International Conference on Innovative Applications of Computational Intelligence on Power, Energy and Controls with their Impact on Humanity (CIPECH14) 28 & 29 November 2014

Training Two Dimensional Cellular Automata for Some Morphological Operations

Anand Prakash Shukla
Department of Computer Science
and Engineering
Krishna Institute of Engineering
and Technology, Ghaziabad,India
Email:anandskla@gmail.com

Suneeta Agarwal
Department of Computer Science
and Engineering
Motilal Nehru National Institute of Technology
Allahabad,India.
Email: suneeta@mnnit.ac.in

***Abstract*—Cellular Automata is significantly applying to image processing operations. This paper describes the application of cellular automata (CA) to various morphological operations such as thinning and thickening of binary images. The description about the use of training of cellular automata for thinning and thickening of binary images is illustrated by this paper. The selection of the best rule set from large search space has been performed on the basis of sequential floating forward search method. The misclassification error between the images obtained by the standard function and the one obtained by cellular automata rule is used as the objective function. The proposed method is also compared with some standard methods.**

I. INTRODUCTION

Due to simple structure of Cellular Automata (CA) to model complex behavior system, it has attracted various researchers from different areas. Cellular automata primarily announced by Ulam[1] and Von Neuman [2] in 1950's and also discussed in the book of Wolfram 'A New Kind of Science'[3] with the purpose of obtaining models of biological self-reproduction.Cellular automata make up a very important class of completely discrete dynamical systems Now a day's Cellular Automata became very popular because of its diverse function and utility as a discrete model for many processes. Cellular Automata also provided a concept for computational automata.

Cellular Automata is also called Systems of Finite Automata, i.e. Deterministic Finite Automata (DFA) arranged in an infinite, regular lattice structure[4]. In cellular automata state of a cell at the next time step is determined by the current states of a surrounding neighborhood of cells along with its own state and is updated synchronously in discrete time steps.

Formally, a (bi-directional, deterministic) cellular automaton is a triplet

$$A = (S; N; \delta),$$

where S is an non-empty state set, N is the neighborhood system, and

$$\delta : S^N \to S$$

is the local transition function (rule).

Commonly used neighborhood systems are the von Neumann and Moore neighborhoods.

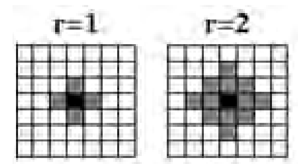


Fig. 1. von Neuman Neighborhood

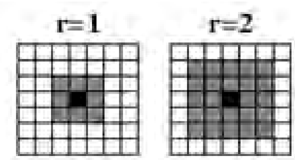


Fig. 2. Moore Neighborhood

1) Von Neumann Neighborhood: A diamond-shaped neighborhood that can be used to define a set of cells surrounding a given cell (x_0, y_0) that may affect the evolution of a two-dimensional cellular automaton on a square grid.The von Neumann neighborhood of range r is defined by the equation 1.

$$N_{x_0,y_0} = \{(x, y) : |x - x_0| + |y - y_0| \leq r\} \quad (1)$$

The von Neumann neighborhood is illustrated in figure I-1

2) Moore Neighborhood: A square-shaped neighborhood that can be used to define a set of cells surrounding a given cell (x_0, y_0) that may affect the evolution of a two-dimensional cellular automaton on a square grid. The Moore neighborhood of range r is defined by the equation 2

$$N_{x_0,y_0} = \{(x, y) : |x - x_0| \leq r, |y - y_0| \leq r\} \quad (2)$$

Moore neighborhoods is illustrated in figure I-2

A. Application of Cellular Automata

Some applications of cellular automata in complex systems' modeling, analyzing and controlling are: The Games of Life[5], biological systems[6], Cellular automata in environmental system and ecological system [7], [8], CA in edge deduction[9], [10], [11], Cellular automata in the traffic system[12], CA in image processing[13], Cellular automata in machine learning and control[14] and CA in cryptography[15]. Digital image processing plays an important role in real life applications such as satellite television, computer tomography and magnetic resonance imaging as well as in areas of research and technology such as biological information systems and astrophysics[16].CA is used in various image processing tasks such in Image Filtering in better way than some existed filters in denoising process[17], [18], [19], Border Detection in Digital Images that provide boundaries of images[20], Connected Set Morphology (applied on more than one image at a time), Thinning and Thickening of images[17], [21], Image Segmentation which is an integral part of image processing applications like medical image analysis and photo editing[22], [20] and in Image Enhancement because of its dynamic behaviour[23]. The advantage of cellular automata is that though each cell has an extremely limited view of the system (just its immediate neighbors) and each cell generally contains a few simple rules, the combination of a matrix of cells with their local interaction leads to more sophisticated emergent global behavior system i.e. The CA provides simplicity in complexity. This paper concentrates on training of CA for image's thinning and thickening process with their applications in different fields.

II. Training of Cellular Automata

Cellular automata training is basically used to acquire knowledge, skill to improve capability, capacity and performance of cellular automata in image processing like- noise filtering, Thinning, Convex Hull, calculating distance features, Template Matching, Image Sharpening, Simple Object Recognition using rule sets which provide specific operations to their states at each step of time.

A. Related work

To find out the desired rule set, feature selection is acquiescent and can be performed using branch and bound algorithms[24]. In automatically learning of rules most of researchers focus in the density classification problem[22] that is recognized as a standard for exploring cellular automata rules with universal properties further applied standard genetic algorithm(GA) for learning rules[22][25].A standard genetic programming scaffold is used in learning and training of cellular automata[26]. There are various feature selection methods that were introduced also consider GA and SFFS (sequential floating forward search)[26], [27], [17], where SFFS have better characteristics over the GA approach. Rule set automatically generated by evolutionary algorithms but probably this procedure was applied to artificial problems like- Majority Problem or for boundary detection in binary images[28], [29]. It depends on a rule set produced exact target output by increasing the size of neighborhood[30]. A form of hill climbing and backtracking algorithms also used to identify the rules. To train the cellular automata for image's thinning and thickening process Paul Rosin has been used RMS criterion and Hausdorff distance error measures with SFFS procedure as objective function but did not produce satisfactory results and produced some lines in fragmented way. Fragmented problem overcame by an algorithm and two cycle cellular automata to produce better results in learning of rules[17].

III. Training of Cellular Automata for Image Thinning and Thickening

Thinning and Thickening words come under the morphology operations that refer to form and structure; in computer visualization it can be used to refer to the shape of a region.

Thinning algorithms on binary images have a long history in image processing, because of their value in deriving higher representations (and compressed encodings) of the information in a bitmap. Thinning is a procedure to remove selected foreground pixels from binary images like erosion or opening .It is normally applied to binary images and produces another binary image as output. It can be used for several applications, but is particularly useful for skeletonization.

Thickening is a morphological operation that is used to breed particular regions of forefront pixels in binary images, that could be seem as dilation or closing. Basically Thickening process only applied to binary images, and by this it produces another binary image as output. The thickening operation is related to the hit-and-miss transform, There are various applications in image processing where thinning and thickening are used for different purposes as- Thinned images possess a subset of the original information that is useful for applications such as segmentation, feature extraction, vectorization, and pattern identification and Thickening includes determining the approximate convex hull of a shape, to study any disease growing cell in spreading manner and determining the skeleton by zone of influenced region.

IV. Methodology

In the current experiments, binary images are considered which means cells have two states i.e. white or black and Moore neighborhood is considered in which the cells are eight way connected and transition rules are only applied to non boundary cells. The initial cell values are considered as the pixel values of the input image. As the Moore neighborhood is considered there are 2^8 rules are possible which are reduced to 51 as shown in figure 3, by taking into account 45^o rotational symmetry and bilateral reflection[17]. Now the aim is to find the rule or set of rules among these selected 51 rules which provides the best thinning and thickening of the given image. Before applying the rules, the thinning and thickening of input image is obtained by the standard function bwmorph() available in the MATLAB and the resulting images are saved for the further references.

In order to select the best rule set for thinning of white portion, first the 3×3 neighborhood pattern of rules is , as shown in figure reffig:ruleset,is compared with the 3×3 neighbors of the current pixel of input image. If both are not same then the central pixel is inverted in case of central white pixel.This operation is performed for all pixels of the image. Each rule among 51 rules is applied at one time and the resulting image is compared with the image obtained by the standard function bwmorph(). the misclassification error defined by the Yasnoff et al.[31] has been consider which is best suited for the classification of the thinned(thickened) white portion with standard function and rule set.The misclassification error is defined as

$$error = 1 - \frac{|B_o \cap B_T| + |F_o \cap F_T|}{B_o + F_o} \tag{3}$$

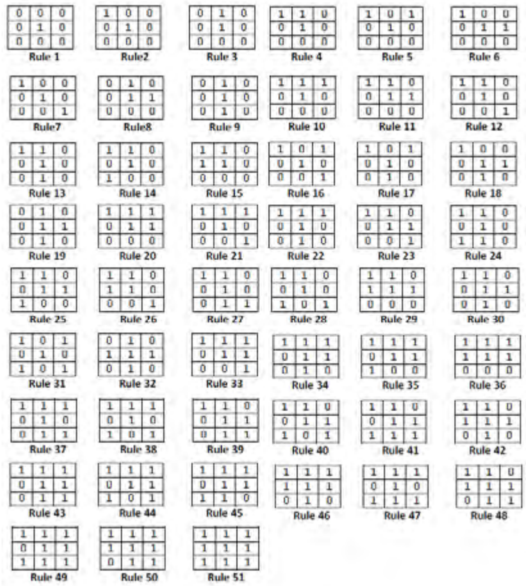


Fig. 3. Rule Set

where F_o and B_o shows the black and white pixels of the image obtained by the CA rule where as F_T and B_T shows the black and white pixels of target image obtained by the standard function.

To compute the rule set for the thickening of the image the same procedure is used for the central black pixel i.e. if the central pixel is black and the encoded neighborhood pattern does not match with the encoded rule then the central pixel is inverted.

Proposed approach is providing Thinning and Thickening of white region on different images (10) taking the advantage of cellular automata rules. The current paper shows only a subpart of test images to shown a fine visible of result.

The procedure CA thinning shows the algorithm for proposed method. The rules are referred in the procedure by the rule number. The rules are numbered from 1 to 51 in figure 3 from left to right and upwards down manner. i.e. The proposed algorithm is initialized by setting the value of m as 0,here m shows the rule number as in step 1.The variable Set is the array of strings which stores all 51 encoded rules in string format. The rules are applied to each pixels in the image and if the neighbors not matched with the rule then the central pixel is inverted in case of the central white pixel.

In each step each image pixel should be processed in parallel. However, since it is used as sequential operation, repeated until the maximum number of pixels matched to the standard method, the processed pixels are stored in a secondary image C. At the end of every iteration it is copied back to original image in step 7. In Step 6, this cycle is repeated until the maximum number of pixels matched to the standard method bwmorph(). The same procedure with minute change can be used for the thickening operation. The only change is required is to change the value of central black pixel in step 3 if it does not match with the encoded rule set. All other steps are same.

To find the optimum set of the rules the sequential floating forward search(SFFS) method is used. The SFFS is a deterministic search method which provide the effectiveness equivalent to the genetic algorithm(GA)[25].The 0 shows the SFFS algorithm

Let Y_k denote the rule set at iteration and its score be $J(Z_k)$. Here, It is defined by the result of the 0 by applying the CA rule set to the input image and computing the fitness function. Step 1 shows that the initial set is empty. In step 2 at each iteration, all rules are considered for addition to the rule set and only the rule giving the maximum score is added the resulting rule set. This process is repeated until no improvements in score are gained by adding rules.In step 3, each rule in rule set found in step 2 is removed to find the rule whose removal provides the resulting rule set with the improved value of objective function. As shown in step 4 if removal of the rule cause the better score of the objective function then it is discarded from the rule set and again next rule is tried for the deletion and process go to step 3. Otherwise, the process go to step 2 for the addition of new rule to the rule set.

V. EXPERIMENTAL RESULTS

Following experiments are performed by using MATLAB 2014a. Considering 51 rules, which are used to provide thin or thick images at different iteration. Aforementioned procedure has been implemented and one pixel thinned image generated by the function bwmorph() available in MATLAB has been considered for the comparison purpose. In order to train the rule set 20 binary sample images are carefully selected in such a way that the images contains all kind of shapes such as lines,circles, cones and some complicated structures. Then the rules are trained by using the objective function misclassification error with the thinned image by the SFFS method as described in the previous section. In all the images the trained rule set which gives the best thinning result is rule number 51 as shown in figure 4.

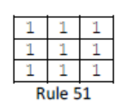


Fig. 4. Rule set learned for edge detection

(a) Rice (b) Cameraman (c) Hand

Fig. 5. Input images

Figure 5 shows the input images rice, cameraman and hand respectively. Then learned rule 51 is applied to these images which produced result in V that are compared to thinned images produced by standard function bwmorph() in 7.

Algorithm 1 Algorithm for Thining by Cellular Automata Rules

procedure CATHINING(A)
  Initilization $m \leftarrow 0$
  $B \leftarrow A$
  $SET \leftarrow Encoded\ 51\ Rules$
  $E \leftarrow bwmorph(B)$ ▷ thin the images by standard function bwmorph(),
  repeat
    $m \leftarrow m + 1$
    $C \leftarrow B$
    For every pixel C[i,j] and Rule number = m in SET
    begin
    $S1 \leftarrow 3 \times 3\ neighbors\ of\ C[i][j]$.
    $S2 \leftarrow 3 \times 3\ neighbors\ of\ SET[m]$.
    if $C[i,j] = white\ and\ S1\ NOT\ matched\ with\ S2$ **then**
      $B[i,j] \leftarrow InvertC[i,j]$
    end for
    $ComputeError[i] = 1 - \frac{|B_E \cap B_B| + |E_E \cap E_B|}{B_B + E_B}$
  until $m = 51$

Algorithm 2 Sequential Floating Forward Search

procedure SFFS
Step 1:
  $Y \leftarrow \{\phi\}$
Step 2:
  Select the best feature
  $x^+ \leftarrow argmax J(Z_k + x) | x \notin Z_k$
  $Z_k \leftarrow Z_k + x^+$
  k=k+1
Step 3:
  Select the worst feature
  $x^- \leftarrow argmax J(Z_k - x) | x \in Z_k$
Step 4:
  if $J(Z_k - x^-) > J(Z_k)$ **then**
    $Z_k \leftarrow Z_k - x^-$
    $k = k + 1$
    Go to Step 3
  else
    Go to Step 2

(a) Rice (b) Cameraman (c) Hand

Fig. 6. Input Images Thinned by Cellular Automata Rules

(a) Rice (b) Cameraman (c) Hand

Fig. 7. Input Images Thinned by Standard Function

(a) Rice (b) Cameraman (c) Hand

Fig. 8. Input Images Thicken by Cellular Automata Rules

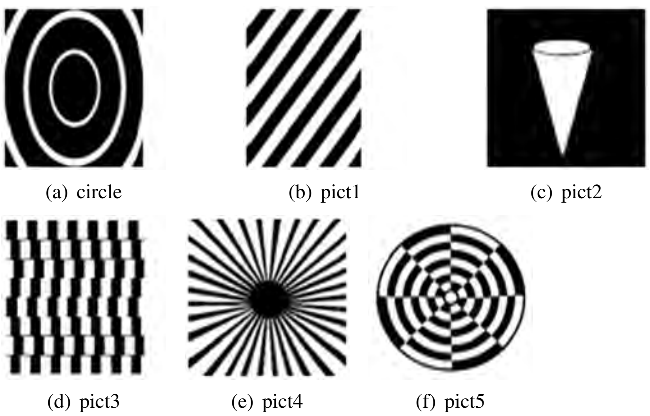
(a) Rice (b) Cameraman (c) Hand

Fig. 9. Input Images Thicken by Standard Function

Similarly, figures 8 and 9 show the results for thickening.

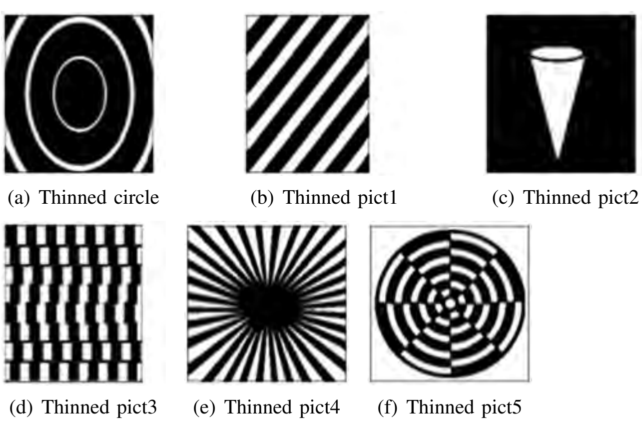
It is also found that rule number 51 have the minimum misclassification error for the thickening operation and the images of figure 8 shows the result of the thickening performed by the rule 51 and figure 9 shows the thickening result of bwmorph().

Besides these images experiment of training of cellular automata for thinning and thickening of images using best rules set at which minimum misclassification error is occurred in corresponding to standard function is done on several other images as- circles, pict1, pict2, pict3, pict4, pict5, pict6 and results are shown in figure 10,11 and 12 where fig 10 is the input images,figure 11 and 12 are the results of thinning and thickening respectively when learned rule is applied.



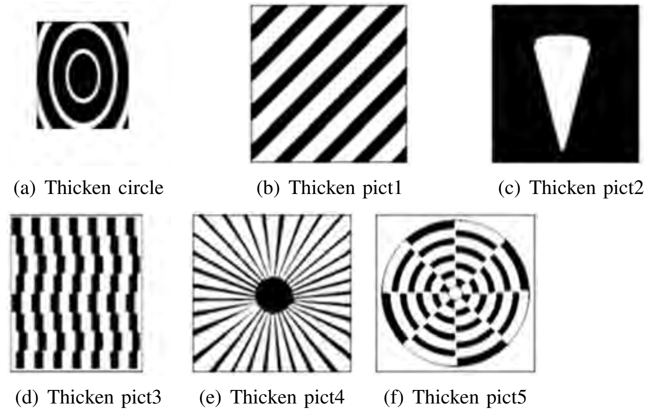
(a) circle (b) pict1 (c) pict2 (d) pict3 (e) pict4 (f) pict5

Fig. 10. Sample images to perform the thinning and thickening by the learned rule .



(a) Thinned circle (b) Thinned pict1 (c) Thinned pict2 (d) Thinned pict3 (e) Thinned pict4 (f) Thinned pict5

Fig. 11. Thinned image using learned rule.



(a) Thicken circle (b) Thicken pict1 (c) Thicken pict2 (d) Thicken pict3 (e) Thicken pict4 (f) Thicken pict5

Fig. 12. Thicken images using learned rule.

Since the proposed algorithm is simulated on the sequential machine in MATLAB 2014a the time required to find the result is high. But once the rule set is trained for the thinning and thickening, then it is quite easier to apply the trained rule set to the images to find the desired result.

VI. Conclusion and Future Work

It is found that the results for the image processing application thinning and thickening of binary images through cellular automata rules are encouraging. The resulting rule sets provides comparable thinning and thickening as compared with the standard function bwmorph(). It is also required to mention that the aim to use the cellular automata is to provide simplicity to solve the complex processes which is fulfilled in our experiment and clearly depicted in the results section. Implementation and use of cellular automata rules is easier and straight forward as compare to other methods. As the work presented in the paper is limited in thinning and thickening with the use of the above can be extended for some other morphological and image processing operations.

References

[1] S. Ulam, "Some ideas and prospects in biomathematics," *Annual review of biophysics and bioengineering*, vol. 1, no. 1, pp. 277–292, 1972.

[2] J. Von Neumann, A. W. Burks *et al.*, "Theory of self-reproducing automata," 1966.

[3] S. Wolfram, *A new kind of science*. Wolfram media Champaign, 2002, vol. 5.

[4] M. Sipper, "The evolution of parallel cellular machines: Toward evolware," *BioSystems*, vol. 42, no. 1, pp. 29–43, 1997.

[5] M. Gardner, "Mathematical games: The fantastic combinations of john conways new solitaire game life," *Scientific American*, vol. 223, no. 4, pp. 120–123, 1970.

[6] H. De Garis, "Cam-brain the evolutionary engineering of a billion neuron artificial brain by 2001 which grows/evolves at electronic speeds inside a cellular automata machine (cam)," in *Towards evolvable hardware*. Springer, 1996, pp. 76–98.

[7] S. Aassine and M. C. El Jaı, "Vegetation dynamics modelling: a method for coupling local and space dynamics," *Ecological modelling*, vol. 154, no. 3, pp. 237–249, 2002.

[8] R. Smith, "The application of cellular automata to the erosion of landforms," *Earth Surface Processes and Landforms*, vol. 16, no. 3, pp. 273–281, 1991.

[9] C.-l. Chang, Y.-j. Zhang, and Y.-Y. Gdong, "Cellular automata for edge detection of images," in *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on*, vol. 6. IEEE, 2004, pp. 3830–3834.

[10] T. Kumar and G. Sahoo, “A novel method of edge detection using cellular automata,” *International Journal of Computer Applications*, vol. 9, no. 4, pp. 0975–8887, 2010.

[11] S. Wongthanavasu and R. Sadananda, “Pixel-level edge detection using a cellular automata-based model,” *Advances in Intelligent Systems: Theory and Applications*, vol. 59, pp. 343–351, 2000.

[12] M. Fukui and Y. Ishibashi, “Traffic flow in 1d cellular automaton model including cars moving with high speed,” *Journal of the Physical Society of Japan*, vol. 65, no. 6, pp. 1868–1870, 1996.

[13] A. Popovici and D. Popovici, “Cellular automata in image processing,” in *Fifteenth International Symposium on Mathematical Theory of Networks and Systems*, vol. 1, 2002.

[14] F. M. Marchese, “A directional diffusion algorithm on cellular automata for robot path-planning,” *Future Generation Computer Systems*, vol. 18, no. 7, pp. 983–994, 2002.

[15] S. Nandi, B. Kar, and P. Pal Chaudhuri, “Theory and applications of cellular automata in cryptography,” *Computers, IEEE Transactions on*, vol. 43, no. 12, pp. 1346–1357, 1994.

[16] L. S. Davis, “A survey of edge detection techniques,” *Computer graphics and image processing*, vol. 4, no. 3, pp. 248–270, 1975.

[17] P. L. Rosin, “Training cellular automata for image processing,” *Image Processing, IEEE Transactions on*, vol. 15, no. 7, pp. 2076–2087, 2006.

[18] A. P. Shukla, S. Chauhan, and S. Agarwal, “Training of cellular automata for image filtering,” *in Proc. Second International Conference on Advances in Computer Science and Application - CSA 2013*, pp. 86–95, 2013.

[19] A. P. Shukla and S. Agarwal, “An enhanced cellular automata based scheme for noise filtering,” *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 7, no. 4, pp. 231–242, 2014.

[20] Y. Y. Boykov and M.-P. Jolly, “Interactive graph cuts for optimal boundary & region segmentation of objects in nd images,” in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 1. IEEE, 2001, pp. 105–112.

[21] A. Shukla, S. Chauhan, S. Agarwal, and H. Garg, “Training cellular automata for image thinning and thickening,” in *Confluence 2013: The Next Generation Information Technology Summit (4th International Conference)*, Sept 2013, pp. 394–400.

[22] M. Mitchell, J. P. Crutchfield, R. Das *et al.*, “Evolving cellular automata with genetic algorithms: A review of recent work,” in *Proceedings of the First International Conference on Evolutionary Computation and Its Applications (EvCA96)*, 1996.

[23] G. Sahoo, T. Kumar, B. Raina, and C. Bhatia, “Text extraction and enhancement of binary images using cellular automata,” *International Journal of Automation and Computing*, vol. 6, no. 3, pp. 254–260, 2009.

[24] P. Somol, P. Pudil, and J. Kittler, “Fast branch & bound algorithms for optimal feature selection,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 7, pp. 900–912, 2004.

[25] A. Jain and D. Zongker, “Feature selection: Evaluation, application, and small sample performance,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 2, pp. 153–158, 1997.

[26] D. Andre, F. H. Bennett III, and J. R. Koza, “Discovery by genetic programming of a cellular automata rule that is better than any known rule for the majority classification problem,” in *Proceedings of the First Annual Conference on Genetic Programming*. MIT Press, 1996, pp. 3–11.

[27] H. Hao, C.-L. Liu, and H. Sako, “Comparison of genetic algorithm and sequential search methods for classifier subset selection.” in *ICDAR*. Citeseer, 2003, pp. 765–769.

[28] A. Adamatzky, “Automatic programming of cellular automata: identification approach,” *Kybernetes*, vol. 26, no. 2, pp. 126–135, 1997.

[29] M. Mitchell, P. Hraber, and J. P. Crutchfield, “Revisiting the edge of chaos: Evolving cellular automata to perform computations,” *arXiv preprint adap-org/9303003*, 1993.

[30] B. Straatman, R. White, and G. Engelen, “Towards an automatic calibration procedure for constrained cellular automata,” *Computers, Environment and Urban Systems*, vol. 28, no. 1, pp. 149–170, 2004.

[31] W. A. Yasnoff, J. K. Mui, and J. W. Bacus, “Error measures for scene segmentation,” *Pattern Recognition*, vol. 9, no. 4, pp. 217–231, 1977.