

# Rough Neural Networks in Adapting Cellular Automata Rule for Reducing Image Noise

Yasser F. Hassan

**Abstract**—The reduction or removal of noise in a color image is an essential part of image processing, whether the final information is used for human perception or for an automatic inspection and analysis. This paper describes the modeling system based on the rough neural network model to adaptive cellular automata for various image processing tasks and noise remover. In this paper, we consider the problem of object processing in colored image using rough neural networks to help deriving the rules which will be used in cellular automata for noise image. The proposed method is compared with some classical and recent methods. The results demonstrate that the new model is capable of being trained to perform many different tasks, and that the quality of these results is comparable or better than established specialized algorithms.

**Keywords**—Rough Sets, Rough Neural Networks, Cellular Automata, Image Processing.

## I. INTRODUCTION

THERE is a various kind of applications in many scientific fields involving image processing. For instance, image enhancement is the processing of images to improve their appearance to human viewers or to enhance the performance of other image processing system [4], [5]. In most applications involving images or image processing one of the most common problems is the presence of noise. Noise can be systematically introduced into digital images, e.g. due to the circumstances of recording, transmission, scanning, etc. The objective of image enhancement (for example improving image quality, intelligibility, visual appearance) is dependent of application context. An image enhancement algorithm that performs well for one class of images may not perform as well for other classes. Classically, image enhancement is formulated in either spatial or transform (basically the Fourier transform) domains. One of the most used spatial domain techniques is that of the so-called convolution masks [6]. Such an example may be the Gaussian filter. In transform domains, perhaps the most famous technique is the Wiener filter [7].

The object space in an image is modeled as a finite space of the Euclidean space [10]. Once the object's image is captured via an imaging system, the model of underlying space is a finite subset of the Cartesian coordinates. In case of the binary images, the image at the intensity coming from the area of pixel  $(i,j)$  and if it is above a certain threshold, the corresponding pixel  $(i,j)$  is assumed to be present; otherwise it is assumed to be absent.

Yasser F. Hassan is with the Department of Mathematics & Computer Science, Faculty of Science, Alexandria University, Egypt (e-mail: y.fouad@sci.alexu.edu.eg).

A Cellular Automaton (CA) [2], [11], as the term is used in this paper, in their simplest form, are made up from a regular grid of cells, each of which can be in one of a finite set of states. At a given time step, the state of each cell is updated in parallel and is determined as a function of the values in the cell's neighborhood during the previous time step, i.e. by a set of cell state transition rules. CA are discrete dynamical systems, and their simplicity coupled with their complex behavior has made them popular for simulating complex systems is a discrete state system consisting of a countable network of cells that interact with their neighbors. Although each cell generally only contains a few simple rules, the combination of a matrix of cells with their local interaction leads to more sophisticated emergent global behavior [3]. That is, although each cell has an extremely limited view of the system (just its immediate neighbors), localized information is propagated at each time step, enabling more global characteristics of the overall CA system. This can be seen in many examples such as Conway's Game of Life.

Cellular computing architectures represent an important class of computation that are characterized by simple processing elements, local interconnect and massive parallelism [8], [9]. These architectures are a good match for many image and video processing applications. From this idea, the paper describes the development of cellular automata model that can be used to support the assessment of design performance in the rough sets and neural network model. Cells represent objects or pixels with their own behavior, think using rough set methods and take the structure of neural network lattice.

The paper is organized as follows: Section II details the basic design of proposed model. Section III provides the details of some illustrative applications of proposed approach. Finally, the paper is concluded in Section IV.

## II. HYBRID ROUGH SETS, NEURAL NETWORKS AND CELLULAR AUTOMATA

This section describes the overall design of proposed model as shown in Fig.1. A digital image is a bi-dimensional array  $Z$  of  $(M \times M)$  pixels. Each pixel can be characterized by the triplet  $(i; j; s)$  where  $(i; j)$  represents its position in the array and  $s$  the associated color. The two-dimensional cellular automata model consists of array of cells  $x(i, j)$ . The image will be considered as a particular configuration state of a cellular automaton that has cellular space  $M \times M$  array defined by the image. Each site in the array corresponds to a pixel. The use of a cell as an image pixel provides an even greater amount of flexibility to the ability and configuration of the

system itself. The neighborhoods of the cell  $x(i, j)$  are the von-Neumann neighborhood of radius  $r$ , i.e.

$$N_r(i, j) = \{x(m, n) : |m - i| + |n - j| \leq r\} \quad (1)$$

where  $N$  is the neighborhood relation function of distance  $r$ . If we assign  $r = 1$ , then the neighbors of a cell  $x$  are the eight orthogonal adjacent cells plus the cell  $x$  itself.

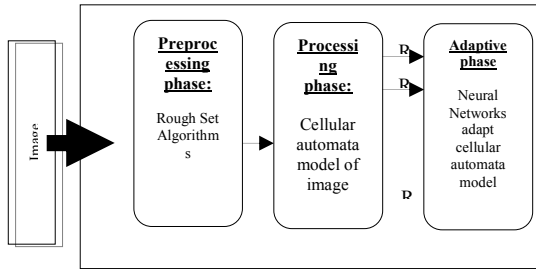


Fig. 1 The proposed model is composed of three fundamental building processes: pre-processing, processing using cellular automata and neural networks to adapt cellular automata results

The pixel at cell  $x$  is called grid pixel if and only if all neighborhood cells exist. Otherwise is called a boundary cell. At any given instant, the cell is assumed to be in one state of the set of states (color)  $S$ . Then abstractly, each cell state function is

$$f : S \times A \rightarrow S \quad (2)$$

where  $A = \{a_1, a_2, \dots\}$  is a set of actions. The intuition is that a cell decides what action to perform based on its state and its environment states. I.e. the cell takes its current state and an action and maps them to a set of states:  $S$ ; those that are called result from performing action  $a \in A$  in state  $s \in S$ .

A pixel color (cell state) is represented by a state in  $\{0, 1, \dots, g\}$  where  $g = 2$  for monochromatic image;  $g=16$  for an image with 16 colors, 0' state is the quiescent state associated to the cell. The cells disposed outside the lattice of  $M \times M$  pixels are assumed to be in the quiescent state.

The approximation space  $Z$  is the image space (i.e. set of Cartesian coordinates) and the original space  $X$  is the object space (i.e. the Euclidean space). Then the pixel size would be different for different camera resolutions and thus led to different representations (approximations) of the same object. The larger the pixel size, the more details is lost in the representation. It is assumed that an image can be classified according to three main parameters:  $A$ ,  $B$  and  $C$ .  $C$  is an amplification factor that affects the size of membership functions resulting in  $A$  and  $B$  varying between 0 and 255.

$$A = |x(i-1, j, o) - x(i+1, j, o)|$$

$$B = |x(i, j-1, o) - x(i, j+1, o)|$$

$$C = [\max(i, j, o) x(i, j, o) - \min(i, j, o) x(i, j, o)] xC / \sum(i, j, o) x(i, j, o) \quad (3)$$

Each image has an associated decision table that defines the combinational actions in its pixels. Hence, the decision table defines the behavior of the pixels and is responsible for state changes.

Most of literature on cellular automata studies the effect of applying manually specified transition rules. The inverse problem of determining appropriate rules to produce a desired effect is hard. The issue with training the CA is the high dimensionality of the search space of possible rules [9]. Evolutionary methods appear to be preferred; for instance to solve the density classification task researchers have used genetic algorithms and genetic programming. Instead, this paper uses a deterministic approach, it proposes an adaptation of dynamical transition rules for cellular automata model based on neural networks algorithm and rough sets. Such that given a noisy image pixels as initial configuration these rules produce a trajectory whose final configuration corresponds to a noise-reduced version of the image [1]. They are desirable that the dynamics be applicable to any kind of images without distinction (monochromatic, gray level or color).

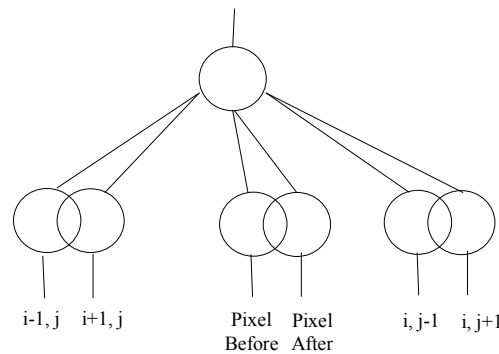


Fig. 2 RNN model to adapt the pixel value after cellular automata process

To adapt the transition rule of cellular automata, rough neural network model is constructed as in Fig. 2. The desired pixel and its neighbors will be considered as input values for rough neural network model (RNN). Using RNN for random pixels  $p=1,2,3,\dots,M \times M$  and the decisions are observed as the output of RNN model so modify the transition rule by this output. The RNN cases for any  $x_i \in Z$  are calculated as follows: first the  $p$  points in  $Z$  closest to  $x_i$  are determined and then the membership of pattern  $x_i$  to class  $C_j$  is calculated through the relative frequencies:

$$L(x_i)_j = k_j / k \quad (4)$$

where  $k_j$  is the number of elements  $x \in Z$  amongst the  $k$  closest neighbors to  $x_i$  which are labeled with classes  $j$ . The image pixel size would be different for different camera resolutions and thus led to different representations

(approximations) of the same image. The larger the pixel size, the more detail is lost in the representation.

Four pixels around the processed pixel must be read before state may be extracted. Then the extracted RNN model is embedded in the position of each processed pixel. The step of categorization of the input attribute involves converting the attributes from numerical to categorical.

Consider  $Z$  to be an RGB image of size  $M \times M$  consisting of three primary components, red  $R$ , green  $G$  and blue  $B$ . For a  $P \times Q$  neighborhood around a pixel  $x(i, j)$ , the total distance of the entire pixel in the neighborhood and the pixel  $x(i, j)$  is given by:

$$E(i, j) = \sum_p \sum_q F(x(i, j), x(p, q)) \quad (5)$$

The pixels in the neighborhood fall in the sphere of the similar color if the distance  $E(i, j)$  is less than  $\epsilon$ . Where  $F$  is the Euclidean distance between two pixels  $x(m, n)$  and  $x(p, q)$  which is given by:

$$F(x(m, n), x(p, q)) = \sqrt{(x(m, n, R) - x(p, q, R))^2 + (x(m, n, G) - x(p, q, G))^2 + (x(m, n, B) - x(p, q, B))^2} \quad (6)$$

The value of  $E$  is calculated for each pixel and each of four neighbor pixels to know the output of the pixel from cellular automata processing step. Each of the  $N$  pixels in the system starts; its desired decision will be determined according to cellular automata. The minimal decision algorithm can be described easily by rules such as "IF conditional part THEN conclusive part (decision)," and used by each object. This leads to dynamic knowledge bases, in which the active sets of RNN models are embedded in the set of the pixels.

One way to construct a simpler model computed from pixels, easier to understand and with more predictive power is to create a set of minimal number of RNN model with each cell in the model of cellular automata.

This method can be considered as semi-unsupervised algorithm. The basic procedure of an unsupervised learning method involves grouping inputs together according to similarity or indiscernibility. This process discovers cases in which a set of pixels typically co-occur and thus tends to  $n^{\text{th}}$ -order associations between pixels. The method can be visualized as an information system building operation which alternates between a statistical exploitation phase in which new competitive objects or pixels are produced and a relational exploitation phase in which new RNN models are produced. This learning method is fully incremental. Objects and rules are added to the information of each image until the end of time iterations.

### III. APPLICATION AND RESULTS

We now turn attention to the model itself and study the performance of the proposed noise detector and remover. This takes the form of a simulation program. We carried out experiments using standard images of  $256 \times 256$  pixels size. The images were corrupted with salt and pepper impulse

noise. In an image that is corrupted with salt and pepper noise, two things can happen to each of the components of a pixel: it remains unchanged, it gets a value 0 or it changes to value  $2k - 1$  where  $k$  is the number of states (color). The corrupted images were treated by noise detection and removal system. The phenomenon is generated while doing so include simulations of models building, learning processes and interactions between pixels in the system.

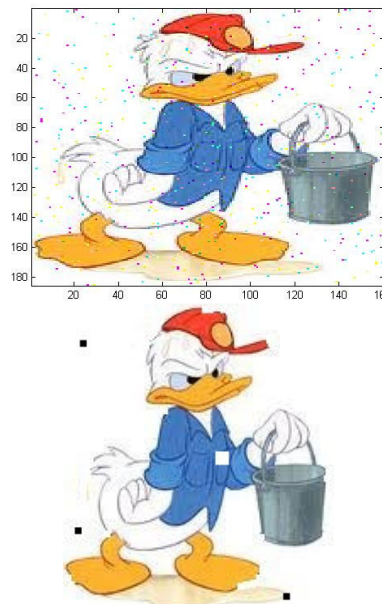


Fig. 3 (a) The noisy image and (b) The reduced-noise image

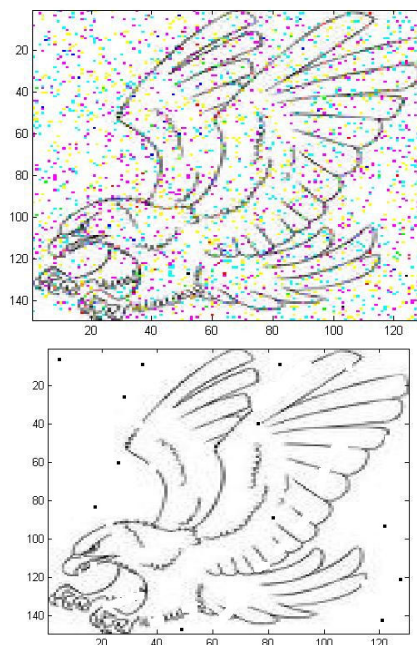


Fig. 4 (a) The noisy image and (b) The reduced-noise image

The proposed model is applied to some noisy image as in Figs.3 and 4. The performance of the noise detector is the measure of its ability to exactly find the noisy pixels. To analyze the performance of the proposed model, test image is corrupted by various noise densities (10 – 70% in steps). To measure of objective similarity between a filtered image and the original one it is using the noise ratio. The noise ratio is a ratio between the number of noisy pixel and the original noise in the image. The number of truly identified noisy pixels decreased drastically with the increase of noise density.

The proposed model has provided a direct approach to studying how dynamical systems perform emergent computation; that is, how the interaction of simple components (pixels) with local information storage gives rise to coordinated global information processing. Whether in real-life situation, the topology of the interconnections that gives meaning to the term immediate neighbor can change frequently. Every pixel participating in the system must be able to communicate with its immediate neighbors. The state transition within each image could be identical throughout the system or unique to each image.

#### IV. CONCLUSION

This paper has described a new approach to building new cellular automata systems such that they can be applied to process grey level and colored images while still enabling good rule sets to be learnt in a practical manner. The proposed model has provided a direct approach to studying how dynamical systems perform emergent computation; that is, how the interaction of simple components with local information storage and communication gives rise to coordinated global information processing. Whether in real-life situation, the topology of the interconnections that gives meaning to the term immediate neighbor can change frequently. Although every pixel participating in the system must be able to communicate with its immediate neighbors, the system itself should not depend on knowledge of the overall system topology. The state transition within each cell could be identical throughout the system or unique to each pixel. In practice, the state transitions within the cells can most conveniently be viewed as shared by all cells, but with local adaptations as a function of either static or dynamic local conditions. Because the model uses processes, which are fully incremental, it is able to show how a pixel/image interaction can modulate and guide an underlying evolutionary process. Unfortunately, in its present manifestation the model has many shortcomings. As a computational implementation, it is not as robust as one would like. However, it is hoped that all these deficiencies will be remedied in the ongoing development of this work.

#### REFERENCES

- [1] Kaliraj, G., Baskar, S., An efficient approach for the removal of impulse noise from the corrupted image using neural network based impulse detector, *Image and vision computing*, 28, pp. 458-466, 2010.
- [2] Yasser Hassan, Rough Set Adaptive in the Model Based of Cellular Automata and Multi-Agents, *Journal of Emerging Trends in computing and Information Sciences*, Vol.2 No.8, 2011.
- [3] Yasser Hassan, rough sets for adapting wavelet neural networks as a new classifier system, *Springer applied intelligence journal* (35), pp. 260-268, 2011.
- [4] Li H., Liao X., Li C., Huang H., and Li Ch., Edge detection of noisy images based on cellular neural networks, *Common nonlinear sci number simulat* 16, pp. 3746-3769, 2011.
- [5] Lin T., Decision-based fuzzy image restoration for noise reduction based on evidence theory, *Expert system with applications* 38, pp. 8303-8310, 2011.
- [6] Musgrif M. and Ray A., Color image segmentation: rough-set theoretic approach, *pattern recognition letters* 29, pp. 483-493, 2008.
- [7] Paul L. Rosin, Image processing using 3-state rough set, *Computer Vision and Image Understanding* 114, 790-802, 2010.
- [8] Porter, R., Frigo, J., Conti, A., Harvey, N., Kenyon, G. and Gokhale, M., A reconfigurable computing framework for multi-scale cellular image processing, *Microprocessors and Microsystems* 31, 546-563, 2007.
- [9] Rosin, P., Image processing using 3-state cellular automata, *Computer vision and image understanding*, 114, pp. 790-802, 2010.
- [10] Sinha D. andLaplante P., A rough set-based approach to handling spatial uncertainty in binary images, *Engineering applications of artificial intelligence* 17, 97-110, 2004.
- [11] Terrazas, G., Slepman, P., Kendall, G., and Krasnogor, N., An evolutionary methodology for automated design of cellular automaton-based complex systems, *Journal of rough set*, vol. pp. 77-102, 2007.