

# **WIDS: PROJECT REPORT**

**Name:** Bhavin Hareesh Jain

**Roll No:** 22B0685

**Project UID:** 20

**Project Name:** Sudoku Solver

## **What this project is about?**

Everyone likes solving Sudokus. Some are quite easy; some are really tough and solving them has always been interesting. But to do it using coding and ML was another task altogether. Basically, this project could be divided in 6 steps:

1. Use Four-Perspective Transform to detect the sudoku puzzle in the image and align in it a standard size rectangle.
2. Then, convert it into Black and White for Standard Purposes.
3. Now, separately, train a model using MNIST dataset and your own validation dataset.
4. Now, continuing step 2, we will need to extract all 81 cells individually. Then, we will preprocess each cell and resize it to a standard size.
5. We will identify the empty and filled cells by using our model.
6. We will first solve the sudoku by filling digits 1 to 9 in the empty cell and keep checking the conditions.
7. Then, eventually our sudoku will be solved.

## **How was ML involved in this project?**

Machine Learning was used in this project in the following ways:

- i) A model was trained on the basis of training data and validation data. This is a CNN Model implemented through TensorFlow and Keras.

- ii) The code uses a backtracking algorithm to solve the sudoku puzzle. Basically, my code recursively tried different digit placements and backtracks if a solution is not valid. It will keep continuing till a valid solution is found.
- iii) After extracting individual cells from sudoku puzzle, the code utilizes the ML model to recognize digits in each cell.

Overall, ML was the main pillar of my project.

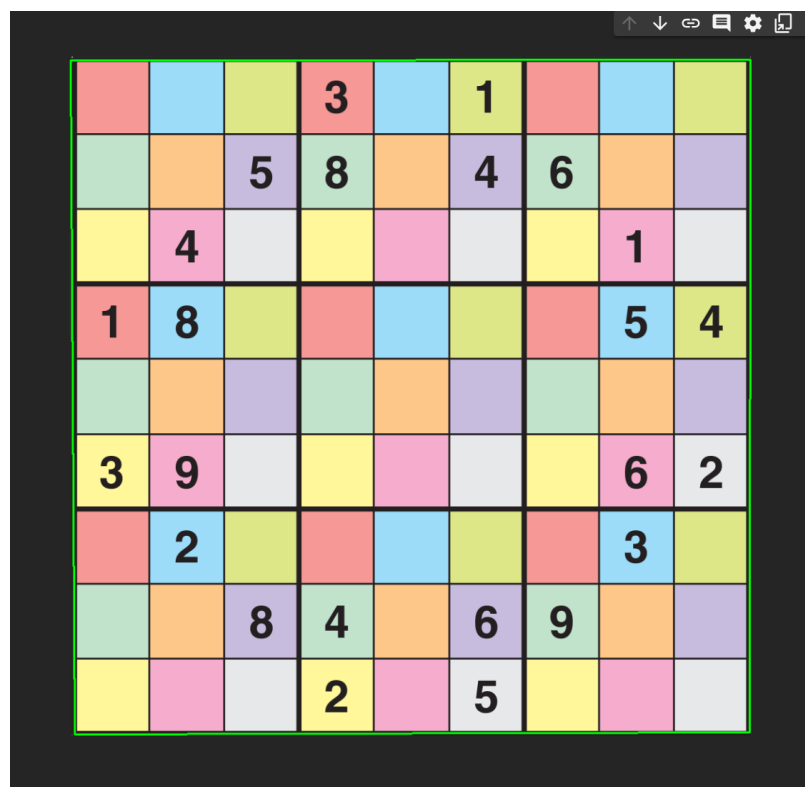
## What new things did I learn?

I learnt a lot of new things while doing this project. I did many of the things for the first time in this project.

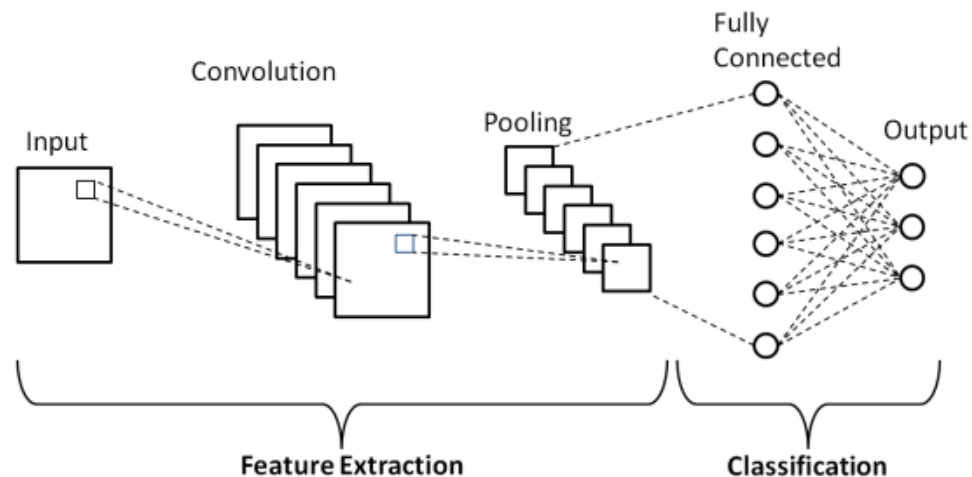
1. Contour Detection and Four Point Transform – I learnt this to outline the sudoku in any image and then extract the sudoku puzzle from the image.

I implemented this by using cv2 and imutils module.

2. Image Processing Techniques – We also used methods such as Gaussian blur and adaptive thresholding to pre-process the Sudoku puzzle image. In the image below, green line can be seen which indicates the boundary from where the image will be extracted.



3. I had learnt briefly about Neural Networks in my AI&DS course but in this model, I learnt Convolutional Neural Networks. My model was basically a CNN implemented using TensorFlow and Keras. These are some of the new modules which I also had to learn.



4. I learnt how to implement a backtracking algorithm.

### Some Key Findings from this project

1. Digit Recognition Model is quite effective.
2. Image preprocessing Techniques are quite useful and can be relied upon.
3. On average, this model takes about 8-15 seconds to implement.
4. The time of implementation of model varies depending upon the hardness level of the sudoku.
5. Contour Detection works quite effectively and till now I have not encountered any case where it may not have worked.
6. My model works for all types of images due to the variety of the training and validation data set used while making the model.

```
1/1 [=====] - 0s 28ms/step
1/1 [=====] - 0s 27ms/step
1/1 [=====] - 0s 26ms/step
1/1 [=====] - 0s 28ms/step
1/1 [=====] - 0s 38ms/step
1/1 [=====] - 0s 28ms/step
1/1 [=====] - 0s 30ms/step
1/1 [=====] - 0s 28ms/step
1/1 [=====] - 0s 28ms/step
1/1 [=====] - 0s 31ms/step
1/1 [=====] - 0s 30ms/step
1/1 [=====] - 0s 30ms/step
1/1 [=====] - 0s 27ms/step
1/1 [=====] - 0s 28ms/step
Solved Sudoku:
[[2 6 7 3 5 1 4 8 9]
 [9 1 5 8 2 4 6 7 3]
 [8 4 3 6 7 9 2 1 5]
 [1 8 2 9 6 3 7 5 4]
 [5 7 6 1 4 2 3 9 8]
 [3 9 4 5 8 7 1 6 2]
 [4 2 1 7 9 8 5 3 6]
 [7 5 8 4 3 6 9 2 1]
 [6 3 9 2 1 5 8 4 7]]
```