

Cellular Neural Networks With Virtual Template Expansion for Retinal Vessel Segmentation

Renzo Perfetti, Elisa Ricci, Daniele Casali, and Giovanni Costantini

Abstract—A retinal vessel segmentation method based on cellular neural networks (CNNs) is proposed. The CNN design is characterized by a virtual template expansion obtained through a multistep operation. It is based on linear space-invariant 3×3 templates and can be realized using existing chip prototypes like the ACE16K. The proposed design is capable of performing vessel segmentation within a short computation time. It was tested on a publicly available database of color images of the retina, using receiver operating characteristic curves. The simulation results show good performance comparable with that of the best existing methods.

Index Terms—Cellular neural networks (CNNs), line detection, retinal imaging, vessel segmentation.

I. INTRODUCTION

AUTOMATIC segmentation of blood vessels in retinal fundus images plays an important role in the diagnosis of several pathologies, like hypertension, diabetes, and cardiovascular disease [1]. Several morphological features of veins and arteries (e.g., diameter, length, branching angle, and tortuosity) have diagnostic relevance. Accurate vasculature segmentation is a difficult task for several reasons: the presence of noise, the low contrast between vessels and background, and the variability of vessel width, brightness, and shape. Moreover, due to the presence of lesions, exudates, and other pathological effects, the image may have large abnormal regions. Several methods have been proposed in the literature to address these problems, including matched filtering [2], tracking methods [3], multithreshold probing [4], and supervised classification [5].

One important aspect which is rarely addressed is the complexity of the algorithm and its efficient hardware implementation in order to reduce the time spent for the segmentation. One attractive paradigm for parallel real-time image processing is represented by cellular neural networks (CNNs) [6], [7].

To the best of our knowledge, retinal vessel extraction with CNNs has been previously proposed only in [8], where segmentation is obtained through histogram modification, local adaptive thresholding, and morphological opening. This method presents two drawbacks. First, it relies on several design parameters: the scaling factors of local mean and variance (α and β), the neighborhood size, and the structuring element

for opening. Since no guidelines are available for their settings, they must be empirically tuned. Moreover, nonlinear CNN templates are required for local estimation of the variance.

In this brief, we present a new CNN-based approach for detecting vessel pixels in color fundus images, avoiding the previously mentioned disadvantages. The effectiveness of the current approach comes from the fact that we exploit geometrical knowledge about the task to be solved whereas the method in [8] just applies generic algorithms.

The proposed method is based on the detection of linear structures through an operator introduced by Dixon and Taylor [9]. This operator, previously used in mammographic image analysis [10], could be realized with simple CNN templates. However, our simulations showed that, to obtain state-of-the-art performance in retinal vessel segmentation, the template size must be 15×15 . Thus, to simplify the physical realization of the CNN, we adopt a multistep operation with virtual template expansion. This network could reduce the segmentation time of some orders of magnitude with respect to a sequential digital realization.

The CNN operation has been simulated on a digital computer, and the corresponding performance has been evaluated on a publicly available database through receiver operating characteristic (ROC) analysis. The ROC curve shows the effectiveness of our method.

The remainder of this brief is organized as follows. In Section II, we illustrate the proposed method to detect vessel pixels. In Section III, we describe the CNN implementation. In Section IV, some simulation results are presented and compared with existing techniques for vessel segmentation. Some comments conclude this brief.

II. LINE DETECTION

Usually, in RGB nonmydriatic (i.e., without pupil dilatation) images, the green channel exhibits the best vessel/background contrast while the red and blue ones tend to be very noisy. Therefore, we work on the inverted green channel images without performing any further preprocessing. An example is shown in Fig. 1: vessels appear brighter than the background. Gray level -1 corresponds to black, and level $+1$ corresponds to white.

The proposed method is as follows. The average gray level is evaluated along lines of fixed length l passing through the target pixel (i, j) at different orientations. We consider 12 orientations (15° of angular resolution). The line with the largest average gray level is found: its average value is denoted by L_{ij} . The difference $S_{ij} = L_{ij} - N_{ij}$ represents the *line strength* of the pixel [10], where N_{ij} is the average gray level in the square window, centered on the pixel, with edge length equal to l .

Manuscript received February 23, 2006; revised June 23, 2006. This work was supported in part by the Italian Ministry for Education and Scientific Research under Grant COFIN 2004. This paper was recommended by Associate Editor J. Suykens.

R. Perfetti and E. Ricci are with the Department of Electronic Engineering, University of Perugia, I-06125 Perugia, Italy (e-mail: perfetti@diei.unipg.it).

D. Casali and G. Costantini are with the Department of Electronic Engineering, University of Rome "Tor Vergata," Rome 0033, Italy.

Digital Object Identifier 10.1109/TCSII.2006.886244



Fig. 1. Inverted green channel of a nonmydiatic retinal image.

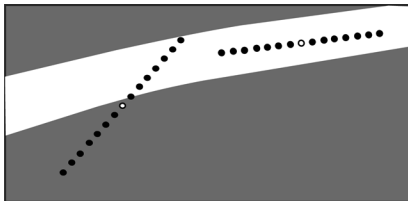


Fig. 2. Proposed line detector.

As shown in Fig. 2, if the central pixel belongs to a vessel, the winning line is aligned with the vessel itself. Otherwise we have a partial overlap and the line strength is lower. This difference allows to discriminate vessel pixels from nonvessel pixels.

In the simulations, we tested different line lengths l . The best results were obtained using $l = 15$ pixels. In our method, the square window is not oriented like the winning line, as in [10], but it is kept fixed in order to simplify the hardware realization. For the same reason, the average line intensity is not obtained by interpolation: the gray values to be averaged are determined by rounding the coordinates of points on the ideal line (see Section III for details). The value of S_{ij} is compared with a threshold T : only if $S_{ij} > T$ does it correspond to a vessel pixel. The effect of the threshold will be discussed in Section IV.

III. CNN REALIZATION

The computation of the line strength S_{ij} is the most time consuming task of the method. In a sequential pixel-by-pixel processing of an image with N pixels, the 12 line averages and the local window average must be repeated N times. Moreover, for each pixel, the maximum line average must be selected. Since in usual images of the eye fundus N is of the order of 300 000, the computational burden is evident. Here, we describe a design exploiting the parallel computation ability and the simple programmability properties of CNNs. We assume the usual piecewise-linear activation function of the cell: $f(x_{ij}) = 1/2(|x_{ij} - 1| - |x_{ij} + 1|)$. The cell input satisfies the constraint $-1 \leq u_{ij} \leq +1$, for every i and j . For details on the dynamic equation and properties of CNNs, the reader is referred to [6].

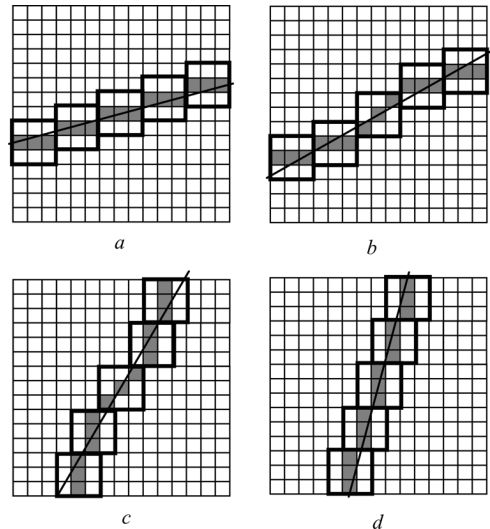


Fig. 3. Approximation of lines with (a) 15°, (b) 30°, (c) 60°, and (d) 75°.

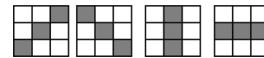


Fig. 4. Four different patterns used to approximate the 12 lines of 15 pixels. The corresponding templates are, respectively, B_1 , B_2 , B_3 , and B_4 .

The proposed method requires a 15×15 neighborhood. The CNN realization can be obtained using a 3×3 physical neighborhood, virtually expanded through a multistep operation [11], [12]. In particular, a shifting approach [12] is suited to the present application. The shift template can be found in [13]. The number of processing steps depends on the template decomposition and the shifting strategy [12].

A. Template Decomposition

To reduce the number of processing steps and the memory requirements, each line of 15 pixels is first approximated as shown in Fig. 3. Lines with orientations -15° , -30° , -60° , and -75° are turned right to left. Lines with orientations 0° , 90° , and $\pm 45^\circ$ are exact. Each line can be decomposed into five 3×3 patterns, highlighted in Fig. 3. In each pattern, there are three pixels to be averaged. The mean gray level for each pattern can be computed by a corresponding 3×3 template. Note that some patterns are reused for different angles. Taking into account all 12 orientations, the number of required different patterns (i.e., templates) is four. They are shown in Fig. 4. Each processing step corresponds to a linear convolution of the input image with a suitable B template (using $A = 0$). As an example, let us consider the leftmost and rightmost patterns in Fig. 4. The corresponding partial outcomes are computed by using the following input templates:

$$\mathbf{B}_1 = \frac{1}{3} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \quad \mathbf{B}_4 = \frac{1}{3} \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}. \quad (1)$$

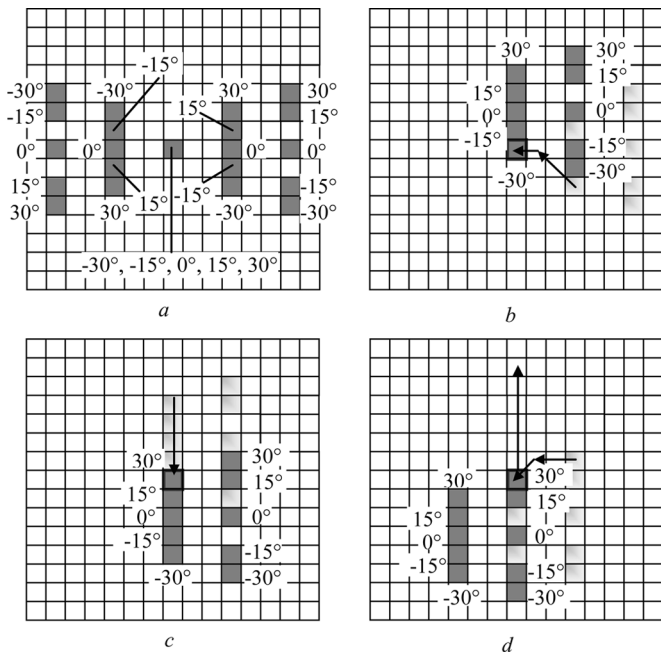


Fig. 5. Cascaded shift example. (a) The 21 positions where template B_4 is required. (b) Shifting the result of the convolution by two pixels to NW and one pixel to W. Only the ten rightmost positions are shown. (c) Consecutive four shifts in the S direction of one pixel. (d) Two shifts to W and one to SW.

For each template, the time evolution is stopped after a fixed number of time constants in order to obtain an almost steady-state output value. The scaling factor $1/3$, common to all templates, guarantees the operation in the linear region for each cell and in each processing step. After the last step, we have the output images $Y_m = [y_{ij}(m)]$, $m = 1, \dots, 4$, stored in distinct local analog memories (LAMs).

B. Shifting Strategy

In the second phase, the partial outcomes must be shifted to give the correct contribution to each line. The shift results are scaled by $1/5$ and accumulated into 12 LAMs, one for each orientation. The final result represents the correct line average since each pixel gray level is divided by 15. The scaling factor $1/15$ is split in order to attenuate the eventual noise arising from the grayscale shift, improving the signal-to-noise ratio.

To illustrate the shifting strategy, we consider template B_4 in (1). It is required for five lines, i.e., $0^\circ, \pm 15^\circ$, and $\pm 30^\circ$, in 21 different positions, shown in Fig. 5(a). The central position does not require shift; thus, we must move 20 partial results to the center of the window. Now, let us consider the ten rightmost positions in Fig. 5(a). The output image Y_4 is first shifted to the NW direction by two pixels and to the west by one pixel, as shown in Fig. 5(b). As a consequence the partial result corresponding to the angle -30° is moved to the center and accumulated into the corresponding LAM. Further four consecutive shifts by one pixel to S, will move to the center the results useful, respectively, for the $-15^\circ, 0^\circ, 15^\circ$ and 30° lines, as shown in Fig. 5(c). Then, by two shifts to W and one shift to SW, we have the situation shown in Fig. 5(d). The result corresponding to 30° is accumulated into the corresponding memory. Finally, we shift

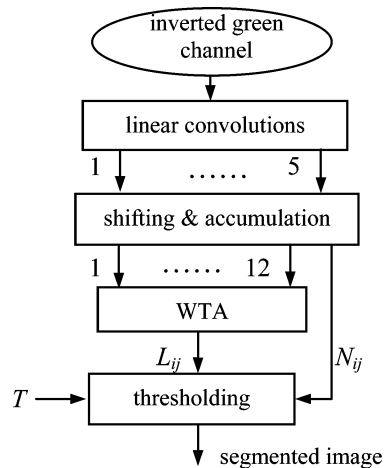


Fig. 6. Block diagram of the proposed CNN algorithm.

to N, in the order by one, two, two, and finally one pixel, storing the results for $15^\circ, 0^\circ, -15^\circ$, and -30° respectively.

At this point, the stored image Y_4 is reloaded as input to the CNN, and, in a similar way, the partial results corresponding to the leftmost ten positions in Fig. 6(a) are moved to the center and accumulated.

Hence, the total number of shift operations needed for this template is $16 \times 2 = 32$. Exactly the same operations are required for the template B_3 with the only need to swap vertical and horizontal shifts. The template B_1 is used by lines $45^\circ, \pm 15^\circ$, and $\pm 60^\circ$, in five different positions, that can be moved with a total of 12 NE and SW shifts; the same number of shifts is required with B_2 . Hence, the total number of elementary shift operations is $32 + 32 + 12 + 12 = 88$, and the total number of steps to obtain the line averages is 92.

The average in the 15×15 square can be computed with a similar multistep operation: one linear convolution and 24 shifts of three pixels each, for a total of 73 additional steps. The maximum line average can be determined by a winner-take-all (WTA) circuit [14], [15]. Taking into account the rather small number of inputs (twelve), the existing chips allow high speed and precision. In principle, the WTA function could be realized also using a CNN [16] even if, to the best of our knowledge, no tested prototype is available. Subtraction of N_{ij} and thresholding can be easily obtained using $A(i, j; i, j) > 1, B(i, j; i, j) = 1$, and the bias $I_{ij} = -T - N_{ij}$, where T is the threshold. A scheme of the proposed system is shown in Fig. 6.

C. Circuit Implementation and Processing Speed

The proposed CNN algorithm requires only linear space-invariant 3×3 templates, so it could be implemented using one of the existing CNN chips. For example, the ACE16K chip is a 128×128 array with 7-bit accuracy, eight analog grayscale memories per cell, and 32 stored templates [17]. The most recent version allows also the space-variant bias term I_{ij} required by the proposed method.¹

¹Further details can be found at http://www.imse.cnm.es/~dictam/2ND_PHASE/Demonstrators/ACE16K/index.htm.

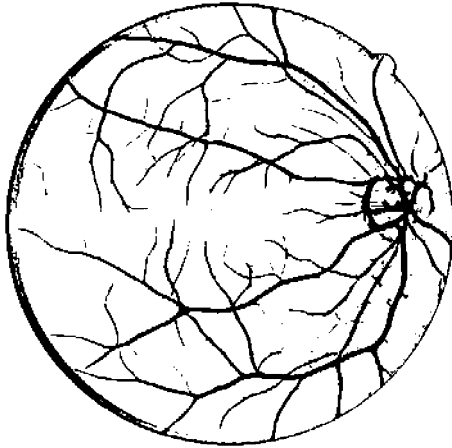


Fig. 7. Vessel segmentation of the image in Fig. 1 ($T = 0.05$).

The proposed segmentation algorithm requires four convolutions for line approximation, eight shifting templates, and one convolution for the local average for a total of 13 templates. The required memories are five for the first phase and 13 for the second phase. Thus, to accommodate the algorithm in the ACE16K chip, only a slight LAM expansion is required. Hence, we take this chip as a reference to estimate the processing speed of the proposed method.

The total number of steps is $92 + 73 = 165$. We assume that each processing step lasts 5τ , where τ is the time constant of the cell circuit. Using the value $\tau = 160$ ns for linear operations [17], we obtain $0.8 \mu\text{s}$ per step.

The images usually employed in the screening of the eye fundus (like that in Fig. 1) must be decomposed into subimages in order to fit with the state-of-the-art CNN chips. Moreover, the subimages must overlap to avoid border effects. Using lines of 15 pixels, the overlap is seven pixels on each side and 25 subimages are required using a 128×128 array. Ignoring the convergence time of the WTA circuit and the time for I/O, we have an execution time of $25 \times 165 \times 0.8 \mu\text{s} \cong 3.3$ ms. This value allows real-time segmentation as an aid to ophthalmic diagnosis.

IV. EXPERIMENTAL RESULTS

The proposed system has been simulated with Matlab (the code is available on request from the authors). An example of vessel segmentation is shown in Fig. 7, concerning the input image in Fig. 1. To test the proposed method, we used a database of nonmydriatic color images, collected by Staal *et al.* [5] with the intent of comparing the performance of different segmentation methods. This database is known as Digital Retinal Images for Vessel Extraction (DRIVE), and it is publicly available.² It consists of 40 images (seven with pathologies) captured by a Canon CR5 3CCD camera with a 45° field of view (FOV).

The images are of size 768×584 pixels with 8 bits per color channel. The FOV in the images is circular with approximately 540 pixels in diameter. The images are compressed in JPEG format, which is a common practice in screening programs. The 40 images are divided into a training set and a test set, each containing 20 images. The test set has four images with pathology.

²[Online]. Available: <http://www.isi.uu.nl/Research/Databases/DRIVE/>

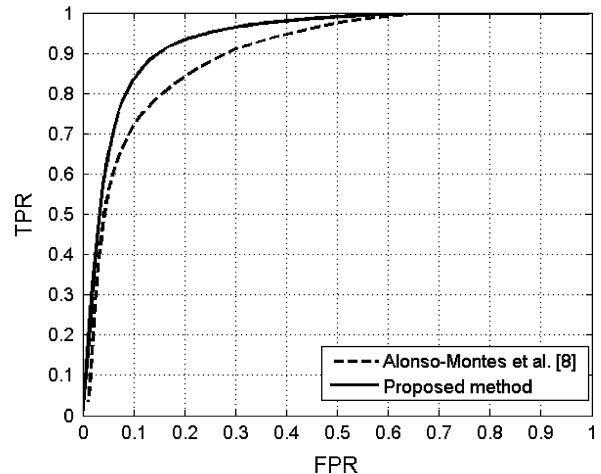


Fig. 8. ROC curves for the proposed CNN algorithm, compared with the method of [8].

TABLE I
COMPARISON OF SEGMENTATION METHODS

Method	AUC	Accuracy
proposed CNN	0.9348	0.9261 ($T=0.05$)
Alonso-Montes et al. [8]	0.9011	0.9185 ($T=0.22$)
Staal et al. [5]	0.9520	0.9441

The training set is useful to design supervised segmentation methods like [5]; in the present study, it has not been used. All of the images were manually segmented. The images of the test set were segmented twice, resulting in a set A and a set B. In set A, 12.7% of pixels were marked as vessel, against 12.3% for set B. For the sake of comparison with supervised methods [5], the performance is evaluated on the test set considering only pixels inside the FOV, using the segmentations of set A as the ground truth.

It is common practice to evaluate the performance of retinal vessel segmentation algorithms using ROC curves [5]. A ROC curve plots the fraction of pixels correctly classified as vessel, namely the true positive rate (TPR), versus the fraction of pixels wrongly classified as vessel, namely the false positive rate (FPR). The closer the curve approaches the top left corner, the better the performance of the system. The most frequently used performance measure extracted from the ROC curve is the value of the area under the curve (AUC), which is 1 for an ideal system. In the present method, the ROC has been traced by varying the threshold T . The curve is shown in Fig. 8. The corresponding value of AUC is given in Table I, along with the maximum accuracy corresponding to an “optimum” threshold value ($T = 0.05$).

An ROC analysis has been performed also for the method [8]. We considered only the first phase of the method (i.e., vessel segmentation), since the results of the following stages are not easily comparable with the existing literature. The method uses a space-variant threshold $\theta = \alpha E_m + \beta E_v + T$, where E_m, E_v are local mean and variance estimations, respectively, and α, β are scale factors. The ROC curve was traced by empirically

tuning α and β on the dataset and varying the value T . We traced the curves with neighborhood size ranging from 3×3 to 15×15 . For the sake of comparison, we show the result for the same neighborhood size of the present method (15×15), which is also the best curve we found. The curve corresponds to $\alpha = 1.1$, $\beta = 7$, and it is depicted in Fig. 8. The values of AUC and maximum accuracy are shown in Table I, along with those of the method [5], which is the best available technique in the literature.

V. COMMENTS AND CONCLUSION

A CNN-based retinal vessel segmentation method has been presented. It is easy to use and can be implemented using existing devices with minor changes. From the ROC curves and Table I, we can see that the proposed approach is slightly worse than that of [5]. However, this last method is very time consuming and not easy to implement in hardware. Moreover, in the present method, most false positive detections correspond to the circular borderline of the FOV (see Fig. 7), which is void of diagnostic relevance. With respect to [8], the results highlight the effectiveness of the proposed technique, which gives better AUC and accuracy for the same neighborhood size.

REFERENCES

- [1] M. E. Martinez-Perez, A. D. Hughes, A. V. Stanton, S. A. Thom, N. Chapman, A. B. Bharath, and K. H. Parker, "Retinal vascular tree morphology: A semi-automatic quantification," *IEEE Trans. Biomed. Eng.*, vol. 49, no. 8, pp. 912–917, Aug. 2002.
- [2] S. Chauduri, S. Chatterjee, N. Katz, M. Nelson, and M. Goldbaum, "Detection of blood vessels in retinal images using two-dimensional matched filters," *IEEE Trans. Med. Imaging*, vol. 8, no. 3, pp. 263–269, Jun. 1989.
- [3] I. Liu and Y. Sun, "Recursive tracking of vascular networks in angiograms based on the detection-deletion scheme," *IEEE Trans. Med. Imaging*, vol. 12, no. 2, pp. 334–341, Apr. 1993.
- [4] X. Jiang and D. Mojon, "Adaptive local thresholding by verification based multithreshold probing with application to vessel detection in retinal images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 1, pp. 131–137, Jan. 2003.
- [5] J. J. Staal, M. D. Abramoff, M. Niemeijer, M. A. Viergever, and B. van Ginneken, "Ridge based vessel segmentation in color images of the retina," *IEEE Trans. Med. Imaging*, vol. 23, no. 4, pp. 501–509, Aug. 2004.
- [6] L. O. Chua and L. Yang, "Cellular neural networks: Theory," *IEEE Trans. Circuits Syst.*, vol. 35, no. 10, pp. 1257–1272, Oct. 1988.
- [7] T. Roska and L. O. Chua, "CNN Universal Machine: An analogic array computer," *IEEE Trans. Circuits Syst.*, vol. 40, no. 3, pp. 163–173, Mar. 1993.
- [8] C. Alonso-Montes, D. L. Vilarino, and M. G. Penedo, "CNN-based automatic retinal vascular tree extraction," in *Proc. 9th Int. Workshop Cell. Neural Networks and Appl.*, Hsin-Chu, Taiwan, R.O.C., 2005, pp. 61–64.
- [9] R. Dixon and C. Taylor, "Automated asbestos fibre counting," in *Proc. Inst. Phys. Conf. Ser.*, 1979, vol. 44, pp. 178–185.
- [10] R. Zwiggelaar, S. M. Astley, C. R. M. Boggis, and C. J. Taylor, "Linear structures in mammographic images: Detection and classification," *IEEE Trans. Med. Imaging*, vol. 23, no. 9, pp. 1077–1086, Sep. 2004.
- [11] K. Slot, "Large-neighborhood templates implementation in discrete-time CNN universal machine with nearest neighbor connection patterns," in *Proc. IEEE 3rd Int. Workshop Cellular Neural Networks*, 1994, pp. 213–218.
- [12] N. A. Fernández, D. L. Vilarino, V. M. Brea, and D. Cabello, "On the emulation of large neighborhood templates with binary CNN-based architectures," in *Proc. 9th Int. Workshop Cell. Neural Networks Appl.*, Hsin-Chu, Taiwan, R.O.C., 2005, pp. 274–277.
- [13] G. Costantini, D. Casali, and R. Perfetti, "Cellular neural network template for rotation of grey-scale images," *Electron. Lett.*, vol. 39, no. 25, pp. 1803–1805, Dec. 2003.
- [14] D. Y. Aksun, "A high-precision high-resolution WTA—MAX circuit of $O(N)$ complexity," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 49, no. 1, pp. 48–53, Jan. 2002.
- [15] J. Ramírez-Angulo, G. Ducoudray-Acevedo, R. G. Carvajal, and A. López-Martín, "Low-voltage high-performance voltage-mode and current-mode wta circuits based on flipped voltage followers," *IEEE Trans. Circuits Syst. II: Exp. Briefs*, vol. 52, no. 7, pp. 420–423, Jul. 2005.
- [16] M. Forti, "A study on WTA cellular neural networks," *Int. J. Circ. Theor. Appl.*, vol. 29, pp. 537–552, 2001.
- [17] G. Linán-Cembrano, A. Rodríguez-Vázquez, S. Espejo, and R. Domínguez-Castro, "ACEK16K: A 128×128 focal plane analog processor with digital I/O," in *Proc. 7th IEEE Int. Workshop Cellular Neural Networks Applic.*, Frankfurt, Germany, Jul. 2002, pp. 132–149.