The CNN Paradigm

Leon O. Chua, *Fellow, IEEE,* and Tamás Roska, *Fellow, IEEE*

***Abstract*— Since its invention in 1988, while retaining the two basic concepts of *local connectedness* and *analog circuit dynamics*, the CNN paradigm has evolved to cover a very broad class of problems and frameworks. In this paper a tutorial summary of the various classes of CNN is presented along with a precise taxonomy. Some examples are presented to demonstrate the inherent richness of this paradigm.**

I. Introduction

MANY complex computational problems can be reformulated naturally as well-defined tasks where the signal values are placed on a regular geometric 2-D or 3-D grid, and the direct interactions between signal values are limited within a finite local neighborhood (sometimes called the receptive field).[1] The invention, called cellular neural network (CNN) [1], [2], is an analog dynamic processor array which reflects just this property: the processing elements interact directly within a finite local neighborhood. Unlike cellular automaton or systolic array, here we have *analog* processors and interactions, and due to dynamic propagation, not only the near neighbors are in interaction. Also differing from general neural networks, our CNN cells (some of them are totally unrelated to the many ideal neuron models) capture the geometric, nonlinear, and/or delay-type properties in the interaction weights.

Since 1988 many researchers have made significant contributions to the CNN paradigm by various generalizations, while retaining two original ideas: cells are analog processors with *continuous* (not binary) *signal values and interactions are local within a finite radius.* Although barely four years since its inception, phenomenal progress on CNN has been made [68]–[71], most of which have been documented in the first two IEEE International Workshops on Cellular Neural Networks and their Applications (CNNA Budapest 1990, Munich 1992), in the Special Issue of the *International Journal of Circuit Theory and Applications* in 1992, in several special sessions at international conferences, and in the first book on this subject to appear in 1993.

Due to their local connectivity, CNN's can be realized as VLSI chips and can operate at a very high speed and complexity: from the first fully tested chip [57] one can extrapolate a 0.3 TeraXPS performance for a 10×10 mm^2 chip using a conservative 2-μm technology. Since the range of dynamics and the connection complexity (connections per cell) are independent of the number of processing elements (cells), the implementation is reliable and robust. The unprecedented speed of the proliferation of the CNN paradigm is partly due to these facts. The relative ease of "programming," i.e., the design of cloning templates, is based on the geometric aspects of these "analog instructions." It turned out that the CNN model is just the paradigm biologists are looking for, namely, a unifying model of many complex neural architectures, especially in the case of various sensory modalities.

In the original paper in 1988, it has been shown that the CNN is also a spatial approximation of a diffusion type partial differential equation. By adding simple circuit elements to the cells it has been shown [7], [63] that some simple wave type equations can be represented as well.

In this paper we give a concise tutorial description of the CNN paradigm along with a precise taxonomy. Section II contains the exact definition of the CNN, and the canonical equations are described in Section III. In Section IV the importance of many independent input signal arrays, adaptive templates, and the multilayer capability is emphasized and motivated by examples. In Section V the CNN paradigm is reviewed, and in Section VI we show how simply the wave-type PDE can be generated. Section VII contains the complete taxonomy. In the Appendix a part of a current CNN software library is also given. Finally, we give a selected list of references. Although it is not complete, it gives a very broad source of information. The table of contents below gives a more detailed outline of the paper.

II. Definition and Remarks

The definition below follows the original one [1], emphasizing the generality of the cell and the interactions, and using "CNN" as a generic term, which could be interpreted as *cellular neural network*, or, in a more general context, as *cellular nonlinear network.*

Definition: The CNN is a

i) 2-, 3-, or n-dimensional **array of**

ii) mainly identical **dynamical systems**, called cells, which satisfies two properties:

iii) most **interactions are local** within a finite radius r, and

iv) all **state variables are continuous valued** signals.

A template specifies the interaction between each cell and all its neighbor cells in terms of their input, state, and output variables.

Remarks:

Manuscript received July 17, 1992; revised November 16, 1992, and December 11, 1992.

L. O. Chua is with the Department of Electrical Engineering and Computer Sciences and the Electronics Research Laboratory, University of California, Berkeley, CA 94720.

T. Roska is with the Dual and Neural Computing Systems Laboratory, Computer and Automation Institute of the Hungarian Academy of Sciences, H-1518, Budapest, Hungary.

IEEE Log Number 9208120.

[1] See also U.S. patent 5140670, issued on August 18, 1992.

1057-7122/93$03.00

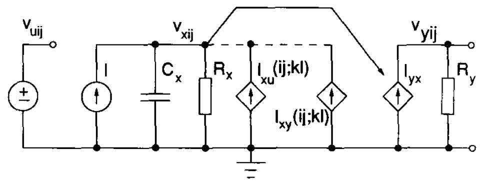


Fig. 1. A simple CNN cell.

1) Each cell is identified by 2, 3 (or n) integers $(i, j, k\ldots, n)$, i.e., the space variable is always discretized.
2) The time variable t may be continuous or discrete.
3) The interconnection effect represented by the cloning template may be a *nonlinear* function of the state $\boldsymbol{x}$, output $\boldsymbol{y}$, and input $\boldsymbol{u}$ of each cell, within the neighborhood N_r of radius r, as well as that of the time t (e.g., time delay or time-varying coefficients). The cloning template is *not* space invariant in general, rather it is a useful special case as it was defined in [1] (the space invariance is not specified in the original constitutive relations).
4) The dynamical system is governed uniquely by an *evolution law* (e.g., ODE, discrete maps, differential-difference equations, functional equations, etc.) such that given $\boldsymbol{x}(t_0)$ and $\boldsymbol{u}(t)$ for all $t \geq t_0$ and given the signals stored in the delay lines at $t = t_0$ (if there is any), $\boldsymbol{x}(t)$ is uniquely determined for all $t \geq t_0$. This includes the boundary conditions.
5) Occasionally, the dynamical system and/or the interconnections may be perturbed by some noise sources of known statistics.

As to the array grid, the cell (or processor) dynamics, the interaction, and the mode of operation, etc., there exist many choices. They will be classified in Section VII.

III. The Main Types of Canonical Equations

Suppose for simplicity that the processing elements are arranged on a 2-D grid (one layer), although this layer can obviously be duplicated to form a multilayer CNN. Without loss of generality, we will follow [1] and write the equations in 2-D. One processing element with a nonlinear template [3] can be seen in Fig. 1. The controlled current sources connected by dotted lines represent the interactions within the neighborhood N_r. Observe that without these interactions, each processing element is just a low-pass amplifier with output saturation.

The dynamics of the array can be described by (setting $C_x = 1$, $R_x = 1$)

$$\dot{v}_{xij}(t) = -v_{xij}(t) + \sum_{kl\in N_r(ij)} \hat{A}_{ij;kl}(v_{ykl}(t), v_{yij}(t)) + \sum_{kl\in N_r(ij)} \hat{B}_{ij;kl}(v_{ukl}(t), v_{uij}(t)) + I_{ij} \quad (1)$$

$$v_{yij} = f(v_{xij}); \quad \text{(in many cases } I_{ij} = I), \quad (2)$$

where ij refers to a grid point associated with a cell on the 2-D grid, and $kl \in N_r(ij)$ is a grid point in the neighborhood within a radius r of the cell ij. $\hat{A}$ and $\hat{B}$ are the nonlinear cloning templates [3].

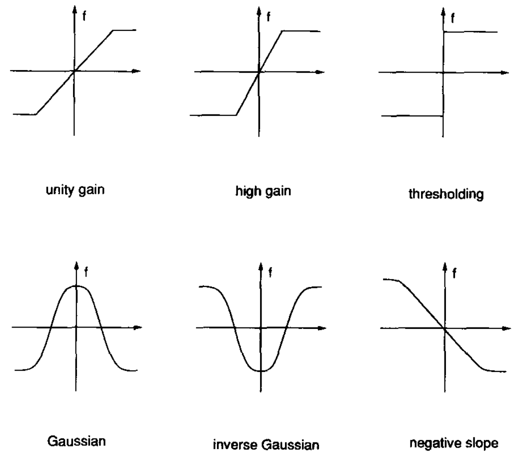


Fig. 2. Some simple useful nonlinear functions in the CNN cells.

Some useful nonlinear output functions f are shown in Fig. 2. In [3, equation (2b′)] we have proposed a dynamic output function in the form of

$$\dot{v}_{xij} = -v_{yij} + f(v_{xij}) \quad (2')$$

as a higher order dynamical system. The simplest case of a dynamical output function consists of a capacitor connected across the cell output.

Delay template elements [3] contribute two additional terms:

$$\sum_{kl\in N_r(ij)} A^{\tau}_{ij;kl} v_{ykl}(t-\tau) + \sum_{kl\in N_r(ij)} B^{\tau}_{ij;kl} v_{ukl}(t-\tau). \quad (3)$$

In many applications, the cloning templates A and B and the threshold current I are translation invariant. Observe that until now, as in the original paper [1], we have not used this restriction (sometimes also called a space-invariant template).

In the case of single variable $\hat{A}$ and $\hat{B}$ functions, the linear (space-invariant) cloning template is represented by the following additive terms [1]

$$\sum_{kl\in N_r(ij)} A_{ij;kl} v_{ykl}(t) + \sum_{kl\in N_r(ij)} B_{ij;kl} v_{ukl}(t). \quad (4)$$

In this case, when the template is space invariant each cell is described by a simple identical cloning template defined by two $(2r + 1) \times (2r + 1)$ real matrices A and B, as well as the constant term I. In addition, as a very special case, if the input and the initial state values are sufficiently small and f is piecewise linear, then the dynamics of the CNN array is linear.

Unlike other standard analog processing arrays, or neural networks, the one-to-one geometric (topographic) correspondence between the processing elements and the processed signal-array elements (e.g., pixels) is of crucial importance. Moreover, the cloning template has geometrical meanings which we can be exploited to provide us with geometric insights and simpler design methods.

Continuous input (output) signal values (e.g., gray scale images) are presented by values in the range [−1, 1]; (e.g.,

-1 is white, $+1$ is black, and gray-scales are in between). Without loss of generality, we can assume

$$| v_{uij}(t) | \leq 1, \qquad | v_{xij}(0) | \leq 1. \tag{5}$$

It has been shown [1], [3] that, if the constraint equations (5) are met, the range of dynamics is bounded by a single number M which can be calculated in terms of the cloning templates:

$$M = \max\{| v_{xij}(t) |\} = \sum_{kl \in N_r(ij)} | \hat{A}_{ij;kl}(v_{ykl}(t), v_{yij}(t)) | + \sum_{kl \in N_r(ij)} | \hat{B}_{ij;kl}(v_{ukl}(t), v_{uij}(t)) | + | I_{ij} | . \tag{5'}$$

Discretizing (1) in time, the numerical solutions of the dynamics of the CNN can be calculated (like in the case of general neural networks). Using [2, equation (11)] with the time step $h = 1$ we get the following equation for the linear template

$$v_{xij}(n+1) = \sum_{kl \in N_r(ij)} A_{ij;kl} v_{ykl}(n) + I_{ij} \tag{6}$$

where $v_{ykl}(n) = f(v_{xkl}(n))$ can be any convenient nonlinear function (e.g., $f(x) = \mathrm{sgn}(x)$ [6]) and I_{ij} contains both I and the feedforward terms. Equation (6) is the *discrete-time CNN equation.* One desirable feature of this class of CNN is that switched-capacitor analog circuits can be used and that digital hardware accelerator boards [21] can be utilized for exact simulation.

Noise current sources can be inserted in parallel with the input, state, or output port, of each cell; even they can be added to the cloning template atoms. Inserting an additive noise current source to the cell state capacitor we get the following equation:

$$\dot{v}_{xij}(t) = -v_{xij}(t) + \sum_{kl \in N_r(ij)} \hat{A}_{ij;kl}(v_{ykl}(t), v_{yij}(t)) + \sum_{kl \in N_r(ij)} \hat{B}_{ij;kl}(v_{ukl}(t), v_{uij}(t)) + I_{ij} + \eta_{ij}(t) \tag{7}$$

where $\eta_{ij}(t)$ is a noise source. This model has been found to be extremely useful in some biological learning processes [74].

If the cell dynamics are described by a multivariable dynamical system the canonical equations can be modified accordingly (see, e.g., [41] in the linear case).

IV. Independent Input Arrays, Adaptive Templates and the Use of Different Layers

To exploit the full capability of the CNN, the possibility of applying 2 or 3 independent inputs cannot be overemphasized. In array processing applications, two independent input signal arrays $S_1(ij)$, $S_2(ij)$ can be mapped onto the CNN as

$$S_1(ij) = v_{uij}(t) \quad \text{and} \quad S_2(ij) = v_{xij}(0)$$

while the output signal array $S_o(ij)$ is associated with v_{yij}. In many important applications these signal arrays are real images or geometric maps of other signals (e.g., thermographic images [40]) and the associated CNN can be used for solving complex processing tasks. The generic input can be time-varying (continuous and discrete-time), but the initial state usually changed in sampled mode only, after the transient has settled.

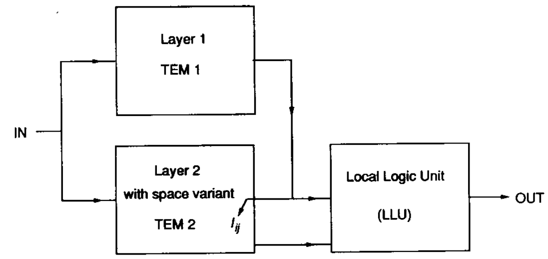


Fig. 3. An example of the use of an adaptive translation variant template.

Moreover, if the constant current term I (bias or threshold current in VLSI implementation) of the cloning template is space-varying (I_{ij}) then we have an additional input signal array. This array, which may be time-varying as well, could play the role of a *spatial geometric program.* This, possibly time-varying, spatial program may be a reference map, or a prescribed route, etc. Since the dynamics of the CNN is very sensitive to this bias term, this provides us with an effective way of programming.

In addition, in a fully transient mode, the time-varying bias term may adapt to the local circumstances calculated dynamically by another template. This is a special case of an adaptive translation-variant CNN.

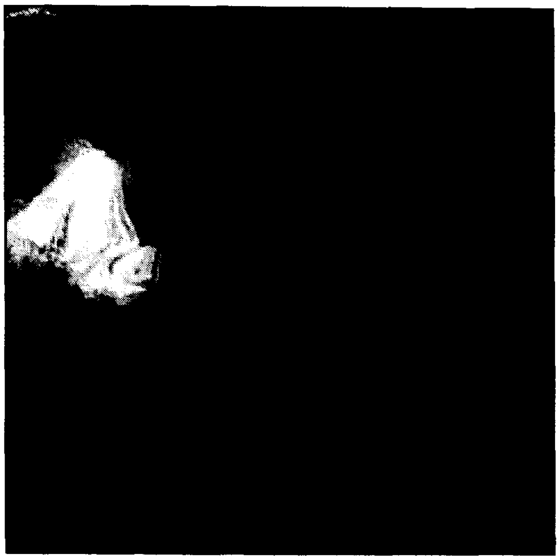
The following example illustrates this generic capability: how to control the template through its bias current term, based on the local properties of the input. A given local property (e.g., average gray-level, or intensity, or texture) is checked by a given template with higher speed and the resulting map is used to control the other template (of lower speed) which is used for checking another property (e.g., contour detection). Even the results of the two templates can be combined logically (if they are black and white).

Example 1: Given a painting (from Rembrandt and Fabritius, Museum of Fine Arts, Budapest) where one part of the painting is much more luminous (the angel) than the other parts. The structure of the CNN algorithm for detecting *contours* from an image is shown in Fig. 3 without the local logic. The input image is shown in Fig. 4(a) and the output images without and with the adaptive template control are shown in Fig. 4(b) and (c), respectively. The two templates performed the following functions.

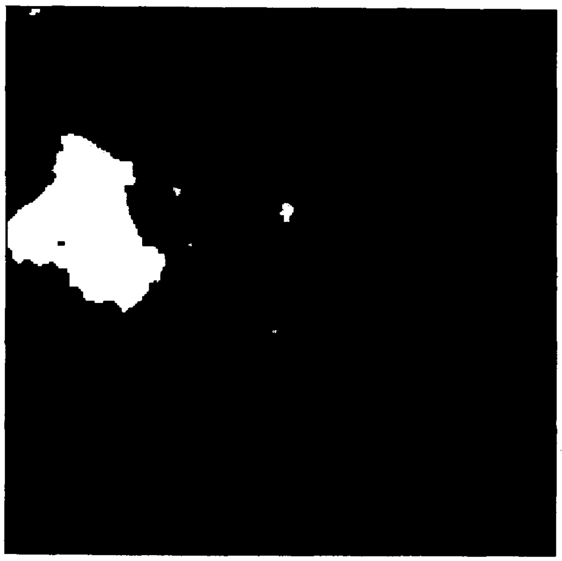
TEM1 calculates the bias map for TEM2 (with initial state equal to the input).

TEM1 ($r = 2$)

$$A = \begin{bmatrix} 0.01 & 0.02 & 0.02 & 0.02 & 0.01 \\ 0.02 & 0.02 & 0.03 & 0.02 & 0.02 \\ 0.02 & 0.03 & 0.04 & 0.03 & 0.02 \\ 0.02 & 0.02 & 0.03 & 0.02 & 0.02 \\ 0.01 & 0.02 & 0.02 & 0.02 & 0.01 \end{bmatrix}$$



(a)



(b)

Fig. 4. Example demonstrating the advantage of using an adaptive template: (a) the gray-scale input, (b) the output without, and (c) with adaptation.

$$B = \begin{bmatrix} 0.01 & 0.02 & 0.02 & 0.02 & 0.01 \\ 0.02 & 0.02 & 0.03 & 0.02 & 0.02 \\ 0.02 & 0.03 & 0.04 & 0.03 & 0.02 \\ 0.02 & 0.02 & 0.03 & 0.02 & 0.02 \\ 0.01 & 0.02 & 0.02 & 0.02 & 0.01 \end{bmatrix} \qquad I = 0.$$

TEM2 ($r = 1$)

$$A = \begin{bmatrix} 0 & 0.5 & 0 \\ 0.5 & 2 & 0.5 \\ 0 & 0.5 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 0.15 & 0 \\ 0.15 & 0.45 & 0.15 \\ 0 & 0.15 & 0 \end{bmatrix}$$

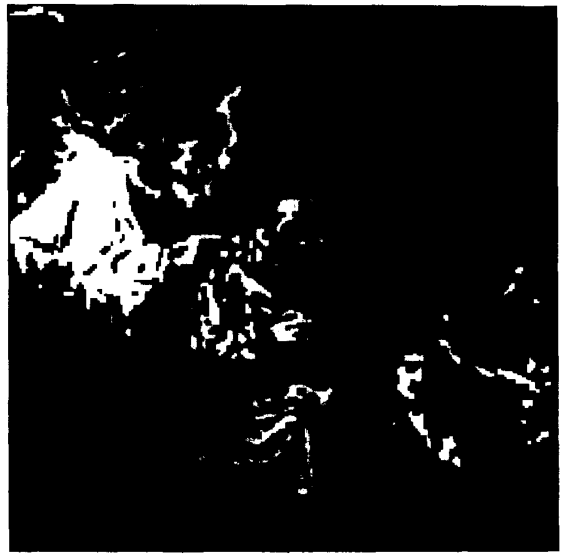


Fig. 4(c)

$$I_{ij} = -v_{yij}, (\text{layer1})$$

with initial states set equal to zero.

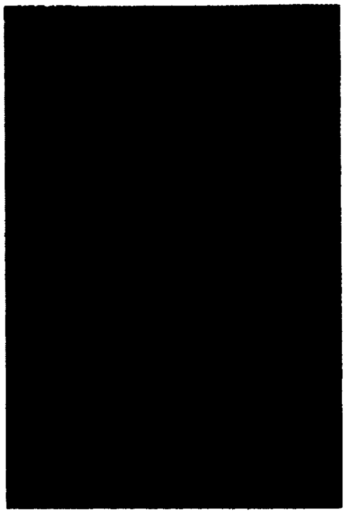
Observe that while contours (edges) can be detected only in one part of the image when only an averaging template is used, the adaptive template-pair TEM1-TEM2 results in a much better detection in the other parts of the picture as well.

The multiple layers of the CNN can be used not only for representing 3-D structures, the 2-D layers may represent different modalities as well. In the case of *color image processing* they may represent different colors. The next example shows briefly this option.

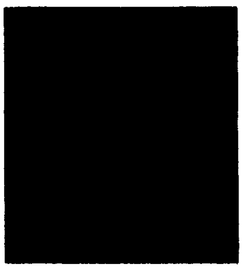
Example 2: The key idea is the following. Each basic color is represented by a CNN layer. For example red, green, and blue (R, G, and B) layers can be used (L_R, L_G, and L_B). The structure of the templates representing the interactions within a neighborhood (receptive field) is determined by modeling the center-surround antagonism between different color-layers. Single opponency means a given cell is excited by a given color input and from the surrounding cells representing another color input the off-center template elements are inhibiting (negative). For example, R_+-G_- means: a cell output in the red layer is increased by the red-cell input and decreased by the green-cells input in the surrounding area within the given neighborhood. Using only feedforward interactions, the template (with $r = 1$) in the L_R layer may be

$$A_R = [0] \qquad B_R = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad I_R = 0$$

(in the center, "red on"), and from the L_G layer, the inhibitory effects are defined by the G_R template, namely, representing the inhibition from the surrounding green cells ("green off"),



(a)



(b)

(c)

(d)

Fig. 5. Single opponency: (R_+-G_-).

(e)

(f)

Fig. 5. (continued)

the interlayer template is

$$A_{GR} = [0] \qquad B_{GR} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & -1 \end{bmatrix} \qquad I_{GR} = 0.$$

(The numerical values are just for illustration.)

As an example, Fig. 5(a) shows a color image [73]. The R, G, and B layers are shown in color in Fig. 5(b), (c), and (d), respectively. Two single opponency (R_+-G_-) and (Y_+-B_-) and gray-scale output is shown in Fig. 5(e). The results of CNN templates acting on the original gray-scale image and on the (R_+-G_-) are shown in Fig. 5(f) and (g). It is quite interesting how many different details can be identified on these last two outputs. Many other surprising results will be reported in a forthcoming paper [72].

V. Key Elements of Programmability

In addition to the variability of the bias term in the cloning template, the additive property of the canonical dynamic equations provides us with a unique capability.

The key circuit element of the whole CNN array is the *cloning template atom* (a single additive interaction term) which is implemented in VLSI technology as a voltage controlled current source (VCCS), transconductance amplifier. Hence, programming is based on this element. The cloning template is the elementary program of the CNN (an instruction or subroutine) and it can be specified by giving prescribed values, and/or characteristics to the template elements (atoms).

The full use of the programmability of CNN is exploited in the CNN Universal Machine [76].

VI. CNN for Solving the Three Basic Types of Partial Differential Equations

In the original paper [1] it was shown how a typical partial differential equation (PDE), the heat equation, can be approximated, on a finite spatial grid, by a CNN with a given simple cell and cloning template. In addition, by adding a capacitor across the output of this simple cell, wave type equations can also be generated [7], [63], [70]. Moreover, at equilibrium, we recover the Laplace equation. Hence we can solve all three basic types of PDS's: the diffusion equation, the Laplace equation, and the wave equation.

The introduction of nonlinear and delay-type templates [3] allows us to generalize the class of functions used in the diffusion equation. Various types of reaction diffusion equations and the autowave equation [44] are important examples of this type of partial differential equations which have been proposed for image processing.

Let us summarize these observations in a 2-D setting. If a planar signal array (a spatio-temporal intensity function) is $I(x, y, t)$, then the nonlinear (reaction) diffusion equation is represented by

$$\partial I/\partial t = D \text{ div } \text{ grad } \{g(I(x, y, t))\} + F(I(x, y, t), L) \quad (8)$$

where g and F represent nonlinear functions, D and L are scalars. F can easily be introduced by adding nonlinear diodes across the capacitors.

As a special case, one obtains the following Laplace equation after the transient has settled.

$$0 = D \text{ div } \text{ grad } \{g(I(x, y, t))\}. \quad (9)$$

The simple wave-type equation

$$\partial^2 I/\partial t^2 = D \text{ div } \text{ grad } \{g(I(x, y, t))\} \quad (10)$$

can also be mapped onto a single layer CNN by using cells augmented by an additional capacitor across the cell output (see (2′)). The mapping of any of these PDE's onto a CNN architecture necessitates a spatial discretization and a numerical integration procedure resulting in a system of ODE's, which can be identified by (1) and (2). This step, which has to be carried out beforehand, is crucial in preserving qualitative properties, such as passivity or stability of the original system in the CNN model.

VII. Glossary of CNN Types, A Taxonomy

Based on the preceding generic definition of CNN, several types of CNN's can be generated. They can be classified according to the types of the **grid**, the **processor (cell)**, the **interaction** (cloning template), and the **mode of operation**. The next table (Table I) shows a summary of this taxonomy.

TABLE I
THE WORLD OF CNN; A TAXONOMY

Grid types	Processor types	Interaction types	Modes of operation
square, hexagonal tridiagonal	linear (or small-signal operation in the piecewise-linear or sigmoid characteristics)	linear of one or two variables, memoryless	continuous-time, discrete-time (synchronous or asynchronous), time-varying
single and multiple grid-size on different layers	sigmoid (including unity gain, high gain, and thresholding)	nonlinear of one, two, or more variables, memoryless	local mode or propagating mode
equidistant and varying grid size (e.g., logarithmic, as in the retina)	Gaussian (inverse Gaussian)	delay-type	fully analog or combined with logic (analogic computing CNN)
planar and circular	first-, second-, and higher order (e.g., one, two or more capacitors)	dynamic (lumped)	transient-, dc steady state, oscillating, or chaotic mode
lattice (3D)	with or without local analog memory	symmetric and nonsymmetric	deterministic or chaotic
	with or without local logic	noiseless or noisy	
	uniform or nonuniform	fixed or programmable template (continuous or in discrete values)	

For almost all cases, useful templates and/or applications have been developed already.

Comments

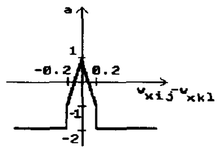
Grid types: Multiple grid-size [3] (e.g., coarse and fine grid) may be useful in artificial systems, not only when mimicking the visual system (magno and parvo cells [75]) but in detection tasks as well. Varying grid size (e.g., logarithmic, like in the retina) may make the CNN design more economical as far as the area-complexity is concerned.

Processor types: Small-signal operation on the linear region is an important special case. Some sigmoid and non-sigmoid nonlinear functions are shown in Fig. 2. Slightly and regularly-varying processor types are useful when less accurate components are used for highly accurate detection tasks (e.g., [59]). Two or three different cell types in a regular grid may be useful, for example in color image processing. The embedding of noise sources is useful in some learning tasks.

Interaction (connection) types: Many interaction types are used. Representing the interactions by cloning templates is a key notion of CNN. Not only translation-invariant templates are used nowadays. Regularly varying templates are becoming useful not only in adaptive time-varying templates, but also in associative memories.

Modes of operation: Besides the usual fixed point mode, the transient, oscillating, chaotic, and stochastic modes are becoming more and more useful.

$$A=\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad B=\begin{bmatrix} a & a & a \\ a & a & a \\ a & a & a \end{bmatrix} \qquad I=\text{threshold}$$



Example: Input and output picture Threshold is -2.9.

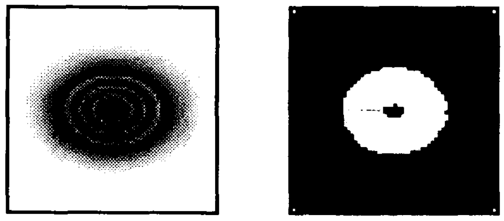


Fig. 6.

$$A=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad B=\begin{bmatrix} -1 & -1 & -0.5 & 0.5 & 1 \\ -1 & -0.5 & 1 & 1 & 0.5 \\ -0.5 & 1 & 5 & 1 & -0.5 \\ 0.5 & 1 & 1 & -0.5 & -1 \\ 1 & 0.5 & -0.5 & -1 & -1 \end{bmatrix} \qquad I=-9$$

Example: Input and output picture.

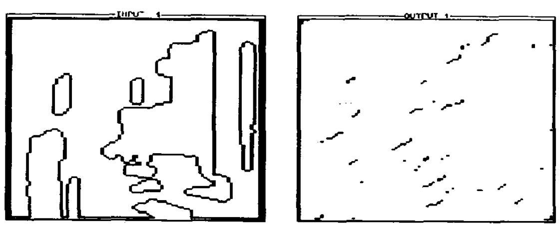


Fig. 7.

VIII. Appendix
Samples from a CNN Template Library

Name: Extreme

Function: Find the locations where the gradient of the field is smaller than a given threshold value, i.e., find the extremities of the picture. See Fig. 6 [40].

Name: DIAG

Function: Detects diagonal lines having an approximately SW-NE direction (like /). See Fig. 7 [46].

Name: Gradient

Function: Find the locations where the gradient of the field is higher than a given threshold value. See Fig. 8 [40].

Name: Movehor

Function: Direction dependent motion detection. Only objects moving horizontally to the left with a speed of 1 pixel/delay time remain on the screen. See Fig. 9 [46].

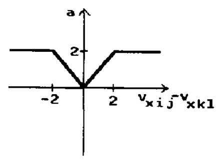
Acknowledgment

The assistance of T. Kozek,Á. Zarándy, and P. Venetiáner is kindly acknowledged.

References

[1] L. O. Chua and L. Yang, "Cellular neural networks: Theory," *IEEE Trans. Circuits Syst.*, vol. 35, pp. 1257–1272, Oct. 1988.

$$\mathbf{A}=\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{B}=\begin{bmatrix} a & a & a \\ a & a & a \\ a & a & a \end{bmatrix} \quad \mathrm{I=threshold}$$



Example: Input and output picture Threshold is -1.8.

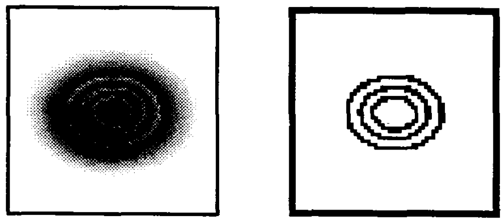


Fig. 8.

$$\mathbf{A}=\begin{bmatrix} -0.1 & -0.1 & -0.1 \\ -0.1 & 0 & -0.1 \\ -0.1 & -0.1 & -0.1 \end{bmatrix} \quad \mathbf{B}=\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathrm{I}=-2$$

$$\mathbf{A}^{\tau}=\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathbf{B}^{\tau}=\begin{bmatrix} 0 & 0 & 0 \\ 1.5 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \tau=10$$

Example: Two snapshots from the input stream followed by the corresponding output pictures. Only the dot moving to the right horizontaally remains on the screen.

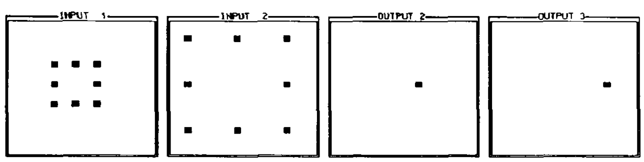


Fig. 9.

[2] ——, "Cellular neural networks: Applications," *IEEE Trans. Circuits Syst.*, vol. 35, pp. 1273–1290, Oct. 1988.

[3] T. Roska and L. O. Chua, "Cellular neural networks with nonlinear and delay-type template elements," in *Proc. IEEE Int. Workshop on Cellular Neural Networks and Their Applications*, pp. 12–25, 1990 (extended version in *Int. J. Circuit Theory and Applications*, vol. 20. pp. 469–481, 1992).

[4] L. O. Chua, T. Roska, P. L. Venetianer, and A. Zarandy, "Some novel capabilities of CNN: Game of life and examples of multipath algorithms," *Rep. DNS-3-1992*, Dual and Neural Computing Systems Laboratory, Hungarian Academy of Sciences, 1992.

[5] J. A. Nossek, G. Seiler, T. Roska, and L. O. Chua, "Cellular neural networks: Theory and circuit design," *Rep. TUM-LNS-TR-90-7*, Technische Universität München, Dec. 1990 (also in *Int. J. Circuit Theory and Applications*, vol. 20. pp. 533–553, 1992).

[6] H. Harrer and J. A. Nossek, "Discrete time cellular neural networks: Architecture, applications and realization," *Rep. TUM-LNS-TR-90-12*, Technische Universität München, Nov. 1990 (also in *Int. J. of Circuit Theory and Applications*, vol. 20. pp. 453–467, 1992).

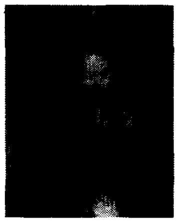
[7] J. Henseler and P. J. Braspennig, "Membrain: a cellular neural network model based on a vibrating membrane," *Int. J. Circuit Theory and Applications*, vol. 20, pp. 483–496, 1992.

[8] A. Radványi, K. Halonen, and T. Roska, "The CNNL simulator and some time-varying CNN templates," *Rep. DNS-9-1991*, Dual and Neural Computing Systems Laboratory, Hungarian Academy of Sciences, 1992.

[9] L. O. Chua and B. E. Shi, "Exploiting cellular automata in the design of cellular neural networks for binary image processing," *Memorandum UCB/ERL M89/130*, Univ. California at Berkeley Electronics Research Laboratory, Nov. 1989.

[10] L. O. Chua and T. Roska, "Stability of a class of nonreciprocal cellular neural networks," *IEEE Trans. Circuits Syst.*, vol. 37, pp. 1520–1527, Dec. 1990.
[11] V. Cimagalli, "A neural network architecture for detecting moving objects II," in *Proc. IEEE Int. Workshop on Cellular Neural Networks and Their Applications*, pp. 124–126, 1990.
[12] N. Frühauf and E. Lüder, "Realizations of CNN's by optical parallel processing with spatial light valves," in *Proc. IEEE Int. Workshop on Cellular Neural Networks and Their Applications*, pp. 281–291, 1990.
[13] K. Halonen, V. Porra, T. Roska, and L. Chua, "VLSI implementation of a reconfigurable cellular neural network containing local logic (CNNL)," in *Proc. IEEE Int. Workshop on Cellular Neural Networks and Their Applications*, pp. 206–215, 1990 (extended version in Int. J. Circuit Theory and Applications, vol. 20, pp. 573–582, 1992).
[14] T. Matsumoto, L. O. Chua, and R. Furukawa, "CNN cloning template: Hole-filler," *IEEE Trans. Circuits Syst.*, vol. 37, pp. 635–638, May 1990.
[15] T. Matsumoto, L. O. Chua, and H. Suzuki, "CNN cloning template: Shadow detector," *IEEE Trans. Circuits Syst.*, vol. 37, pp. 1070–1073, Aug. 1990.
[16] ——, "CNN cloning template: Connected component detector," *IEEE Trans. Circuits Syst.*, vol. 37, pp. 633–635, May 1990.
[17] T. Matsumoto, T. Yokohama, H. Suzuki, and R. Furukawa, "Several image processing examples by CNN," in *Proc. IEEE Int. Workshop on Cellular Neural Networks and Their Applications*, pp. 100–111, 1990.
[18] T. Matsumoto, L. O. Chua, and T. Yokohama, "Image thinning with a cellular neural network," *IEEE Trans. Circuits Syst.*, vol. 37, pp. 633–635, May 1990.
[19] J. A. Nossek and G. Seiler, "An equivalence between multi-layer perceptrons with step function type nonlinearity and a class of Cellular Neural Networks," *Rep. TUM-LNS-TR-90-7*, Technische Universität München, Dec. 1990.
[20] T. Roska, T. Boros, P. Thiran, and L. O. Chua, "Detecting simple motion using cellular neural networks," in *Proc. IEEE Int. Workshop on Cellular Neural Networks and Their Applications*, pp. 127–138, 1990.
[21] T. Roska *et al.*, "A digital multiprocessor hardware accelerator board for cellular neural networks: CNN-HAC," in *Proc. IEEE Int. Workshop on Cellular Neural Networks and Their Applications*, pp. 160–168, 1990 (extended version in *Int. J. Circuit Theory and Applications*, vol. 20. pp. 589–599, 1992).
[22] T. Roska and A. Radványi, *CNND simulator, Cellular neural network embedded in a simple dual computing structure, User's Guide Version 3.0*, Computer and Automation Institute, Hungarian Academy of Sciences, Budapest, 1990.
[23] G. Seiler, "Small object counting with cellular neural networks," in *Proc. IEEE Int. Workshop on Cellular Neural Networks and Their Applications*, pp. 114–123, 1990.
[24] K. Ślot, "Determination of cellular neural network parameters for feature detection of two dimensional images," in *IEEE Int. Workshop on Cellular Neural Networks and Their Applications*, pp. 82–91, 1990.
[25] L. Vandenberghe and J. Vandewalle, "Finding multiple equilibrium points of cellular neural networks without enumeration," in *Proc. IEEE Int. Workshop on Cellular Neural Networks and Their Applications*, pp. 45–55, 1990 (also in *Int. J. of Circuit Theory and Applications*, vol. 20. pp. 519–531, 1992).
[26] J. E. Varrientos, J. Ramirez-Angulo, and E. Sanchez-Sinencio, "Cellular neural networks implementation: A current-mode approach," in *CNNA*, pp. 216–225, 1990.
[27] L. Yang, L. O. Chua, and K. R. Krieg, "VLSI implementation of cellular neural networks," in *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 2425–2427, 1990.
[28] G. G. Yang, "Optical associative memory with invariance," in *Proc. IEEE Int. Workshop on Cellular Neural Networks and Their Applications*, pp. 291–302, 1990.
[29] F. Zou, S. Schwarz, and J. A. Nossek, "Cellular neural network design using a learning algorithm," in *Proc. IEEE Int. Workshop on Cellular Neural Networks and Their Applications*, pp. 73–81, 1990.
[30] F. Zou and J. A. Nossek, "Stability of cellular neural networks with opposite sign templates," *Report TUM-LNS-TR-15*, Technische Universität München, Dec. 1990.
[31] M. Balsi, "Remarks on the stability and functionality of CNN's with one-dimensional templates," *Report DNS-5-1991*, Dual and Neural Computing Systems Laboratory, Hungarian Academy of Sciences, 1991 (to be published in *Int. J. Circuit Theory and Applications*).
[32] T. Boros, K. Lotz, A. Radványi, and T. Roska, "Some useful, new, nonlinear and delay-type templates," *Report DNS-1-1991*, Dual and Neural Computing Systems Res. Lab., Comp. Aut. Inst., Hung. Acad. Sci. (MTA SzTAKI), Budapest, 1991.
[33] L. O. Chua and B. E. Shi, "Multiple layer cellular neural networks: A tutorial," in *Algorithms and Parallel VLSI Architectures*, E. F. Deprettere and A. van der Veen, Eds., vol. A: Tutorials. New York: Elsevier, 1991, pp. 137–168.
[34] L. O. Chua and P. Thiran, "An analytic method for designing simple cellular neural networks," *IEEE Trans. Circuits Syst.*, vol. 38, pp. 1332–1341, 1991.
[35] L. O. Chua, L. Yang, and K. R. Krieg, "Signal processing using cellular neural networks," *J. VLSI Signal Processing*, vol. 3, pp. 25–52, 1991.
[36] P. P. Civalleri, M. Gilli, and L. Pandolfi, "On stability of cellular neural networks with delay," report, Politecnico di Torino, Nov. 1991, to be published.
[37] K. R. Crounse, T. Roska, and L. O. Chua, "Image halftoning with cellular neural networks," *Memorandum UCB/ERL M91/106*, Univ. California at Berkeley Electronics Research Laboratory, Nov. 1991, submitted for publication.
[38] J. M. Cruz and L. O. Chua, "A CNN chip for connected component detection," *IEEE Trans. Circuits Syst.*, vol. 38, pp. 812–817, July 1991.
[39] N. Frühauf, E. Lüder, M. Gaiada, and G. Bader, "An optical implementation of space invariant Cellular Neural Networks," in *Proc. European Conf. Circuit Theory and Design*, pp. 42–51, 1991.
[40] S. Fukuda, T. Boros, and T. Roska, "A new efficient analysis of thermographic images by using Cellular Neural Networks," *Tech. Rep. DNS-11-1991*, Dual and Neural Computing Systems Laboratory, Hungarian Academy of Sciences, 1990, to be published.
[41] C. Guzelis and L. O. Chua, "Stability analysis of generalized cellular neural networks," *Int. J. Circuit Theory and Applications*, vol. 21, pp. 1–33, 1993.
[42] H. Harrer and J. A. Nossek, "An analog CMOS compatible convolution circuit for analog neural networks," *Rep. TUM-LNS-TR-91-11*, Technische Universität München, 1991.
[43] H. Harrer, J. A. Nossek, and R. Stelzl, "An analog implementation of discrete-time cellular neural networks," *Rep. TUM-LNS-TR-91-14*, Technische Universität München, June 1991 (extended version in *IEEE Trans. Neural Networks*, vol. 3, pp. 466–477, 1992).
[44] V. I. Krinsky, V. N. Biktashev, and I. R. Efimov, "Autowave principles for parallel image processing," *Physica D*, vol. 49, pp. 247–253, 1991.
[45] A. Radványi and T. Roska, "The CNN workstation-CNND version 4.1," *Tech. Rep. DNS-12-1991*, Dual and Neural Computing Systems Laboratory, Hungarian Academy of Sciences, 1991.
[46] T. Roska, A. Radványi, T. Kozek, and T. Boros, "Dual CNN software library," *Rep. DNS-7-1991*, Dual and Neural Computing Systems Laboratory, Hungarian Academy of Sciences, 1991.
[47] T. Roska, "Cellular neural networks: a state-of-the-art review," in *Proc. European Conf. on Circuit Theory and Design (ECCTD-91)*, pp. 1–9, 1991.
[48] T. Roska and P. Szolgay, "A comparison of various cellular neural network (CNN) realization—A review," in *Proc. 2nd Int. Conf. Microelectronics for Neural Networks*, pp. 423–421, 1991.
[49] T. Roska, "Dual computing structures containing analog cellular neural networks and digital decision units," in *Proc. IFIP Workshop on Silicon Architectures for Neural Nets*, pp. 233–244, 1991.
[50] S. Schwarz and W. Mathis, "A design algorithm for cellular neural networks," in *Proc. 2nd Int. Conf. Microelectronics for Neural Networks*, pp. 53–59, 1991.
[51] G. Seiler and M. Hasler, "Convergence of reciprocal cellular neural networks," *Report TUM-LNS-TR-91-12*, Technische Universität München, June 1991.
[52] B. E. Shi and L. O. Chua, "A generalized cellular neural network of a novel edge detection algorithm," *Memorandum UCB/ERL M91/25*, Univ. California at Berkeley Electronics Research Laboratory, Apr. 1991.
[53] P. Szolgay and T. Kozek, "Optical detection of layout errors of printed circuit boards using learned CNN templates," *Report DNS-8-1991*, Dual and Neural Computing Systems Res. Lab., Comp. Aut. Inst., Hung. Acad. Sci. (MTA SzTAKI), Budapest, 1991.
[54] T. Roska, C. W. Wu, M. Balsi, and L. O. Chua, "Stability and dynamics of delay-type and nonlinear Cellular Neural Networks," *Memorandum UCB/ERL M91/110*, Univ. California at Berkeley Electronics Research Laboratory, Dec. 1991.
[55] L. O. Chua and C. W. Wu, "On the universe of stable cellular neural networks," *Int. J. Circuit Theory and Applications*, pp. 497–572, vol. 20, 1992.
[56] F. Zou and J. A. Nossek, "A chaotic attractor with Cellular Neural Networks," *IEEE Trans. Circuits Syst.*, vol. 38, pp. 811–812, July 1991.
[57] J. M. Cruz and L. O. Chua, "Design of high speed high density CNN's in CMOS technology," *Int. J. Circuit Theory and Applications*, pp. 555–572, vol. 20, 1992.
[58] G. Erös *et al.*, "Optical tracking system for automatic guided vehi-

cle (AGV) using cellular neural networks," *Rep. DNS-15-1992*, Dual and Neural Computing Systems Laboratory, Hungarian Academy of Sciences, 1992 (also in *Proc. IEEE CNNA-92*, pp. 216–221, 1992).
[59] W. Heiligenberg and T. Roska, "On biological sensory information processing principles relevant to dually computing CNN's," *Rep. DNS-4-1992*, Dual and Neural Computing Systems Laboratory, Hungarian Academy of Sciences, 1992.
[60] T. Kozek, T. Roska, and L. O. Chua, "Generic algorithm for CNN template learning," *Memorandum UCB/ERL/M92/82*, Univ. California at Berkeley Electronics Research Laboratory, 1992, and *IEEE Trans. Circuits Syst.*, to be published.
[61] V. Pèrez-Muñuzuri, V. Pèrez-Villar, and L. O. Chua, "Autowaves for image processing on a two-dimensional CNN array of Chua's circuits: flat and wrinkled labyrinths," *Ieee Trans. Circuits Syst.*, this issue.
[62] T. Roska *et al.*, "The use of CNN models in the subcortical visual pathway," *Rep. DNS-16-1992*, Dual and Neural Computing Systems Laboratory, Hungarian Academy of Sciences, 1992.
[63] T. Roska and L. O. Chua, "The dual CNN analog software," *Tech. Rep. DNS-2-1992*, Dual and Neural Computing Systems Laboratory, Hungarian Academy of Sciences, 1992.
[64] T. Roska, "Programmable cellular neural networks—A state-of-the-art," in *State-of-the-Art in Computer Systems and Software Engineering*, P. Dewilde and J. Vandewalle, Eds. Boston: Kluwer, 1992, pp. 151–168.
[65] T. Roska, C. W. Wu, M. Balsi, and L. O. Chua, "Stability and dynamics of delay-type general and Cellular Neural Networks," *IEEE Trans. Circuits Syst.*, vol. 39, pp. 487–490, June 1992.
[66] T. Roska, T. Boros, A. Radványi, P. Thiran, and L. O. Chua, "Detecting moving and standing objects using cellular neural networks," *Int. J. Circuit Theory and Applications*, vol. 20, pp. 613–628, 1992.
[67] M. Tanaka, K. Crounse, and T. Roska, "Template synthesis of cellular neural networks for information coding and decoding," in *Proc. IEEE CNNA-92*, pp. 29–35, 1992.
[68] *Proc. IEEE Int. Workshop on Cellular Neural Networks and their Applications (CNNA-90)*, 1990, Budapest, IEEE Cat. No. 90TH0312-9.
[69] *Proc. 2nd IEEE Int. Workshop on Cellular Neural Networks and Their Applications (CNNA-92)*, 1992, Munich.
[70] *Proc. Special Session on Cellular Neural Networks European Conf. Circuit Theory and Design (ECCTD-91)*, Sept. 1991.
[71] *Int. J. Circuit Theory and Applications*, Special issue on Cellular Neural Networks, vol. 20, 1992.
[72] T. Roska, A. Zarándy, and L. O. Chua, "Color image processing by CNN," in preparation.
[73] "Lady Dressed in Lilac," by P. Szinnyei-Merse.
[74] J. Buhmann and K. Schulten, "Influence of noise on the function of a 'physiological' neural network," *Biol. Cybern.*, vol. 56, pp. 313–327, 1987.
[75] E. R. Kandel, J. H. Schwarz, and T. M. Jessel, *Principles of Neural Science*, third edition. Amsterdam: Elsevier, 1991.
[76] T. Roska and L. O. Chua, "The CNN universal machine," *IEEE Trans. Circuits Syst.—II*, vol. 40, Mar. 1993.



Leon O. Chua (S'60–M'62–SM'70–F'74) received the S.M. degree from the Massachusetts Institute of Technology in 1961 and the Ph.D. degree from the University of Illinois, Urbana, in 1964.

He is currently a Professor of electrical engineering and computer sciences at the University of California, Berkeley. His research interests are in the areas of general *nonlinear* network and system theory. He has been a consultant to various electronic industries in the areas of nonlinear network analysis, modeling, and computer-aided design. He is the author of *Intrduction to Nonlinear Network Theory* (McGraw-Hill, 1969), and a coauthor of the books *Computer-Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques* (Prentice-Hall, 1975) , *Linear and Nonlinear Circuits* (McGraw-Hill, 1987), and *Practical Numerical Algorithms for Chaotic Systems* (Springer-Verlag, 1989). He has published many papers in the area of nonlinear networks and systems. He served as editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS from 1973 to 1975 and as President of the IEEE Society on Circuits and Systems in 1976. He is presently the editor of the *International Journal of Bifurcation and Chaos* and a deputy editor of the *International Journal of Circuit Theory and Applications.*

Professor Chua is the holder of five U.S. patents. He is also the recipient of several awards and prizes, including the 1967 IEEE Browder J. Thompson Memorial Prize Award, the 1973 IEEE W. R. G. Baker Prize Award, the 1974 Frederick Emmons Terman Award, the 1976 Miller Research Professorship, the 1982 Senior Visiting Fellowship at Cambridge University, the 1982/83 Alexander von Humboldt Senior U.S. Scientist Award at the Technical University of Munich, the 1983/84 Visiting U.S. Scientist Award at Waseda University, Tokyo, the IEEE Centennial Medal in 1985, the 1985 Myril B. Reed Best Paper Prize, and both the 1985 and 1989 IEEE Guillemin–Cauer Prizes. He was also awarded a Doctor Honoris Causa from the Ecole Polytechnique Federale-Lausanne, Switzerland, in 1983, an Honorary Doctorate from the University of Tokushima, Japan, in 1984, and an Honorary Doctorate from the Technische Universitat Dresden in 1992. In 1986 he was awarded a *Professor Invite* International Award at the University of Paris-Sud from the French Ministry of Education.

Tamas Roska (M'87–SM'90–F'92), for a photograph and biography please see page 146 of this issue.