SUDOKU SOLVER USING IMAGE PROCESSING TECHNIQUES

1Kamuju Abhi Subrahmanyam, 2Mandava Sai Vineeth, 3Tatikonda Lakshman, 4Sathvik Yadlapalli

Abstract—Sudoku Image Solver is a Python program that takes in an image, scans for a Sudoku puzzle, and solves it using image processing methods. Sudoku Image Solver uses image processing techniques to extract Sudoku Puzzle from an image, Convolutional Neural Network to parse digits, and a backtracking search algorithm to solve the parsed puzzle. When an image is being input into the solver, the solver will first preprocess the input image using a bilateral filter and thresholding. The Sudoku puzzle is located by finding the largest feature/ largest contour in the input image. The puzzle is then flattened (warp perspective) to obtain a top-down view of the puzzle. The flattened puzzle is being extracted and preprocessed. The extracted puzzle is separated evenly into pieces for cell extraction. For each cell in the puzzle, the digit (located by finding the largest feature in the cell) is centred and resized (28x28) to conform to the format of the MNIST dataset. The digits are then passed into an 8-layer Convolutional Neural Network (built using TensorFlow) for digit recognition. The parsed puzzle is being solved using the Backtracking search algorithm.

Index Terms—image, CNN, Char74k.

I. INTRODUCTION

The Sudoku game is something almost everyone plays either daily or at least once in a while. Sudoku is a logic-based, combinatorial number-placement puzzle with a 9x9 grid with digits so that each column, each row, and each of the nine 3x3 subgrids that compose the grid contain all of the digits from 1 to 9. The goal is to fill the blank spaces with suitable numbers. These numbers can be filled in keeping in mind some rules. The rule for filling these empty spaces is that the number should not appear in the same row, same column, or the same 3x3 grid. Many are interested in solving Sudoku puzzles in their free time but there are some Sudoku which are unsolvable so we will be looking into the online Sudoku solver in which we have to enter numbers into the cell of predefined GUI present. So we aim to take a Sudoku image as an input to detect the bounding box of the Sudoku detect the cells inside the Sudoku and solve it. This process requires some image processing techniques which we will be discussing below. There are some Sudoku which is extremely hard to solve so we aim to make a Sudoku solver which solves the puzzle by inputting the image into it. This is easier rather than typing all the digits in an online Sudoku solver and takes less time. Just scan the image and we will the result.

II. LITERATURE SURVEY:

a) A Novel Automated Solver for Sudoku Images: This paper attempts to explore the solving of Sudoku puzzles (as commonly found in newspapers and mobile games) using image processing, machine learning algorithms for OCR, and an efficient solving algorithm to compute the correct answer. We use various image processing techniques, such as Image Thresholding, Erosion, Dilation, etc. to convert a high-resolution and colored camera-generated image of the physical Sudoku puzzle into a format that can be digitally operated upon, to isolate the contents of the 9 x 9 puzzle grid correctly. We then use our custom Optical Character Reader (OCR), which is based on the k-NN machine learning algorithm, to correctly identify the digits contained in the grid and place them in the respective positions in a digital copy of the Sudoku grid, which is then processed by an efficient Sudoku solving algorithm, to compute the correct solution.

b) Sudoku solver using Artificial Intelligence: In this paper, on the high-level three steps are performed:

1. See/Capture the image in real-time.
2. Identify/Detect whether the captured image is the same object or the puzzling query.
3. Solving the puzzle by making use of Artificial intelligence-based strategies.
This paper not only involves the use of Artificial Intelligence but also makes use of Computer vision, thus combining the two major concepts and increasing the future scope. In this project, the computer tries to analyze the environment by capturing the multiple image bursts in real-time, and from those images, it would detect the Sudoku grid. For the detection of the grid, the use of the Hough Transform technique has been made. Then the numbers are detected using OCR i.e. Optical Character Recognition. Thus, the system gets the total knowledge of the puzzle and then computes the final solution by making the use of Artificial Intelligence based strategies for getting the optimal solution to the Sudoku puzzle problem.

Solve Sudoku Puzzle Using Deep Learning, OpenCV, And Backtracking: In this paper, we learn how to solve a game of sudoku using simple concepts of deep learning, OpenCV, and backtracking. This model can be used on harder puzzles as well as sufficient training. Now, since this game involves numbers we will make use of the simple MNIST dataset and build a neural network on it. We will use Keras to build this neural network.

Visual Sudoku Solver: In this report, the design, implementation, and testing of a visual Sudoku solver app for android written in Kotlin are discussed. The produced app is capable of recognizing a Sudoku puzzle using the phone’s camera and finding its solution(s) using a backtracking algorithm. To recognize the puzzle, multiple vision and machine learning techniques were employed, using the OpenCV library. Techniques used include grayscaling, adaptive thresholding, Gaussian blur, contour edge detection, and template matching. Digits are recognized using AutoML, giving promising results. The chosen methods are explained and compared to possible alternatives. Each component of the app is then evaluated separately, with a variety of methods. A very brief user evaluation was also conducted. Finally, the limitations of the implemented app are discussed and future improvements are proposed

Image Detection and Digit Recognition to solve Sudoku as a Constraint Satisfaction Problem: This paper addresses the problem of partitioning the Sudoku image into a 1-D array, recognizing digits from the array, and representing it as a Constraint Satisfaction Problem (CSP). In this paper, we introduce new feature extraction techniques for recognizing digits, which are used with our benchmark classifiers in conjunction with the CSP algorithms to provide performance assessment. Experimental results show that the application of CSP techniques can decrease the solution’s search time by eliminating inconsistent values from the search space.

III. EXISTING WORK:
The existing approach that we’ve taken here is a paper named “A Novel Automated Solver for Sudoku Images” by Kshitij Gupta; Sagar Khatri; Muneeb Hasan Khan published in 2019. This paper attempts to explore the solving of Sudoku puzzles (as commonly found in newspapers and mobile games) using image processing, machine learning algorithms for OCR, and an efficient solving algorithm to compute the correct answer. They used various image processing techniques, such as Image Thresholding, Erosion, Dilation, etc. to convert a high-resolution and colored camera-generated image of the physical Sudoku puzzle into a format that can be digitally operated upon, in order to isolate the contents of the 9 x 9 puzzle grid correctly. Then used their custom Optical Character Reader (OCR), which is based on the k-NN machine learning algorithm, in order to correctly identify the digits contained in the grid and place them in the respective positions in a digital copy of the Sudoku grid, which is then processed by an efficient Sudoku solving algorithm, to compute the correct solution.

IV. PROBLEM DESCRIPTION:
In the research paper mentioned above, it is mentioned that the Sudoku on which we are taking pictures needs to be clean in order to detect the image from plain paper. But the images will not be that clean if we take an image from a newspaper (as most people solve Sudoku that is published in the newspaper). So we are making a model which can solve the Sudoku even if there is little noise present.

V. CNN:
CNNs are an extension of multilayer perceptions, which can learn filters that need to be computed by the machine learning models, as experimented earlier in [Yann LeCun et al. 1989] using back-propagation. Convolutional networks are mainly applied to visual imagery. Since the training process involves learning about patterns from smaller patterns, computations usually are time-consuming and may require GPU- based implementation. CNN’s work well with small-size images, but with large-resolution images, weights with full-scale connectivity cannot be processed efficiently.

![Fig.2 CNN Architecture](image-url)
A convolutional neural network consists of distinct hidden layers in addition to the input and output layers. These distinct layers usually consist of the convolutional layer with filters that can be learned, rectified linear unit layer for application of activation function, pooling layer for down-sampling, and loss layer for specification of penalization for incorrect output. We use Keras with TensorFlow as a backend to process the CNN model and provide a comparison between the distinct CNN implementations formed by choosing different hyperparameters associated with each layer.

VI. PROPOSED MODEL:
The proposed model can be divided into three parts namely image enhancement, image segmentation, and solving the Sudoku puzzle.

![Fig.3 Proposed Model]

VII. IMAGE ENHANCEMENT:
Here we will enhance the given image. Firstly we convert the image into a grayscale image. As we aim to detect the edges we apply Gaussian Blur to the grayscale image. Next is thresholding the blurred image, here we’ve used Adaptive Thresholding because there may be many colors on the image. One newspaper may have one type of border color and another newspaper may have another type of color so to remove this type of Confusion we use an adaptive threshold where the threshold value will be the average of the segment of the image.
VIII. IMAGE SEGMENTATION:

We will use the inbuilt OpenCV Contour algorithm. There are 2 types of Contour algorithms is RETER_Internal and RETER_External. From the above 2 RETER_External is the best algorithm for our model as it takes only External Counter algorithms so if there is some Noise Internally it is unnoticed. Now after getting the counters we needed to extract the Sudoku box so we use approx PloyDp method with the Help of Arc length which is used to predict where the curve closes or Not. As the all images taken are not clear we made a feature to transform an image from one plane to another. We use the Transformation of coordinates from linear Algebra where we convert points from one plane to another plane.

Now extracting numbers from the images first after extraction we need to prepare the neural network model. So any CNN model takes binary Images as the input for the detection of numbers. So next we need to compress the image because while training the model we have compressed the image into 224x224 so that more features from our model are extracted but to do that we need to make our image into 224x224 so that it guesses the correct number. So we will be using DCT (Discrete cosine transform for JPEG) images and Entropy Encoding for all other image compressions. As our image is a grey image it will be the most lossless compression.

So the aim is to create a rectangle and convert it into 900x900 square so that any image whatever the size maybe will be converted into a 900x900 image so there are 9 rows and 9 columns in Sudoku so each cell is of size 100x100. Now after compressing the image then we draw the lines to throw some clarity on it.
IX. DATASETS:

Publicly available datasets from the UCI repository, MNIST repository, and Kaggle were used to retrieve the digits dataset, consisting of images and the resultant digit they have, to recognize the digits in the picture. We used The Chars74 dataset as it has over 7000 images of printed characters with dimensions 128*128 for different digits. The dataset proved effective as we were able to obtain greater accuracy than achieved with the MNIST dataset with our features. Sudoku images for the experiments were chosen from the public domain of the commons repository. With this, we have achieved 99.25% test accuracy after 12 epochs.

X. SOLVING THE SUDOKU:

After getting the numbers we’ll store each cell Detected in an array and will be given to solve the Sudoku function. We will be using a standard backtracking Mechanism to detect boxes and underline the Sudoku. We used the Backtracking Search algorithm. This
algorithm is a version of depth-first search (DFS) with the modifications of checking and assigning a value to one variable at a time and of checking constraints after each variable assignment. It is the standard algorithm used to solve any CSP, not just Sudoku. Now after solving Sudoku from the array the next thing is to write the Sudoku from the array to the image. Also, a function that will take the image and Sudoku where the image is an open image object and Sudoku is just an array and will take an image as a 900x900 clean image.

XI. RESULT:
We have tested with around 10 Sudoku papers which are from newspapers that have some noise in which 9 pictures showed 100% accuracy in predicting the model (reading number in the cell). The accuracy of our model is 97% which is better than the research paper we have referred to.

XII. CONCLUSION:
In this work, we have proposed methods for the identification of Sudoku images, recognition of digits from the Sudoku image, and solving the Sudoku puzzle. We have observed that the combined use of these algorithms has highly useful applications. We can see that CNN provides higher accuracy for digit recognition, while models like KNN depend on custom features for achieving comparable results.
REFERENCES