

ANALYSIS:

For this assignment, the work was divided into 3 phases, first being to extract sudoku grid from given image, second being extracting digits and recognising digits from the image of each extracted digit and third being to solve the sudoku and show its solution in the original image.

1) GRID EXTRACTION:

Here, i divided the work into certain stages to help debug easily if errors pop up. It took a quite bit of work and i took 3 stages to do this, first being to pre process the image, second being to detect the sudoku using the largest contour and applying four point transform and third being to extract each cell from the rescaled sudoku and checking whether a digit is present or not , again using contour detection. Here, the third stage took quite a bit of time and for a few sudokus, its wasn't detecting whether a digit is present or not correctly due to huge border. So i had a manual rescale the sudoku in a few cases for it to work properly.

2) DIGIT RECOGNIZER:

This part also took a quite bit of time since my old model trained on MNIST dataset was a simple NN with around 97.63 accuracy, which isn't enough to detect printed digits well. So i made a new model by implementing CNN with 2 convolution layers. This helped to get my accuracy up to 98.66 percent and it could detect digits well from the grid now. It still wasn't 100 percent accurate in detecting digits due to various reasons like printed digits are kinda diff from the MNIST dataset where the model was trained, then during digit extraction, many times the digits couldn't be centred perfectly and this caused few inefficiencies. Most common error from these being detecting 1 and 7, this happens because printed 1 looks like a handwritten 7 than a handwritten 1 and this caused a few issues. Another issue was 5 and 9, again whose printed forms are little different from their handwritten forms.

During this part, i also found out a few sudokus which had more than 3 errors while detecting digits, this were mainly sudokus whose numbers had different fonts, hence looked different from the the handwritten digits in MNIST dataset. This part can probably be improved by training the model on a custom dataset of printed numbers, with different fonts. One more issue was orientation of the digits. Some of the digits weren't properly oriented after the contour detection and this led to inaccurate results. This part can probably fixed by rescaling and reorienting each digit to the centre of its cell after extraction.

3) SUDOKU SOLVING:

This part was fairly simpler than the other 2 parts. I implemented a simple algorithm to solve the sudoku using recursive backtracking. This part didn't give many issues and worked pretty well. After solving, i also reprojected the solved digits to the original image in a diff colour(blue) than the other digits to signify the added digits.

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