Training Cellular Automata with Extended Neighborhood for Edge Detection

Safia Djemame[a]

[a]Computer Science Department, College of Sciences, Ferhat Abbes University, Setif-1, Setif 19000, Algeria

Abstract

Edge detection refers to the process of identifying and locating sharp discontinuities in an image. Since edge detection is in the forefront of image processing for object detection, it has attracted much attention from scientific research. More accurate results and less time consuming are there still the main issues when extracting edges from images. To cope with this challenge, we propose a complex system: Cellular Automata (CA) that has proven high performances in image processing domain. Unlike previous works, which used in majority Von Neumann or Moore neighborhood, We use a particular kind of CA, with extended Moore neighborhood. This allows a large exploration of the search space. We trained a QPSO algorithm for extracting the adequante subset of rules. Experiments were carried on several images from Mathworks and Berkeley dataset. Visual and numerical results show that our CA provides excellent performances, and edges with high accuracy.

Keywords

Artificial intelligence, Complex systems, Cellular Automata, Rule selection, Image Processing, Edge detection

1. Introduction

Edge detection is one of the fundamental image processing tasks that has been widely investigated since technology allowed people to digitally process visual data. Information about edges is the basis of many computer vision systems such as object recognition, pattern classification, robotic vision and medical diagnosis[1] . The quality of detected edges has a direct and high influence on the performance of mentioned systems.

There are many ways to perform edge detection. However, the majority of methods may be grouped into two categories [2]:

* Gradient-based edge detection: the gradient method detects the edges by looking for the maximum and minimum in the first derivative of the image.

* Laplacian-based edge detection: the Laplacian method searches for zero crossings in the second derivative of the image to find edges. An edge has the one-dimensional shape of a ramp, and calculating the derivative of the image can highlight its location [3] .

ICCSA'2021: International Conference on Complex Systems and Applications, May 25–21, 2021, Oum-el-Bouaghi, Algeria

djemame@univ-setif.dz (S. Djemame)

The main techniques used in the literature for edge detection are Canny [4] , Sobel, Deriche, Prewitt, Roberts edge detectors and Laplacian of Gaussian (LoG).
Given the importance of edge detection, many different methods were introduced to overcome different problems that occur in classical methods. Our research focuses on Cellular Automata, a complex system that presents interesting properties in solving image processing problems, due to its simplicity and local interactions.
This study focuses on edge detection based on Cellular Automata. Cellular Automata, introduced by John Von Neumann [5] in the 1950's, is a spatially and temporally discrete dynamical system composed of cells arranged in a lattice. Each cell can be in one of a finite number of states. The transition between states is dependent on the cell value, the state of its neighbourhood and the transition rule. The most characteristic feature of CA is that using simple rules interacting with a local neighbourhood can produce very complex behaviour.
Image is viewed as a two dimensional CA model with initial configuration in which each pixel is represented by a cell and the pixel value is represented by the state of the cell. So, any updating rule can be applied once in an image at a particular time and its intensity of the pixels change simultaneously in the successive time interval. Due to this kind of behavior, CA model influences a large application in image processing.
Based on the above idea, many algorithms have been developed for edge detection, however this is still a challenging and an unsolved problem.
In this paper, we propose a CA with extended Moore neighborhood, which allows to have larger set of transition rules to explore. Among these rules, we extract the subset which gives better results for edge detection than some traditional methods.
The remaining parts of this paper are organized as follows. In Section 2, we present a brief overview of CA. Related works are presented in Section 3. In section 4, we explain the proposed method. In section 5, we present and discuss the obtained results for a set of images. Finally the paper is concluded in Section 6.

2. Overview of Cellular Automata

Cellular automata (CA) are mathematical models for systems consisting of large numbers of simple identical components with local interactions. The simple components act together to produce complex emergent global behavior [6] .
Cellular automata perform complex computation with high degree of efficiency and robustness. They are especially suitable for modeling natural systems that can be described as massive collections of simple objects interacting locally with each other [7]. Cellular automata is called cellular, because it is made up cells like points in the lattice and it is called automata, because it follows a simple local rule [8] . Each cell can assume a state from finite set of states. The cells update their states synchronously on discrete steps according to a local rule. The new state of each cell depends on the previous states of a set of cells, including the cell itself, and constitutes its neighborhood [9].
Formally, Cellular Automata is quadruples (d, S, N,I):
* The integer d is the dimension of the space the CA will work on,
* S = $s_1, s_2, ..., s_k$ is a finite set of states,

* The neighborhood N is a v-tuple of distinct vectors of Z^d * N = ($x_1, x_2, ..., x_v$): the Xi 's are the relative positions of the neighbor cells with respect to a given center cell,
* f : $S^v \to S$ is the local transition rule.

The states of all cells in the lattice are described by a configuration. A configuration can be described as the state of the whole lattice. The rule and the initial configuration of the CA specify the evolution of CA that tells how each configuration is changed in one step.

The reason behind the popularity of cellular automata can be traced to their simplicity, and to the enormous potential they hold in modeling complex systems. Cellular automata can be viewed as a simple model of a spatially extended decentralized system made up of a number of individual components (cells). The communication between cells is limited to local interaction. Each individual cell is in a specific state which changes over time depending on the states of its local neighbors. The overall structure can be viewed as a parallel processing device. However, this simple structure when iterated several times produces complex patterns displaying the potential to simulate different sophisticated natural phenomena[7].

2.1. Neighborhood structure

The neighborhood of a cell, called the core cell (or central cell), made up of the core cell and those surrounding cells whose states determine the next state of the core cell. There are different neighborhood structures for cellular automata.The two most commonly used neighborhoods are Von Neumann and Moore neighborhood, shown in "Fig. 1".

Von Neumann neighborhood has five cells, consisting of the cell and its four immediate non-

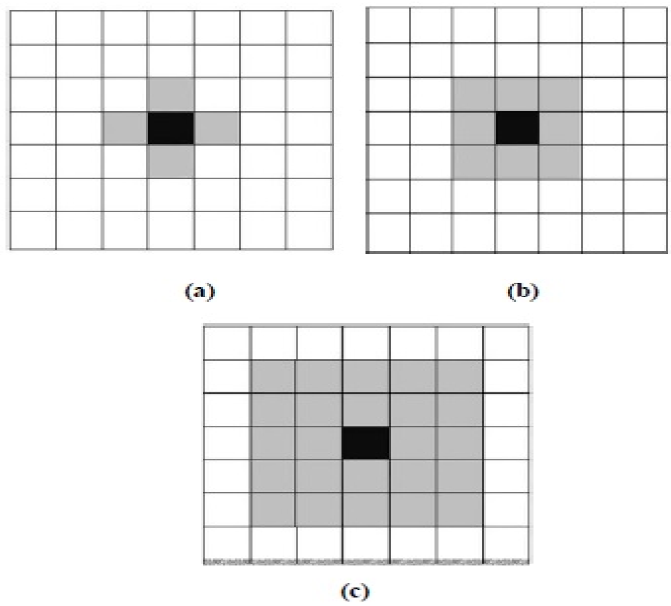

Figure 1: Neighborhood models (a)Von Neumann (b) Moore (c) Extended Moore.

diagonal neighbors and has a radius of 1. The radius of a neighborhood is defined to be the maximum distance from the core cell, horizontally or vertically, to cells in the neighborhood.

Moore neighborhood has nine cells, consisting of the cell and its eight surrounding neighbors and has a radius of 1. Extended Moore neighborhood composed of the same cells as the Moore neighborhood, but the radius of neighbourhood is increased to 2. In this paper, we explore the abilities of a 2D-CA with extended Moore neighborhood (25 cells), to extract efficiently edges on an image.

2.2. Relationship of 2D CA with image

An image may be described as a two-dimensional function I, $I = f(x, y)$, where x and y are spatial coordinates. Amplitude of f at any pair of coordinates (x, y) is called intensity I or gray value of the image. When spatial coordinates and amplitude values are all finite, discrete quantities, the image is called digital image. The digital image I is represented by a single 2-dimensional integer array for a gray scale image and a series of three 2- dimensional arrays for each color bands. As the digital image is a two-dimensional array of $m * n$ pixels, so we are interested in two- dimensional CA model. An image is viewed as a two dimensional CA where each cell represents a pixel in the image and the intensity of the pixel is represented by the state of that cell. The color values of the pixels are updated synchronously at a discrete time step. So very less time is required to solve any image processing task.

3. Related Works

The most characteristic feature of CA is that using simple rules interacting with a local neighbourhood can produce very complex behaviour. Many researchers have investigated the possibility of using Cellular Automata for image processing. [10],[11] and a few focused on edge detection with either binary, greyscale or colour images as inputs. The linear set of rules applied to binary images were recently investigated by several authors. Qadir and Khan [12] divided all 512 rules for Moore neighbourhood into three groups depending on their ability to detect edges. However, they did not cover the different behaviour of rules or compared them. Uguz et. al. [13] focused on the benefits of implementation of the transition function in the form of matrix multiplication. They have presented the results for four rules but focused more on the speed benefits.

Aydogan [14] formulated a cellular neural network based edge detection of 2D data. Diwakar et. al. [15] presented an application of totalistic rules with Moore neighbourhood for edge detection. Wongthanavasu and Sadananda [16] proposed a Weighted Cellular Automata method (WCA) based on von Neumann neighbourhood that can deal with both binary and greyscale images and can be implemented efficiently and it does not require selection of rules or any user input. Djemame et. al. [17] presented a method using a Continuous Cellular Automata for edge detection.

Chang et. al. [18] proposed a method, where an Orientation Information Measure is used to process a greyscale image into binary, and then a Cellular Automata with semi-neighbourhood is used to detect edges. Chen and Yan [19] presented a method that combines the diffusion model with CA.

Recently, a lot of attentions were attracted to the work of Rosin [20]. He proposed to use a Sequential Floating Forward Search (SFFS), which is a deterministic algorithm, to search for the

best set of rules that would allow performing different tasks like denoising, thinning and finding the convex hull. Later he proposed an extension of his method [21] to tackle edge detection. This method can generate edge intensity images with simultaneous removal of impulse noise. However, this method is relatively time-consuming since processing has to be done for a set of 255 images.
In opposite to deterministic SFFS, a heuristic can be applied, with most interest presented in Genetic Algorithms. An example of searching for an optimal packet of rules is presented in work of Batouche et. al. [22] and Slatnia et. al.[23]. Similarly to Rosin's method, they searched for a set of rules that would change the central pixel state, but they did not restrict them to central white pixel. Their publications claim that great results can be obtained using only a single rule. Djemame and Batouche [24] used Particle Swarm Optimization heuristic to determine the best rules without enumerating the complete search space.
Apart from simple cellular automata, Fuzzy Cellular Automata-based edge detector has also been studied because it incorporates fuzzy logic into transition rules, which results in a good performance when used for greyscale images [25]. Patel and More incorporated fuzzy logic and cellular learning automata [26]. Sinaie et. al. presented a method for enhancement of 65 edges acquired by fuzzy edge detector [27]. Some others CA-based methods have been developed to focus on grey and color image [28],[29],[30].

4. Proposed Approach

This work target finding a discrete uniform CA that performs edge detection. A choice is established to use a CA as model, its lattice has as initial configuration the input image, and in one-time iteration the computed lattice represents the output edge image. The reason behind these choices is to attend an edge detector of fast performance. Unlike the precedent works which use Von Neumann or Moore neighborhood, we explore the use of CA with extended Moore neighbor model (of radius r=2).

4.1. Neighborhood Configuration

Our binary CA has 25 neighbor cells (extended Moore neighbor model), so the number of possible neighbor configurations is $2^{25} = 33554432$. The possible number of CA that can be conceived using combination of these configurations is $2^{33554432}$. Our problem is to find the rules that perform edge detection the best within this large number of CA. A choice is established to only use CA with linear rules (only one neighbor configuration rule for each CA) which is beneficial in terms of computational performance of the CA and the substantial reduction in the search space to 33554432.
The rule convention shown in "Fig. 2" is used to designate rules. The number within each box represents the rule number associated with a neighbor configuration that only has that particular neighbor. So, if the next state of a cell depends only on its present state, it is represented as Rule 1. Similarly, if the next state of a cell is dependent only on its bottom neighbor, then it is represented as Rule 8 and so on. These twenty-five rules are known as fundamental/basic rules. All linear rules are derived using these basic rules which are expressed as the sum of the basic

1048576	2097152	4194304	8388608	16777216
524288	64	128	256	512
262144	32	1	2	1024
131072	16	8	4	2048
65536	32768	16384	8192	4096

Figure 2: The rule convention model

rules. For example, Rule 71, Rule 130, Rule 262176 can be expressed as follows:

$$Rule71 = Rule64 \oplus Rule4 \oplus Rule2 \oplus Rule1$$
$$Rule130 = Rule128 \oplus Rule2$$
$$Rule262176 = Rule262144 \oplus Rule32$$

Likewise, we can express all the possible 33554432 linear rules CA.

4.2. The Transition Rule

The transition function of our CA switches the state of each white cell to black only if the neighbor cells defined by its rule are white. The black cells are unchanged. The boundary conditions for our CA is set arbitrarily to use white boundary conditions, where all border cells are considered white. Later experimentation proven that this choice produces better results. The steps implementing this CA are:

- Iterate on each pixel of the input image, if this pixel is white,
- iterate on each of its neighbors and check if all of them are white.
- If all of them are white it changes the pixel to black.

It is found that a code optimization can be applied to the algorithm. The value of the output pixel can be computed using “(1)”:

$$P_{t+1} = P_t \wedge (\neg N_1 \neg N_2) \neg N_n \tag{1}$$

P_{t+1} is the new logical value of the pixel (True for white, False for black).
P_t is the original value of the pixel.
n is the number of rule neighbors.
N_i is the value of the N_i neighbor.
From this we can deduce the matrixial logical calculation of the whole image “(2)”:

$$\begin{aligned} Img_{t+1} = Img_t AND NOT(Shift(Img_t, N_1), ..., \\ NOT(Shift(Img_t, N_n))) \end{aligned} \tag{2}$$

Img_t is the input image as a logical matrix. Img_{t+1} is the output image as a logical matrix.
AND, OR are matrixial logical operators that do the respective logical operation ∧,∨ between each two corresponding elements of the matrixes.
NOT is matrixial logical operators that do logical ¬for each element of the matrix.
Shift(Img_t, N_i) is a function that returns the matrix Img_t shifted in both x,y direction by x_i, y_j the positions of the neighbor i relative to the core cell.

4.3. The method

In this section, the proposed method is discussed. A digital image is assumed to be a two dimensional array of $(m * n)$ pixels, each with a particular gray value or color. An image can be considered as the lattice configuration of a 2D CA where each cell corresponds to an image pixel, and the possible states are the different gray values or colors.
As the possible values of transition rules are a huge number (about $2^{33554432}$), we used an optimization algorithm based on a quantum PSO metaheuristic (QPSO), to extract the subset of rules, capable to give a satisfactory result. The algorithm is detailed in [31].

We take advantage of the calculating faculties of the CA, to transform the initial configurations defined by a numerical image lattice as discrete input data in order to find its edges. The search space is defined by all the transition rules of the CA. The evolutionary process trained by QPSO has the effect of extracting the subset of rules which leads to an edge detection with good quality. In this context, a rule is a particle of the swarm, and the best rule which gives rise to a good segmentation corresponds to the particle with the best fitness.
In this algorithm, the input image and the ground-truth image are uploaded. The QPSO process is initialized by setting the number of iterations and the swarm size. At the beginning of the process, the value of parameter beta is set to 1.0; then, it is linearly decreased during the execution of the algorithm. Beta is the only parameter tuned automatically in the QPSO process. QPSO has relatively better performance by varying linearly the value of beta from 1.0 to 0.5 in order to balance between exploration and exploitation. The particles are randomly initialized in the search space. Each particle of the swarm (a rule) is converted in binary representation and applied on the input image pixel by pixel, according to the transition function defined in (Eq. 2). For each particle, an output image is obtained. The quality of edge detection is measured by evaluating two fitness functions: SSIM and RMSE. The best position is identified. The mean of the best positions mbest is computed. For each particle,the new position X is computed. The fitness of the new particle is evaluated, and the new rule is applied on the image. The process is repeated until reaching a predefined maximum number of iterations. At the end of the algorithm, the best rule, the best segmentation and the best fitness are displayed.

5. Experimental Results

In this section, we present the results of the evolutionary algorithm in the search process, and the results of the extracted rules on different images.
Experiments were carried on several test images from Mathworks and the database of Berkeley University. In this paper, we illustrate some examples: Cameraman, Lena, X-ray images from Mathworks and swan, church, woman, bird and plane images from Berkeley database.

After dozens of experiments, we observed that six rules appeared most frequently than others and allowed extracting good edges after only one application on the input image. The rules identified were rule 38, rule 42, rule 175, rule 935, rule 1273, rule 1511. It is important to note that once the best rules emerge, they may be directly applied to an image, quickly leading to the desired result.

The results in "Fig. 3","Fig. 4" and "Fig. 5", clearly demonstrate that rules 38, 42, 175, 935, 1273, and 1511 extracted by the QPSO algorithm, produce satisfactory results. Edges are continuous, clean and fine. They are one pixel wide. The external contour is accurate, continuous and without noise. The rules provide good edges with a fine level of details.

Figure 3: Edge detection of Cameraman image. a) Original image b) Binary image c) Rule 38 d) Rule 42 e) Rule 175 f) Rule 935 g) Rule 1273 h) Rule 1511

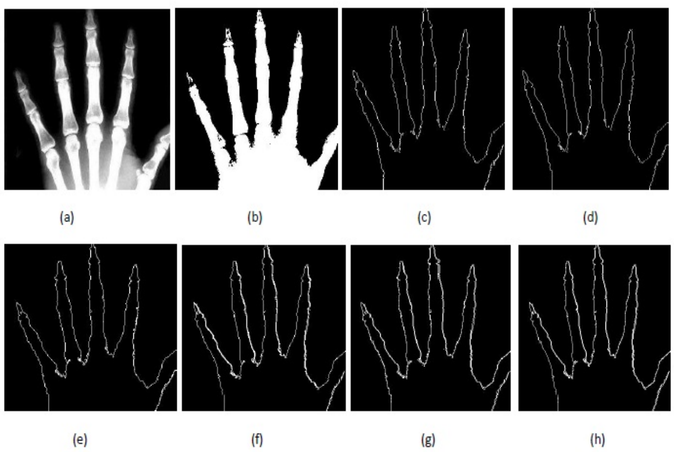

Figure 4: Edge detection of X-ray image. a) Original image b) Binary image c) Rule 38 d) Rule 42 e) Rule 175 f) Rule 935 g) Rule 1273 h) Rule 1511

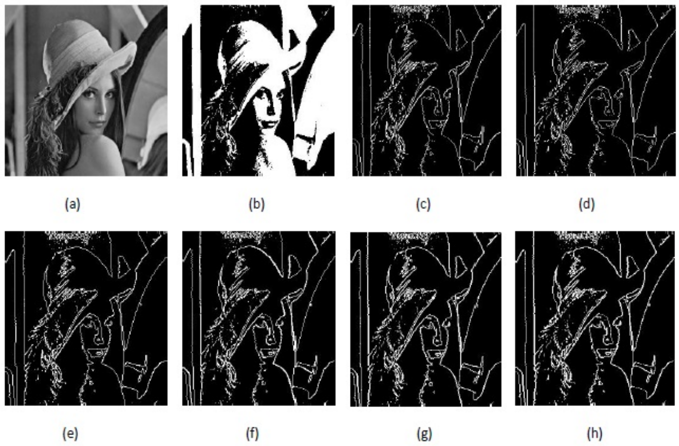

Figure 5: Edge detection of Lena image. a) Original image b) Binary image c) Rule 38 d) Rule 42 e) Rule 175 f) Rule 935 g) Rule 1273 h) Rule 1511

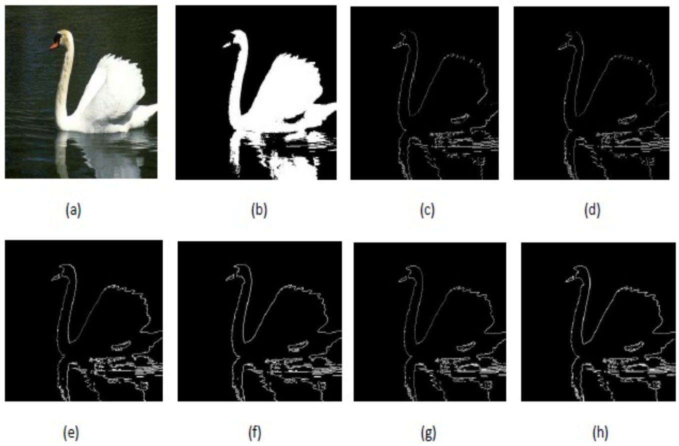

Figure 6: Edge detection of Swan image. a) Original image b) Binary image c) Rule 38 d) Rule 42 e) Rule 175 f) Rule 935 g) Rule 1273 h) Rule 1511

5.1. Images with Ground Truth

In this section, we present five images with their hand-made ground truth, provided from the Berkeley database: woman, plane, swan, bird and church. Figures "Fig. 6","Fig. 7", "Fig. 8","Fig. 9" and "Fig. 10" show the edges after application of rules 38, 42, 175, 935, 1273, 1511. We can easily conclude that the edges provided by the CA rules are very satisfactory, with a high visual quality. They provide a great level of continuity and smoothness. Edges are fine and accurate.

5.2. Fitness fuctions

In order to evaluate the quality of edges produced by a CA rule, we need a function that measures how close are the results produced by this CA to the ground truth.

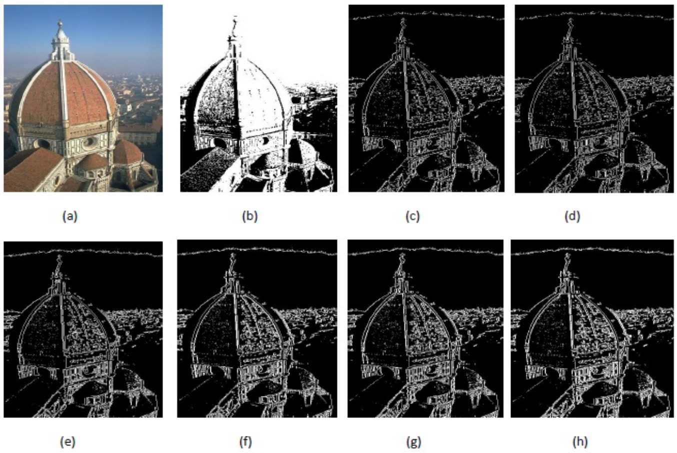

Figure 7: Edge detection of church image. a) Original image b) Binary image c) Rule 38 d) Rule 42 e) Rule 175 f) Rule 935 g) Rule 1273 h) Rule 1511

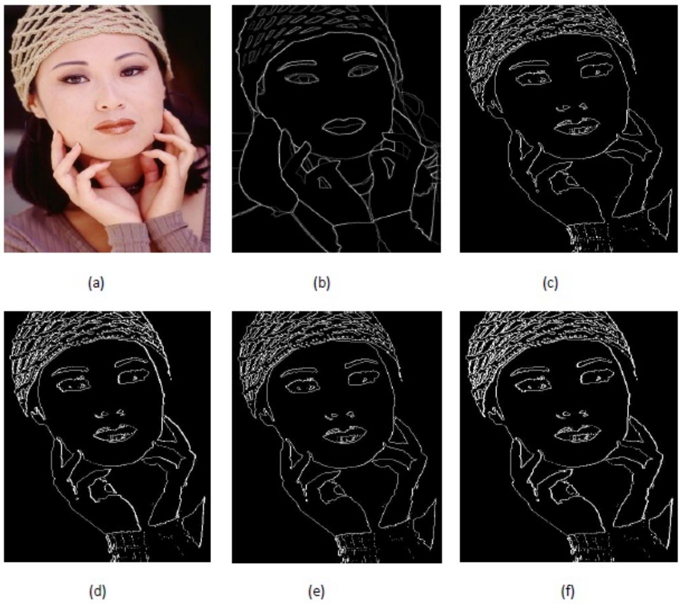

Figure 8: Edge detection of woman image. a) Original image b) Ground truth c) Rule 175 d) Rule 935 e) Rule 1273 f) Rule 1511

Whichever optimization method is used, an objective function is required, and its quality obviously has a crucial effect on the final results. In this work, We considered two fitness functions: the Structural Similarity Index (SSIM) and Root-Mean-Square Error (RMSE). The role of these functions is to measure the difference of quality between the CA resulting image and the reference image.

For images with more intensity values, the RMSE between the input and target image is a straightforward measure. However, it is well known that RMSE values have limitations. In

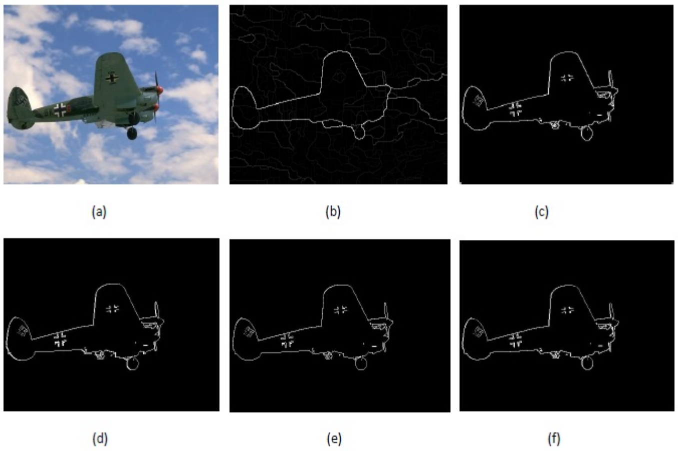

Figure 9: Edge detection of plane image. a) Original image b) Ground truth c) Rule 175 d) Rule 935 e) Rule 1273 f) Rule 1511

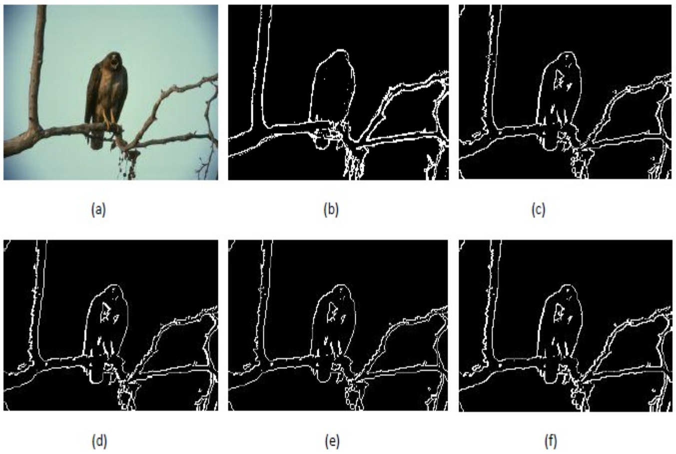

Figure 10: Edge detection of bird image. a) Original image b) Ground truth c) Rule 175 d) Rule 935 e) Rule 1273 f) Rule 1511

particular, given that they do not involve inter-pixel relationships they often do not capture perceptual similarity.
SSIM measures the image similarity taking into account three independent channels including luminance, contrast and structure [32]. It is the well suited measure for gray level images.
The SSIM metric between two images x and y is defined as "(3)" :

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{3}$$

where $\mu_x, \mu_y, \sigma_x^2, \sigma_y^2, \sigma_{xy}$ are the mean of x, the mean of y, the variance of x, the variance of y, and the covariance of x and y, respectively. Following [32], C_1 is set to $(0.01 * 255)^2$ and $C_2 = (0.03 * 255)^2$.

The RMSE is calculated according to "(4)":

$$RMSE = \sqrt{\frac{1}{MN}\sum_{r=0}^{M-1}\sum_{c=0}^{N-1}[E(r,c) - O(r,c)]^2} \tag{4}$$

where $O(r,c)$ is the original image (in our case, the ground-truth image) and $E(r,c)$ is the reconstructed image.

"Tab. 1" shows the best fitness values obtained for Woman, Bird, and Plane input images. For each image, they are tested using the fitness functions RMSE and SSIM. The best fitness values are collected in the table. The numerical results clearly show that rules 175, 935, 1273 and 1511 provide high performances and very good values of fitness. They consolidate the visual results obtained by these rules.

Table 1
Fitness Values for Images: Woman, bird and plane

	Rule 175		Rule 935		Rule 1273		Rule 1511	
	RMSE	*SSIM*	*RMSE*	*SSIM*	*RMSE*	*SSIM*	*RMSE*	*SSIM*
Woman	0.213	0.99	0.216	0.99	0.221	0.99	0.211	0.99
Bird	0.118	0.99	0.127	0.99	0.124	0.99	0.115	0.99
Plane	0.021	0.99	0.022	0.99	0.021	0.99	0.021	0.99

6. Conclusion

This paper presents a novel method of edge detection based on a cellular automata with extended Moore neighborhood. Although the rule space have become larger, we used an evolutionary algorithm to extract the optimal rules which provide the best results. This process allowed to extract a subset of simple rules which produce very good edges in only one iteration. Experiments are carried on several images, from Mathworks, and comparisons are made with images from Berkeley database which have ground truth edges. The visual results show that the extracted rules enhance the contrast of the output images, and smoothes the edge of the objects present in the image. The fitness values of SSIM and RMSE are calculated and confirm the good quality of obtained results.

Possible future research directions could be extended to search about rules that perform image denoising, or exploring quantum metaheuristics for optimizing huge sets of rules.

References

[1] M. Juneja, P. S. Sandhu, Performance evaluation of edge detection techniques for images in spatial domain, International journal of computer theory and Engineering 1 (2009) 614.

[2] D. Ziou, S. Tabbone, Edge detection techniques-an overview, Pattern Recognition and Image Analysis 8 (1998) 537–559.
[3] Y. Z. S. Amrogowicz, Y. Zhao, An edge detection method using outer totalistic cellular automata, Neurocomputing 214 (2016) 643–653.
[4] J. Canny, A computational approach to edge detection, IEEE Transactions on pattern analysis and machine intelligence (1986) 679–698.
[5] A. Burks, Theory of Self-Reproducing Automata, University of Illinois Press, Champaign, IL, USA, 1966.
[6] J. Mohammed, D. R. Nayak, An efficient edge detection technique by two dimensional rectangular cellular automata, in: In International Conference on Information Communication and Embedded Systems (ICICES2014), 2014, pp. 1–4.
[7] N. H. Packard, S. Wolfram, Two-dimensional cellular automata, Journal of Statistical Physics 38 (1985) 901–946.
[8] E. Fredkin, Digital machine: A informational process based on reversible cellular automata, Physica D 45 (1990) 254–270.
[9] J. Kari, Reversability of 2d cellular automata is undecidable, Physica D 45 (1990) 379–385.
[10] N. Silva, P. Ween, B. Baets, Bruno, Improved texture image classification through the use of a corrosion-inspired cellular automaton, Neurocomputing, Part C (2015) 1560–1572.
[11] Y. Yang, S. Tian, H. Lei, Y. ZhoU, W. Shi, Novel quantum image encryption using one-dimensional quantum cellular automata, Information Sciences (2016) 257–270.
[12] F. Qadir, K. Khan, Investigations of cellular automata linear rules for edge detection, International Journal of Computer Network and Information Security 4 (2012) 47–53.
[13] S. Uguz, U. Sahin, F. Sahin, Uniform cellular automata linear rules for edge detection, in: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, 2013, pp. 2945–2950.
[14] D. Aydogan, Cnnedgepot: Cnn based edge detection of 2d near surface potential field data, Computers and Geosciences 46 (2012) 1–8.
[15] M. Diwakar, P. Pate, K. Gupta, Cellular automata based edge detection for brain tumor, in: International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2013, pp. 53–59.
[16] S. Wongthanavasu, R. Sadananda, A ca-based edge operator and its performance evaluation, Journal of Visual Communication and Image Representation 14 (2003) 83–96.
[17] S. Djemame, O. Djidel, M. Batouche, Image segmentation using continuous cellular automata, in: 10th International Symposium on Programming and Systems (ISPS), 2011, pp. 94–99.
[18] C. Chang, Y. Zhang, Y. Gdong, Cellular automata for edge detection of images, in: Third International Conference on Machine Learning and Cybernetics, 2004, pp. 3830–3834.
[19] Y. Chen, Z. Yan, A cellular automatic method for the edge detection of images, in: D.-S. Huang, I. Wunsch, DonaldC., D. L. K.-H. J. (Eds.) (Eds.), Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence,, volume 5227 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2008, pp. 935–942.
[20] P. Rosin, Training cellular automata for image processing, IEEE Transactions on Image Processing 15 (2006) 2076–2087.
[21] P. Rosin, Image processing using 3d-state cellular automata, Computer Vision and Image

Understanding 114 (2010) 790–802.
[22] M. Batouche, S. Meshoul, A. Abbassene, On solving edge detection by emergence, in: R. D. E. M. Ali (Ed.), Advances in Applied Artificial Intelligence, volume 4031 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2006, pp. 800–808.
[23] S. Slatnia, M. Batouche, K. E. Melkemi, Evolutionary cellular automata based-approach for edge detection, in: G. P. E. F. Masulli, S. Mitra (Ed.), Applications of Fuzzy Sets Theory, volume 4578 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, 2007, pp. 404–411.
[24] S. Djemame, M. Batouche, Combining cellular automata and particle swarm optimization for edge detection, International Journal of Computer Applications 57 (2012) 16–22.
[25] M. Mraz, N. Zimic, I. Lapanja, I. Bajec, Fuzzy cellular automata: from theory to applications, in: IEEE International Conference on Tools with Artificial Intelligence, 2000, pp. 320–323.
[26] D. Patel, S. More, Edge ddetection technique by fuzzy logic and cellular learning automata using fuzzy image processing, in: International Conference on Computer Communication and Informatics (ICCCI), 2013, pp. 1–6.
[27] S. Sinaie, A. Ghanizadeh, E. Majd, S. Shamsuddin, A hybrid edge detection method based on fuzzy set theory and cellular learning automata, in: International Conference on Computational Science and Its Applications, ICCSA 09, 2009, pp. 208–214.
[28] P. Dollar, C. Zitnick, Structured forests for fast edge detection, in: ICCV, International Conference on Computer Vision, (2013), 2013, pp. 1841– 1848.
[29] M. Mofrad, S. Sadeghi, A. Rezvanian, M. Meybodi, Cellular edge detection: Combining cellular automata and cellular learning automata, International Journal of Electronics and communications 69 (2015) 1282–1290.
[30] M. Han, X. Yang, Y. Jiang, An extreme learning machine based on cellular automata of edge detection for remote sensing images, Neurocomputing 198 (2016) 27–34.
[31] S. Djemame, M. Batouche, H. Oulhadj, P. Siarry, Solving reverse emergence with quantum pso, application to image processing, Soft Computing 23 (2019) 6921–6935.
[32] Z. Wang, A. Bovik, H. Sheikh, E. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Transactions on Image Processing 13 (2004) 600–612.