

# Sudoku Solver Using Minigrad Based Backtracking Algorithm

M. Prabha<sup>1</sup>, S. Radha<sup>2\*</sup>, P. M. Priya<sup>3</sup>, B. S. Dhivya<sup>4</sup>

<sup>1</sup>Assistant Professor-III, Department of Information Technology, Velammal College of Engineering and Technology, Madurai, India

<sup>2,3,4</sup>Student, Department of Information Technology, Velammal College of Engineering and Technology, Madurai, India

**Abstract:** Sudoku is a famous Japanese puzzle game. It helps to enhance our logical thinking. Most of the people have a habit of solving sudoku puzzle in a daily routine. It consists of number ranging from 1 to 9. This has to be done in such a way that each number appears only once per row, once per column and once per zone. Solving sudoku puzzles are helpful in the fields such as steganography, secret image sharing with necessary reversibility, encrypting messages and image authentication. In this application, we used minigrad based backtracking algorithm for solving sudoku puzzles which is less time consuming.

**Keywords:** Sudoku puzzle, Minigrad, Backtracking algorithm.

## 1. Introduction

In today's busy world, it is difficult to find efficient solution for sudoku puzzle in a less amount of time. Hence, it is a good idea to develop an application where people can solve the sudoku puzzles in computerized manner.

Gaming plays a vital role in day-to-day life. It helps to build sound body and mind. Solving sudoku puzzles tunes our thinking. It improves our analytical thinking, reasoning ability and decision making which results in happy and healthy lifestyle.

It consists of rows and columns. The size of puzzle can be number of rows x number of columns. It consists of clues. Based on the given clues we have to fill the empty spaces with numbers. In our, application, we are using 9 x 9 model. There are 81 cells in total.

## 2. Literature Survey

This section reviews already published papers, which are found to be relevant to build the proposed application.

"An FPGA based Sudoku Solver based on Simulated Annealing Methods [1] reviewed on The Sudoku simulated annealing solver -SSAS is a probabilistic Sudoku solver.

"Solving Sudoku Puzzles using Improved Artificial Bee Colony Algorithm" [2] provides Sudoku puzzles belong to a set of hard problems called NP-Complete problems. A Sudoku puzzle is a logic-based combinatorial puzzle with rules that are relatively simple. Various algorithms have been applied to solve this combinatorial problem.

"A Deep Learning approach to solve sudoku puzzle" [3] explains the methodology involves several stages such as image

preprocessing and image extraction using OpenCV, OCRing and finally feeding the numerical data extracted to the neural network (tensorflow) model to get the desired output.

A Novel Automated Solver for Sudoku Images [4] addressed the attempts to explore the solving of Sudoku puzzles (as commonly found in newspapers and mobile games) using image processing, machine learning algorithms for OCR and an efficient solving algorithm to compute the correct answer.

Mixed handwritten and printed digit recognition in Sudoku with Convolutional Deep Belief Network [5] provides a method to recognize Sudoku puzzles containing both handwritten and printed digits from images taken with a mobile camera. The grid and the digits are detected using various image processing techniques including Hough Transform and Contour Detection.

Detection of Sudoku puzzle using image processing and solving by Backtracking, Simulated Annealing and Genetic Algorithms [6] helps to recognize any sudoku puzzle captured from a digital camera the digits are recognized using Optical Character Recognition (OCR) and based on their pixel locations in the image, they are stored in corresponding locations in the 9x9 matrix.

Camera-based Sudoku recognition with deep belief network [7] a method to detect and recognize a Sudoku puzzle on images taken from a mobile camera. The lines of the grid are detected with a Hough transform.

Solving Sudoku with Boolean Algebra [8] Sudoku solving technique named Boolean Sudoku Solver (BSS) using only simple Boolean algebras. Use of Boolean algebra increases the execution speed of the Sudoku solver.

## 3. Design and Implementation

### A. Design

In the Design Process, the flow of events in which the application works is designed. The flow chart (Figure 1) shows the flow of events in the application.

A user registration system has designed to get necessary information from the user. This is designed using Wamp server. It consists of the Apache web server, OpenSSL for SSL support, MYSQL, PHP programming language. MYSQL database is used to store and manage user data for authentication.

It gets name, phone number, gender, email id and password

\*Corresponding author: radhasethu.raman01@gmail.com

from user. After successful signup, users can login using registered email id and password.

The users can see a dashboard with options such as Home, Sudoku Solver and Logout. The user can choose from one of the options. Users can use these features to solve Sudoku and logging off.

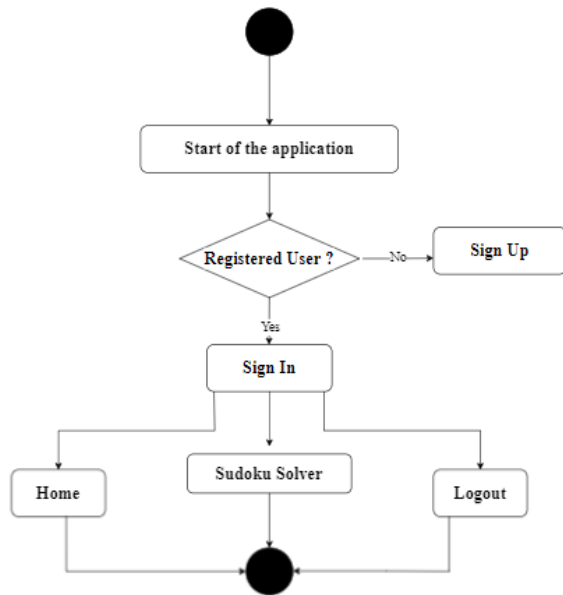


Fig. 1. Flowchart of the application

**B. Implementation**

The application is built using Python, Django, Keras, OpenCV which helps in implementing sudoku solver based minigrd using back tracking algorithm.

Python is an object-oriented programming language and we used to develop sudoku solver application using Pycharm IDE.

Django is a web framework which is used to develop frontend and backend.

OpenCV is an open-source library. It is used for face recognition and computer vision.

Keras is deep learning API used for implementing neural networks.

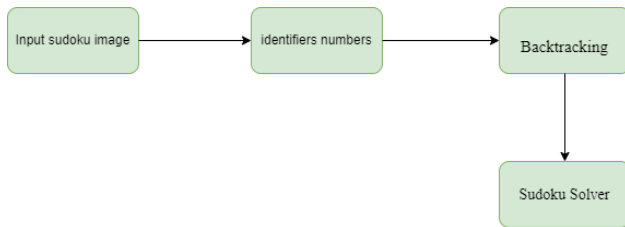


Fig. 2. Work flow

**4. Features of Application**

**A. Authentication**

**Registered Users** – Users can sign in using email id and password. Once the sign in button is clicked, it redirects user to the home page figure 3.

**New Users** – Users can sign up by providing details such as name, email id, phone number, gender, and password. Once the

sign-up button is clicked, it redirects user to the home page figure 4.

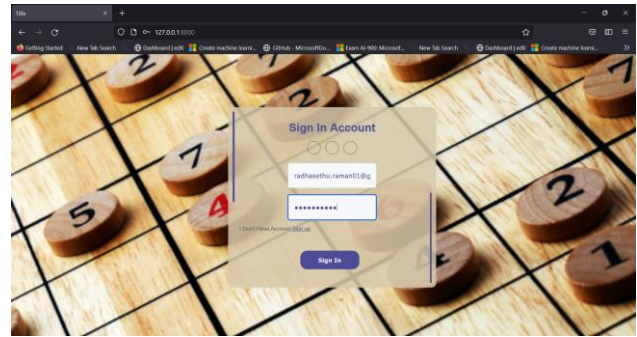


Fig. 3. Sign In page

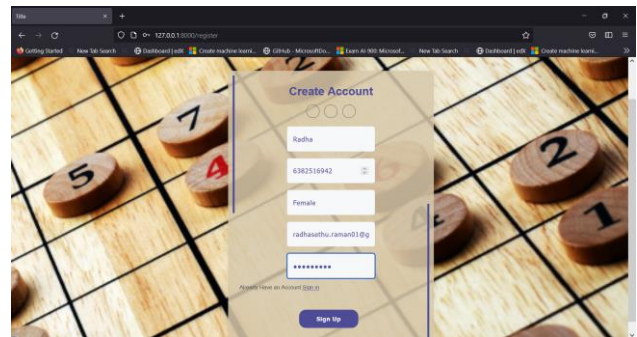


Fig. 4. Sign Up page



Fig. 5. Home page

**B. Sudoku Solver**

In our application, we developed an algorithm which performs 9\*9 is converted to 3\*3 minigrd and perform backtracking only on those places. The processing of sudoku solver consist of four steps image recognition, optimized character recognition, calculating threshold value, evaluating values for each cell.

*Image Recognition:*

User can feed an input image to this application figure 5.

In this step, Digits are recognized in the image.

*Optimized Character Recognition:*

OCR technique identifies the character correctly which is presented in the cell irrespective of the font size and font style.

*Calculating Threshold Value:*

In this step it will assign threshold value to each and every cell in the grid. If it is an empty cell then it assigns threshold value to zero otherwise it assigns threshold value to one. At the end of the process the cells which contains threshold value one

is replaced by input values.

#### Evaluating values for each cell:

In this step the 9\*9 grids are separated in to 3\*3 grids. This 3\*3 grids are handled as a single grid. The empty cells have been filled with correct values with step-by-step manner using the backtracking algorithm. The final evaluated values are appeared as green colour and given input values are appear as black colour.

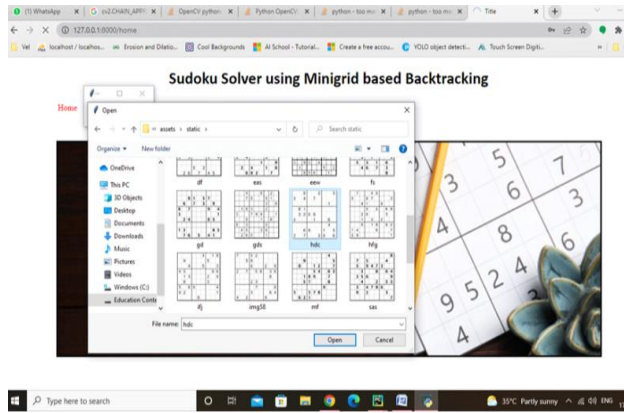


Fig. 6. Fetching input image

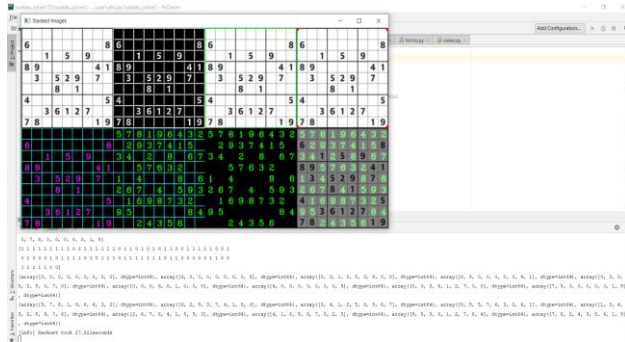


Fig. 7. Final output

## 5. Conclusion

This application is really helpful for people who are really enthusiastic in solving sudoku puzzles even though they are very busy with their schedules. These people can start using this application to attain solution with less amount of time. Minigrad based back tracking algorithm optimizes solution and time. Our application shows how much time taken to solve each puzzle.

## 6. Future Scope

The application can be further improved by suggestions and feedbacks. In our application, size of the minigrad will be enhanced above 9 x 9. In our application, complexity level will be increased.

## References

- [1] Rajneesh Tiwari, Aritra Sen, Arindam Banerjee, "Generalization through augmentation in deep neural networks for handwritten character recognition", International Journal of Scientific & Engineering Research Volume 11, Issue 9, September 2020.
- [2] Azhar Talha, Syed, Suresh Merugu, "Augmented Reality on Sudoku Puzzle using Computer Vision and Deep Learning", International Journal of Innovative Technology and Exploring Engineering, Volume 8, Issue 11S2, September 2019.
- [3] Kshitij Gupta, Sagar Khatri, M. Khan, "A Novel Automated Solver for Sudoku Images", IEEE 5th International Conference for Convergence in Technology (I2CT), March 2019.
- [4] P. J. Simha, K. V. Suraj, T. Ahobala, "Recognition of Numbers and Position using Image Processing Techniques for Solving Sudoku Puzzles", Image and Graphics: 8th International Conference, ICG 2015, Tianjin, China.
- [5] Baptiste Wicht, J. Hennebert, "Mixed handwritten and printed digit recognition in Sudoku with Convolutional Deep Belief Network", Thirteenth International Conference on Document Analysis and Recognition (ICDAR), August 2015.
- [6] Dhanya Job, Varghese Paul, "Recursive backtracking for solving 9\*9 sudoku puzzle", 2016.
- [7] Sunanda Jana, Arnab Kumar Maji, Rajat Kumar Pal, "A novel sudoku solving technique using column-based permutation," 2015 International Symposium on Advanced Computing and Communication (ISACC), pages 71–77, 2015.
- [8] A. Van Horn, "Extraction of sudoku puzzles using the Hough transform", University of Kansas, Department of Electrical Engineering and Computer Science, Tech. Rep., 2012.
- [9] B. Wicht, Hennebert, "Camera-based sudoku recognition with deep belief network in Soft Computing and Pattern Recognition (SoCPar)", 2014 6th International Conference, Aug. 2014.
- [10] Hung-Hsuan Lin, I-Chen Wu, "Solving the minimum sudoku problem", 2010 International Conference on Technologies and Applications of Artificial Intelligence, pp. 456–461, 2010.
- [11] Tanh Minh Ly, Dung Trung Vo, "A novel and automatic character extraction and recognition for sudoku puzzle solving," 2015 International Conference on Advanced Technologies for Communications (ATC), pp. 546–550, 2015.
- [12] Arnab Kumar Maji, Rajat Kumar Pal, "Sudoku solver using minigrad based backtracking," 2014 IEEE International Advance Computing Conference (IACC), pp. 36–44, 2014.
- [13] S. M. Shamim, Mohammad Badrul, Alam Miah, Angona Sarker, Masud Rana, Abdullah Al Jobair, "Handwritten Digit Recognition using Machine Learning Algorithm," in Global Journal of Computer Science and Technology: D Neural & Artificial Intelligence, Volume 18, Issue 1, 2018.