MODULAR MORPHOLOGICAL NEURAL NETWORK TRAINING VIA ADAPTIVE GENETIC ALGORITHM FOR DESIGNING TRANSLATION INVARIANT OPERATORS

Ricardo de A. Araújo, Francisco Madeiro, Robson P. de Sousa and Lúcio F. C. Pessoa*[1]

Statistics and Computer Science Department, Catholic University of Pernambuco
Rua do Príncipe, 526, Boa Vista, 50050-900, Recife - PE - Brazil
[1]Freescale Semiconductor, Inc., Austin, Tx, 78729, USA

ABSTRACT

In the present paper, adaptive genetic algorithm (AGA) is used for training a modular morphological neural network (MMNN) for designing translation invariant operators via Matheron decomposition and via Banon and Barrera decomposition. The operators are applied to restoration of images corrupted by salt and pepper noise. The AGA is used to determine the weights, architecture and number of modules of the MMNN. Results in terms of noise to signal ratio show that the method proposed in the present work lead to a better operators performance when compared to other methods previously proposed in the literature.

1. INTRODUCTION

The design of translation invariant operators is a relevant problem in mathematical morphology, with applications in image processing, such as image restoration, edge extraction and object recognition. Many works have focused on the design of morphological operators. Dougherty and Loce [1] designed sub-optimal operators satisfying Matheron theorem [2] for binary image processing. Yang and Maragos [3] designed operators (min-max classifiers) according to Matheron decomposition [2] by using the mean square error for minimizing the cost function. Pessoa and Maragos [4] generalized Yang and Maragos operators [3] for a neural network architecture involving morphological/rank/linear operators. Harvey and Marshall [5] used simple genetic algorithm (SGA) for designing morphological filters for gray level images. Oliveira [6] generalized the work of Harvey and Marshall [5] by implementing Banon and Barrera decomposition [7] via SGA for operators non necessarily linear. Davidson and Hummer [8] used morphological neural networks (MNN) for designing morphological filters, differing from the classical neural networks [9] in the sense that the computation in each node of the MNN is carried out by simple morphological operators in the context of Algebra of Images [10]. Herwing and Shalkoff [11] presented a MNN with learning based on the delta rule for designing filter for binary images. Sousa [12] presented a general network architecture, referred to as modular morphological neural network (MMNN), based on Matheron decomposition [2] and in the more general Banon and Barrera decomposition [7]. The MMNN training is via SGA or via back propagation algorithm, which uses the methodology of Pessoa and Maragos [13] for estimating the derivatives of the training equation. Each module of the MMNN represents a morphological operation: dilation, erosion, anti-dilation and anti-erosion.

The purpose of the present paper is to use adaptive genetic algorithm (AGA) [14] for training the MMNN for designing translation invariant operators via Matheron decomposition [2] for dilations and erosions and via Banon and Barrera decomposition [7] for sup-generators and inf-generators based on the methodology described in Sousa [12]. The AGA is used to determine the weights, architecture and number of modules of the MMNN.

*E-mails: randrade@dei.unicap.br (Ricardo de A. Araújo), madeiro@dei.unicap.br (Francisco Madeiro), robson@dei.unicap.br (Robson P. de Sousa) and lucio.pessoa@freescale.com (Lúcio F. C. Pessoa).

2. BACKGROUND

2.1. Mathematical Morphology

The following equations are used in MMNN for designing translation invariant operators [12]:

$$\text{Dilation: } \delta_k = max\left(\vec{x} + \vec{a}_k\right); \quad (1)$$

$$\text{Erosion: } \epsilon_k = min\left(\vec{x} - \vec{a}_k\right); \quad (2)$$

$$\text{Anti-Dilation: } \delta_k^a = 1 - min\left(\vec{x} - \vec{b}_k\right); \quad (3)$$

$$\text{Anti-Erosion: } \epsilon_k^a = 1 - max\left(\vec{x} + \vec{b}_k\right), \quad (4)$$

where $\vec{x}$ is the input signal and $\vec{a}_k$ and $\vec{b}_k$ represent the structuring element (terms $\vec{b}_k$ represent the reflection of the complement of the structuring elements of anti-dilation or anti-erosion).

2.2. Adaptive Genetic Algorithm

The AGA of Mitsuo and Cheng [14] differs from SGA by using adaptive methods applied to crossover and mutation operators. The method adopted in the present work is the deterministic adaption [14] and consists in modifying the operators rate according to a pre-stablished rule. The operators rates are gradually decreased in each population evolution. The following equation defines the rule adopted as the adaptive parameter in the rates of crossover and mutation:

$$Tx_a = Tx_i - (Tx_i - Tx_f) * \frac{g_a}{G}, \quad (5)$$

where Tx_a, Tx_i and Tx_f represent the current, initial and final rates. The terms G and g_a represent, respectively, the maximum number of generations and the current generation.

2.3. Operators Design by Matheron Decompositon

Sousa [12] uses the MMNN for designing translation invariant operators that satisfy Matheron decomposition theorem [2] for dilations as well as for erosions. The theorem states that each increasing and translation invariant operator may be decomposed by a union or a intersection of erosions or dilations operators. Figure 1 presents the MMNN architecture for Matheron decomposition [2] by dilations. The following equations define the MMNN architecture for Matheron decomposition [2] via dilations according to Sousa [12].

$$v_k = \delta_k = max\Big(\vec{x} + \vec{a}_k\Big), \tag{6}$$

where $\vec{x}$ is the input image of the MMNN.

$$\textbf{Network output: } Y = min(\vec{v}), \tag{7}$$

where

$$\vec{v} = (v_1,\ v_2,\ \dots\ , v_k). \tag{8}$$

The weight matrix, A, of the MMNN is defined by

$$A = (\vec{a}_1,\ \vec{a}_2,\ \dots,\ \vec{a}_k), \tag{9}$$

where $\vec{a}_k \in \mathcal{R}^k$, $k = 1, 2, \dots, N$, represent the MMNN weights. Symbol $\wedge$ represents the minimum operation.

In a dual manner, the architecture for Matheron decomposition [2] via erosions is defined by substituting dilations by erosions and symbol $\wedge$ by $\vee$, where $\vee$ represents the maximum operation.

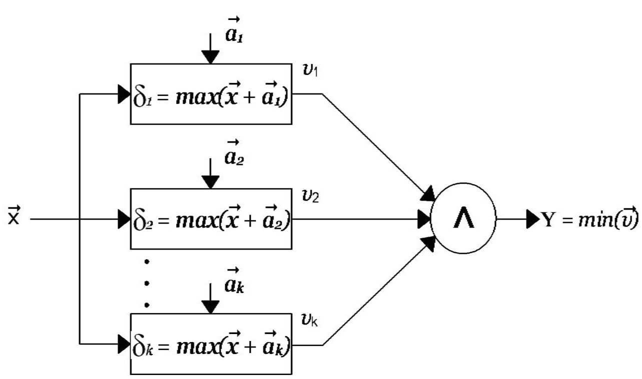

Fig. 1. MMNN architecture used for Matheron decomposition via dilations.

2.4. Operators Design by Banon and Barrera Decomposition

Sousa [12] uses the MMNN for designing translation invariant operators satisfying Banon and Barrera decomposition theorem [7] for sup-generators as well as for inf-generators. The theorem states that each operator, not necessarily increasing, and translation invariant, may be decomposed by a union of sup-generators or intersection of inf-generators. Figure 2 presents the MMNN architecture for Banon and Barrera decomposition [7] via sup-generators. The following equations define the MMNN architecture for Banon and Barrera decomposition [7] via sup-generators according to Sousa [12].

$$u_{k1} = \epsilon_k = min\Big(\vec{x} - \vec{a}_k\Big), \tag{10}$$

$$u_{k2} = \delta_k^a = 1 - max\left(\vec{x} + \vec{b}_k\right). \tag{11}$$

$$\text{Sup-Generator: } v_k = min\left(\vec{u_k}\right),\quad k = 1, 2, \dots, N, \tag{12}$$

where

$$\vec{u_k} = (u_{k1}, u_{k2}),\quad k = 1, 2, \dots, N. \tag{13}$$

$$\textbf{Network output: } Y = max(\vec{v}), \tag{14}$$

where

$$\vec{v} = (v_1,\ v_2,\ \dots\ , v_N). \tag{15}$$

The weight matrices, A and B, of the MMNN are defined by

$$A = (\vec{a}_1,\ \vec{a}_2,\ \dots,\ \vec{a}_N), \tag{16}$$

$$B = (\vec{b}_1,\ \vec{b}_2,\ \dots,\ \vec{b}_N), \tag{17}$$

where $\vec{a}_k$ and $\vec{b}_k \in \mathcal{R}^k$, $k = 1, 2, \dots, N$, represent the MMNN weights. Symbol $\wedge$ represents the minimum operation in the sub-integrators units and symbol $\vee$ represents the maximum operation in the general integrator unit.

In a dual manner, the architecture for Banon and Barrera decomposition [7] via inf-generators is defined by substituting dilations by erosions, anti-dilations by anti-erosions, symbol $\wedge$ by $\vee$ in the sub-integrators units, and symbol $\vee$ by $\wedge$ in the general integrator unit.

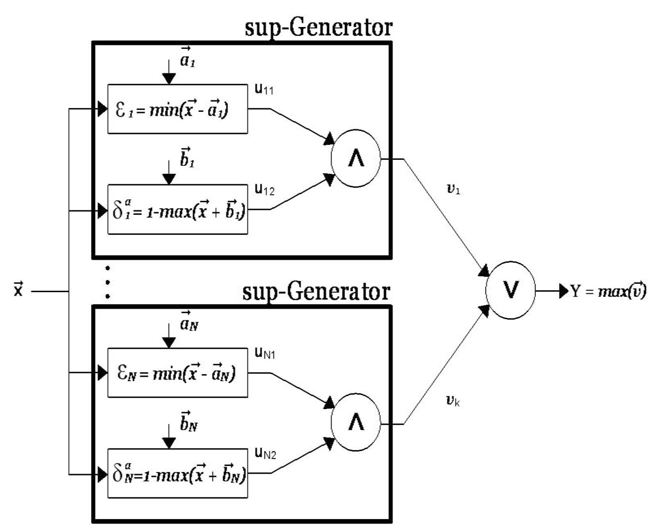

Fig. 2. MMNN architecture used for Banon and Barrera decomposition via sup-generators.

3. PROPOSED METHODOLOGY

The proposed methodology trains the MMNN via AGA for designing translation invariant operators via Matheron decomposition [2] by dilations or erosions and Banon and Barrera decomposition [7] by sup-generators and inf-generators. The main objective is determining the weights, architecture and number of modules of the MMNN for designing translation invariant operators. As an example, Figure 3 represents an element of the AGA population, where $se_{(i)}, i = 1, 2, \ldots, N$, is the set of structuring elements for the design of translation invariant operators. Terms $arch$ and mod represent the architecture and number of MMNN modules, respectively. Table 1 presents an example of coding used for identifying the MMNN architectures. Eight bits are used for determining the number of MMNN modules.

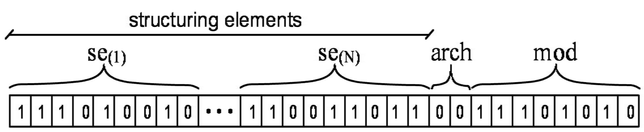

Fig. 3. Coding of the chromosome.

Table 1. Example of code for the MMNN architectures.

Code	MMNN architecture
00	Matheron decomposition via dilations
01	Matheron decomposition via erosions
10	Banon and Barrera decomposition via sup-generators
11	Banon and Barrera decomposition via inf-generators

4. SIMULATIONS AND RESULTS

For training the AGA for designing translation invariant operators, the fitness function, E, is defined by

$$E = D - Y, \tag{18}$$

where D is the desired image and Y is the output of the filter designed by AGA.

The noise to signal ratio (NSR) is used for assessing the performance of the designed operators. It is defined by

$$NSR = 10\, log_{10} \frac{\overline{(D-Y)^2}}{\overline{(D)^2}}, \tag{19}$$

where $\overline{(D-Y)^2}$ and $\overline{(D)^2}$ represent the mean energy of the error (second moment of the error) and the mean energy of the desired output (second moment of the target).

4.1. Application in Restoration of Images

A classical problem in image processing is restoration of images corrupted by noise [15]. The present paper considers salt and pepper noise. The classical median filter [15] is an alternative commonly used for restoring imagens corrupted by that noise.

The purpose of the present paper is the design of translation invariant operators by training the MMNN via AGA for restoring images corrupted by salt and pepper noise.

4.1.1. MMNN Training via AGA

For designing translation invariant operators, the MMNN was trained via AGA with an initial population of 100 elements, maximum number of generations (cicles of the AGA) corresponding to 100, with an interval of adaptive variation $Tx_i = 1.0$ to $Tx_f = 0.5$ for crossover probability and $Tx_i = 0.05$ to $Tx_f = 0.0001$ for mutation probability, according to [14]. In all simulations the noisy image X in Figure 4 (a) was used for training the AGA, considering as the desired output the image in Figure 4 (b).

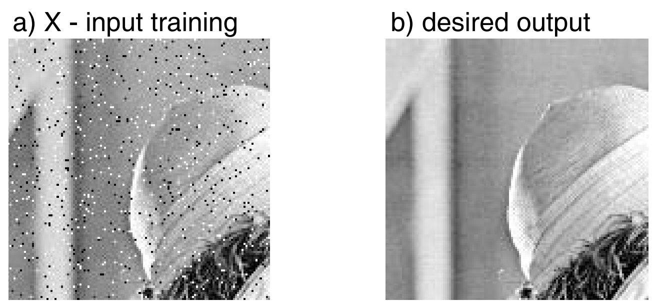

Fig. 4. Images considered. (a) input image for training, (b) desired output.

The following notation is considered in Table 2: MMNN (AGA) denotes the MMNN training via AGA by using predetermined architecture and number of modules; MMNN (AGA-MOD) denotes the MMNN training via AGA by using weights, architecture and number of modules determined by the AGA; MMNN (SGA) denotes the MMNN training via simple (non-adaptive) genetic algorithm. Results of the median filter are also presented in the table. It is observed that for a noisy density corresponding to 5%, MMNN (AGA) and MMNN (AGA-MOD) overperform MMNN (SGA) and median filter. It is worth to mention that MMNN (AGA-MOD) overperforms MMNN (AGA) by using a smaller number of decompositions. MMNN (AGA-MOD) led to NSR=-24.52dB with 22 decompositions, while MMNN (AGA) led to a noise to signal ratio of -24.27dB with 100 decompositions. Figure 5 shows the image corresponding to the operator obtained by MMNN (AGA-MOD) for Matheron decomposition by 22 dilations.

5. CONCLUSIONS

Results have shown that the training of the MMNN via AGA for designing translation invariant operators is more efficient than the median filter and MMNN trained by SGA for restoring images corrupted by salt and pepper noise. Training via MMNN (AGA-MOD) has reduced the number of decompositions and the computational complexity in the operators

Table 2. NSR Results for a noise density = 5%.

Method	Final NSR (dB)
MEDIAN FILTER	-21.55
MMNN(SGA) (8 erosions)	-17.63
MMNN(SGA) (8 dilations)	-19.00
MMNN(SGA) (25 erosions)	-18.40
MMNN(SGA) (25 dilations)	-19.70
MMNN(AGA) (8 erosions)	-22.21
MMNN(AGA) (8 dilations)	-23.22
MMNN(AGA) (25 erosions)	-23.11
MMNN(AGA) (25 dilations)	-23.76
MMNN(AGA) (50 erosions)	-23.27
MMNN(AGA) (50 dilations)	-24.07
MMNN(AGA) (75 erosions)	-23.31
MMNN(AGA) (75 dilations)	-24.13
MMNN(AGA) (100 erosions)	-24.11
MMNN(AGA) (100 dilations)	-24.46
MMNN(AGA-MOD) (22 dilations)	**-24.52**

design when compared to MMNN (AGA). Future works will consider the proposed methodology in image segmentation and pattern recognition.

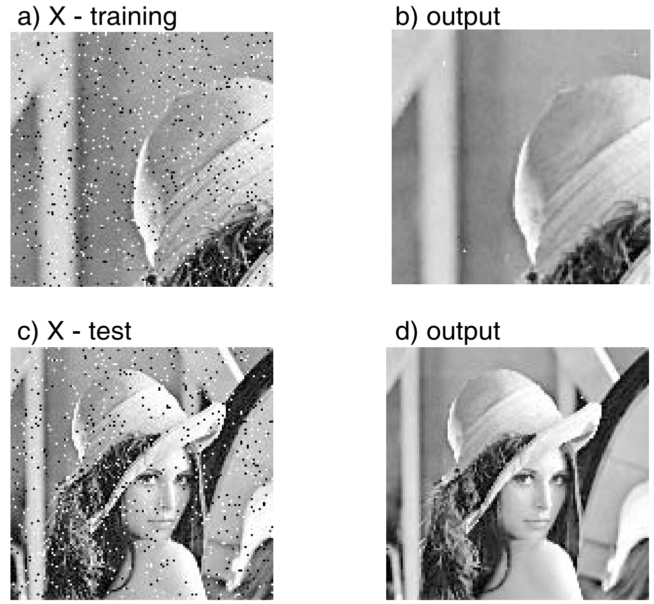

Fig. 5. Training and testing images. Results of MMNN (AGA-MOD) for Matheron decomposition by 22 dilations. (a) input for the training, (b) output by the end of the training, (c) testing input and (d) testing output.

6. REFERENCES

[1] E. R. Dougherty and R. P. Loce, "Efficient design strategies for the optimal binary digital morphological filter: Probabilities, constraints, and structuring-element libraries," in *Mathematical Morphology In Image Processing*, Edward R. Dougherty, Ed., pp. 43–92. Marcel Decker, Inc., New York, 1993.

[2] G. Matheron, *Random sets and integral geometry*, Wiley, New York, 1975.

[3] P. Yang and P. Maragos, "Character recognition using min-max classifiers designed using an LMS algorithm," *Visual Communications and Image Processing*, vol. 92, no. 1818, pp. 674–685, nov. 1992.

[4] L. F. C. Pessoa and P. Maragos, "Morphological rank neural networks and their adaptive optimal design for image processing," in *Proc. of the 1996 IEEE Intl Conference on Acoustics, Speech and Signal Processing*, Atlanta, 1996, IEEE.

[5] N. R. Harvey and S. Marshall, "The use of genetic algorithms in morphological filter design," *Signal Processing Image Communication*, vol. 8, pp. 55–71, 1996.

[6] J. R. de F. Oliveira, *O uso de algoritmos genéticos na decomposição morfológica de operadores invariantes em translação aplicados a imagens digitais*, Ph.D. thesis, INPE, 1998.

[7] G. J. F. Banon and J. Barrera, "Minimal representation for translation invariant set mappings by mathematical morphology," *SIAM J. Appl. Math.*, vol. 51, no. 6, pp. 1782–1798, 1991.

[8] J. L. Davidson and F. Hummer, "Morphology neural networks: An introduction with applications," *Circuits, System and Signal Process*, vol. 12, no. 2, pp. 179–210, 1993.

[9] S. Haykin, *Neural networks: A comprehensive foundation*, Prentice Hall, New Jersey, 1998.

[10] G. X. Ritter, "Recent developtments in image algebra," in *Advances in Electronics and Eletron Physics*, vol. 80. Academic Press, 1991.

[11] C. B. Herwing and R. J. Shalkoff, "Morphological image processing using artificial neural networks," in *Control and Dynamic Systems, Vol. 67*, C. T. Leondes, Ed., pp. 319–379. Academic Press, 1994.

[12] R. P. de Sousa, J. M. de Carvalho, F. M. de Assis, and L. F. C. Pessoa, "Designing translation invariant operations via neural network training," in *Proc. of the 2000 IEEE Intl Conference on Image Processing*, Vancouver, Canada, 2000, IEEE.

[13] L. F. C. Pessoa and P. Maragos, "Neural networks with hybrid morphological/rank/linear nodes: A unifying framework with applications to handwritten character recognition," *Pattern Recognition*, vol. 33, pp. 945–960, 2000.

[14] M. Gen and R. Cheng, *Genetic algorithms and engineering optimization*, John Wiley, New York, 2000.

[15] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison-Wesley Publishing Company, New York, 1992.