the training patterns used to train the 47 cell planes of layer U_{S4}. As with U_{S1} and U_{S2}, the cell at the center of each cell plane is appointed as the seed cell. We chose $r_4(k) = 1.0$.

Some characters have two or more different styles of writing, which are difficult to detect by means of single cell planes. In this case, two S cell planes are used to detect the single character, and their outputs are joined together to a single C cell in the recognition layer, U_{C4}. As can be seen in Fig. 8, however, only a few characters actually required such joining.

IV. Response of the Network

This system is simulated on a SUN workstation (SPARCstation 1+). The program is written in FOR-

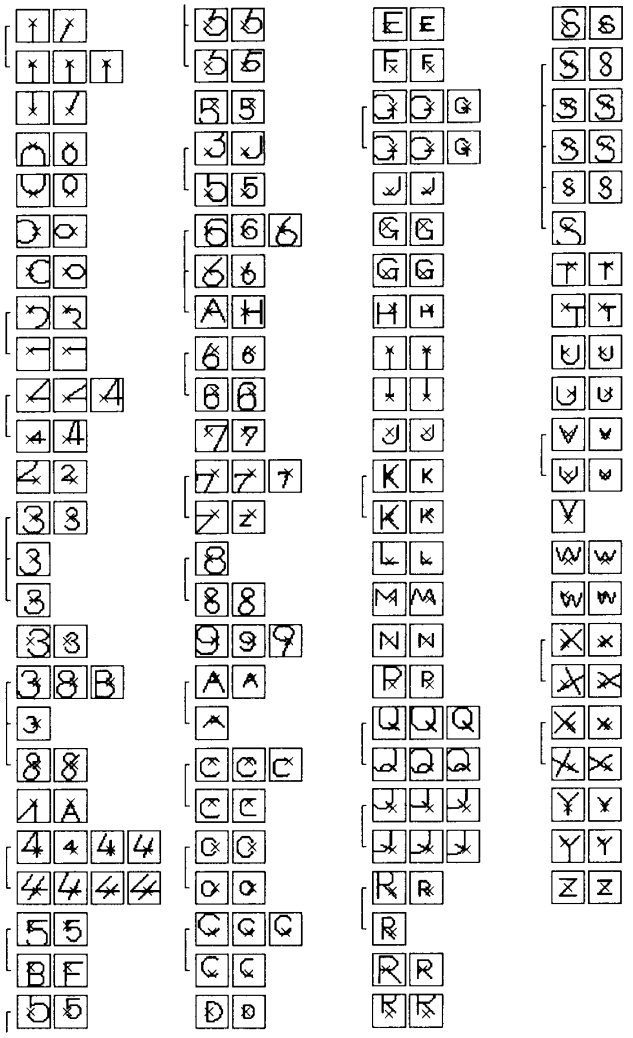


Fig. 7. Training patterns used to train the 93 cell planes of layer U_{S3}. Receptive field centers of the seed cells are marked by crosses in the figure.

TRAN. The response of the network after finishing training has been tested. In this experiment, a mouse is used to input test characters. Although a mouse is used, the system does not use any temporal information about the order of the strokes of the characters. Static patterns which have already been drawn are used as input patterns to the system.

This system has been designed to accept input characters consisting of lines of one pixel width. (This is not an essential restriction of the neocognitron and will be solved as discussed in the next section.) In order to draw characters with lines of one pixel width, the following simple algorithm is used. The input plane, on which characters are drawn by a mouse, is divided into 19 × 19 square pieces, which correspond to the 19 × 19 cells of the input layer U_0. A cell is activated and yields an output of 1.0 if, and only if, the locus of the mouse crosses its surface more than a half of the length of one side of the square, in either the horizontal or the vertical direction. Hence a cell for which the locus of the mouse touches only at a corner, for example, is not activated and remains at 0.0.

Any preprocessing for the input pattern, such as normalizing scale or rotation, is not made at all. The patterns drawn with the mouse are directly used as test patterns to the system. One character is presented at a time.

Fig. 9 shows an example of the response of the network, which has finished training. The input pattern and the responses of all C cells are displayed together. A pattern "A" has been presented to the input layer U_0. In the recognition layer U_{C4}, shown at the extreme right, only cell "A" is activated. This means that the neocognitron recognizes the input pattern correctly.

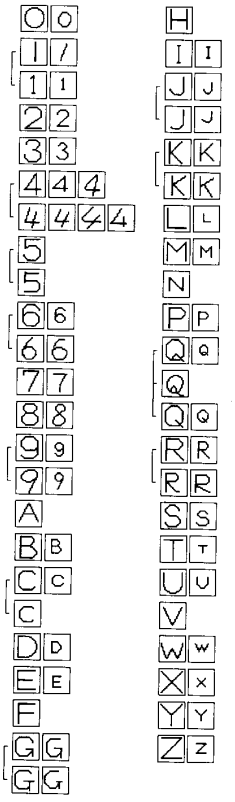


Fig. 8. Training patterns used to train the 47 cell planes of layer U_{S4}.

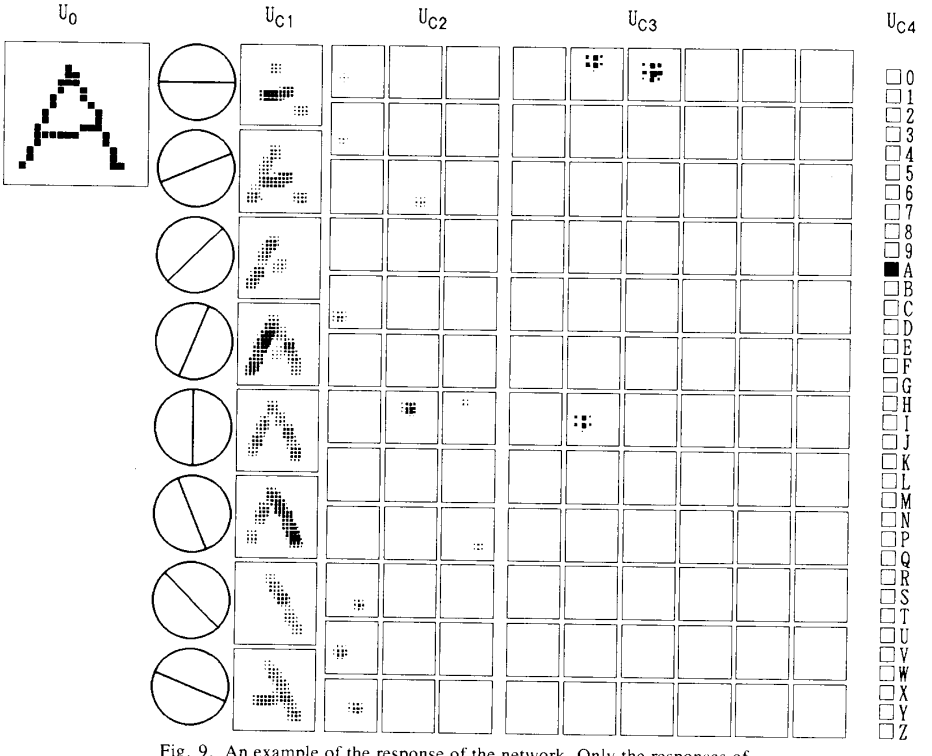


Fig. 9. An example of the response of the network. Only the responses of the C cells are displayed.

Fig. 10. Some examples of deformed input patterns which the neocognitron has recognized correctly.

Fig. 10 shows some example of deformed input patterns which the neocognitron has recognized correctly. As can be seen from this figure, the neocognitron recognizes the input pattern correctly without being affected by deformation, changes in size, shifts in position, or contaminating noise.

V. Discussion

The present system, which can recognize 35 alphanumeric characters, is somewhat larger in scale than a previously reported system [3]–[5], which can recognize only ten numerals. However, the increase in scale is not so great, as shown in Table I: the number of cells in the present system is 70 045 (including S, C, and V cells, as well as the receptor cells), while that of the previous system is 36 321. The number of characters to be recognized is increased by a factor of 3.5, but the number of cells is increased by a factor of only around 1.9. To be more specific, the numbers of cell planes for U_{S1}, U_{S2}, and U_{S3} in the present system are 12, 80, and 97, respectively, while the numbers of the previous system are 12, 38, and 35, respectively. These numbers are increased by factors of only by 1.0, 2.1, and 2.8, respectively. This is because the local features to be extracted in the lower stages are common, and are usually contained in many patterns of different categories.

TABLE I
Comparison of the Numbers of Cells and Cell Planes in the Present and the Previous System

Layer	Recognition of 35 Alphanumeric Characters		Recognition of 10 Numeric Characters		Ratio 35/10
	Number of Cell Planes	Number of Cells*	Number of Cell Planes	Number of Cells*	
U_0	1	361	1	361	1.0
U_{S1}	12	4693	12	4693	1.0
U_{C1}	8	3528	8	3528	1.0
U_{S2}	80	35 721	38	17 199	2.1
U_{C2}	33	5577	19	3211	1.7
U_{S3}	97	16 562	35	6084	2.7
U_{C3}	64	3136	23	1127	2.8
U_{S4}	47	432	11	108	4.0
U_{C4}	35	35	10	10	3.5
Total	—	70 045	—	36 321	1.9

*Layers U_S include V cells as well as S cells.

The computation time to recognize one alphanumeric character is 3.3 s on average on the SUN SPARCstation. Incidentally, it is about 1.5 s for the ten-numeric-character recognition system.

In order to test the performance of the system, the authors and their colleagues tried to write characters using various styles. It was found that the system recognizes them robustly unless the deformation from training patterns is too large. However, it is difficult to state quantitatively to what degree the system can cope with deformation in patterns, because we do not have an appropriate mathematical measure to correctly express the psychological feeling of the deformation.

The possibility of confusion between similar patterns generally increases with the number of characters to be recognized. Some of them cannot be recognized correctly even by a human observer. In order to reduce confusion between similar characters of different categories, we placed certain restrictions on the way of writing when creating a training pattern set. For instance, the character "I" has been taught to the system with serifs at both ends of the vertical bar. A vertical bar without serifs is taught as the numeral "1." Most of the characters have been taught without serifs.

Although a skillful choice of training patterns can make the neocognitron discriminate between similar patterns of different categories, a process of constructing a good training pattern set requires hard labor with an increase in the number of characters to be recognized. On the other hand, the conventional technique of unsupervised learning for the neocognitron [1], [2], with which all the training processes progress automatically, shows a somewhat lesser ability to recognize deformed patterns. An im-

provement of the technique of unsupervised learning or the development of a new technique combining supervised and unsupervised learning is a problem left to be solved in the future.

One of the advantages of the supervised learning used for this system is a very short training time. Once the training pattern set had been created, the training time was only 13 min on the SUN SPARCstation. This is extremely short compared with other training methods, such as back-propagation. For instance, LeCun *et al.* [9] used the back-propagation algorithm to train a network similar to the neocognitron to recognize ten numerals. They reported a training time of 3 days on a SUN workstation.

The system discussed in this paper has been designed to accept input characters consisting of lines of one pixel width. Hence line-extracting u_{Sl} cells are designed to have receptive fields as small as 3×3 cells in size, and training patterns as shown in Fig. 5 are used to generate line-extracting cells. However, this is not an essential restriction on the neocognitron itself. We can design a system which can accept some variation in line thickness by increasing the density of the cells of the input layer and also increasing the number of cells in the receptive field of a line-extracting u_{Sl} cell if necessary.

Appendix
Mathematical Description of the Network

The notation $u_{Sl}(\boldsymbol{n}, k)$, for example, is used to denote the output of an S cell in layer U_{Sl}, where $\boldsymbol{n}$ is a two-dimensional set of coordinates indicating the position of the cell's receptive-field center in the input layer U_0, and k is a serial number of the cell plane. For S cells, k is in the range of $1 \le k \le K_{Sl}$; for C cells, it is in the range of $1 \le k \le K_{Cl}$. The values of K_{Sl} and K_{Cl} and the size of each cell plane in the present system are indicated at the bottom of Figs. 1 and 3.

The output of an S cell is given by

$$u_{Sl}(\boldsymbol{n}, k) = r_l(k) \cdot \varphi\left[\frac{1 + \sum_{\kappa=1}^{K_{Cl-1}} \sum_{\nu \in A_l} a_l(\mathbf{v}, \kappa, k) \cdot u_{Cl-1}(\boldsymbol{n} + \mathbf{v}, \kappa)}{1 + r_l(k) \cdot \{1 + r_l(k)\}^{-1} \cdot b_l(k) \cdot u_{Vl}(\boldsymbol{n})} - 1\right] \quad (1)$$

where

$$\varphi[x] = \begin{cases} x & \text{if } x \ge 0 \\ 0 & \text{if } x < 0. \end{cases} \quad (2)$$

In the case of $l = 1$ in (1), $u_{Cl-1}(\boldsymbol{n}, \kappa)$ stands for $u_0(\boldsymbol{n})$, the output of a receptor cell of the input layer, and we have $K_{Cl-1} = 1$.

The quantity $a_l(\mathbf{v}, \kappa, k)$ (≥ 0) is the strength of the variable excitatory connection coming from C cell $u_{Cl-1}(\boldsymbol{n} + \mathbf{v}, \kappa)$ in the preceding layer, and A_l denotes the summation range of $\mathbf{v}$, that is, the size of the spatial spread of the input connections to one S cell. In the present system, it is chosen as a square. Its size can be read from Fig. 3: 3×3 for A_1 and 5×5 for A_2, A_3, and A_4. The quantity $b_l(k)$ (≥ 0) is the strength of the variable inhibitory connection coming from the subsidiary V cell $u_{Vl}(\boldsymbol{n})$. Since all of the S cells in a cell plane have identical sets of input connections, $a_l(\mathbf{v}, \kappa, k)$ and $b_l(k)$ do not explicitly contain the argument $\boldsymbol{n}$, which represents the position of the receptive field of the cell $u_{Sl}(\boldsymbol{n}, k)$.

The positive parameter $r_l(k)$ determines the efficiency of the inhibitory input to this S cell and controls the selectivity in feature extraction [8]. A larger $r_l(k)$ corresponds to a smaller tolerance of noise and deformation of the feature.

The subsidiary V cell, which sends an inhibitory signal to this S cell, yields an output proportional to the weighted root mean square of the signals from the preceding C cells. That is,

$$u_{Vl}(\boldsymbol{n}) = \left[\sum_{\kappa=1}^{K_{Cl-1}} \sum_{\nu \in A_l} c_l(\mathbf{v}) \cdot \{u_{Cl-1}(\boldsymbol{n} + \mathbf{v}, \kappa)\}^2\right]^{1/2} \quad (3)$$

where $c_l(\mathbf{v})$ (≥ 0) represents the strength of the fixed excitatory connections and is a monotonically decreasing function of $|\mathbf{v}|$. In the present system, $c_l(\mathbf{v}) = \gamma_l^{|\mathbf{v}|}$. Quantitatively, the values of the parameters are $\gamma_1 = \gamma_2 = \gamma_3 = 0.9$ and $\gamma_4 = 0.8$ (if the pitch of the array of the cells in the preceding layer, from which the connections lead, is taken as the unit of length for $\mathbf{v}$).

The output of a C cell is given by

$$u_{Cl}(\boldsymbol{n}, k) = \psi\left[\sum_{\kappa=1}^{K_{Sl}} j_l(\kappa, k) \sum_{\nu \in D_l} d_l(\mathbf{v}) \cdot u_{Sl}(\boldsymbol{n} + \mathbf{v}, \kappa)\right], \quad (4)$$

where $\psi[\ ]$ is a function specifying the characteristic of saturation of the C cell, and is defined by

$$\psi[x] = \frac{\varphi[x]}{1 + \varphi[x]}. \quad (5)$$

Parameter $d_l(\mathbf{v})$ (≥ 0) denotes the strength of the fixed excitatory connections, and is a monotonically decreasing function of $|\mathbf{v}|$. In the present system, $d_l(\mathbf{v}) = \bar{\delta}_l \cdot \delta_l^{|\mathbf{v}|}$, where $\bar{\delta}_1 = \bar{\delta}_2 = 4.0$, $\bar{\delta}_3 = 2.5$, $\bar{\delta}_4 = 1.0$; $\delta_1 = 0.9$, $\delta_2 = 0.8$, $\delta_3 = 0.7$, and $\delta_4 = 1.0$ (if the pitch of the array of the S cells in the preceding layer, from which the connections lead, is taken as the unit of length for $\mathbf{v}$). D_l is the area to which these connections spread. In the present system, it is chosen as a square. Its size can be read from Fig. 3: 3×3 for D_1, 7×7 for D_2, 5×5 for D_3, and 3×3 for D_4.

The outputs of several S cell planes are sometimes joined together and made to converge to a single C cell plane. This condition of joining is represented by $j_l(\kappa, k)$. Depending on whether or not the kth C cell plane receives signals from the κth S cell plane, $j_l(\kappa, k)$ takes a positive value (1.0 in the present system) or zero, accordingly. Hence, for each κ, $j_l(\kappa, k)$ is usually zero except for one

particular value of k. These values can be read from the bracket drawn to the left of the training patterns in Figs. 5 through 8.

During training, variable connections $a_l(\mathbf{v}, \kappa, k)$ and $b_l(k)$ are reinforced depending on the intensity of the input to the seed cell. Let a training pattern be presented to the input layer, and let an S cell $u_{Sl}(\hat{\mathbf{n}}, \hat{k})$ be selected as a seed cell. The variable connections $a_l(\mathbf{v}, \kappa, \hat{k})$ and $b_l(\hat{k})$ to this particular seed cell, and consequently to all the S cells in the same cell plane as this seed cell, are reinforced by the following amount:

$$\Delta a_l(\mathbf{v}, \kappa, \hat{k}) = q_l \cdot c_l(\mathbf{v}) \cdot u_{Cl-1}(\hat{\mathbf{n}} + \mathbf{v}, \kappa) \quad (6)$$

$$\Delta b_l(\hat{k}) = q_l \cdot u_{Vl}(\hat{\mathbf{n}}). \quad (7)$$

In these equations, the response of the C cells $u_{Cl-1}(\hat{\mathbf{n}} + \mathbf{v}, \kappa)$ and of the V cell $u_{Vl}(\hat{\mathbf{n}})$ are determined (or can be calculated) by simply presenting the training pattern to the input layer, since the training of the preceding stages has already been completed. The quantity q_l is a positive constant determining the speed of reinforcement. In the present case of supervised learning, a sufficiently large value (10^4) is given to q_l so that the reinforcement of the input connections for each training pattern can be completed in one step. The concrete value of this parameter is not so important provided it is large enough.

REFERENCES

[1] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biol. Cybern.*, vol. 36, no. 4, pp. 193-202, 1980.

[2] K. Fukushima and S. Miyake, "Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position," *Pattern Recog.*, vol. 15, no. 6, pp. 455-469, 1982.

[3] K. Fukushima, "Neocognitron: A hierarchical neural network capable of visual pattern recognition," *Neural Networks*, vol. 1, no. 2, pp. 119-130, 1988.

[4] K. Fukushima, S. Miyake, T. Ito, and T. Kouno, "Handwritten numeral recognition by the algorithm of the neocognitron—An experimental system using a microcomputer" (in Japanese), *Trans. Inform. Process. Soc. Japan*, vol. 28, no. 6, pp. 627-635, 1987.

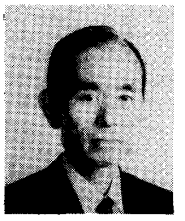
[5] T. Ito, K. Fukushima, and S. Miyake, "Realization of a neural network model neocognitron on a hypercube parallel computer," *Int. J. High Speed Computing*, vol. 2, no. 1, pp. 1-16, 1990.

[6] K. Fukushima and N. Wake, "Alphanumeric character recognition by the neocognitron," in *Advanced Neural Computers*, R. Eckmiller, Ed. Amsterdam, New York, Oxford, Tokyo: Elsevier Science Publishers B. V. (North-Holland), 1990, pp. 263-270.

[7] K. Fukushima and N. Wake, "A hierarchical neural network model for pattern recognition," presented at Int. Conf. Fuzzy Logic & Neural Networks (Iizuka, Fukuoka, Japan), July 1990.

[8] K. Fukushima, "Analysis of the process of pattern recognition by the neocognitron," *Neural Networks*, vol. 2, no. 6, pp. 413-420, 1989.

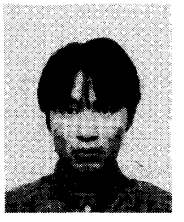
[9] Y. LeCun *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural Computat.*, vol. 1, pp. 541-551, 1989.



Kunihiko Fukushima received the B.S. degree in electronics in 1958 and the Ph.D. degree in electrical engineering in 1966, both from Kyoto University, Kyoto, Japan.

He is a Professor in the Department of Biophysical Engineering at Osaka University, Osaka, Japan. Until May 1989, he was a Senior Research Scientist at the NHK Science and Technical Research Laboratories. His major research interest is the mechanism of information processing in the brain, and he is engaged in the synthesis of neural network models of the mechanism of visual and auditory pattern recognition, selective attention, learning, self-organization, and memory. He is the author of three books in this field (in Japanese): *The Physiology and Bionics of the Visual System* (IEICE Japan, 1976), *Neural Networks and Self-Organization* (Kyoritsu, 1979), and *Neural Networks and Information Processing* (Asakura, 1989).

Prof. Fukushima is president of the Japan Neural Network Society and serves on the Governing Board of the International Neural Network Society.



Nobuaki Wake received the B.S. degree in biophysical engineering in 1990 from Osaka University, Osaka, Japan. He is currently studying toward the master's degree in the Department of Biophysical Engineering at Osaka University. His research focuses on neural networks.