**SUDOKU GAME WITH AI**

# BACHELOR OF TECHNOLOGY in

**COMPUTER SCIENCE AND ENGINEERING**

By

**RAVI KUMAR**

**Registration number-11905393**

**Roll No: B44**

**Section: K18 GX**



**School of Computer Science and Engineering**

Lovely Professional University

Phagwara, Punjab (India)

**Table of Content**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TOPIC** | **PAGE NO** |
| **1** | **Acknowledgements** | **3** |
| **2** | **Abstract** | **4** |
| **3** | **Introduction** | **5** |
| **4** | **Problem Statement & Objectives** | **5** |
| **5** | **Objectives** | **6** |
| **6** | **Constraints** | **6** |
| **7** | **Scope of Sudoku** | **7** |
| **8** | **References** | **10** |

**Acknowledgements**

I am profoundly grateful to **Sagar pande** for his expert guidance and continuous encouragement throughout to see that this project rights its target. I would like to express deepest appreciation towards.

HOD Computer Department whose invaluable guidance supported me in this project. At last I must express my sincere heartfelt gratitude to all the staff members of Computer Engineering Department who helped us directly or indirectly during this course of work.

**Name of the Candidate**

**RAVI KUMAR**

**R. No.-44**

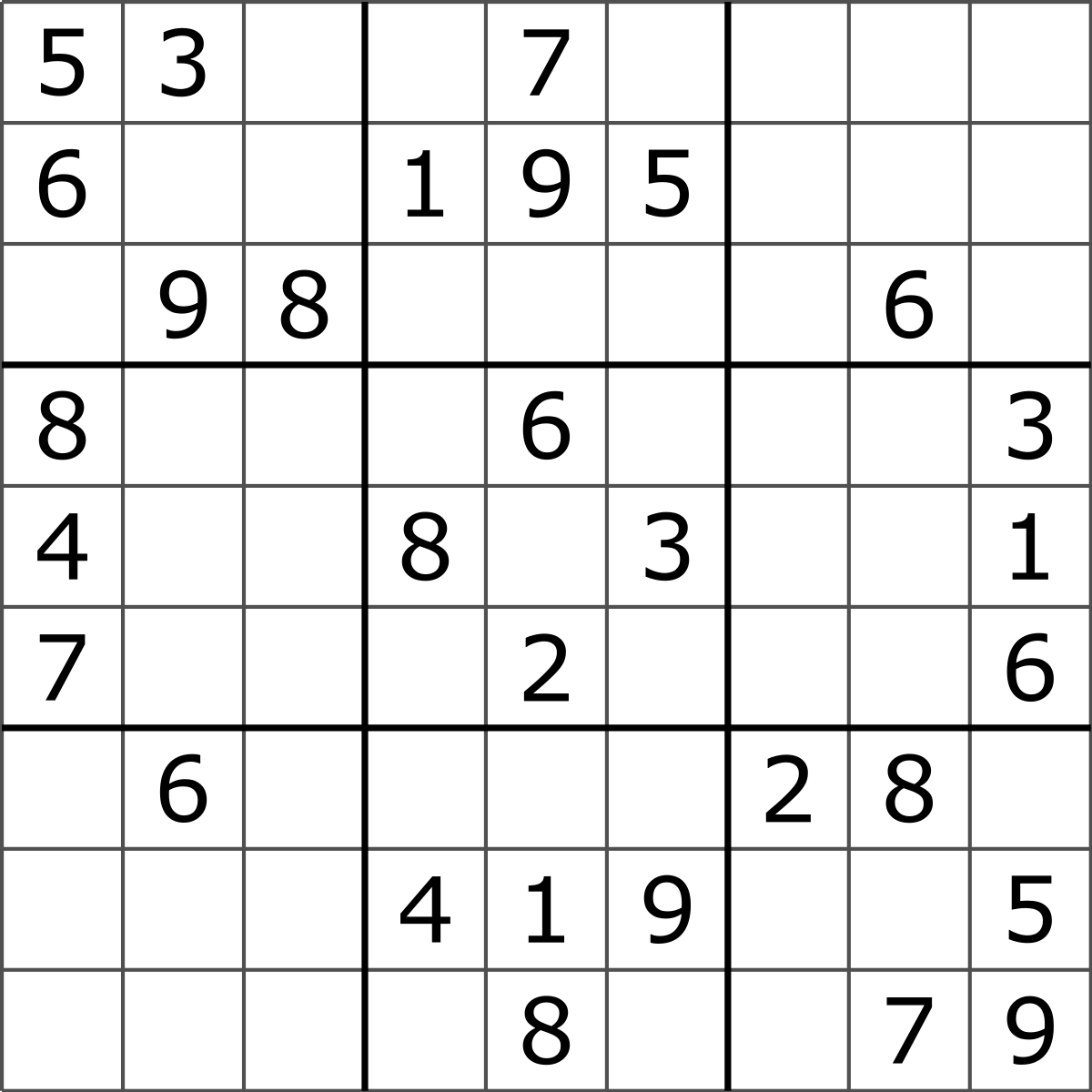
**ABSTRACT**

In our project we propose a method of detecting the elements of a Sudoku Puzzle and providing a digital copy of the solution for it. The method involves a vision-based Sudoku solver. A virtual grid is then created to identify the digit positions. The actual solution is computed using a backtracking algorithm. Experiments conducted on various types of Sudoku questions demonstrate the efﬁciency and robustness of our proposed approaches in real-world scenarios. The algorithm is found to be capable of handling cases of translation, perspective, illumination gradient, scaling, and background clutter.

**Introduction**

In real life we come across Sudoku puzzles of varying difﬁculty levels in newspapers and other text and digital media. It is a common leisure activity for a lot of people. However, it is observed that the solution is not always immediately available for the users veriﬁcation. In most cases, people have to wait till the next day to check the solutions of the Sudoku they just solved. Hence our motivation for this project was to develop an application on an android device for this purpose. In our application which then returns the complete solution of the same.

Figure 1.1: Example of Sudoku Puzzle



Sudoku is a logic based puzzle with the goal to complete a 9x9 grid so that each row, each column and each of the nine 3x3 boxes contain the numbers 1 through 9, given a partial fling to a unique solution. The problem itself is a popular brainteaser but can easily be used as an algorithmic problem, with similarities to the graph colouring problem only with ﬁxed size and dependencies. The modern version of Sudoku was introduced in Japan in the eighties and has since then attracted a lot of programmers and therefore there are a great deal of different Sudoku solving algorithms available on the Internet, both implementations and construction methods. Sudoku solvers are interesting because they are practical applications of very theoretical problems, and most people are familiar with them.

**Problem Statement & Objectives**

The task is to solve the Sudoku puzzle in as much less time as possible. It can be done by investigating different techniques for solving Sudoku and comparing them for the most efﬁcient solution. Sudoku itself can be solved using brute-force in a reasonable amount of time in most cases, but there are special cases where it takes a long time to brute-force. Therefore, our task is to try to ﬁnd efficient algorithms for all instances of the problem and evaluate them while using the optimal solver to get the solution for the Sudoku puzzle. There are two main constraints that determine the efﬁciency of an optimal solver. They are

Time consumption and Memory consumption whose degree of satisfaction determines the quality of an optimal Sudoku solver. These constraints usually vary from algorithm to algorithm.

**Objectives**

The objective of this project is to solve Sudoku with the use of recursive back tracking algorithm while satisfying the game constraints.

**Constraints:**

• Sudoku puzzle can only contain the numbers 1 through 9.

• A position constraint: Only 1 number can occupy a cell

• A row constraint: Only 1 instance of a number can be in the row

• A column constraint: Only 1 instance of a number can be in a column

• A region constraint: Only 1 instance of a number can be in a region

**SCOPE OF SUDOKU**

The objectives of proposed Project is to increase the thinking

Capability.

The Game having all the records which u perform in playing you can select easy, hard level according to your choice. You

Can make your own Sudoku and at any step you can go back to one step as well as you can see the solution of it.

It is manually a very difficult job to perform and it need a lot of recallying, reminding and mathematical calculation. The game of Sudoku help to increase mental thinking, vision e.t.c

**Code of Sudoku game**

A screenshot of a video game

Description automatically generated

A screenshot of a video game

Description automatically generated

A screenshot of a video game

Description automatically generated

**References**

1. [**https://www.geeksforgeeks.org/sudoku-backtracking-7/**](https://www.geeksforgeeks.org/sudoku-backtracking-7/)
2. <https://www.youtube.com/sudoku%20game>
3. https://www.google.com/search?q=sudoku+game+with+ai+in+python&ie=&oe=