

# **COMP4102 Group Project Proposal**

**Title: Sudoku Solver**

**Team Members:**

Wenyu Zhang    100941511

Bowen Zuo     100951180

**Winter, 2020**

**Carleton University**

## Summary:

This project is designed to recognize any Sudoku from an image and display as digital format. Then, the program is aiming for solving the Sudoku and display answer on the image.

## Background:

Modern Sudoku was firstly introduced and got popular in Japan in 1800s. This fun and logical game was soon spread to all over the world and started to present on newspapers and books. Standard Sudoku is a 9x9 grid comprised of nine 3x3 sub-grids. To complete the puzzle, each row, column and sub-grid should contain all of the digit from 1 to 9. Moreover, some irregular and more advanced Sudoku are developed with irregular sub-grid or more complicated rule. For example, some position can only hold odd or even numbers (Fig. 1)<sup>1</sup>. This project is inspired by the existing mobile application called iPhone Sudoku Grab<sup>[2]</sup> which could search grid and read number in a standard Sudoku. This project will be built with more solving function that aim to help users to solve their Sudoku as quick as possible.

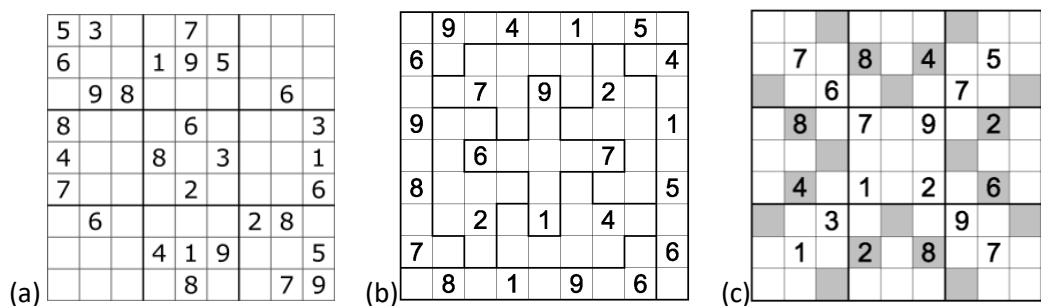


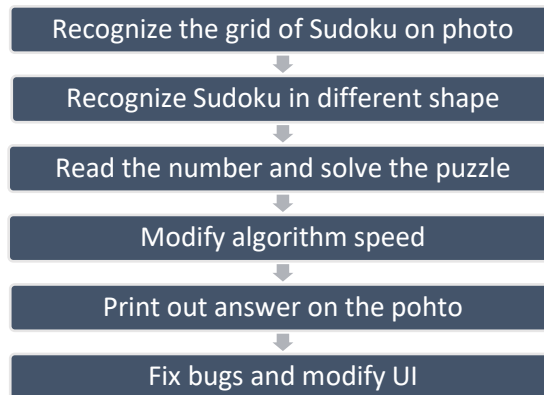
Fig. 1. Illustration of standard Sudoku (a), irregular Sudoku (b) and even/odd number Sudoku (c).

## The Challenge:

The challenge of this project is to recognize and solve the Sudoku correctly. Though there are some source of reference online, it would be still challengeable to recognize number correctly, especially for those pictures with low resolution. Also, we would like to challenge some irregular shaped Sudoku, which makes the reading part much harder than standard Sudoku. The Sudoku recognition can be implemented by using some features in OpenCV, while the solving algorithm is what we need to work with. Also, the program should be designed to have a rapid response time and solving time. Thus, the modification on either the recognition algorithm or the solving algorithm is desirable. After the completion of this project, we hope we could get very familiar to the OpenCV, not only focused on those feature we need, but also some interesting and useful features that could be helpful in solving other queries.

<sup>1</sup> <http://www.worldpuzzle.org/championships/types-of-puzzles/wsc/>

<sup>2</sup> <http://sudokugrab.blogspot.com/2009/07/how-does-it-all-work.html>



## Goals and Deliverables:

- PLAN TO ACHIEVE:

- Auto-detect a standard Sudoku from an imported image by user
- Grid and cut the image to be a square
- Print out the Sudoku image in plain text
- E.g.

```

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

```

```

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

```

```

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

```

- Solve the Sudoku and print out in plain text shown as above
- Print the answer on top of the photo

- HOPE TO ACHIEVE:

- Improve the algorithm to be able to complete more complicated Sudoku (e.g. special shape)
- Improve the algorithm speed

- Validation:

- The program should be able to deal grey-scale and colored pictures and recognize the Sudoku shape correctly.
- When dealing with complicated Sudoku, the program should detect bolded line from regular line.
- The developed program should solve the Sudoku correctly

- When printing out answer on the picture, the designed program should print out each number on correct position.
- Implementation:
  - There are a couple of improvement this program could have. However, within a limit of time, this program will be built by starting with a prompt user interface on terminal-end, and print out any output in plain text.
  - If time allows, a better user interface on either mobile OS or PC will be implement.
  - This program will firstly focus on solving traditional Sudoku. Any functionalities of solving more advanced Sudoku will be added subsequently

**Schedule:**

Week	Team members	
	Wenyu Zhang	Bowen Zuo
<b>Feb 3, 2020</b>	Determine UI language, find the library in OpenCV for gridding.	Research on available source and find tools for processing the image.
<b>Feb 10, 2020</b>	Write draft coding for grid recognition	Write draft coding for grid recognition
<b>Feb 17, 2020</b>	Implement grid recognition	Implement number recognition
<b>Feb 24, 2020</b>	Prepare for midterm	Prepare for midterm
<b>Mar 2, 2020</b>	Design user interface	Implement Sudoku solving
<b>Mar 9, 2020</b>	Add features	Optimize the solving algorithm
<b>Mar 16, 2020</b>	Fix bugs and polish syntax	Fix bugs and polish syntax
<b>Mar 23, 2020</b>	Prepare for demo + testing	Prepare for demo + testing
<b>Mar 30, 2020</b>	Demo + Report editing	Demo + Report editing
<b>Apr 6, 2020</b>	Fix minor errors	Finish writing report