Available online at www.sciencedirect.com

ScienceDirect

Expert Systems with Applications 36 (2009) 2645–2650

Expert Systems with Applications

www.elsevier.com/locate/eswa

Efficient edge detection in digital images using a cellular neural network optimized by differential evolution algorithm

Alper Baştürk[a,*], Enis Günay[b]

[a] Department of Computer Engineering, Erciyes University, Kayseri 38039, Turkey
[b] Department of Electrical and Electronics Engineering, Erciyes University, Kayseri 38039, Turkey

Abstract

A cellular neural network (CNN) based edge detector optimized by differential evolution (DE) algorithm is presented. Cloning template of the proposed CNN is adaptively tuned by using simple training images. The performance of the proposed edge detector is evaluated on different test images and compared with popular edge detectors from the literature. Simulation results indicate that the proposed CNN operator outperforms competing edge detectors and offers superior performance in edge detection in digital images.
© 2008 Elsevier Ltd. All rights reserved.

Keywords: Cellular neural networks; Cloning template; Differential evolution algorithm; Edge detection

1. Introduction

Cellular neural network (CNN) is a massive parallel computing paradigm defined in n-dimensional regular array of elements, cells (Chua, 1998; Chua & Yang, 1988a, 1988b). Cells are multiple input–single output processors and the dynamic behaviors of the cells are determined by 19 free parameters called, cloning template (Matsumoto, Chua, & Suzuki, 1990; Matsumoto, Chua, & Furukawa, 1990).

Determining the robust cloning templates has been the most serious problem among the studies on image processing via CNN (Crounse & Chua, 1995; Matsumoto, Chua, & Yokohoma, 1990). Among the template design studies in literature, the intuitive thinking of the designer seems to be the quickest way. Most of the times it does not guarantee to find the desired templates, also doing lots of experiments in both the template learning and the array dynamics are the main disadvantages. Another way of CNN template design is the direct template. However, it seems to be depend on the particular template class, needs some computational power template learning requirements. The most widely studied way of template design is the template learning method. Besides classic neural network training, the heuristic ways as genetic algorithm, simulated annealing etc., are used to find the correlation between the input and the desired output pairs (Zarandy, 1999).

An edge in an image is the boundary between two different regions. Edge detection is one of the most important steps in image processing, analysis and pattern recognition systems. Its importance arises from the fact that edge often provides an indication of the physical extent of object within the image. Sufficient information to characteristic feature is provided by the detection of edge because the size of the image data is reduced into a size that is more suitable for image analysis. The performance of the later tasks such as image segmentation, boundary detection, object recognition and classification, image registration, and so on are depend on the success of the edge characterization step (Yüksel, 2007; Yüksel & Yildirim, 2004).

The image intensity shows abrupt changes at edges. Therefore, edge detection usually involves the calculation of the derivative of the image intensity function at a given pixel location. If the magnitude of the derivative of the

* Corresponding author. Tel.: +90 352 437 49 01x32552; fax: +90 352 437 57 84.
E-mail address: ab@erciyes.edu.tr (A. Baştürk).

0957-4174/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2008.01.082

image intensity function is relatively high at a given pixel location, then the pixel at that image location is classified as an edge pixel (Yüksel, 2007; Yüksel & Yildirim, 2004).

A number of methods for edge detection implementing different approaches are available in the literature such as the statistical methods (Haberstroh & Kurz, 1993; Hansen & Elliot, 1982; Huang & Tseng, 1988; Nahi & Assefi, 1972; Stern & Kurz, 1988), the difference methods (Kirsch, 1971; Marr & Hidreth, 1980; Prewitt, 1970; Sobel, 1978), and curve fitting (Haralick, 1984; Haralick & Watson, 1981; Huechel, 1971; Nalwa & Binford, 1986; Ji & Haralick, 2002). The classical methods such as Sobel (1978), Prewitt (1970) and Kirsch (1971) detectors calculate the first directional derivative to determine the locations of the edges. The zero-crossing edge detectors (Bovik, 1998; Umbaugh, 1998) use the second derivative along with the Laplacian operator. The Canny detector (Canny, 1986), which is a Gaussian edge detector, is one of the most popular edge detectors in the literature and it has been widely used in many applications (Ali & Clausi, 2001; Hongjian & Ward, 2002; Linares, Maersereau, & Smith, 1996). Although the Gaussian detectors exhibit relatively better performance, they are computationally much more complex than classical derivative based edge detectors (Yüksel, 2007; Yüksel & Yildirim, 2004).

In recent years, there has been a growing research interest in the applications of evolutionary algorithms in many diverse fields of science and engineering. Among these algorithms, the differential evolution (DE) algorithm (Price, 1996; Price, Storn, & Lampinen, 2005; Storn & Price, 1997) is a relatively novel optimization technique for efficiently solving numerical-optimization problems. The algorithm has successfully been applied in many different problems, and gained a wide acceptance and popularity because of its simplicity, robustness, and good convergence properties (Chang, 2006; Liu, Wang, Jin, Huang, & Tang, 2007).

This paper presents a CNN cloning template learning method uses DE algorithm for edge detection in digital images. The remainder of this paper is organized as follows: Section 2 provides a brief introduction to architecture of the CNN, while Section 3 provides some information on DE algorithm. In Section 4, CNN cloning template learning via the DE algorithm is presented. Experimental studies are conducted in Section 5, including comparisons of the performances of proposed method and well known edge detection operators such as Sobel, Prewitt and Canny for both binary and gray level test images. Conclusions are drawn in Section 6.

2. Architecture of the cellular neural network

Every CNN is a spatial arrangement of locally-coupled cells. Each cell is a dynamical system and has the state equation is given in Eqs. (1) and (2), where input, threshold, state and output are demonstrated as $u_{ij}, z_{ij}, x_{ij}, y_{ij}$ (Chua, Roska, Kozek, & Zarandy, 1993). Also each x_{ij} cell has the feed-forward synapses as $b_{kl}u_{kl}$ that is the input and has the feed-back synapses as $a_{kl}y_{kl}$ that is the output of each neighbor cells, as shown in Eq. (1)

$$\dot{x}_{ij} = -x_{ij} + \sum_{kl \in S_{ij}(\tau)} a_{kl}y_{kl} + \sum_{kl \in S_{ij}(\tau)} b_{kl}u_{kl} + z_{ij} \tag{1}$$

where

$$y_{ij} = f(x_{ij}) = \frac{1}{2}(\mid x_{ij}+1 \mid - \mid x_{ij}-1 \mid);$$
$$i = 1,2,\ldots,M; \quad j = 1,2,\ldots N; \quad 1 \leqslant k \leqslant M; \quad 1 \leqslant l \leqslant N. \tag{2}$$

CNN cells are coupled locally to its neighbor cells by the sphere of influence $S_{ij}(\tau)$ of radius τ, in Eq. (3).

$$S_{ij}(\tau) = \{x_{kl} : \max(\mid k-i \mid, \mid l-j \mid) \leqslant \tau, \quad 1 \leqslant k \leqslant M, \quad 1 \leqslant l \leqslant N\}. \tag{3}$$

For example, 3×3 sphere of influence has the $\tau = 1$ radius, where x_{ij} is coupled to its first degree neighborhoods x_{kl} as $(k,l) = (i+1,j+1), (i+1,j), (i+1,j-1), (i,j+1), (i,j-1), (i-1,j+1), (i-1,j), (i-1,j-1)$.

The characteristic of a CNN with 3×3 sphere of influence ($\tau = 1$ radius) can be specified by using 19 real numbers as: one threshold z_{ij}, nine control (feed-forward) b_{kl}, and nine feed-back coefficients a_{kl}. These 19 numbers forms a cloning template diagram that can be shown as a vector μ in Eq. (4)

$$\mu = [a_1a_2a_3a_4a_5a_6a_7a_8a_9b_1b_2b_3b_4b_5b_6b_7b_8b_9z]. \tag{4}$$

Complete stability that is all trajectories of the standard CNN defined in Eqs. (1) and (2) with constant thresholds, constant input, converge to an equilibrium state, can be given under the assumptions as given in Eq. (5) (Chua & Roska, 1993).

$$\left.\begin{array}{l} A(i,j;k,l) = A(k,l;i,j) \\ B(i,j;k,l) = B(k,l;i,j) \end{array}\right\}, \quad \mid x_{ij}(0) \mid \leqslant 1, \quad \mid u_{ij} \mid \leqslant 1. \tag{5}$$

So completely stable CNN can be formed by choosing the cloning template as

$$a_1 = a_9, \quad a_2 = a_8, \quad a_3 = a_7, \quad a_4 = a_6,$$
$$b_1 = b_9, \quad b_2 = b_8, \quad b_3 = b_7, \quad b_4 = b_6. \tag{6}$$

So

$$A = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_4 \\ a_3 & a_2 & a_1 \end{bmatrix} \tag{7}$$

and

$$B = \begin{bmatrix} b_1 & b_2 & b_3 \\ b_4 & b_5 & b_6 \\ b_7 & b_8 & b_9 \end{bmatrix} = \begin{bmatrix} b_1 & b_2 & b_3 \\ b_4 & b_5 & b_4 \\ b_3 & b_2 & b_1 \end{bmatrix}. \tag{8}$$

In this paper, MatCNN InstantVision Toolbox for Matlab (Matcnn instantvision toolbox for matlab) is used to simulate the CNN architecture.