SUDOKU SOLVER

Abstract

Sudoku Image Solver is a Python program that takes an image, scans a Sudoku puzzle, and solves it using image processing. Sudoku image solvers use image processing technology to extract Sudoku puzzles from images, use neural networks to analyse numbers, and use reverse search to solve parsed matches. When images are fed into the solver, the solver pre-processes the input data using both filtering and thresholding. Sudoku puzzles are solved by finding the largest feature/highest contour in the input image. The puzzle is then flattened (distorting the perspective) to get a top-down view of the puzzle. The flat puzzle is extracted and pre-processed. Divide the puzzle into equal pieces by hand. For each cell in the puzzle, the number is averaged (finding the largest cell) and resized to fit the MNIST dataset (28x28). These numbers are fed into an 8-layer convolutional neural network (built using TensorFlow) to find the number. The solution problem is solved by the reverse search algorithm.

I. Introduction

Sudoku is a game that almost everyone plays every day or at least once in a while. Sudoku is a combination of mixed numbers consisting of a 9×9 grid in which each column, row, and 9 3×3 subgrids that make up the grid have numbers from 1 to 9. Appropriate numbers are written in the blanks. There are a few rules to remember when writing these numbers. The rule written in this text is that numbers cannot be located in the same row, column or in the same 3×3 square. Many people enjoy solving Sudoku problems in their spare time, but there are some Sudokus that cannot be solved, so we will look at online Sudoku solvers where we need to enter numbers into cells in the predefined GUI. Therefore, our goal is to use the Sudoku image as a strategy to capture Sudoku's bounding box, capture and solve cells in Sudoku. This process requires some imaging techniques, which we will discuss below. Some Sudoku puzzles are very difficult to solve, so our goal is to make a Sudoku solver that solves puzzles by adding pictures into it. It's easier and less time-consuming than entering all the numbers into an online Sudoku solver. Just scan the image and we will get the result.

II. LITERATURE SURVEY:

New automatic Sudoku image solver: This article explores the use of image processing, OCR machine learning algorithms, and efficient problem solving to calculate the correct answer to solve Sudoku puzzles (mostly in newspapers and mobile games). Image thresholding, corrosion, expansion, etc. are used to convert high-resolution colour camera-generated images of physical Sudoku puzzles into a digitally manipulable format. We use various image processing techniques such as. A 9×9 Puzzle grid is correct for separating the elements of a Sudoku puzzle. Then, using the optical character recognition (OCR)-based k-NN machine learning algorithm, we accurately identify the numbers in the grid and place them in their corresponding positions in the text in the copies of the Sudoku grid, and then finish by working efficiently. Sudoku solving algorithms to calculate solutions. b) Sudoku solver using artificial intelligence: Three high-level steps have been taken in this article: 1. View/capture images instantly. 2. Check/identify if the captured images are the same object or if the query is inconsistent. 3. Use creative strategies to solve complex problems. This article is not only about the use of artificial intelligence, but also about computer vision, thus connecting the two ideas and guiding the future. In this project, the computer tries to analyse the environment by capturing many images broken in time and captures the Sudoku grid from these images. Hough transform technique is used to analyse the grid. Codes are detected using OCR or Optical Character Recognition. Therefore, the system takes all the information of the puzzle and calculates the final solution using artificial intelligence to find the best solution to the Sudoku puzzle problem. Solving Sudoku Puzzles Using Deep Learning, OpenCV and Backtracking: In this article, we learn how to solve Sudoku puzzles using the basics of deep learning, OpenCV and backtracking. This model can be used for tough competition with adequate training. Now, since this game involves numbers, we will take a simple MNIST dataset and build a neural network on it. We will use keras to create this neural network. Visual Sudoku Solver: This report covers the design, implementation and testing of a visual Sudoku solver application for Android, written in Kotlin. The Results app can use the phone's camera to analyse Sudoku puzzles and find their solutions using backtracking algorithms. OpenCV library, which works with multi vision and machine learning, was used to analyse this complex problem. Techniques used include grayscale, thresholding, Gaussian blur, edge detection, and pattern matching. Code recognition using AutoML has been very successful. Options are explained and compared to possible options. Each part of the application is evaluated individually using a variety of methods. A short user survey was also conducted. Finally, the limitations of the application are discussed and future developments in image detection and number recognition are planned to solve Sudoku as a problem of interest: This article solve the problem of dividing the Sudoku

image into an array by displaying numbers. It uses sequences from Sudoku images and constructs them as a satisfaction problem (CSP). In this paper, we introduce a new extraction method for code recognition, which is used together with the CSP algorithm along with our baseline algorithm to provide performance measurement. Experimental results show that the use of CSP technology can reduce solution search time by eliminating inconsistent results in the search space.

III. EXISTING WORK:

The current method we follow here is Kshitij Gupta's paper "A New Automatic Solver for Sudoku Images"; Sagar Khatri; and published "SUDOKU SOLVER IMAGE PROCESSING TECHNIQUES" by Muneeb Hasan Khan, Kamuju Abhi Subrahmanyam, Mandava Sai Vineeth, Tatikonda Lakshman, Sathvik Yadlapalli in 2019. This paper explores the use of image processing, OCR machine learning algorithms, and good problem solving to calculate the correct answer to solve Sudoku puzzles (often found in magazines and mobile games). Image thresholding, erosion, expansion, etc. are used to convert color camera images of physical Sudoku puzzles into a format that can be digitally manipulated. They use various image processing techniques such as. The details of the 9×9 grid are correct. Then use special Optical Character Readers (OCR) based on k-NN machine learning algorithm to accurately identify numbers in the grid and place them in their corresponding positions in copies of the Sudoku grid, and then use Sudoku solving algorithms. It is done to calculate the correct dose.

IV. PROBLEM DESCRIPTION:

In the research article above, it is said that Sudoku requires three pictures to be distinguished from ordinary letters. However, if we take pictures from newspapers, the pictures will not be clean (because most people solve Sudoku printed in newspapers). In other words, we are building a model that can solve Sudoku problems, albeit with some noise.

V. CNN:

CNN is an extension of multilayer perception that can learn filters that need to be calculated by machine learning models, as previously tested in [Yann LeCun et al., 2017]. 1989] using backpropagation. Convolutional networks are often used in image visualization. Since the training process involves learning models from small samples, it is often time consuming and may require the use of GPU. CNNs can handle small images well but cannot handle full connectivity for high-resolution images. Convolutional neural networks have different hidden processes as well as input and output processes. These layers usually consist of a convolutional layer with a learnable filter, a layer for the refinement function, a layer for reduction, and a dropout to penalize false products. We use Keras and TensorFlow as backends to model the CNN and enable comparison of different CNN implementations created by selecting different hyperparameters associated with each layer.

VI. PROPOSED MODEL:

The proposed model can be divided into three parts, such as image enhancement, image segmentation, and solving Sudoku puzzles.

a) Image Enhancement:

Here we will enhance the given image. First, we convert the image to grayscale. We use Gaussian Blue for grayscale images when our goal is to detect edges. The next step is to blur the image, here we use transition because there will be more colors in the image. One newspaper may have different types of edge color while another may have different color, so to eliminate such confusion we use a transformation that will start from the center of the broken image.

b) Image segmentation:

We will use the built-in OpenCV contour algorithm. There are two types of contour algorithms: RETER_Internal and RETER_External. From point 2 above, RETER_External is the best algorithm for our example because it only uses the external counter algorithm, so if there is some internal noise it won't be seen. Now we need to remove the Sudoku box after getting the counter, so we use the approximate PloyDp method with the help of the arc length used to

estimate where the curve will close. Since not all images taken are clear, we created a special feature that changes the image from one plane to another. We use coordinate transformations from algebraic equations to transform elements from one plane to another. Now, first of all, we need to prepare the neural network model after extracting the numbers from the image. So all CNN models use binary images as input to capture the count. Then we need to compress the image because while training the model we compressed the image to 224x224 to extract more features from the model but for this we need to compress the image to 224x224 so that it predicts the correct number. Therefore, we will use DCT (JPEG Discrete Cosine Transform) images and entropy coding for all other image compression. Since our image is a grayscale image, it will be the most lossless compression possible. So, the goal is to create a rectangle and turn it into a 900x900 square, so each image size will turn into a 900x900 image. Since there are 9 rows and 9 rows in Sudoku, the size of each cell is 100x100. Now we draw a clear line after compressing the image.

Datasets:

Public datasets from the UCI repository, MNIST repository, and Kaggle are used to store digital data (including images and their resulting codes) for the purpose of analysing numbers in images. We use the Chars74 dataset because it contains more than 7000 different numbers of printed images of size 128*128 compared to the MNIST dataset. The Sudoku images used for the experiment were selected from publicly available collections in public archives. In this way, we achieved 99.25% test accuracy after 12 trials.

Solving Sudoku:

After getting the numbers, we will store each cell in an array and use this to solve the Sudoku task. We will use the backtracking mechanism to detect boxes and draw lines in Sudoku. We use the reverse search algorithm. This algorithm is a modified version of Depth-First Search (DFS) to check one variable at a time and assign a value to it, and check the limit after each change. It is not just Sudoku; it is a standard algorithm for solving CSP problems. Now after solving the sudoku in the array, the next step is to write the sudoku in the array into the picture. Additionally, there is a function that will take the image and sudoku where the image is the clear image object and the sudoku is just a string and get the image as a 900x900 clear image.

Results:

We tested with about 10 Sudoku data from newspapers with some noise and 9 of the images showed 100% accuracy in predicting the gender of the pattern (reading it in the mind). Our model achieves an accuracy of 97%, which is better than the scientific literature.

Conclusion:

In this work, we proposed a way to recognize Sudoku images, determine the number of Sudoku images, and solve Sudoku images. We have found that a combination of these algorithms is very effective. We can see that CNN provides greater accuracy in number recognition, while models such as KNN rely on rules to obtain comparable results.

Automatic Drug Dispenser and AI Sudoku Solver

ORIGINALITY REPORT

2%	2%	0%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	analyticsindiamag.com Internet Source	2%
2	ijrti.org Internet Source	<1%

Exclude quotes	On	Exclude matches	< 6 words
Exclude bibliography	On		