Neural Networks 37 (2013) 103–119

Contents lists available at SciVerse ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Artificial vision by multi-layered neural networks: Neocognitron and its advances

Kunihiko Fukushima *

Fuzzy Logic Systems Institute, Iizuka, Fukuoka, Japan

ARTICLE INFO

Keywords:
Artificial vision
Neocognitron
Hierarchical network
Bottom-up and top-down
Modeling neural networks

ABSTRACT

The *neocognitron* is a neural network model proposed by Fukushima (1980). Its architecture was suggested by neurophysiological findings on the visual systems of mammals. It is a hierarchical multi-layered network. It acquires the ability to robustly recognize visual patterns through learning. Although the neocognitron has a long history, modifications of the network to improve its performance are still going on. For example, a recent neocognitron uses a new learning rule, named *add-if-silent*, which makes the learning process much simpler and more stable. Nevertheless, a high recognition rate can be kept with a smaller scale of the network. Referring to the history of the neocognitron, this paper discusses recent advances in the neocognitron. We also show that various new functions can be realized by, for example, introducing top-down connections to the neocognitron: mechanism of selective attention, recognition and completion of partly occluded patterns, restoring occluded contours, and so on.

1. Introduction

In the visual systems of mammals, visual scenes are analyzed in parallel by separate channels. Loosely speaking, information concerning object shape is mainly analyzed through the temporal pathway in the cerebrum, while information concerning visual motion and location is mainly analyzed through the occipito-parietal pathway. The neocognitron is an artificial neural network, whose architecture was initially suggested from neurophysiological findings on the temporal pathway: retina → LGN → area V1 (primary visual cortex) → area V2 → area V4 → IT (inferotemporal cortex).

In area V1, cells respond selectively to local features of a visual pattern, such as lines or edges in particular orientations (Hubel & Wiesel, 1962, 1965). In areas V2 and V4, cells exist that respond selectively to complex visual features (e.g., Ito and Komatsu (2004), von der Hydt, Peterhans, and Baumgartner (1984) and Desimone and Schein (1987)). In the inferotemporal cortex, cells exist that respond selectively to more complex features, or even to human faces (e.g., Fujita, Tanaka, Ito, and Cheng (1992), Bruce, Desimone, and Gross (1981) and Yamane, Kaji, and Kawano (1988)). Thus, the visual system seems to have a hierarchical architecture, in which simple features are first extracted from a stimulus pattern, and then integrated into more complicated ones. In this hierarchy, a cell in a higher stage generally has a larger receptive field, and is more insensitive to the location of the stimulus. This kind of physiological evidence suggested the network architecture of the neocognitron.

In the 1960s, Hubel and Wiesel classified cells in the visual cortex into simple, complex and hypercomplex cells. They hypothesized that visual information is processed hierarchically through simple cells → complex cells → lower-order hypercomplex cells → higher-order hypercomplex cells (Hubel & Wiesel, 1962, 1965). They suggested that, in this hierarchy, the relation between lower-order hypercomplex cells to higher-order hypercomplex cells resembles that between simple cells to complex cells. Although classifying hypercomplex cells into lower-order and higher-order is not popular among neurophysiologists recently, it is this hypothesis that suggested the original architecture of the neocognitron model when it was first proposed by Fukushima (1980).

In the neocognitron, there are two major types of cells, namely *S-cells* and *C-cells*. *S*-cells, which are named after simple cells, correspond to simple cells or lower-order hypercomplex cells. Similarly, *C*-cells, which are named after complex cells, correspond to complex cells or higher-order hypercomplex cells. As shown in Fig. 1, the neocognitron consists of cascaded connection of a number of modules, each of which consists of a layer of *S*-cells followed by a layer of *C*-cells.

Although the neocognitron has a long history, modifications of the network to improve its performance are still going on. Referring to the history of the neocognitron, this paper discusses recent advances in the neocognitron.

Sections 2 and 3 discuss the basic architecture of the neocognitron and the principles for robust recognition of visual patterns. Section 4 discusses the mechanism of feature extraction by *S*-cells, comparing several learning rules adopted in the neocognitron of recent versions. Among them, a new learning rule named *add-if-silent* makes the learning process much simpler and more stable.

* Correspondence to: 634-3, Miwa, Machida, Tokyo 195–0054, Japan. Tel.: +81 44 988 5272; fax: +81 44 988 5272.
E-mail address: fukushima@m.ieice.org.
URL: http://www.www4.ocn.ne.jp/~fuku_k/index-e.html.

0893-6080/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2012.09.016

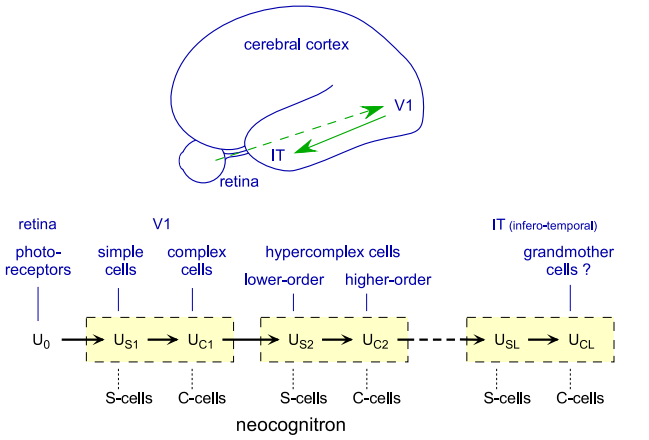


Fig. 1. Relation between the architecture of the neocognitron and the classical hypothesis of Hubel and Wiesel.
Source: (modified from Fukushima (1980)).

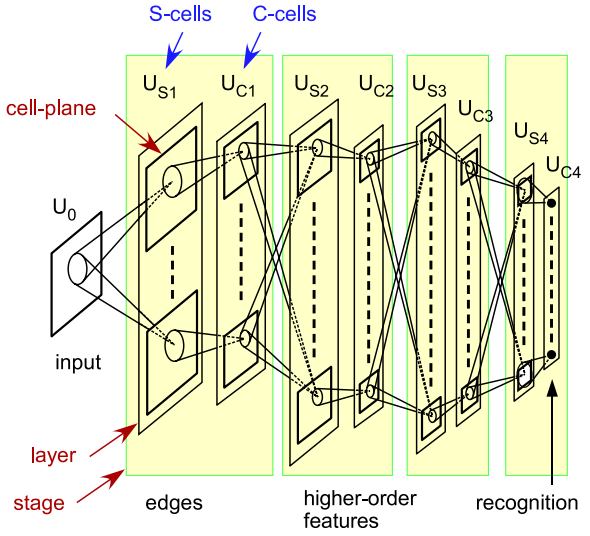


Fig. 2. A typical architecture of the neocognitron. The neocognitron consists of a number of *stages* of modules connected in a cascade in a hierarchical manner. Each stage consists of a *layer* of *S*-cells followed by a layer of *C*-cells. Each layer is divided into a number of sub-layers, called *cell-planes*, depending on the feature to which cells respond preferentially.

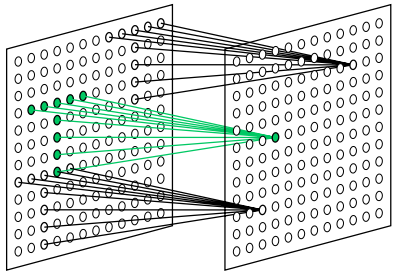


Fig. 3. An illustration of shared connections between two cell-planes. All cells in a cell-plane share the same set of input connections (Fukushima, 1980).

Nevertheless, a high recognition rate can be kept with a smaller scale of the network. Section 5 discusses the blurring operation by *C*-cells. Section 6 discusses the process of pattern classification at the highest stages of the network. We show that the method of *interpolating-vector* can greatly increase the recognition rate.

Section 7 discusses several networks extended from the neocognitron. We show that various new functions can be realized by, for example, introducing top-down connections to the neocognitron: mechanisms of selective attention, recognition and completion of partly occluded patterns, restoring occluded contours, and so on.

Incidentally, varieties of modifications, extensions and applications of the neocognitron, as well as varieties of related networks, have also been reported so far by several groups other than the author's (e.g., LeCun, Bottou, Bengio, and Haffner (1998), Mutch and Lowe (2008), Riesenhuber and Poggio (1999), Satoh, Kuroiwa, Aso, and Miyake (1999) and Serre, Oliva, and Poggio (2007)). They are all hierarchical multi-layered networks and have an architecture of *shared connections*, which is sometimes called a *convolutional net*. They also have a mechanism of pooling outputs of feature-extracting cells. The pooling operation can also be interpreted as a blurring operation. In the neocognitron, the pooling operation, which is done by *C*-cells, is performed by a weighted sum of the outputs of feature-extracting *S*-cells. In some networks, the pooling is realized by simply reducing the density of cells in higher layers. In some other networks, it is replaced by a MAX operation.

2. Outline of the network

The neocognitron is a multi-layered network, which consists of layers of *S*-cells and *C*-cells. These layers of *S*-cells and *C*-cells are arranged alternately in a hierarchical manner.

S-cells work as feature-extracting cells. Their input connections are variable and are modified through learning. After learning, each *S*-cell comes to respond selectively to a particular visual feature presented in its receptive field. The features extracted by *S*-cells are determined during learning. Generally speaking, *local* features, such as edges or lines in particular orientations, are extracted in lower stages. More *global* features, such as parts of learned patterns, are extracted in higher stages.

C-cells are inserted in the network to allow for positional errors in the features of the stimulus. The input connections of *C*-cells, which come from *S*-cells of the preceding layer, are fixed and invariable. Each *C*-cell receives excitatory input connections from a group of *S*-cells that extract the same feature, but from slightly different locations. The *C*-cell responds if at least one of these *S*-cells yields an output. Even if the stimulus feature shifts and another *S*-cell comes to respond instead of the first one, the same *C*-cell keeps responding. Thus, the *C*-cell's response is less sensitive to a shift in location of the input pattern. We can also express that *C*-cells make a blurring operation, because the response of a layer of *S*-cells is spatially blurred in the response of the succeeding layer of *C*-cells.

There are several versions of the neocognitron, which have slightly different architectures. Fig. 2 shows a typical architecture of the network. The hierarchical network has a number of *stages* of modules, each of which consists of a layer of *S*-cells followed by a layer of *C*-cells. Here we use notation like U_{Sl}, for example, to indicate the layer of *S*-cells of the *l*th stage.

There are retinotopically ordered connections between cells of adjoining layers. Each cell receives input connections that lead from cells situated in a limited area on the preceding layer. Since cells in higher stages come to have larger receptive fields, the density of cells in each layer is designed to decrease with the order of the stage.

Each layer of the network is divided into a number of sub-layers, called *cell-planes*, depending on the feature to which cells respond preferentially. In Fig. 2, each rectangle drawn with thick lines represents a cell-plane. Incidentally, a cell-plane is a group of cells that are arranged retinotopically and share the same set of input connections (Fukushima, 1980). Namely, all cells in a cell-plane share the same set of input connections, as illustrated in Fig. 3. In other words, the connections to a cell-plane have a translational symmetry. As a result, all cells in a cell-plane have identical receptive fields but at different locations. The modification of variable

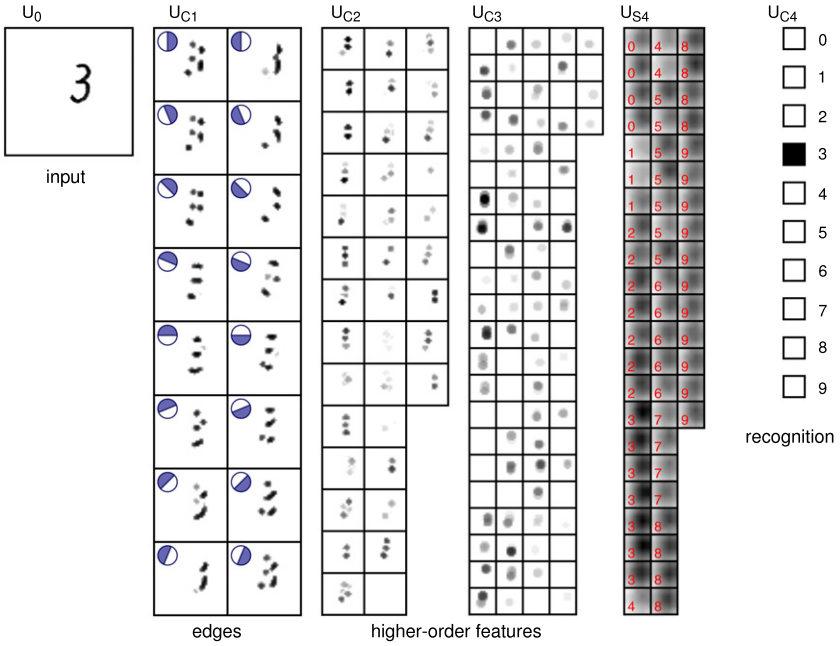


Fig. 4. An example of the response of a neocognitron that has been trained to recognize handwritten digits. Layers U_{S1}, U_{S2} and U_{S3} are abbreviated from the display, but their responses can easily be estimated from the responses of U_{C1}, U_{C2} and U_{C3}, where blurred versions of their responses appear. The half disk drawn in each cell-plane of U_{C1} shows the orientation of the edge to be extracted by the cell-plane. The numerals drawn in layer U_{S4} indicate the labels (class names) assigned to individual cell-planes. The rightmost layer, U_{C4}, shows that the input pattern is recognized correctly as '3'.

connections during learning progresses also under the constraint of shared connections.

The lowest stage of the hierarchical network is the input layer U_0 consisting of a two-dimensional array of cells, which correspond to photoreceptors of the retina. Stimulus patterns are presented to the input layer, U_0.

In the network shown in Fig. 2, the output of U_0 is sent to U_{S1}. Each S-cell of U_{S1} resembles a simple cell in the primary visual cortex, and responds selectively to an edge at a particular orientation. As a result, contours in the input image are decomposed into edges of every orientation in U_{S1}.

At each stage of the network, the output of layer U_{Sl} is fed into layer U_{Cl}. The response of layer U_{Cl} is then fed to U_{Sl+1}, the layer of S-cells of the next stage, where more global features are extracted.

S-cells at the highest stage (U_{SL}; $L = 4$ in the network of Fig. 2) are trained by supervised learning using labeled training data. As the network learns varieties of deformed training patterns, more than one cell-plane per class is usually generated in U_{SL}. Each cell-plane is assigned a label indicating the class of the training pattern that the cell-plane has learned. During recognition, the label of the input stimulus is inferred from the response of U_{SL}. The response of C-cells at the highest stage (U_{CL}) shows the inferred label.

Fig. 4 shows an example of the response of a neocognitron, which has learned to recognize handwritten digits. The responses of all layers in the network, excluding U_{S1}, U_{S2} and U_{S3}, are displayed in series from left to right. Incidentally, the responses of U_{S1}, U_{S2} and U_{S3} can easily be estimated from those of U_{C1}, U_{C2} and U_{C3}, which are blurred versions of the responses of U_{S1}, U_{S2} and U_{S3}, respectively. The numerals drawn in layer U_{S4} indicate the labels (class names) assigned to individual cell-planes. The rightmost layer, U_{C4}, shows the final result of recognition. In this example, the input pattern is recognized correctly as '3'.

3. Principles of robust recognition

3.1. Tolerating shift by C-cells

In the whole network, with its alternate layers of S-cells and C-cells, the process of extracting features by S-cells and tolerating shift by C-cells is repeated. During this process, local features extracted in lower stages are gradually integrated into more *global* features, as illustrated in Fig. 5.

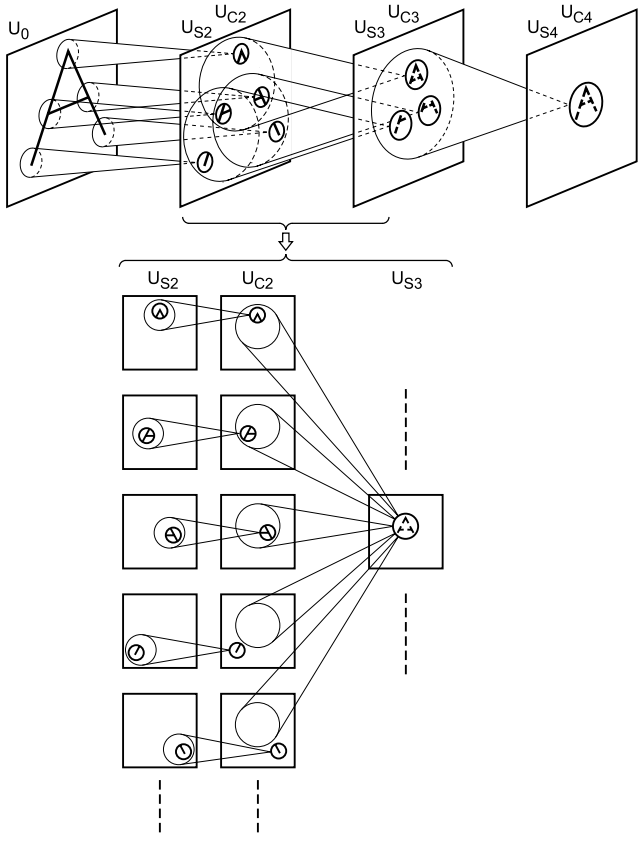


Fig. 5. The process of pattern recognition in the neocognitron. The lower half of the figure is an enlarged illustration of a part of the network.
Source: (modified from Fukushima (1980)).

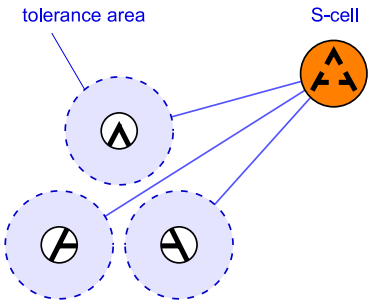


Fig. 6. Connections converging to an *S*-cell that has learned a global feature consisting of three local features of a training pattern '*A*'.

Since small amounts of positional errors of local features are absorbed by the blurring operation by *C*-cells, each *S*-cell in a higher stage comes to respond robustly to a specific feature even if the feature is slightly deformed or shifted.

Let an *S*-cell in an intermediate stage of the network have already been trained to extract a global feature consisting of three local features of a training pattern '*A*' as illustrated in Fig. 6. By the function of its presynaptic *C*-cells, the *S*-cell tolerates positional error of each local feature if the deviation falls within the dotted circle. Hence, the *S*-cell responds to any of the deformed patterns shown in Fig. 7(b) in a similar way as to Fig. 7(a).

The toleration of positional errors, however, should not be too large at this stage. If large errors are tolerated at any one step, the network may come to respond erroneously, such as by recognizing a stimulus like Fig. 7(c) as an '*A*' pattern. Thus, tolerating positional error a little at a time at each stage, rather than all in one step, plays an important role in endowing the network with the ability to recognize even distorted patterns.

The process of tolerating positional errors is repeated at every stage of the hierarchical network. The cells in a higher stage thus acquire a larger ability to accept deformation.

3.2. Blur by *C*-cells

The role of *C*-cells can also be understood from a different point of view. As illustrated in Fig. 8, the operation made by connections from *S*- to *C*-cells can also be interpreted as a blurring operation, as well as an operation of tolerating shift.

Each *S*-cell measures the similarity between the stimulus feature and the feature that the *S*-cell has learned during learning. As will be discussed later in Section 4.1, the similarity, which is defined by the inner product of two feature vectors, is determined by the degree of overlap between the two vectors. The two patterns in the left and the center of Fig. 9(a) are perceived quite similar to each other when observed visually by human beings. *S*-cells, however, judge them completely different, because their similarity defined by the inner product is zero. This is quite different from our natural feelings. If the patterns are blurred like Fig. 9(b), they come to overlap largely, and *S*-cells also judge that they are similar to each other. This coincides with our natural feelings.

If input patterns are blurred directly, however, fine details of the patterns are lost. Hence in the neocognitron, the blurring operation by *C*-cells is performed after extracting local features by *S*-cells. Namely, responses of individual cell-planes of *S*-cells are blurred in the succeeding cell-planes of *C*-cells.

We can summarize the function of *C*-cells that, by averaging their input signals, *C*-cells exhibit some level of translation invariance. As a result of averaging across location, *C*-cells encode a blurred version of their input. The blurring operation is essential for endowing the neocognitron with an ability to recognize patterns robustly, with little effect from deformation, change in size, or shift in the location of input patterns.

The averaging operation, which produces blur, is important, not only for endowing neural networks with an ability to recognize deformed patterns robustly, but also for smoothing additive random noise contained in the responses of *S*-cells. It thus increases robustness against background random noise in the input image.

The blurring operation by *C*-cells also helps reducing aliasing noise caused by coarse sampling, by which the density of cells in a cell-plane is reduced between layers of *S*- and *C*-cells.

4. *S*-cells

4.1. Feature extraction by *S*-cells

S-cells work as feature-extracting cells. Their input connections are determined through learning. After learning, each *S*-cell comes to respond selectively to a particular visual feature presented in its receptive field.

To show the essence of the process of feature extraction, we watch the circuit converging to a single *S*-cell and analyze its behavior. Fig. 10 shows the circuit. The *S*-cell of layer U_{Sl} ($l \geq 2$) receives excitatory signals directly from a group of *C*-cells, which are cells of the preceding layer U_{Cl-1}. Let x_n be the response of the nth presynaptic *C*-cell, and let a_n be the strength of the connection

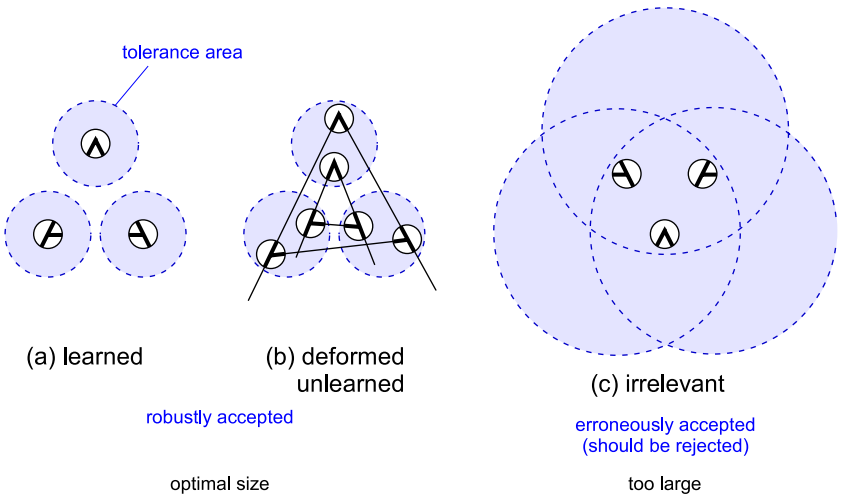


Fig. 7. Optimal size of the tolerance areas.
Source: (modified from Fukushima (1988)).

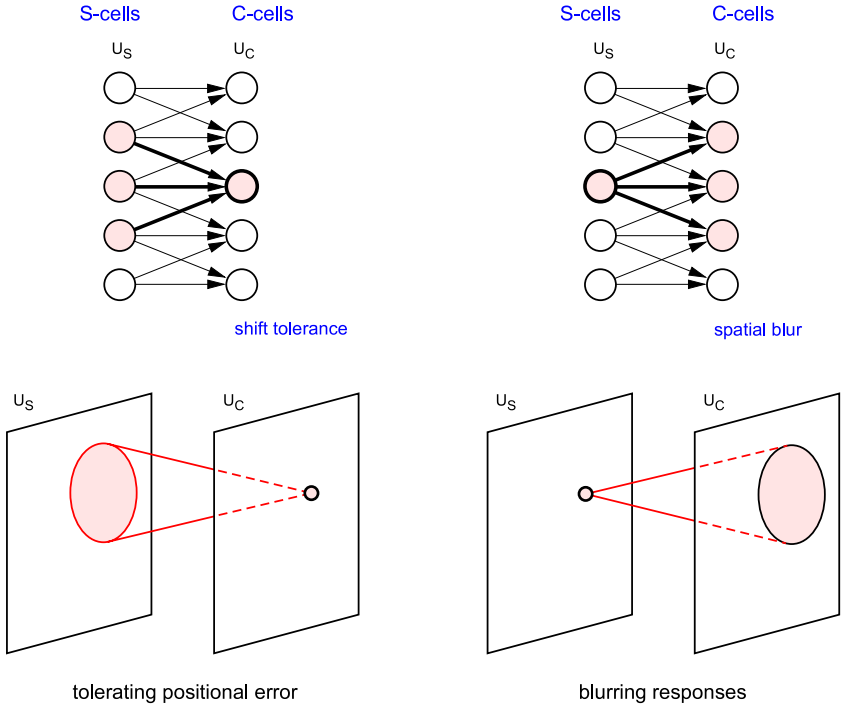
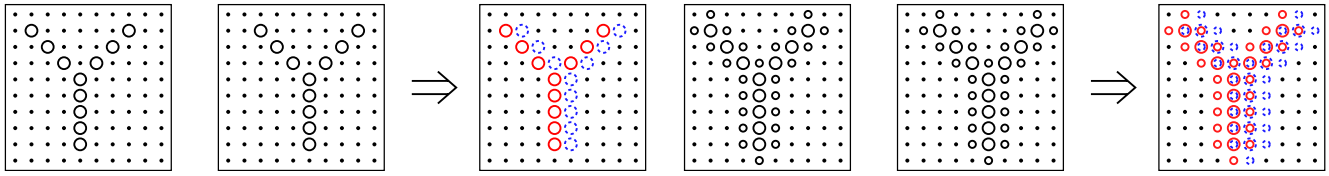


Fig. 8. Connections from *S*- to *C*-cells: one-dimensional cross section (upper half) and three-dimensional view (lower half). Two different interpretations of the same function of *C*-cells: tolerating shift (left) and spatial blur (right).
Source: (modified from Fukushima (1989)).



(a) There is no similarity (no overlap) between two patterns when they are not blurred.

(b) There is a large similarity (overlap) between two patterns after blurring operation.

Fig. 9. Similarity between patterns, which is measured with the degree of overlap, is largely increased by the blurring operation.
Source: (modified from Fukushima (1989)).

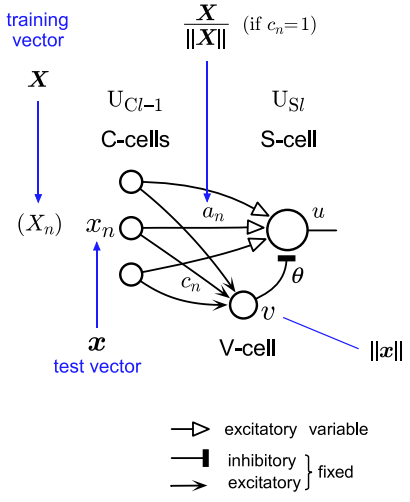


Fig. 10. Input connections converging to an *S*-cell. This figure shows the case of subtractive inhibition. To help intuitive understanding, the vector notation of the value of a_n shows the case of $c_n = 1$.

from the *C*-cell. We use vector notation $\boldsymbol{x} = (x_1, x_2, \ldots, x_n, \ldots)$ to represent these input signals. We sometimes call $\boldsymbol{x}$ the test vector.

The *S*-cell also receives an inhibitory signal through a *V*-cell, which accompanies the *S*-cell. The *V*-cell receives fixed excitatory connections from the same group of *C*-cells as does the *S*-cell, and always responds with the average intensity of the output of the *C*-cells. Let c_n be the strength of the input connection from the *n*th *C*-cell to the *V*-cell. As shown in (3) below, the average is taken, not by an arithmetic mean, but by a root-mean-square (*L*2-norm).

We define *weighted* inner product of two arbitrary vectors $\boldsymbol{y}$ and $\boldsymbol{z}$ by

$$(\boldsymbol{y}, \boldsymbol{z}) = \sum_n c_n y_n z_n, \tag{1}$$

where the strength of the input connections to the *V*-cell, c_n, is used as the weight for the inner product. We also define the norm of an arbitrary vector $\boldsymbol{y}$, using the *weighted* inner product, by $\|\boldsymbol{y}\| = \sqrt{(\boldsymbol{y}, \boldsymbol{y})}$. Incidentally, if $c_n = 1$, (1) represents a conventional inner product.

Input connections to the *S*-cell are determined by learning, where the response of the presynaptic *C*-cells works as a training stimulus. Let $\boldsymbol{X} = (X_1, X_2, \ldots, X_n, \ldots)$ be the training stimulus (training vector) that the *S*-cell has learned. Then the connection a_n is given by $a_n = c_n X_n / \|\boldsymbol{X}\|$.

The training stimulus could be either the response of the presynaptic *C*-cells at the moment when the *S*-cell is generated, or a linear combination of responses of the *C*-cells at several moments, depending on the training methods. Here we first analyze the response of the *S*-cell where $\boldsymbol{X}$ has already been given, and will discuss training methods later.

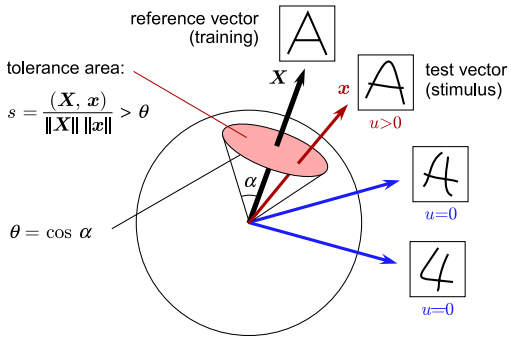


Fig. 11. Tolerance area of an S-cell in the multi-dimensional feature space.

4.1.1. S-cell with subtractive inhibition

Although several types of inhibitory mechanisms are used for S-cells depending on the versions of the neocognitron, here we discuss the case where the inhibitory signals from the V-cells work in a subtractive manner (Fukushima, 2011). (See Section 4.1.2 for other types of inhibition.)

The output u of the S-cell is given by

$$u = \frac{1}{1-\theta} \cdot \varphi \left[\sum_n a_n x_n - \theta v \right]$$
$$= \frac{1}{1-\theta} \cdot \varphi \left[\left(\frac{\boldsymbol{X}}{\|\boldsymbol{X}\|}, \boldsymbol{x} \right) - \theta \|x\| \right]$$
$$= \frac{\|\boldsymbol{x}\|}{1-\theta} \cdot \varphi \left[\frac{(\boldsymbol{X}, \boldsymbol{x})}{\|\boldsymbol{X}\| \cdot \|\boldsymbol{x}\|} - \theta \right], \quad (2)$$

where v is the response of the V-cell:

$$v = \sqrt{\sum_n c_n {x_n}^2} = \|\boldsymbol{x}\| . \quad (3)$$

In (2), $\varphi[\ ]$ is a function defined by $\varphi[x] = \max(x, 0)$. Namely, $\varphi[\ ]$ is a nonlinear function like a half-wave rectifier. The strength of the inhibitory connection is θ, which determines the threshold of the S-cell ($0 \leq \theta < 1$).

We now define similarity s between the training vector $\boldsymbol{X}$ and the stimulus vector $\boldsymbol{x}$ in the multi-dimensional feature space by the following normalized inner product:

$$s = \frac{(\boldsymbol{X}, \boldsymbol{x})}{\|\boldsymbol{X}\| \cdot \|\boldsymbol{x}\|}. \quad (4)$$

Then (2) reduces to

$$u = \|\boldsymbol{x}\| \cdot \frac{\varphi[s-\theta]}{1-\theta}. \quad (5)$$

The S-cell thus yields a non-zero response, only when similarity s is larger than threshold θ.

This situation is illustrated in Fig. 11 for the case of a three-dimensional feature space. The range of similarity values for which $s > \theta$ is called the *tolerance area* of the S-cell. We sometimes call $\boldsymbol{X}$ the *reference vector* of the S-cell. Using a neurophysiological term, we can also express that $\boldsymbol{X}$ is the preferred feature of the S-cell.

Thus the selectivity of the response of the S-cell to a feature that is slightly different from its preferred feature can be controlled by the threshold θ. A higher value of θ produces a smaller tolerance area. If the threshold is low, the radius of the tolerance area becomes large, and the S-cell responds even to features somewhat deformed from its reference vector.

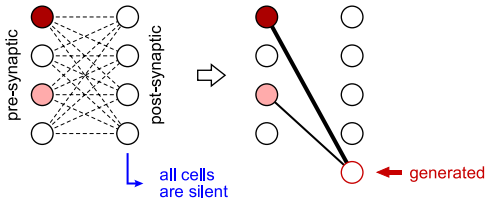


Fig. 12. The add-if-silent rule. A new cell is generated when all postsynaptic cells are silent. In the figure, the response strength of each cell is represented by the depth of the color.
Source: (modified from Fukushima (2011)).

4.1.2. Other types of inhibition

Different from S-cells discussed above, in the original neocognitron (Fukushima, 1980, 2003), the inhibitory signal from the V-cell works in a shunting manner. If the strength of the inhibition is small enough, the shunting inhibition comes to work in a subtractive manner. On the contrary, if the strength of the inhibition is large enough, the shunting inhibition comes to work actually in a divisional manner. In most neocognitrons of previous versions, which used shunting inhibition, parameters were set in such a way that inhibition to S-cells works in the range of divisional inhibition.

When the inhibition from V-cells works in a divisional manner, response of an S-cell is given, not by (5), but by

$$u = \frac{\varphi[s-\theta]}{1-\theta}. \quad (6)$$

It should be noted here that we can have the same tolerance area shown in Fig. 11, regardless of the type of inhibition. This characteristic of S-cells is obtained by the use of root-mean-square operation when calculating the average intensity by the V-cells.

If there is no background noise in the stimulus pattern, the characteristics of (6) is desirable for feature-extracting S-cells, when used, say, for recognizing handwritten characters. The response of an S-cell is determined only by similarity s between the input stimulus $\boldsymbol{x}$ and the reference vector $\boldsymbol{X}$. It is not affected by the strength of the input stimulus $\boldsymbol{x}$. Hence S-cells can extract features robustly without being affected by a gradual non-uniformity in thickness, darkness or contrast in an input pattern. If an stimulus pattern is contaminated by noise, however, interference from the background noise becomes serious.

Under a noisy background, the neocognitron consisting of S-cells with subtractive inhibition can be much more insensitive to interference by noise (Fukushima, 2011). Hence neocognitrons of recent versions use S-cells with subtractive inhibition.

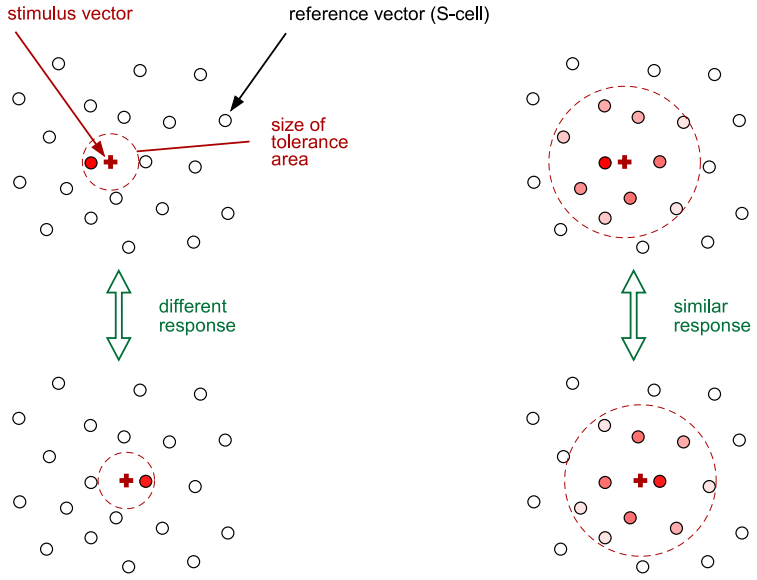
4.2. Add-if-silent rule

Connections to S-cells of intermediate stages of the hierarchical network (U_{S2} and U_{S3} in the network of Fig. 2) are determined by unsupervised learning. Although various methods of training have been used so far for the neocognitron, we first discuss the *add-if-silent* rule, which is used in the newest neocognitron (Fukushima, submitted for publication). We then discuss some other learning rules in 4.4 below.

For the sake of simplicity, we discuss a case where training (or learning) of the network is performed from lower stages to higher stages: after the training of a lower stage has been completely finished, the training of the succeeding stage begins.

During learning, training patterns from a training set are presented one by one to the input layer U_0, and the response of layer U_{Cl-1} works as a training stimulus for layer U_{Sl}.

If all post-synaptic S-cells of U_{Sl} are silent for a training stimulus, as shown in Fig. 12, a new S-cell is generated and added to layer U_{Sl}. Hence we call this learning rule *add-if-silent*. The strength of



(a) High threshold. (b) Low threshold.

Fig. 13. Response of *S*-cells in the multi-dimensional feature space during recognition. The dotted circle around a stimulus vector shows the size of the tolerance areas of *S*-cells. (a) When the threshold of *S*-cells is as high as in the learning phase, only one *S*-cell can responds to a stimulus vector. A slight shift of the stimulus vector produces a completely different response of the layer. (b) When the threshold is low, many *S*-cells respond to a stimulus vector. Even if the stimulus vector shifts a little, most of the *S*-cells keep responding, and the response of the entire layer does not change so much.

the input connections of the generated *S*-cell is determined to be proportional to the response of the pre-synaptic *C*-cells at this moment.

Different from other learning rules shown in Fig. 14 below, however, once the *S*-cell is generated and added to the network, the input connections to the *S*-cell do not change any more.

When applying the add-if-silent rule to the neocognitron, a slight modification is required, because each layer of the neocognitron consists of a number of cell-planes. In a cell-plane, all cells are arranged retinotopically, and share the same set of input connections. This condition of shared connections has to be kept even during learning.

Suppose a training pattern is presented to the input layer during learning. If all *S*-cells, whose receptive fields are located in a certain small area, are silent in spite of non-zero training stimulus, a new *S*-cell is generated and is added to the network. The newly added *S*-cell learns this training stimulus.

In the neocognitron, generation of a new *S*-cell means a generation of a new cell-plane. To keep the condition of shared connections, all *S*-cells in the generated cell-plane are organized so as to have the same input connections as the added *S*-cell. Since the added *S*-cell thus works like a seed in crystal growth, we sometimes call it a *seed-cell*.

After the generation of the new cell-plane, if there still remains any area in which all post-synaptic *S*-cells are silent in spite of non-zero training stimulus, the same process of generating a cell-plane is repeated until the whole area is covered by non-silent *S*-cells.

After that, we proceed to the presentation of the next training pattern.

4.3. Dual threshold for S-cells

The ability to recognize patterns robustly is influenced by the selectivity of feature-extracting *S*-cells, which is controlled by the threshold of the *S*-cells.

For *S*-cells of intermediate stages of the neocognitron, we use *dual threshold*. Namely, we use a lower threshold for *S*-cells during recognition than during learning (Fukushima & Tanigawa, 1996).

As shown in (5) above, an *S*-cell is active, if and only if $s > \theta$. We now represent the distance between two vectors using the angle between them in the multi-dimensional feature space (see also Fig. 11). Then we can express that the *S*-cell is active, only when the distance $\alpha = \cos^{-1} s$ between the reference vector and the training vector is smaller than $\cos^{-1}\theta$.

Under the add-if-silent rule (as well as under other competitive learning rules discussed below), new *S*-cells are generated if, and only if, all *S*-cells are silent. We can express that this learning rule aims to produce a situation where each training vector elicits a response from only one *S*-cell. This means that *S*-cells come to behave like grandmother cells in the layer. If sufficient numbers of training vectors have been presented during learning, *S*-cells come to distribute uniformly in the feature space: the distance α between adjoining *S*-cells (or their reference vectors) is always kept within the range of $\cos^{-1}\theta \leq \alpha < 2\cos^{-1}\theta$ in the feature space.

For recognizing deformed patterns robustly, however, a behavior like grandmother cells is not desirable for *S*-cells of intermediate layers. If the threshold is as high as in the learning phase, each stimulus feature might elicit a response from only one *S*-cell. When the stimulus feature is slightly deformed, the *S*-cell stops responding, and another *S*-cell comes to respond instead of the first one. This decreases the ability of the network to recognize deformed patterns robustly.

Fig. 13 illustrates this situation in the feature space. The dotted circle around a stimulus vector shows the size of the tolerance areas of *S*-cells. In other words, *S*-cells that are located within the dotted circle can respond to the stimulus vector.

Suppose the threshold of *S*-cells is as high as in the learning phase, as illustrated in Fig. 13(a). The size of the tolerance area is small, and only one (or at most a small number of) *S*-cell can responds to a stimulus vector. It should be noted here that, in the learning phase, we have managed to produce this situation, where each training vector elicits a response from only one *S*-cell.

If, as a result of deformation of the input image, the stimulus vector shifts slightly, the *S*-cell stops responding and another *S*-cell comes to respond instead. A slight shift of the stimulus vector thus produces completely different responses, which have no similarity to each other. Hence the feature-extracting cells in the succeeding stage cannot judge that they are similar responses.

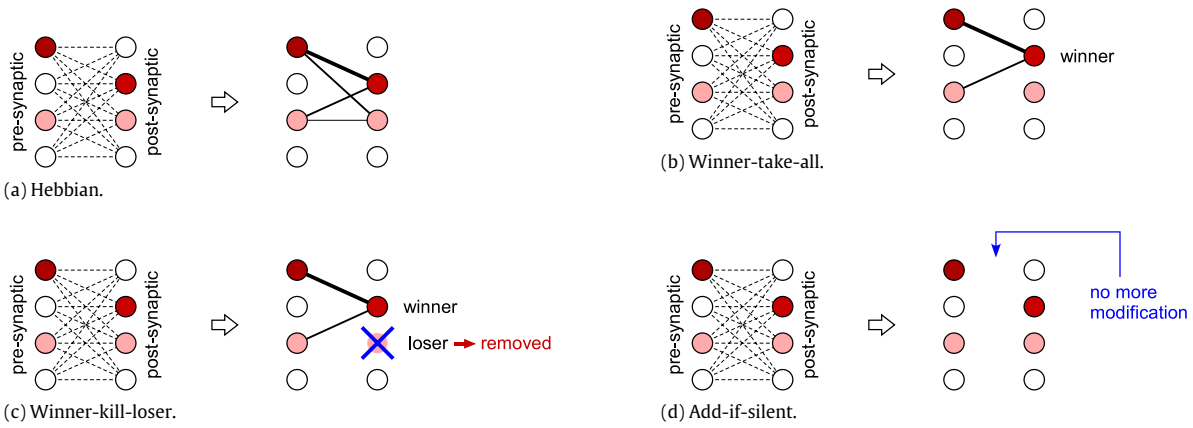


Fig. 14. Several learning rules in comparison with add-if-silent rule. The figure shows how the cells that have already been generated have their input connections modified under different learning rules. In this figure, the response of each cell is represented by the depth of the color.
Source: (modified from Fukushima (2011)).

If the threshold is lowered, however, *S*-cells respond even to features somewhat deformed from their reference vectors. This makes a situation like a population coding of features rather than grandmother-cell theory: many *S*-cells respond to a single feature if the response of an entire layer is observed. In other words, each stimulus elicits non-zero responses from a number of *S*-cells, and these *S*-cells jointly represent the stimulus. Even if the feature is slightly deformed, many of the *S*-cells still keep responding, as shown in Fig. 13(b). Only a small number of *S*-cells change their responses. This situation of lowered threshold in the recognition phase usually endows the network with an ability of generalization and produces a better recognition rate of the neocognitron.

Hence we use the dual threshold of *S*-cells for the learning and the recognition phases (Fukushima & Tanigawa, 1996). In the recognition phase after having finished learning, the threshold of *S*-cells is set to a lower value θ^R than the threshold θ^L for the learning.

Comparing Figs. 9 and 13, we can express that a low threshold of *S*-cells produces a blur in the multi-dimensional feature space, while *C*-cells produce a blur in the retinotopic space.

4.4. Comparison with other learning rules

Several methods, other than add-if-silent, have ever been proposed and are used in the neocognitron to train intermediate layers of *S*-cells. Most of them use unsupervised competitive learning.

The process of generating new *S*-cells for these competitive learning rules is the same as that for the add-if-silent rule. What is different from the add-if-silent rule is the process of modifying input connections to *S*-cells after their generation.

Fig. 14 illustrates and compares several rules for unsupervised learning. The figure shows how the cells that have already been generated have their input connections modified under various learning rules. Different from the add-if-silent rule shown in Fig. 14(d), under all other learning rules shown in Fig. 14(a)–(c), input connections to *S*-cells continue to be modified after their generation throughout the learning phase.

It should be noted here, some other learning rules, such as the one that accepts incremental learning (Fukushima, 2004), have also been proposed but are not shown in Fig. 14.

4.4.1. Winner-take-all rule

Fig. 14(a) shows the Hebbian rule, which is one of the most commonly used learning rules for artificial neural networks (Hebb, 1949). During learning, each synaptic connection is strengthened by an amount proportional to the product of the responses of the pre- and post-synaptic cells.

In the *winner-take-all* rule, shown in Fig. 14(b), post-synaptic cells compete with each other, and the cell from which the largest response is elicited becomes the winner. Different from the Hebbian rule, only the winner can have its input connections renewed. The amount of strengthening of a connection to the winner is proportional to the response of the pre-synaptic cell from which the connection is leading. To keep the condition of shared connections in a cell-plane, the winner takes the place of a seed-cell. As a result, all other cells in the cell-plane follow the winner and come to have the same set of input connections as the winner. Incidentally, most of the conventional neocognitrons use this learning rule (Fukushima, 1980, 2003).

4.4.2. Winner-kill-loser rule

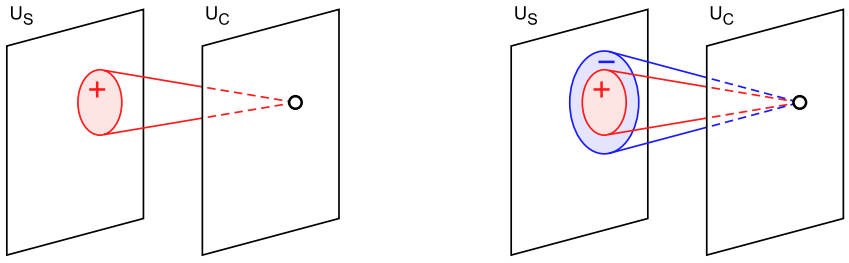
The *winner-kill-loser* rule, shown in Fig. 14(c), resembles the winner-take-all rule in the sense that only the winner learns the training stimulus. In the winner-kill-loser rule, however, not only does the winner learn the training stimulus, but also the losers are simultaneously removed from the network (Fukushima, 2010b; Fukushima, Hayashi, & Léveillé, 2011). Losers are defined as cells whose responses to the training stimulus are smaller than that of the winner, but whose activations are nevertheless greater than zero.

The idea behind this learning rule is as follows. If a training stimulus elicits non-zero responses from two or more *S*-cells, it means that the preferred features (reference vectors) of these cells resemble each other, and that they work redundantly in the network. To reduce this redundancy, only the winner has its input connections renewed to fit more to the training vector, while the other active cells, namely the losers, are removed from the network.

Since silent *S*-cells (namely, the *S*-cells whose responses to the training stimulus are zero) do not join the competition, they are not removed. These cells are expected to work toward extracting other features.

It should be noted here that the constraint for keeping shared connections has to be satisfied also under the winner-kill-loser rule. This means that the constraint is also applied to losers. If there are non-silent cells at the location of the winner, they are losers. The cell-planes, to which losers belong, are removed from the layer.

In the learning phase, a number of training stimuli are presented sequentially to the network. During this process, generation of new cells (cell-planes) and removal of redundant cells (cell-planes) occurs repeatedly in the network. Similarly to the case under the add-if-silent rule, new cells are generated to cover areas of the multi-dimensional feature space that were not previously covered by existing cells. In the areas where similar cells exist in duplicate, redundant cells are removed. By repeating this process



(a) No inhibitory surround. (b) With inhibitory surround.

Fig. 15. Inhibitory surround in the connections to *C*-cells.

for a long enough time, the preferred features (reference vectors) of *S*-cells gradually become distributed uniformly over the multi-dimensional feature space.

The winner-kill-loser rule can train neocognitrons more efficiently than the winner-take-all rule, and an ability to robustly recognize patterns can be obtained with a smaller scale of the network.

4.5. Merits of add-if-silent rule

Under the add-if-silent rule, shown in Fig. 14(d), input connections to a cell are created only at the moment when the cell is generated. Once the cell has been generated and added to the network, their input connections are not modified any more afterward, whatever training stimuli are given to the network.

One of the largest merits of the add-if-silent rule is the simplicity of its algorithm. As a result, computational cost during learning can be made much smaller than other learning rules: once an *S*-cell is generated and added to the network, we need not train it anymore. When designing a neocognitron, we can remove several parameters that control the training after generation of *S*-cells. This makes designing neocognitrons much easier.

Another advantage of the add-if-silent rule is that the process of learning progresses more stably. With the winner-kill-loser rule (Fukushima, 2010b; Fukushima et al., 2011), which produces a higher recognition rate than the winner-take-all rule, the number of cell-planes continues to increase and decrease throughout the learning period and does not completely stabilize. With the add-if-silent rule, the number of cell-planes just increases monotonically, and the increase stops when the multidimensional feature space has been covered with reference vectors.

By the use of the add-if-silent rule, the learning algorithm can thus be simplified, and the computational cost can be reduced. Nonetheless, we can have a slightly higher recognition rate with a slightly smaller scale of the network (Fukushima, submitted for publication).

We now discuss why a good recognition rate can be obtained with a simpler algorithm of the add-if-silent rule.

The process of generating new cells is the same, both under the add-if-silent rule and under the conventional competitive learning rules (winner-take-all and winner-kill-loser). What is different from them is the learning process performed after the generation of individual cells. Under the add-if-silent rule, once an *S*-cell is generated and added to the network, the input connections to the *S*-cell need not change any more.

On the contrary, under the conventional competitive learning rules, an *S*-cell continues to learn training stimuli that are presented after the generation of the *S*-cell. This means that *S*-cells keep learning so as make their reference vectors fit to the training vectors more accurately.

The final classification of input patterns, however, is not made by an intermediate layer, but by the highest stage of the network. The role of the intermediate layer is to represent an input pattern accurately, not by the response of a single cell, but by the population coding, where the stimulus pattern given to the input layer is represented by the response of a number of cells. In the case of population coding, best-fitting of individual cells to training stimuli is not necessarily important. It is enough if the input stimulus can be accurately represented by the response of the population of the whole cells in the layer.

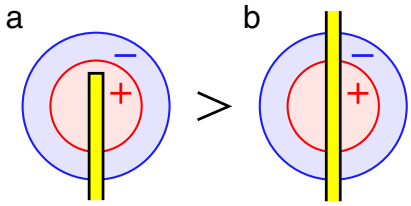


Fig. 16. Inhibitory surround in the connection to a *C*-cell produces a response like an end-stopped cell. Stimulus (a) elicits a larger response than (b).
Source: (modified from Fukushima (2003)).

5. *C*-cells

5.1. Averaging by *C*-cells

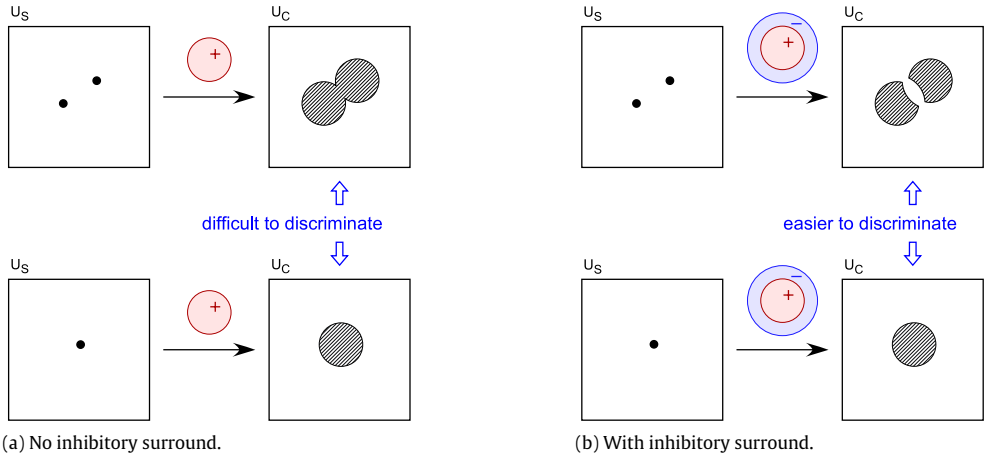
As discussed in Section 3, a *C*-cell has fixed excitatory connections from a group of *S*-cells of the corresponding cell-planes of *S*-cells. Through these connections, each *C*-cell averages the responses of *S*-cells whose receptive field locations are slightly deviated. In other words, *S*-cells' response is spatially blurred in the succeeding cell-planes of *C*-cells. Incidentally, some people call this averaging operation spatial pooling. The averaging operation is important, not only for endowing neural networks with an ability to recognize deformed patterns robustly, but also for smoothing additive random noise contained in the responses of *S*-cells.

In the neocognitron of old versions, the averaging is performed by arithmetic mean. In the neocognitron of recent versions, it is performed by root-mean-square (Fukushima, 2011). Averaging by root-mean-square can reduce the fluctuation in the response of *C*-cells caused by spatial shift of a stimulus feature. As a result, it produces a slightly better recognition rate. (However, the difference between the two averaging methods is not so large).

5.2. Inhibitory surround in the connections to *C*-cells

As mentioned above, the response of an *S*-cell layer U_{Sl} is spatially blurred in the succeeding *C*-cell layer U_{Cl}. In the original neocognitron, the input connections to a *C*-cell consists of only excitatory components of a circular spatial distribution as shown in Fig. 15(a).

Introduction of an inhibitory surround around the excitatory connections as shown in Fig. 15(b) increases the recognition rate of the neocognitron. The concentric inhibitory surround endows the *C*-cells with the characteristics like end-stopped cells, and *C*-cells behave like hypercomplex cells in the visual cortex (Fig. 16). In other words, an end of a line elicits a larger response from a



(a) No inhibitory surround. (b) With inhibitory surround.

Fig. 17. The inhibitory surround in the connections to C-cells increases separation of the blurred responses produced by two independent features.
Source: (modified from Fukushima (2003)).

C-cell than a middle point of the line. Bend points and end points of lines are important features for pattern recognition. C-cells, whose input connections have inhibitory surrounds, thus participate in extraction of bend points and end points of lines while they are making a blurring operation. Incidentally, this kind of response is observed neurophysiologically in neurons of the primary visual cortex (e.g., Walker, Ohzawa, and Freeman (2000)). The inhibitory surround is effective especially for C-cells of lower stages (for U_{C1} and U_{C2}, but not for U_{C3}).

The inhibitory surrounds in the connections also have another benefit. The blurring operation by C-cells, which usually is effective for improving robustness against deformation of input patterns, sometimes makes it difficult to detect whether a lump of blurred response is generated by a single feature or by two independent features of the same kind (Fig. 17(a)). For example, a single line and a pair of parallel lines of a very narrow separation generate a similar response when they are blurred. The inhibitory surround in the connections to C-cells creates a non-responding zone between the two lumps of blurred responses (Fig. 17(b)). This silent zone makes the S-cells of the next stage easily detect the number of original features even after blurring.

6. The highest stage

S-cells at the highest stage (U_{SL}; $L = 4$ in the network of Fig. 2) are trained by supervised learning using labeled training data. When each cell-plane is generated and learns a training pattern at first, the class name of the training pattern is assigned to the cell-plane. As the network learns varieties of deformed training patterns, more than one cell-plane per class is usually generated in U_{SL}. In the recognition phase, the response of U_{SL} is analyzed to classify input patterns. C-cells at the highest stage (U_{CL}) shows the inferred label.

Several methods have been used for analyzing the response of U_{SL} to classify input patterns. Depending on the classification methods used in the recognition phase, different training methods are used in the learning phase. In most neocognitrons of old versions, the *supervised winner-take-all* is used for learning, and the winner-take-all is used for recognition (Fukushima, 2003). In the neocognitron of the newest version, however, the method of *supervised interpolating-vector* is used for learning, and the *interpolating-vector* is used for recognition (Fukushima, submitted for publication).

Although the method of interpolating-vector produces a better recognition rate with a smaller scale of the network, we first discuss the method of supervised winner-take-all, whose algorithm is simpler.

In both learning and recognition phases, the response of C-cells of U_{CL-1} becomes input signals to S-cells of layer U_{SL}. Similarly to the case of intermediate stages, we use vector notation $\boldsymbol{x}$ to represent these input signals. During learning, $\boldsymbol{x}$ works as a training vector. Each training vector has a label indicating the class to which the vector belongs. During recognition, $\boldsymbol{x}$ becomes a test vector, which is to be recognized.

6.1. Supervised winner-take-all

The neocognitron of old versions uses the *supervised winner-take-all* for training S-cells of the highest stage (Fukushima, 2003). The learning rule resembles the competitive learning used to train S-cells of intermediate stages, but the class names of the training patterns are also used for the learning. When a cell-plane is generated and learns a training pattern at first, a label indicating the class name of the training pattern is assigned to the cell-plane.

Every time a training pattern is presented during learning, competition occurs among all S-cells in the layer. If the winner of the competition has the same label as the training pattern, the winner becomes the seed-cell and learns the training pattern. However, if the winner has a wrong label (or if all S-cells are silent), a new cell-plane is generated.

During recognition, the label of the maximally activated S-cell in U_{SL} determines the final result of recognition. The C-cells of U_{CL} show the inferred label of the input stimulus.

Different from intermediate layers, in most neocognitrons of recent versions, the threshold value of S-cells of the highest stage is zero for both recognition and learning phases. Hence the process of finding the largest-output S-cell is equivalent to the process of finding the nearest reference vector in the multi-dimensional feature space. Each reference vector has its own territory determined by the Voronoi partition of the feature space (Fig. 18). The recognition process in the highest stage resembles the vector quantization (Gray, 1984; Kohonen, 1990) in this sense.

During learning, training vectors that are misclassified usually come from near class borders in the feature space. If a particular training vector is misclassified in the learning, the reference vector of the winner, which caused a wrong recognition for this training vector, is not renewed this time. A new cell-plane is generated instead, and the misclassified training vector is adopted as the reference vector of the new cell-plane. Generation of a new reference vector causes a shift of decision borders in the feature space, and some of the training vectors, which have been recognized correctly before, are now misclassified and additional reference vectors have to be generated again to readjust the borders. Thus, the decision borders are gradually adjusted to fit the real borders between classes. During this learning process, reference vectors come

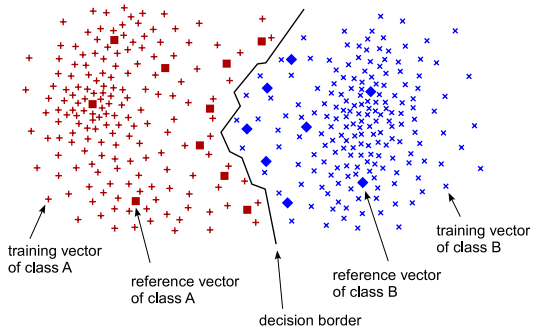


Fig. 18. A schematic illustration of the distribution of training and reference vectors in the multi-dimensional feature space, when supervised winner-take-all is used for training U_{SL}.
Source: (modified from Fukushima (2007)).

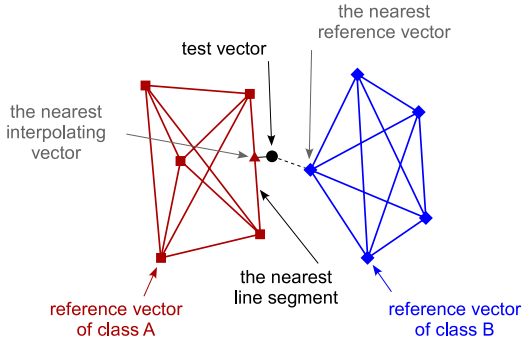


Fig. 19. Recognition by the method of interpolating-vector. The label of the nearest line segment, instead of the nearest reference vector, shows the result of pattern recognition.

to distribute more densely near the class borders, but their density remains low in the locations distant from class borders. Thus a small set of reference vectors (S-cells) can represent the larger set of training vectors.

6.2. Interpolating-vector

The method of *interpolating-vector* is another rule for analyzing the response of S-cells of the highest stage (U_{SL}) to classify input patterns (Fukushima, 2007). It usually produces a higher recognition rate than the supervised winner-take-all.

6.2.1. Classification by interpolating-vector

Before explaining the method of creating reference vectors, we first discuss the recognition phase, where labeled reference vectors have already been produced.

The basic idea of the method of interpolating-vector is as follows. We assume a situation where virtual vectors, which are named *interpolating vectors*, are densely placed along the line segments connecting every pair of reference vectors of the same label. From these interpolating vectors, we choose the one that has the largest similarity to the test vector $\boldsymbol{x}$. The label (or the class name) of the chosen vector is taken as the result of pattern recognition.

Actually, we do not need to generate infinite numbers of interpolating vectors. We just assume line segments connecting every pair of reference vectors of the same label. The line segments are assigned the same labels as the reference vectors on both sides. We then measure distances (based on similarity) to these line segments from the test vector, and choose the nearest line segment (Fig. 19). The label of the line segment, instead of the nearest reference vector, shows the result of pattern recognition.

Mathematically, this process can be expressed as follows. Let $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ be two reference vectors of the same label. An interpolating vector $\boldsymbol{\xi}$ for this pair of reference vectors is given by a linear combination of them:

$$\boldsymbol{\xi} = p_i \frac{\boldsymbol{X}_i}{\|\boldsymbol{X}_i\|} + p_j \frac{\boldsymbol{X}_j}{\|\boldsymbol{X}_j\|}, \quad (p_i + p_j = 1). \tag{7}$$

Similarity s between the interpolating vector $\boldsymbol{\xi}$ and the test vector $\boldsymbol{x}$ takes a maximum value

$$s_{\max} = \sqrt{\frac{s_i^2 - 2s_i s_j s_{ij} + s_j^2}{1 - s_{ij}^2}} \tag{8}$$

at

$$p_i = \frac{s_i - s_j s_{ij}}{(s_i + s_j)(1 - s_{ij})}, \qquad p_j = \frac{s_j - s_i s_{ij}}{(s_i + s_j)(1 - s_{ij})} \tag{9}$$

where

$$s_i = \frac{(\boldsymbol{X}_i, \boldsymbol{x})}{\|\boldsymbol{X}_i\| \cdot \|\boldsymbol{x}\|}, \qquad s_j = \frac{(\boldsymbol{X}_j, \boldsymbol{x})}{\|\boldsymbol{X}_j\| \cdot \|\boldsymbol{x}\|},$$
$$s_{ij} = \frac{(\boldsymbol{X}_i, \boldsymbol{X}_j)}{\|\boldsymbol{X}_i\| \cdot \|\boldsymbol{X}_j\|}. \tag{10}$$

Since threshold θ of S-cells is always zero in layer U_{SL}, we can see from (5) that s_i and s_j are proportional to the responses of S-cells whose reference vectors are $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$, respectively. To be more specific, the response of the ith and jth S-cells are given by

$$u_i = \|\boldsymbol{x}\| \cdot \frac{(\boldsymbol{X}_i, \boldsymbol{x})}{\|\boldsymbol{X}_i\| \cdot \|\boldsymbol{x}\|} = \|\boldsymbol{x}\| \cdot s_i,$$
$$u_j = \|\boldsymbol{x}\| \cdot \frac{(\boldsymbol{X}_j, \boldsymbol{x})}{\|\boldsymbol{X}_j\| \cdot \|\boldsymbol{x}\|} = \|\boldsymbol{x}\| \cdot s_j. \tag{11}$$

Hence, if we have calculated s_{ij} in advance, we can easily get $\|\boldsymbol{x}\| \cdot s_{\max}$ from the responses of S-cells, u_i and u_j. Since the value of $\|\boldsymbol{x}\|$ is the same for all S-cells that have receptive fields at the same location, we can easily find the pair of S-cells (reference vectors) that produces the maximum similarity $s_{\max}$.

We can interpret that $s_{\max}$ represents similarity between test vector $\boldsymbol{x}$ and the line segment that connects a pair of reference vectors $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ (Fig. 20(a)). Among all line segments that connect every pair of reference vectors of the same label, we choose the one that has the largest similarity to the test vector. The label (or the class name) of the chosen line segment is taken as the result of pattern recognition. (If there exists only one reference vector of a class, similarity between the reference vector and the test vector is taken as $s_{\max}$).

In the neocognitron, each cell-plane of layer U_{SL}, like other layers, contains S-cells that have receptive fields at different locations. For every group of S-cells that have receptive fields at the same location, we calculate $\|\boldsymbol{x}\| \cdot s_{\max}$. We then choose the line segment that yields the largest value of $\|\boldsymbol{x}\| \cdot s_{\max}$ among all locations of the receptive fields. The label of the chosen line segment is taken as the final result of pattern recognition by the neocognitron.

6.2.2. Creating reference vectors

Every time a training vector $\boldsymbol{x}$ is presented during learning, we first try to classify it using the method of interpolating-vector.

If the result of the classification is wrong, or if all S-cells are silent, the training vector $\boldsymbol{x}$ is adopted as a new reference vector and is assigned a label of the class name. When applying this process to layer U_{SL} of the neocognitron, the training vector $\boldsymbol{x}$ is chosen from the retinotopic location, at which the intensity of the training vector (weighted sum of the elements of vector $\boldsymbol{x}$) is the largest.

If the result of the classification is correct, we choose the line segment that shows the largest similarity to the training vector $\boldsymbol{x}$. The two reference vectors, $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$, on both sides of the line segment learn the training vector $\boldsymbol{x}$ (Fig. 20(b)). The amounts of

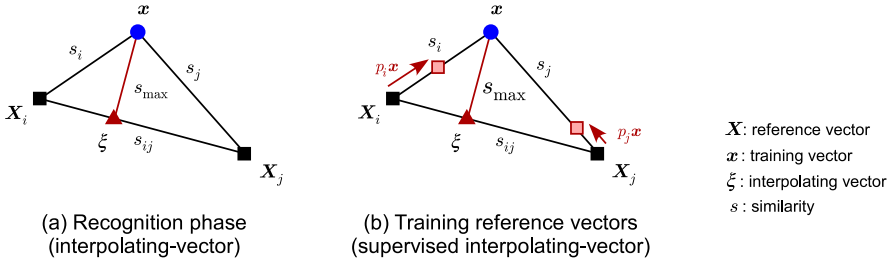


Fig. 20. The method of interpolating-vector.

increase of $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ are $p_i\boldsymbol{x}$ and $p_j\boldsymbol{x}$, respectively, where p_i and p_j are given by (9).

We call this training method *supervised interpolating-vector*. Computer simulation shows that the supervised interpolating-vector, if combined with the interpolating-vector for recognition, produces a high recognition rate with a smaller scale of the network than the supervised winner-take-all method (Fukushima, submitted for publication).

7. Networks extended from the neocognitron

Various extensions and modifications of the neocognitron have been proposed to endow it with further abilities or to make it more biologically plausible.

7.1. Selective attention model

7.1.1. Outline of the selective attention model

Although the neocognitron has considerable ability to recognize deformed patterns, it does not always recognize patterns correctly when two or more patterns are presented simultaneously. The *selective attention model* has been proposed to eliminate these defects (Fukushima, 1987). In the selective attention model, top-down (i.e., backward) connections were added to the neocognitron-type network, which had only bottom-up (i.e., forward) connections.

When a composite stimulus, consisting of two patterns or more, is presented, the model focuses its attention selectively to one of the patterns, segments it from the rest, and recognizes it. After the identification of the first segment, the model switches its attention to recognize another pattern. The model also has the function of associative recall. Even if noise or defects affect the stimulus pattern, the model can recognize it and recall the complete pattern from which the noise has been eliminated and defects corrected. These functions can be successfully performed even for deformed versions of training patterns, which have not been presented during learning.

This model has some similarity to the ART model (Carpenter & Grossberg, 1987), but the most important difference between the two is the fact that the selective attention model has the ability to accept patterns deformed in shape and shifted in location. With the selective attention model, not only recognition of patterns, but also the filling-in process for defective parts of imperfect input patterns works on the deformed and shifted patterns themselves. The selective attention model can repair a deformed pattern without changing its basic shape and its location. The deformed patterns themselves can thus be repaired at their original locations, preserving their deformation.

7.1.2. Architecture of the selective attention model

We now discuss the architecture of the model in more detail. As illustrated in Fig. 21, cells in the top-down path are arranged making pairs with the cells in the bottom-up path. In the figure, W indicates a layer of cells in the top-down path, while U indicates a layer of cells in the bottom-up path. The top-down connections also make a mirror image with the bottom-up connections. The

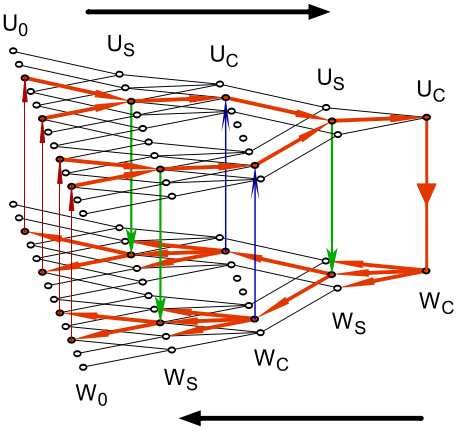


Fig. 21. Interaction of bottom-up and top-down signals in the selective attention model.

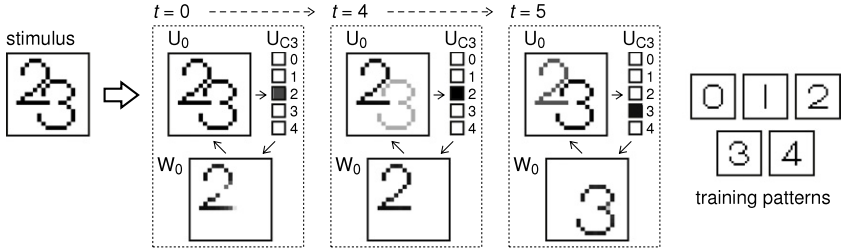
difference between the top-down and bottom-up connections is only in the direction of signal flow.

The bottom-up signals manage the function of pattern recognition, while the top-down signals manage the function of selective attention and associative recall. The output of the highest stage of the bottom-up path is sent back to lower stages through the top-down path and reaches the recall layer at the lowest stage. The bottom-up and top-down signals interact with each other at every stage of the hierarchical network, and the top-down signals are controlled so as to trace the same route as the bottom-up signals.

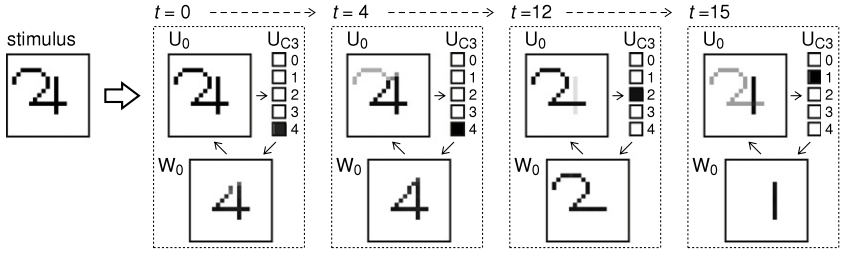
In the bottom-up path, which has the same architecture as the neocognitron, a C-cell receives excitatory connections from a group of S-cells. In a usual operating condition, however, it is only a small number of S-cells that actually send non-zero bottom-up signals. If top-down signals from a W_{Cl}-cell (C-cell in the lth stage of the top-down path) simply flow through strong connections, we have only blurred signals in layer W_{Sl}. To make the top-down signals flow retracing the same route as the bottom-up signals, U_{Sl}-cells send gate signals to corresponding W_{Sl}-cells.

At the same time, the top-down signals, that is, the signals for selective attention, have a facilitating effect on the bottom-up signals by controlling the gain of U_C-cells. When two or more patterns are simultaneously presented to the input layer, a number of cells (recognition cells) might be activated at first in the highest stage of the bottom-up path. However, these recognition cells, except one, stop responding gradually while signals are circulating through the feedback loop because of competition by lateral inhibition. Then only the bottom-up signals relevant to a single pattern are kept flowing by the facilitation from the top-down signals. This means that attention is selectively focused on only one of the patterns in the stimulus.

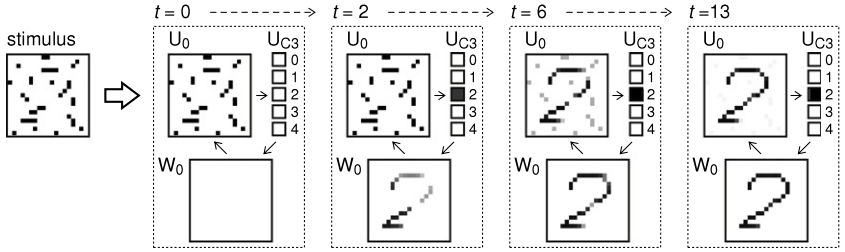
The lowest stage W_0 of the top-down path works as the recall layer, where the output of associative recall and the result of segmentation appear. Guided by the bottom-up signal flow, the top-down signals reach exactly the same locations at which the input pattern is presented. The response of the recall layer W_0 is fed back positively to the input layer U_0.



(a) Each pattern in the stimulus is sequentially recognized (U_{C3}) and segmented (W_0).



(b) From an ambiguous stimulus, several patterns are recognized sequentially. Each pattern is segmented and missing parts are gradually restored in W_0.



(c) From a noisy stimulus, pattern '2' is recognized (U_{C3}) and gradually restored in W_0.

Fig. 22. Some examples of the response of the selective attention model shown in a time sequence. Training patterns that were used to train the network are shown in the right of figure (a).

When some part of the input pattern is missing and a feature which is supposed to exist there fails to be extracted in the bottom-up path, the top-down signal flow is interrupted there and cannot go down any more, because gate signals from the bottom-up path do not come. In such a case, the threshold of U_{SI}-cells around there is automatically lowered, and the U_{SI}-cells try to extract even vague traces of the undetected feature. Incidentally, the fact that a feature has failed to be extracted is detected by the condition that a W_{CI}-cell in the top-down path is active but that the corresponding U_{SI}-cells in the bottom-up path are not. Once a feature is thus extracted in the bottom-up path, the top-down signal now can be transmitted further to lower stages through the route unlocked by the newly activated bottom-up cell. Hence a complete pattern in which defective parts are interpolated emerges in the recall layer. From this pattern, noise and blemishes have been eliminated, because top-down signals are not fed back there.

Fig. 22 shows some examples of the response of the selective attention model in a time sequence. Layer U_{C3} (C-cell layer of the highest stage of the bottom-up path) shows the result of pattern recognition. The segmented and/or restored pattern appears one by one in W_0. Incidentally, training patterns that were used to train the network are shown in the right of Fig. 22(a).

Fig. 22(a) shows the response to a stimulus consisting of two juxtaposed patterns, '2' and '3'. In the recognition layer U_{C3}, the cell corresponding to pattern '2' happens to be activated at first ($t = 0$). This signal is fed back to the recall layer W_0 through a top-down path, but the middle part of the segmented pattern '2' is missing because of interference from the closely adjacent '3'. However, the interference soon decreases and the missing part recovers, because the signals for pattern '3', which is not being attended to, are gradually attenuated without receiving facilitation by gain-control signals ($t = 4$). At $t = 5$, the top-down signal-flow is interrupted for a moment to switch attention. Since the gain-control signals from the top-down cells stop, the bottom-up routes for pattern '2', which have so far been facilitated, now lose their conductivity because of fatigue. The recognition cell for pattern '3' is now activated. Since top-down signals are fed back from this newly activated recognition cell, pattern '3' is segmented and emerges in W_0.

Fig. 22(b) shows how several patterns in an ambiguous stimulus are recognized and segmented sequentially. Pattern '4' is isolated first, pattern '2' next, and finally pattern '1' is extracted. The recalled pattern '4' initially has one part missing ($t = 0$), compared with the training pattern shown in the right of Fig. 22(a). However, the missing part is soon restored ($t = 4$). Each pattern is thus segmented and missing parts are gradually restored in W_0.

Fig. 22(c) shows the response to a greatly deformed pattern with several parts missing and contaminated by noise. Because of the large difference between the stimulus and the training pattern, no response is elicited from the recognition layer U_{C3} at first ($t = 0$). Accordingly, no top-down signal reaches the recall layer W_0. The no-response detector detects this situation, and a threshold-control signal is sent to all feature-extracting cells (U_S-cells) in the network, which makes them respond more easily even to incomplete features. Thus, at time $t = 2$, the recognition cell for '2' is activated in U_{C3}, and top-down signals are fed back from it. In the pattern now sent back to the recall layer W_0, noise has been completely eliminated, and some missing parts have begun to be restored. This partly restored signal, namely the output of the recall layer W_0, is again fed back positively to the input

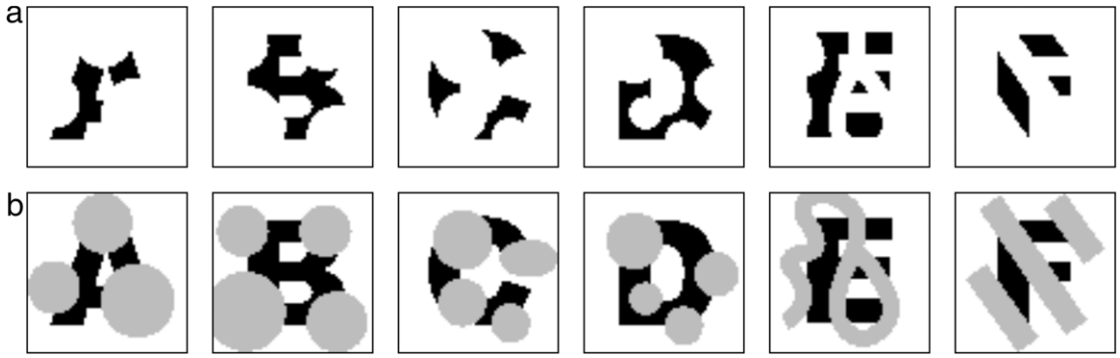


Fig. 23. (a) Patterns partly occluded by invisible masking objects are difficult to recognize. (b) It becomes much easier to recognize when the occluding objects are visible (Fukushima, 2001).

layer U_0. The interpolation continues gradually while the signal circulates through the feedback loop, and finally the missing parts of the stimulus are completely filled in. It should be noted here that the horizontal bar at the bottom of pattern '2' is shorter in the restored pattern than in the training pattern. This means that the length of the bar of the stimulus pattern is kept intact even after restoration. The missing parts are restored quite naturally, where the style of writing of the stimulus pattern is kept as faithful as possible, and only indispensable missing parts are restored.

7.1.3. Application of the selective attention model

The principles of this selective attention model can be extended to be used for several applications, such as recognition and segmentation of connected characters in cursive handwriting of English words (Fukushima & Imagawa, 1993).

We can also design an artificial neural network that recognizes and segments a face and its components (e.g., eyes and mouth) from a complex background (Fukushima & Hashimoto, 1997). It consists of two channels of a selective attention model with different resolutions. The high-resolution channel can analyze input patterns in detail, but usually lacks the ability to get global information because of small receptive fields of the cells in it. On the other hand, the low-resolution channel, whose cells have large receptive fields, can capture global information but only roughly. The network analyzes an object by the interaction of both channels. Even after having learned only a small number of facial front views, the network can recognize and segment faces, eyes and mouths correctly from images containing a variety of faces against complex backgrounds.

7.2. Recognizing and restoring occluded patterns

Human beings are often able to read or recognize a letter or word contaminated by ink stains that partly occlude the letter. If the stains are completely erased and the occluded areas of the letter are changed to white, however, we usually have difficulty in reading the letter, which now has some missing parts. For example, the patterns in Fig. 23(a), in which the occluding objects are not visible, are almost illegible, but the patterns in Fig. 23(b), in which the occluding objects are visible, are much easier to read.

Visual patterns have various local features, such as edges and corners. The visual system of animals extracts these features in its lower stages and tries to recognize a pattern using information of extracted local features. When a pattern is partly occluded, a number of new features, which do not exist in the original pattern, are generated (Fig. 24).

If the occluding objects are visible, the visual system can easily distinguish relevant from irrelevant features, and can ignore irrelevant features. Since the visual system has a large tolerance to partial absence of relevant features, it can recognize the partly occluded patterns correctly, even though some relevant features are missing. The same is true for the neocognitron model.

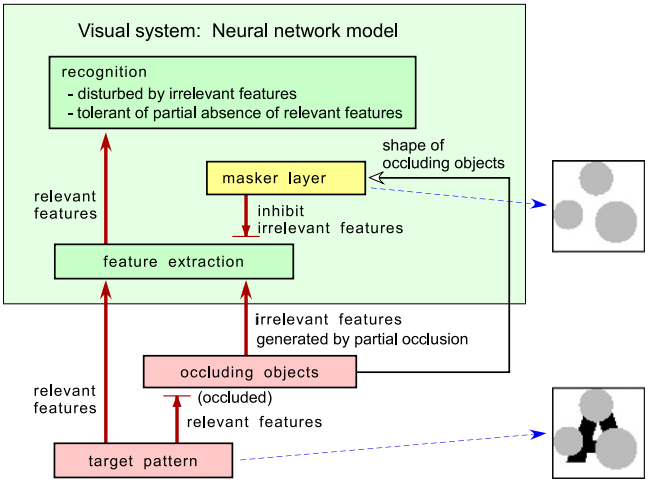


Fig. 24. Process of recognizing an occluded pattern.
Source: (modified from Fukushima (2005b)).

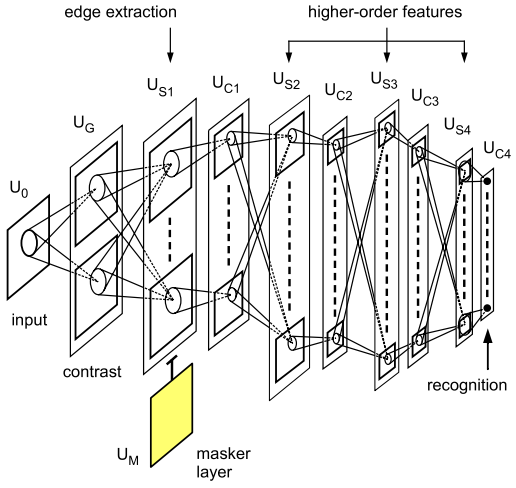


Fig. 25. If a layer U_M, which is called the *masker layer*, is added to a neocognitron, the neocognitron can recognize partly occluded patterns correctly.
Source: (modified from Fukushima (2001)).

On the other hand, if the occluding objects are not visible, the visual system has difficulty in distinguishing which features are relevant to the original pattern, and which are not. These irrelevant features largely disturb the correct recognition by the visual system.

To eliminate irrelevant features, which are usually generated near the contours of the occluding objects, a new layer U_M, named a *masker layer*, is added to a neocognitron as shown in Fig. 25

Fig. 26. Identical patterns are perceived differently by the placement of different gray objects (Fukushima, 2005b).

(Fukushima, 2001). The masker layer detects and responds only to occluding objects. The shape of the occluding objects appears in U_M, in the same shape and at the same location as in the input layer U_0. There are retinotopically ordered and slightly diverging inhibitory connections from layer U_M to all cell-planes of layer U_{S1}. The inhibitory signals from U_M thus suppress the responses to features irrelevant to the occluded pattern. Hence only local features relevant to the occluded pattern are transmitted to higher stages of the network, and the occluded input pattern can be recognized correctly.

Fig. 26 shows another example of stimuli, in which the perception is largely affected by the placement of occluding objects. The black parts of the patterns are actually identical in shape between the left and right figures. We feel as though different black patterns are occluded by gray objects. Namely, we perceive as though pattern '*R*' is occluded in the left figure, while pattern 'B' is occluded in the right. The neocognitron with a masker layer can recognize these patterns correctly as '*R*' and '*B*', like human beings.

By further adding top-down (i.e., backward) connections to the network, for example, the network comes to have an ability, not only to recognize occluded patterns correctly, but also can restore the occluded parts of the patterns (Fukushima, 2005b).

It is reported that, in area $V2$ of monkeys, there are cells that show highly selective responses to a particular angle of bend of line stimuli (Ito & Komatsu, 2004; Pasupathy & Connor, 1999). If cells of this kind, which we call *bend-extracting cells*, are built into U_{S2} of the bottom-up path, the network can acquire an ability to restore occluded contours more smoothly (Fukushima, 2010a). The network shows a function like amodal completion. Using the responses of bend-extracting cells, the network predicts the curvature and location of the occluded contours. Missing contours are gradually extrapolated and interpolated from the visible contours. Fig. 27 shows some examples of the responses of the network. From the images presented to the input layer U_0, occluded contours are gradually completed in layer W_0.

7.3. Other applications

The architecture of cascaded connection of *S*- and *C*-cell layers is useful, not only for recognizing visual patterns, but also for various types of image processing.

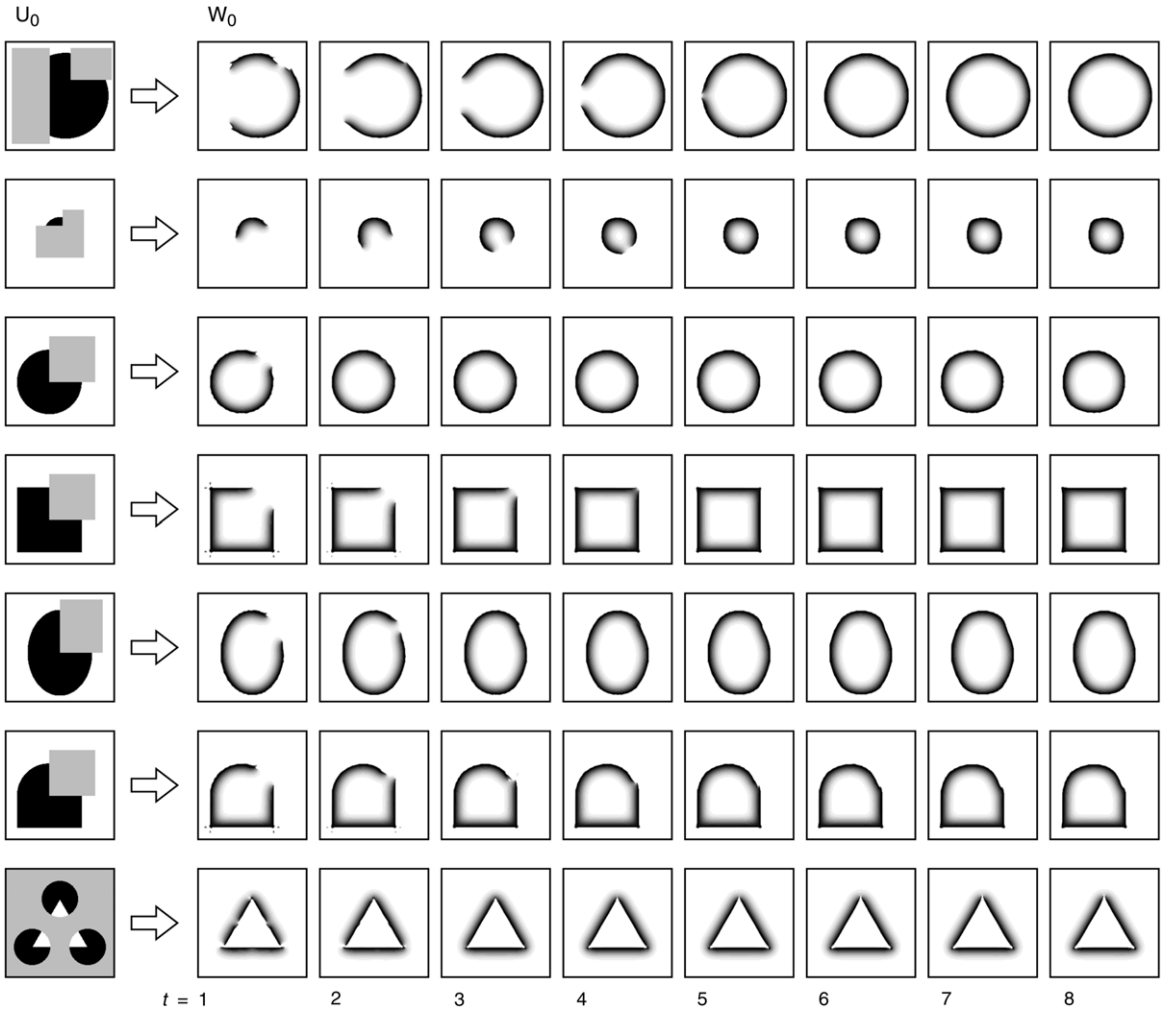


Fig. 27. Progress of amodal completion. From the images presented to input layer U_0, occluded contours are gradually completed in layer W_0. Time t shows the number of circulation of signals through the feedback loop, which is composed by the bottom-up and top-down paths.
Source: (modified from Fukushima (2010a)).

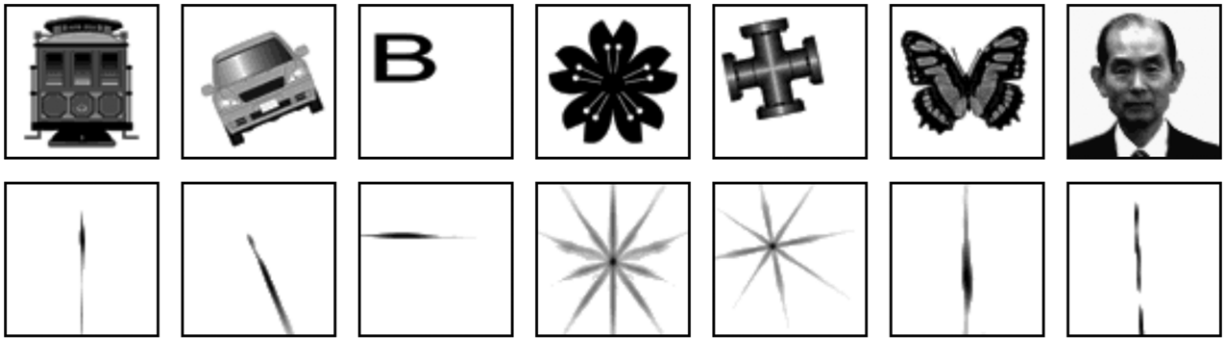


Fig. 28. Symmetry axis extraction by a network of multi-resolution channels of cascaded connection of *S*- and *C*-cell layers. Symmetry axes (shown in the bottom) are extracted correctly even from complicated figures (shown in the top), where a small amount of asymmetry can be tolerated.
Source: (modified from Fukushima (2005a)).

For example, by introducing non-uniform blur (namely, multi-resolution channels) into cascaded connections of *S*- and *C*-cell layers, we can construct a network that extracts symmetry axes of visual patterns (Fukushima, 2005a). Fig. 28 shows some examples of the response of the network. We can see that symmetry axes are extracted correctly, where a small amount of asymmetry can be tolerated. Even if a pattern has a number of symmetry axes, all of them are extracted.

8. Discussion

Referring to the history of the neocognitron, this paper has discussed recent advances in the neocognitron. Various extensions and modifications of the neocognitron have also been proposed to endow it with further abilities. Powerful abilities of the neocognitron and related networks are acquired from the architecture of cascaded connection of *S*-and *C*-cell layers in a hierarchical network. This architecture is useful, not only for recognizing visual patterns robustly, but also for various types of image processing.

S-cells work as feature-extracting cells. *C*-cells are inserted in the network to allow for positional errors in the features of the stimulus. The response of a *C*-cell is less sensitive to a shift in retinotopic location of the input pattern. We can express that *C*-cells make a blurring operation, because the response of a layer of *S*-cells is spatially blurred in the response of the succeeding layer of *C*-cells.

The blurring operation is essential for endowing the network with an ability to recognize or process visual patterns robustly. In the neocognitron-type networks, blur is produced, not only by *C*-cells, but also by *S*-cells, which have a low threshold for extracting features. A low threshold of *S*-cells produces a blur in the multi-dimensional feature space, while *C*-cells produce a blur in the retinotopic space.

Although the neocognitron has a long history, modifications of the network to improve its performance are still going on.

Acknowledgment

This work was partially supported from Kansai University by the Strategic Project to Support the Formation of Research Bases at Private Universities: Matching Fund Subsidy from MEXT, 2008–2012.

References

Bruce, C. J., Desimone, R., & Gross, C. G. (1981). Visual properties of neurons in a polysensory area in superior temporal sulcus of the macaque. *Journal of Neurophysiology*, *46*, 369–384.

Carpenter, G. A., & Grossberg, S. (1987). ART 2: self-organization of stable category recognition codes for analog input patterns. *Applied Optics*, *26*, 4919–4930.

Desimone, R., & Schein, S. J. (1987). Visual properties of neurons in area *V*4 of the macaque: sensitivity to stimulus form. *Journal of Neurophysiology*, *57*(3), 835–868.

Fujita, I., Tanaka, K., Ito, M., & Cheng, K. (1992). Columns for visual features of objects in monkey inferotemporal cortex. *Nature*, *360*, 343–346.

Fukushima, K. (1980). Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, *36*(4), 193–202.

Fukushima, K. (1987). Neural network model for selective attention in visual pattern recognition and associative recall. *Applied Optics*, *26*(23), 4985–4992.

Fukushima, K. (1988). A neural network for visual pattern recognition. *IEEE Computer*, *21*(3), 65–75.

Fukushima, K. (1989). Analysis of the process of visual pattern recognition by the neocognitron. *Neural Networks*, *2*(6), 413–420.

Fukushima, K. (2001). Recognition of partly occluded patterns: a neural network model. *Biological Cybernetics*, *84*(4), 251–259.

Fukushima, K. (2003). Neocognitron for handwritten digit recognition. *Neurocomputing*, *51*, 161–180.

Fukushima, K. (2004). Neocognitron capable of incremental learning. *Neural Networks*, *17*(1), 37–46.

Fukushima, K. (2005a). Use of non-uniform spatial blur for image comparison: symmetry axis extraction. *Neural Networks*, *18*(1), 23–32.

Fukushima, K. (2005b). Restoring partly occluded patterns: a neural network model. *Neural Networks*, *18*(1), 33–43.

Fukushima, K. (2007). Interpolating vectors for robust pattern recognition. *Neural Networks*, *20*(8), 904–916.

Fukushima, K. (2010a). Neural network model for completing occluded contours. *Neural Networks*, *23*(4), 465–582.

Fukushima, K. (2010b). Neocognitron trained with winner-kill-loser rule. *Neural Networks*, *23*(7), 926–938.

Fukushima, K. (2011). Increasing robustness against background noise: visual pattern recognition by a neocognitron. *Neural Networks*, *24*(7), 767–778.

Fukushima, K. (2012). Training multi-layered neural network neocognitron. Neural Networks (submitted for publication).

Fukushima, K., & Hashimoto, H. (1997). Recognition and segmentation of components of a face by a multi-resolution neural network. In W. Gerstner, A. Germond, M. Hasler, & J.-D. Nicoud (Eds.), *Lecture notes in computer science, LNCS*: *vol. 1327*. *Artificial neural networks—ICANN 97* (pp. 931–936). Berlin, Heidelberg, New York: Springer-Verlag.

Fukushima, K., Hayashi, I., & Léveillé, J. (2011). Neocognitron trained by winner-kill-loser with triple threshold. In B.-L. Lu, L. Zhang, & J. Kwok (Eds.), *Lecture notes in computer science, LNCS*: *vol. 7063*. *Neural information processing—ICONIP 2011, part II* (pp. 628–637). Berlin, Heidelberg: Springer-Verlag.

Fukushima, K., & Imagawa, T. (1993). Recognition and segmentation of connected characters with selective attention. *Neural Networks*, *6*(1), 33–41.

Fukushima, K., & Tanigawa, M. (1996). Use of different thresholds in learning and recognition. *Neurocomputing*, *11*(1), 1–17.

Gray, R. M. (1984). Vector quantization. *IEEE ASSP Magazine*, *1*(2), 4–29.

Hebb, D. O. (1949). *Organization of behavior*. New York, London, Sydney: John Wiley & Sons.

Hubel, D. H., & Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal of Physiology (London)*, *106*(1), 106–154.

Hubel, D. H., & Wiesel, T. N. (1965). Receptive fields and functional architecture in two nonstriate visual areas (18 and 19) of the cat. *Journal of Neurophysiology*, *28*(2), 229–289.

Ito, M., & Komatsu, H. (2004). Representation of angles embedded within contour stimuli in area *V*2 of macaque monkeys. *Journal of Neuroscience*, *24*(13), 3313–3324.

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, *78*(9), 1464–1480.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Mutch, J., & Lowe, D. G. (2008). Object class recognition and localization using sparse features with limited receptive fields. *International Journal of Computer Vision*, *80*(1), 45–57.

Pasupathy, A., & Connor, C. E. (1999). Responses to contour features in macaque area *V*4. *Journal of Neurophysiology*, *82*, 2490–2502.

Riesenhuber, M., & Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nature Neuroscience*, *2*(11), 1019–1025.

Satoh, S., Kuroiwa, J., Aso, H., & Miyake, S. (1999). Recognition of rotated patterns using a neocognitron. In L. C. Jain, & B. Lazzerini (Eds.), *Knowledge based intelligent techniques in character recognition* (pp. 49–64). CRC Press.

Serre, T., Oliva, A., & Poggio, T. (2007). A feedforward architecture accounts for rapid categorization. *Proceedings of the National Academy of Sciences of the United States of America*, *104*(15), 6424–6429.

von der Hydt, P., Peterhans, E., & Baumgartner, G. (1984). Illusory contours and cortical neuron responses. *Science*, *224*(4654), 1260–1262.

Walker, G. A., Ohzawa, I., & Freeman, R. D. (2000). Suppression outside the classical cortical receptive field. *Visual Neuroscience*, *17*, 369–379.

Yamane, S., Kaji, S., & Kawano, K. (1988). What facial features activate face neurons in the inferotemporal cortex of the monkey? *Experimental Brain Research*, *73*, 209–214.