SOLVING SUDOKU PUZZLE with NUMBERS RECOGNIZED by USING ARTIFICIAL NEURAL NETWORKS

Selcuk SEVGEN, Emel ARSLAN, Ruya SAMLI

Department of Computer Engineering, Istanbul University, Istanbul, Turkey
{sevgens, earslan, ruyasamli}@istanbul.edu.tr

Abstract: *This paper proposed a method to solve 9×9 SUDOKU puzzles automatically. To this end, a captured puzzle image is used, the numbers in this image are recognized by using Artificial Neural Networks (ANN) and a 9×9 number array with these numbers is constituted, respectively. Then, the proposed method is applied to the prepared numerical array for solving the puzzle. The validity of the proposed method is demonstrated with results from an example 9 × 9 SUDOKU puzzle image.*

Keywords: *SUDOKU, Puzzle Solving, Artificial Neural Networks, Image Recognition, Training.*

1. Introduction

Computer games which have various mathematical, algorithmic and visual properties are important research issues of computer science. Hence, they can be used for solving real world problems such as education [1-7], mathematics [8-15] and fitting problems [16-18].

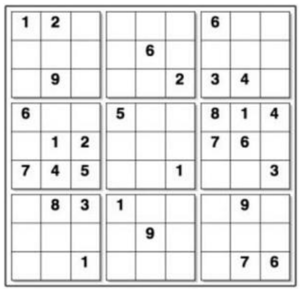
SUDOKU is a puzzle-typed game not related to mathematics directly although it has numerical components. The game is based on filling a 9×9 grid so that each column, each row and each of the nine 3×3 sub-grid contains all of the numbers from 1 to 9. Examples of standard and modified SUDOKU grids are shown in Fig.1.

There are various studies which examine SUDOKU in the related literature. A hybrid AC3-tabu search algorithm is used for solving the puzzle [14]. Permutations are generated for candidate values of empty cells in SUDOKU puzzle [15]. Some studies analyzed SUDOKU considering it as a subclass of the Latin squares [19, 20] and the rules of the puzzle for solving are rewritten [21].

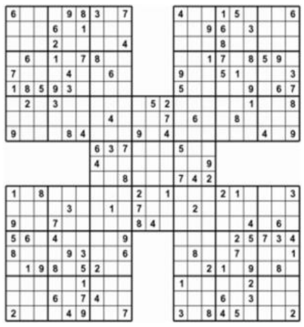
This study focuses on proposing a method for solving a standard SUDOKU puzzle by making use of image processing and ANN. The rest of the paper is organized as follows: in Section 2, the ANN structure is explained; in Section 3, our proposed method is presented and finally in Section 4, the conclusions and future work are given.

2. Artificial Neural Networks

In recent years, image processing has found many application areas such as medicine, engineering, security etc. ANN is an eficient tool used in image processing. The ANN structure and the training algorithm used in this study are back-propagation artificial neural network (BP-ANN) and Levenberg-Marquardt (L-M) algorithm [22, 23].



(a) Standart SUDOKU grid



(b) Modified SUDOKU grid

Figure 1. Example SUDOKU images

The ANN model used in the study consists of three layers of neurons as input, hidden and output as depicted in Fig. 2. The input layer of this system consists of the number images which are represented by matrices. In the training process of this type of network, the connection weights are

Received on: 27.09.2016
Accepted on: 31.01.2017

updated to minimize the error between the correct and estimated values of the system variables [24].

A hidden or output unit in the ANN operates as follows :

$$y_j = f(\sum_i w_{ji} x_i + b_j) \tag{1}$$

where

y_j: transformed output by the jth hidden or output node,

f : activation function,

w_{ji} : the synaptic weight from the ith node to jth node,

x_i : input node,

b_j : bias at jth node

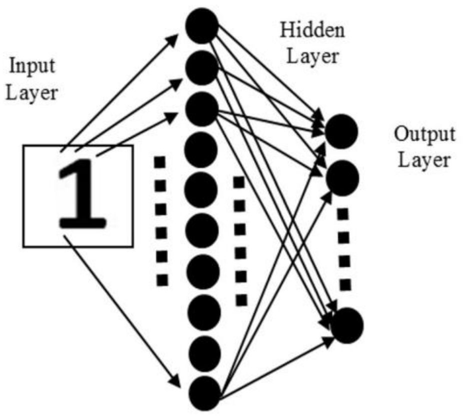


Figure 2. ANN structure used in this study

3. Proposed Method

3.1. Image Recognition

In this section, the steps for image recognition as image segmentation, system training and numerical array constitution are explained. The resolution of input SUDOKU image is 532 × 474 pixels.

Image segmentation: The aim of this step is dividing a one-piece grid image into 81 cells (sub-images). The cells have the possibility of containing 1-9 numbers or they can be empty. Sub-images are transformed to binary images, then, edge detection process is applied to determine whether there is an object or not. If any, the numerical value of the object is recognized by ANN.

System training: In the ANN, there are 3135 inputs representing 55 × 57 pixels in every image of the training set, 9 outputs (1-9 numbers) and a hidden layer with 10 neurons. The input image is transformed to a binary image, then, the matrices for numbers are transformed to a column vector form. The 9 column vectors are collected in an input array. As output value, a 9 × 9 identity matrix is constituted. The input values are used as 70%, 15% and 15% for training, validation and test, respectively. The error value is chosen as 10^{-7}. The 9 output neurons produce outputs which must be 0 or 1. The value of 1 in the column represents the desired number.

Numerical array constitution: The sub-images in the study are transformed to a 9 × 9 numerical array via the principles below:

- if there is a number in the sub-image, this determined number value is placed to the corresponding index in the array,
- if any number cannot be determined in the sub-image, the corresponding value in the array is 0,

otherwise -1 value is placed to the array.

3.2. Puzzle Solving

The algorithm must firstly decide if the current element of the array is 0 or not. Since 0 value means it is an empty cell in the SUDOKU and the appropriate value must be replaced to corresponding element, a candidate vector as ([1 2 3 4 5 6 7 8 9]) is constituted. If the element is not 0, a [X 0 0 0 0 0 0 0 0] vector is constituted, where X is the numerical value of the element. The whole algorithm is depicted in Fig. 3.

The explanation of the proposed algorithm is given below:

A[i][j]: The matrix which has all the recognized numbers in the SUDOKU grid.

Temp A[i][j][ ]: The temporary array which has vectors as [1 2 3 4 5 6 7 8 9] or [X 0 0 0 0 0 0 0 0] instead of each element in A[i][j] according to being 0 or not.

CANDIDATE: The [1 2 3 4 5 6 7 8 9] vector which consists of all the possible choices.

STEP 1 : For all i and j, check whether A[i][j] is 0 or not.

If so,

Temp A[i][j][ ] = [1 2 3 4 5 6 7 8 9] (it means, the appropriate value of the position will be searched from this row)

Otherwise

Temp A[i][j][ ]=[[A[i][j]] 0 0 0 0 0 0 0 0] (it means the appropriate value is settled to the position)

STEP 2: Eliminate the known values in the ith row of Temp A from the CANDIDATE vector.

STEP 3: Eliminate the known values in the jth column Temp A from the CANDIDATE vector.

STEP 4: Divide Temp A array into 9 3 × 3 sub-arrays.

STEP 5: Eliminate the known values in all 3 × 3 sub-arrays in Temp A from CANDIDATE vector.

STEP 6: Put the 3×3 sub-arrays together as Temp A[i][j][ ] array again.

STEP 7: If any Temp A[i][j][ ] element from CANDIDATE vector ≠ 0 GOTO Step 1

Else it means the puzzle is solved properly.

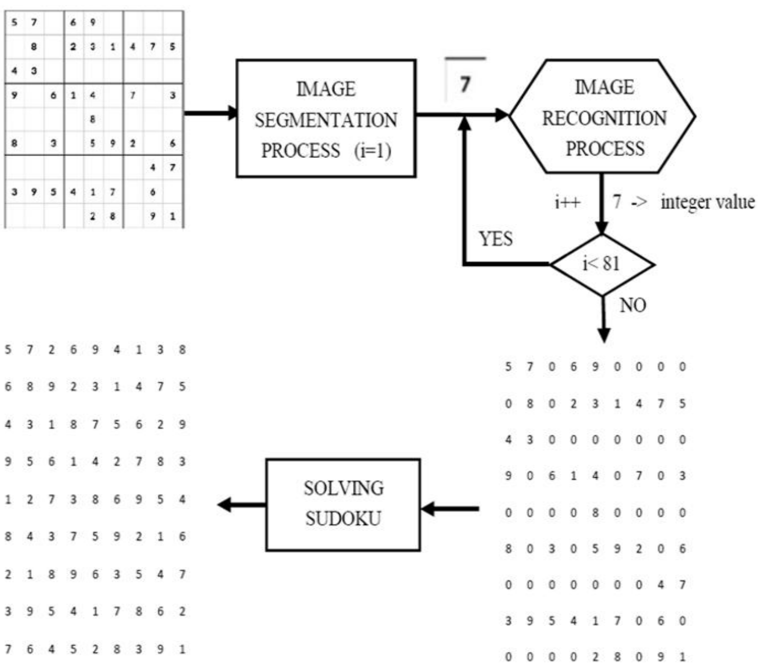


Figure 3. The flowchart of the proposed algorithm

3.3. Experimental Results

The captured 9 × 9 SUDOKU image of 32 × 474 pixels is given in Fig. 4. By the segmentation of that image, 81 sub-images (with or without a number) are obtained. Example sub-images obtained from this process are given in Fig. 5. Next step includes the transformations between RGB form and binary form of the image. Additionally, edge detection and number recognition are also implemented in this step. For instance, the recognition steps of number 5 are depicted in Fig 6.

The number recognition step of this study in the Fig. 6 can be also called training procedure. We use the number set for the training process which is shown in Fig. 7. The fonts of the number set in SUDOKU image and in ANN training procedure differ from each other for the purpose of providing the independency of fonts. An example of ANN training results is shown in Fig. 8. As it can be seen easily from the figure, the system reached the desired error value in 1613th iteration.

The recognition results of numbers 1, 2 and 5 are shown in Fig. 9 as an example. In each column, the row whose numerical value is closest to 1 represents the desired number. Then, the recognized numbers constitute an array as explained in Section 3.1 and this array is given in Fig.10. As explained in Section 3.2, the puzzle solving algorithm is implemented to the array. The temporary array (Temp A in the algorithm) is constituted as in Fig. 11.

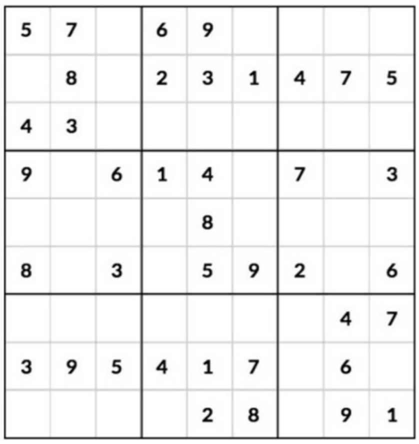


Figure 4. The SUDOKU image of 532 × 474 pixels

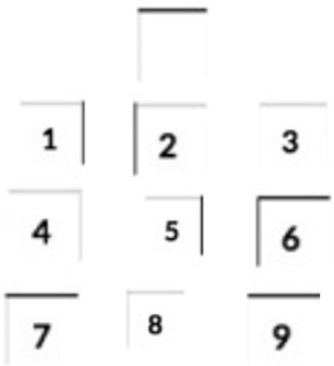


Figure 5. An example of sub-images obtained by using segmentation process

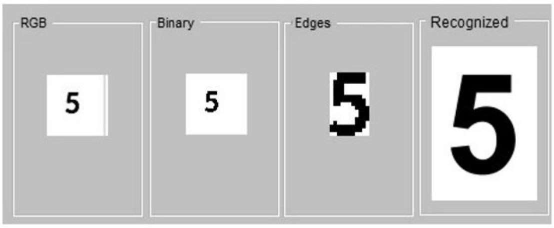


Figure 6. Image processing steps of number 5

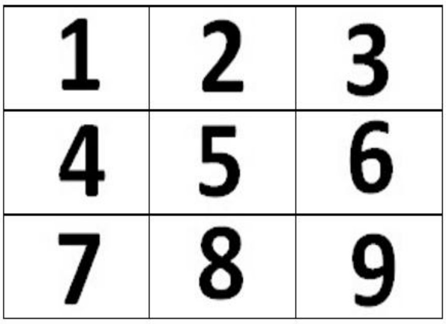


Figure 7. The font of ANN training number set

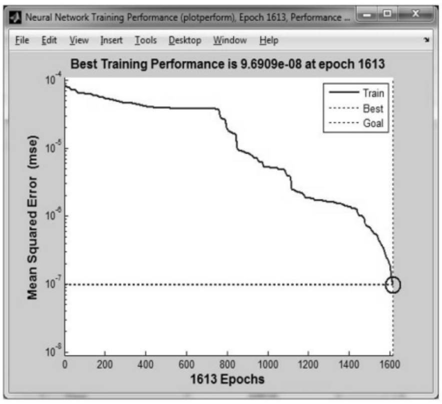


Figure 8. An example of ANN training result

x =	x =	x =
0.9555	0.0004	0.2927
0.0000	0.9827	0.0000
0.0000	0.0026	0.0000
0.0031	0.0019	0.1344
0.0366	0.0000	0.7936
0.0000	0.0000	0.0000
0.0037	0.0007	0.0014
0.0000	0.0007	0.0000
0.0000	0.0000	0.0000

Figure 9. The recognition results of numbers 1, 2 and 5

5	7	0	6	9	0	0	0	0
0	8	0	2	3	1	4	7	5
4	3	0	0	0	0	0	0	0
9	0	6	1	4	0	7	0	3
0	0	0	0	8	0	0	0	0
8	0	3	0	5	9	2	0	6
0	0	0	0	0	0	0	4	7
3	9	5	4	1	7	0	6	0
0	0	0	0	2	8	0	9	1

Figure 10. The numerical array

500000000	700000000	123456789	600000000	900000000	123456789	123456789	123456789	123456789
123456789	800000000	123456789	200000000	300000000	100000000	400000000	700000000	500000000
400000000	300000000	123456789	123456789	123456789	123456789	123456789	123456789	123456789
900000000	123456789	600000000	100000000	400000000	123456789	700000000	123456789	300000000
123456789	123456789	123456789	123456789	800000000	123456789	123456789	123456789	123456789
800000000	123456789	300000000	123456789	500000000	900000000	200000000	123456789	600000000
123456789	123456789	123456789	123456789	123456789	123456789	123456789	400000000	700000000
300000000	900000000	500000000	400000000	100000000	700000000	123456789	600000000	123456789
123456789	123456789	123456789	123456789	200000000	800000000	123456789	900000000	100000000

Figure 11. Temporary array

In the next step, the rows of temporary array are checked and the known numbers are eliminated since they cannot be a candidate for the solution. The new version of the temporary array is given below (Fig. 12). The elimination process is implemented to columns similar to rows. Fig. 13 shows the new version of the temporary array. The elimination process is implemented to 3 × 3 sub-arrays similar to rows and columns. In Fig. 14, the final version of the temporary array is shown. These elimination steps must be repeated until each CANDIDATE vector has a single value. The final solution of the puzzle in this study is shown in the Fig. 15.

4. Conclusion and Future Work

In this study, we have implemented a hybrid SUDOKU puzzle solving algorithm for the purpose of recognizing the numbers in a SUDOKU image and finding the solution of the puzzle. Our study differs from similar studies in the literature via the reasons below:

- it considers both number images and empty square images in the same way,
- it transforms all images (numbers and empty cells) to a numerical array.

We have also observed that the resolution of the image, the noise in the image and font of the texts have an important effect on the performance of the proposed algorithm. We should mention here that our current paper is an extensively improved version of [25].

In future work, the proposed algorithm may be improved by using new images and image recognition methods. Also, it is known that some computer games such as Tetris and SOKOBAN are used for real-world fitting problems. Therefore, we think that there is a possibility for using SUDOKU for the same purpose. This study constitutes the first step of our thought and this algorithm may be improved for real-world fitting problems.

5	7	12348	6	9	12348	12348	12348	12348
69	8	69	2	3	1	4	7	5
4	3	1256789	1256789	1256789	1256789	1256789	1256789	1256789
9	258	6	1	4	258	7	258	3
12345679	12345679	12345679	12345679	8	12345679	12345679	12345679	12345679
8	147	3	147	5	9	2	147	6
1235689	1235689	1235689	1235689	1235689	1235689	1235689	4	7
3	9	5	4	1	7	28	6	28
34567	34567	34567	34567	2	8	34567	9	1

Figure 12. The new version of the temporary array after row elimination

7	1248	6	9	234	13	1238	248
8	9	2	3	1	4	7	5
3	12789	5789	67	256	1569	1258	289
25	6	1	4	25	7	258	3
12456	12479	3579	8	23456	13569	1235	249
14	3	7	5	9	2	1	6
1256	1289	3589	6	2356	13569	4	7
9	5	4	1	7	8	6	28
456	47	357	2	8	356	9	1

Figure 13. The new form of the temporary array after column elimination

5	7	12	6	9	4	13	1238	28
6	8	9	2	3	1	4	7	5
4	3	129	578	7	5	169	1258	289
9	25	6	1	4	2	7	58	3
127	1245	1247	37	8	236	59	15	49
8	14	3	7	5	9	2	1	6
126	126	128	359	6	356	35	4	7
3	9	5	4	1	7	8	6	28
67	46	47	35	2	8	35	9	1

Figure 14. The final version of the temporary array after 3 × 3 sub-array elimination

5	7	2	6	9	4	1	3	8
6	8	9	2	3	1	4	7	5
4	3	1	8	7	5	6	2	9
9	5	6	1	4	2	7	8	3
1	2	7	3	8	6	9	5	4
8	4	3	7	5	9	2	1	6
2	1	8	9	6	3	5	4	7
3	9	5	4	1	7	8	6	2
7	6	4	5	2	8	3	9	1

Figure 15. The final solution

Acknowledgment

The authors wish to thank for the support of the Research Fund of Istanbul University under Project N-53905.

5. References

[1] M. Simkova, "Using of Computer Games in Supporting Education", *Procedia - Social and Behavioral Sciences*, 141, pp.1224-1227, 2014.

[2] Y.G. Butler, "The use of computer games as foreign language learning tasks for digital natives", *System*, 54, pp.91-102, 2015.

[3] G.G. Smith, M. Li, J. Drobisz, H.R. Park, D. Kim, and S.D. Smith, "Play games or study? Computer games in eBooks to learn English vocabulary", *Computers and Education*, 69, pp.274-286, 2013.

[4] N. Alias, F. Rosman, M.N.A. Rahman and D. Dewitt, "The Potential of Video Game in Malay Language Learning for Foreign Students in a Public Higher Education Institution", *Procedia - Social and Behavioral Sciences*, 176, pp.1020-1027, 2015.

[5] I. Ahmad and A. Jaafar, "Computer Games: Implementation into Teaching and Learning", *Procedia - Social and Behavioral Sciences*, 59, pp.515-519, 2012.

[6] H.D. Mo and R.G. Xu, "Sudoku Square - a New Design in Field", *Acta Agronomica Sinica*, 34(9), pp.1489-1493, 2008.

[7] R. Kampf and E. Cuhadar, "Do computer games enhance learning about conflicts? A cross-national inquiry into proximate and distant scenarios in Global Conflicts", *Computers in Human Behavior*, 52(C), pp.541-549, 2015.

[8] H. Mahmoudi, M. Koushafar, J.A. Saribagloo, and G. Pashavi, "The Efect of Computer Games on Speed, Attention and Consistency of Learning Mathematics among Students", *Procedia - Social and Behavioral Sciences*, 176, pp.419-424, 2015.

[9] M. Bakker, M. Heuvel-Panhuizen and A. Robitzsch, "Effects of playing mathematics computer games on primary school students multiplicative reasoning ability", *Contemporary Educational Psychology*, 40, pp.55-71, 2015.

[10] F. Ke, "Computer-game- based tutoring of mathematics", *Computers and Education*, 60(1), pp.448-457, 2013.

[11] F. Ke, "An implementation of design-based learning through creating educational computer games: A case study on mathematics learning during design and

computing", *Computers and Education*, 73, pp.26-39, 2014.
[12] J. Cooper and A. Kirkpatrick, "Critical sets for Sudoku and general graph colorings", *Discrete Mathematics*, 315-316, pp.112-119, 2014.
[13] S.K. Jones, S. Perkins and P.A. Roach, "Properties, isomorphismsand enumeration of 2-Quasi- Magic Sudoku grids", *Discrete Mathematics*, 311(13), pp.1098-1110, 2011.
[14] R. Soto, B. Crawford, C. Galleguillos, E. Monfroy and F. Paredes, "A hybrid AC3-tabu search algorithm for solving Sudoku puzzles", *Expert Systems with Applications*, 40(15), pp.5817-5821, 2013.
[15] A.K. Maji, S. Jana and R.K. Pal, "An Algorithm for Generating only Desired Permutations for Solving Sudoku Puzzle", *Procedia Technology*, 10, pp.392-399, 2013.
[16] D. Dor and U. Zwick, "SOKOBAN and other motion planning problems. *Computational Geometry*, 13(4), pp.215-228, 1999.
[17] R.A. Hearn and E.D. Demaine, "PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraintlogic model of computation", *Theoretical Computer Science*, 343(1-2), pp.72-96, 2005.
[18] T. Anthony, D. Polani and C.L. Nehaniv, "General Self-Motivation and Strategy Identification: Case Studies Based on Sokoban and Pac-Man", *IEEE Transactions on Computational Intelligence and AI in Games*, 6(1), pp.1-17, 2014.
[19] M. Mahdian and E.S. Mahmoodian, "Sudoku Rectangle Completion", *Electronic Notes in Discrete Mathematics*, 49, pp.747-755, 2015.
[20] R. Bejar, C. Fernandez, C. Mateu and M. Valls, "The Sudoku completion problem with rectangular hole pattern is NP-complete", *Discrete Mathematics*, 312, pp.3306-3315, 2012.
[21] G. Santos-Garcia and M. Palomino, "Solving Sudoku Puzzles with Rewriting Rules", *Electronic Notes in Theoretical Computer Science*, 176, pp.79-93, 2007.
[22] K. Levenberg, "A method for the solution of certain problems in least squares", *Quarterly of Applied Mathematics*, 5, pp.164-168, 1944.
[23] D. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters", *SIAM Journal on Applied Mathematics*, 11, pp.431-441, 1963.
[24] S. Haykin, "Neural Networks and Learning Machines" (3rd ed.). Prentice Hall.
[25] S. Sevgen, E. Arslan and R. Samli, "Number Recognition of Sudoku Grid Image with Artificial Neural Networks", *In Proceedings of 22nd International Conference on Neural Information Processing (ICONIP 2015)*, pp.540-547, 2015.

Selcuk Sevgen is currently an Assistant Professor at the Department of Computer Engineering in Istanbul University Istanbul, Turkey. He received his M.Sc. and Ph.D. degree in same department in 2003 and in 2009, respectively. His main interests are Neural Networks, CNNs.

Emel Arslan was born in Istanbul, Turkey, in 1977. She received the B.Sc. and M.Sc. degrees from Trakya University, Edirne, Turkey, and Ph.D. degree from Istanbul University, Istanbul, Turkey, in 2001, 2004 and 2011, respectively.
She is currently working as an assistant professor in the Department of Computer Engineering, Istanbul University.
Her research interests are artificial neural networks, natural language processing, image processing applications and intelligent systems..

Ruya Samli is currently a Associative Professor at the Department of Computer Engineering in Istanbul University, Istanbul, Turkey. She received her M.Sc. and Ph.D at the same department in 2006 and 2011, respectively about stability of different types of neural networks. Her main interests are Neural Networks and modelling techniques.