

Computer Vision based Sudoku Solving with Augmented Reality

Gummadi Sai Dheeraj¹, K.B.V.Lakshmi², K.Anjan Krishna³

^{1,3}Department of Electronics and Communication Engineering, SRM University, Andhra Pradesh, Guntur

Abstract - The computer vision absolutely alludes to picture handling, these pictures are just the stuff of a way comprehensive, the indistinguishable as endeavors to mimic human perceptual capacities, Sudoku is one in all the foremost popular puzzle rounds ever, Sudoku might be a combinatorial puzzle during which numbers are be set in a really 9*9 lattice, which is split into 3*3 sub-networks. The matrix is somewhat finished, and each Sudoku envelops a distinct arrangement. The target of the riddle is to fill the matrix with numbers 1-9, without the reiteration of variety in very line, section or the sub-lattice. Trouble of the matter depends on the incompletely finished lattice. All the work was created in Python utilizing Pycharm IDE, OpenCV, Keras and Tensorflow Libraries and furthermore the code to determine the Sudoku is open source

Key Words: Sudoku, OpenCV, Deep Learning, Augmented Reality, OCR

1. INTRODUCTION

The part of computers in puzzle world is currently turning out to be increasingly significant, since PCs changed riddle makers, yet additionally puzzle solvers [1]. Allmost all riddles are only for recreation. A sort of puzzle is Sudoku (pencil and paper puzzles). The word Sudoku signifies 'the digits must stay single.

Limitation fulfillment may be defined as a cycle with a collection of factors having requirements forced on them that need to be fulfilled. There are different genuine models where CSP is utilized, like automated arranging and booking of classes for the semester. Speaking to Sudoku as a Constraint Satisfaction Problem and utilization of spread procedures permit the riddles to be comprehended with polynomial time thinking.

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

Figure 1

Some works are done to recognition the numbers of Sudoku and solved it like [4], in [5] a minigrad based novel technique is developed to unravel the Sudoku puzzle in

guessed free manner. Additionally, other works studied that randomized guess choices are way more efficient, on average, in solving Sudoku puzzles than ordered ones, and will be considered as further optimization factor[6]. Papers like [7] examine the effect of stochastic approaches when solving Sudoku games.

Humans are thinking of machines with the power to "read" and "decipher" printed literary documents, in request that they'll be consequently changed over into a substitute medium or configuration[8]. Optical Character Recognition(OCR) let machines to acknowledge characters. As a model, OCR is employed to acknowledge optically handled printed character in car place number which is predicated on format coordinating [9]. An audit of the OCR history and consequently the various procedures utilized for OCR development within the chronological request is being exhausted [10].

These days the computers includes a lot of learning ability even is regular that has a coordinated webcam, the necessity of clear and reasonable images brought humanity to make techniques to enhance a picture, if important decrease commotion, obscure, modify the splendor, shading transformation, and so on

Our work builds up a program to identify the Sudoku region tackle it and print the answer with expanded reality on a screen. We move just the areas of numbers within the grid and distinguish if encompasses a number or is unfilled, if variety is present no action is completed but if the matrix position is vacant, consequently we put a zero.

2. Information

The remainder of this paper is composed as follows. Section 2.1: brief view of components utilized in the implementation of algorithm. Section 3 gives a proof about the steps for image processing. Finally Section 4 gives conclusion

2.1 Principal Components

The major components for the development of the algorithm are:

- 1) Python
- 2) OpenCV
- 3) Tensorflow
- 4) OCR(optical recognition)
- 5) Sudoku Solver Algorithm

2.1.1 Python Language

We program in general Python language that may be executed on the foremost common working systems, as an example in Linux, Mac OSX and Windows, the particular work was created in Windows climate.

2.1.2 OpenCV Libraries

OpenCV could be a library of programming capacities mainly geared toward real time computer vision, created by Intel Russian exploration, was intended for computational effectiveness, the library can make the most of the equipment increasing speed of the basic heterogeneous process stage [13]

2.1.3 Optical Character Recognition (OCR)

So as to peruse the numbers we use OpenCV because it maybe the foremost accurate open source Optical character acknowledgment (OCR) motor accessible. Joined with theLeptonica Image Processing Library it can read a large style of image formats and convert them to message in more than 60 languages. It had been one in all the highest 3 engines within the 1995 University of Nevada city Accuracy test. Somewhere in the range of 1995 and 2006 it had little work done on that, however since then it's been improved widely by Google. It released under the Apache License 2.0 [14], likewise as OpenCV is upheld by the foremost commons stages, the library chips away at Linux, Windows and Mac OSX, it may also compiled for other plat- structures, including Android and likewise the iPhone

2.1.4 Sudoku Solver

We utilized the code in [15] where explain Sudoku by individually appointing numbers to purge cells. Before assigning variety, the calculation checks whether it's safe to dole out. It fundamentally checks that the identical number isn't present in current line, current segment and current 3 × 3 sub lattice.

Subsequent to checking for security, it assigns the amount, and recursively checks whether this assignment ends up in an answer. On the off chance that the task doesn't cause an answer, at that point the calculation attempt next number for current void cell. What's more, if none of number (1 to 9) cause solution, we return bogus.

3. Problem Description

So as to tackle the sudoku issue, we initially need to part the problem up into 6 parts,

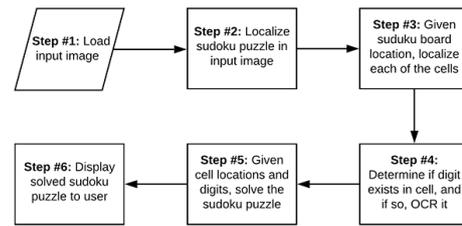


Figure2: Work flow

3.1 Sudoku Net

Each sudoku puzzle begins with a NxN lattice, where a few cells are clear and different cells as of now contain a digit. The objective is to utilize the information about existing digits to properly infer the opposite digits.

Yet, before we tackle sudoku puzzles with openCV , we first have to implement a neural specification that may handle OCR'ing the digits on the sudoku puzzle board. We name it SudokuNet

3.2 Convolutional neural network

CNNs are an expansion of multilayer perceptrons, which might learn filters that require to be processed by the machine learning models, as tested prior in [Yann LeCun et al. 1989] utilizing back-propagation. Convolutional networks are primarily applied on visual symbolism. Since the preparation cycle includes finding out about examples from more modest examples, A convolutional neural organization comprises of particular covered up layers additionally to the information and yield layers.

These particular layers usually incorporates convolutional layer with filters which will be learned, rectified long measure layer for utilization of enactment work, pooling layer for down-testing and misfortune layer for determination of punishment for wrong output. We use Keras with TensorFlow as backend to handle the CNN model and supply a examination between the particular CNN usage framed by picking distinctive hyperparameters related with each layer.

3.3 Training phase

We can begin preparing our SudokuNet model, we will utilize the MNIST informational index for perceiving digits. We start our preparation with a little modest bunch of imports. Most eminently, we import MNIST informational collection of transcribed digits, which is incorporated right with Keras informational collections module.

We took hyperparameters as in Table.1

Hyper -Parameters	Values
Learning Rate	1e-3
Epochs	50
Batch Size	128

We fundamentally split our informational collection into training information and testing information. We add a channel measurement to the digits to demonstrate that they are in grayscale. We scale them into a scope of 0 and 1, we utilize One-hot encoder to isolate and perceive the numbers with 0 and 1. for instance number3 is represented as [0,0,0,1,0,0,0,0,0] in one-hot encoding.

Preparing is dispatched through a call to the fit strategy, when preparing is finished, we will assess the model and fare our model.

3.4 Sudoku Puzzle Board

Now, we have a model that can perceive digits in the picture. Nonetheless, that digit recognizer doesn't do us much good in the event that we cannot find the sudoku puzzle board in a picture.

"find_puzzle", finds and concentrates the sudoku puzzle board from the information picture. "extract_digits", inspects every cell of the sudoku puzzle board and concentrates the digit from the cell.

Moreover, we will utilize "clear_boarder" routine in our "extract_digit" capacity to tidy up the edges of sudoku cell. Most tasks will be driven with OpenCV with a smidgen of help from Numpy and imutils.

3.5 find_puzzle

"find_puzzle" work starts things out and acknowledges two boundaries, "input picture" and "troubleshoot". "investigate" is a discretionary boolean sign whether to show middle advances so you are more likely envision what's going on in the engine of our computer vision .

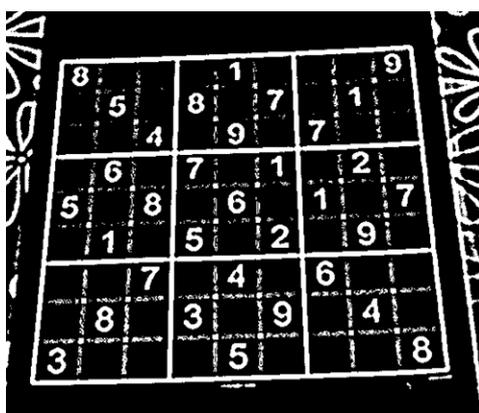


Figure3

We can see something similar to Fig3, if we set debug to true.

One of our forms will compare to the diagram of the sudoku lattice. In the event that the sudoku puzzle isn't discovered we raise Exception.

We will imagine what is happening in the engine by drawing the riddle form layout on the picture

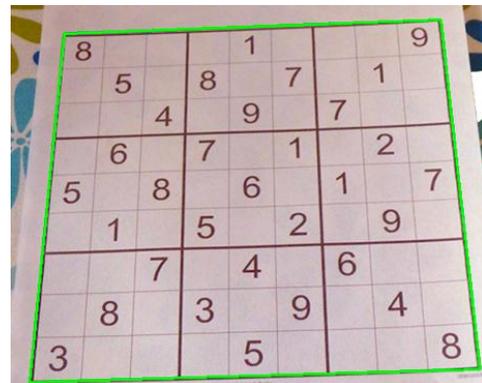


Figure 4

With the shape of the riddle close by, we can deskew the picture to acquire a top-down feathered creatures eye perspective on the riddle.

Applying twist point of view change viably deskews our sudoku puzzle matrix, making it a lot simpler for us to decide lines and sections, and cells as we push ahead. This activity is performed on the first RGB and dim pictures.

Final image after applying find_puzzle function is shown below



Figure 5

Our find_puzzle returns mark of a 2-tuple of the first RGB picture and grayscale picture after all activities, including the last four-point viewpoint change

3.6 Extracting Digits

In our previous section, we extracted sudoku puzzle board from an image with OpenCV

In this segment, we will inspect every one of the individual cell in the sudoku board, recognize if there is digit in the cell and assuming so, remove the digit.

We characterize "extract_image" capacity to acknowledge two boundaries, "cell" and "troubleshoot"

Cell: A Roi speaking to an individual cell of the sudoku puzzle

Troubleshoot: A boolean sign whether halfway stem perceptions ought to be appeared on screen.

Our initial step, is to clear the frontal area pixels that are contacting the fringes of the phone and discover the shapes in edge cell, if no forms are discovered, we return "None".

From our forms we locate the biggest shape by pixle region and build a related cover

Isolating the pixel region in to cover by the region of tyhe cell itself gives us the percentfilled esteem.

Expecting we don't have commotion in the cell, we apply veil to the thresholded cell. This veil is alternatively appeared on screen.

3.7 Sudoku Solver

At this point we are armed with SudokuNet model, warp perspective transform, extracting digits, sudoku solver.

Accounting for every cell in t he sudoku puzzle, we loop over rows and columns in a nested fashion

Inside, we use our step values to determine the starting and ending coordinates of the current cell.

Now, we append the coordinates as a tuple to this particular row. Each row will have Nine entries.

Now, we are ready to crop out the cell and recognize digits.

Harvest the cell from the changed picture and concentrate digit. In the event that the digit isn't none, at that point we know there is genuine digit in the cell, we pre-measure the digit in a similar way that we did in preparing and characterize the digit with SudokuNet and update the sudoku puzzle board cluster with the anticipated estimation of the phone.

As should be obvious, first we show sudoku puzzle board as it was deciphered by means of OCR.

To clarify our picture with the arrangement numbers, we just circle over cell areas and the board and unload cell organizes and figure directions of where text explanation will be drawn.

Presently, we draw each yield digit on our riddle board photograph and show our settled sudoku puzzle picture until any key is squeezed.

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

Figure 6

4. CONCLUSIONS

- OpenCV is the best free device for character acknowledgment yet it is exceptionally delicate to the commotion
- Characters misread likewise create blunders in the arrangement of Sudoku.
- We make a capacity to discover precisely if each crate was at that point full or we need to fill it.
- We built up a framework to deal with numbers and position individually on the grounds that help to found all the crates on the matrix.
- The proposed calculation functions admirably for both: turn and planar uprooting insidethe picture caught by the camera yet isn't proposed for cases in which there is point of view, can raise this as additional reproducing work.
- At Fig. 6 shows the calculation goal, the numbers are printed vertically in light of the fact that are not designed to pivot with the picture, are arranged to follow the right position to be printed however not for have a point of view.
- The execution season of the calculation is affected by the phase of mathematical goal, which thusly relies upon how well they have been perused the numbers present in the Sudoku, if all stages are well the settling is appeared progressively

REFERENCES

[1] Higashida, H.: The role of computers in puzzle world. In: Second International Conference on Culture and Computing (Culture Computing), pp. 141–142, 20–22 Oct 2011

[2] Sullivan, F.: Born to compute. Comput. Sci. Eng. **8**(4), 88 (2006) Augmented Reality for Automatic Identification and Solving Sudoku Puzzles 613

- [3] Moraglio, A., Togelius, J., Lucas, S.: Product geometric crossover for the sudoku puzzle. In: 2006 IEEE Congress on Evolutionary Comp. (CEC 2006), pp. 470–476, Vancouver, BC, Canada, 16–21 Jul 2006
- [4] Simha, P.J., Suraj, K.V., Ahobala, T.: Recognition of numbers and position using image processing techniques for solving sudoku puzzles. In: 2012 International Conference on Advances in Engineering, Science and Management (ICAESM), pp. 1–5, 30–31 Mar 2012
- [5] Maji, A.K., Pal, R.K.: Sudoku solver using minigrid based backtracking. In: 2014 IEEE International Advance Computing Conference, 2014
- [6] Schottlender, M.: The effect of guess choices on the efficiency of a backtracking algorithm in a Sudoku solver. 2014 IEEE Systems, Applications and Technology Conference (LISAT), pp. 1–6, Long Island, 2–2 May 2014. doi:10.1109/LISAT.2014.6845190
- [7] Perez, M., Marwala, T.: Stochastic optimization approaches for solving sudoku. School of Electrical and Information Engineering, University of the Witwatersrand, p. 13, 6 May 2008. doi:10.1016/j.eswa.2012.04.019. <<http://arxiv.org/abs/0805.0697>>
- [8] Nagy, G.: Twenty years of document image analysis is PAMI. IEEE Trans. Pattern Anal. Mach. Intell. **22**(1), 38–62 (2000)
- [9] Hidayatullah, P., Syakrani, N., Suhartini, I., Muhlis, W.: Optical character recognition improvement for license plate recognition in Indonesia. In: 2012 Sixth UKSim/AMSS European Symposium on Computer Modeling and Simulation (EMS), pp. 249–254, 14–16 Nov 2012
- [10] Berchmans, D., Kumar, S.S.: Optical character recognition: an overview and an insight. In: International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), pp. 1361–1365, 10–11 Jul 2014
- [11] Heimo, O.I., Kimppa, K.K., Helle, S., Korkalainen, T., Lehtonen, T.: Augmented reality - towards an ethical fantasy? In: 2014 IEEE International Symposium on Ethics in Science, Technology and Engineering, pp. 1–7, 23–24 May 2014
- [12] MacIntyre, B., Lohse, M., Bolter, J.D., Moreno, E.: Integrating 2D video actors into 2D augmented reality systems. Presence Teleoper. Virtual Environ. **11**, 189–202 (2002)
- [13]<http://docs.opencv.org/modules/core/doc/intro.html>
- [14]<http://code.google.com/p/tesseract-ocr/>
- [15] <http://www.geeksforgeeks.org/backtracking-set-7-sudoku/>