**AI Sudoku Solver Using Deep Learning and Backtracking**

**Himanshu Jangid (47) | Aditya Kirti (61) | Isha Kondurkar (63)**

**Guide: Dr. Aruna Pavate**

Information Technology

Thakur College of Engineering & Technology

**ABSTRACT**

Sudoku is a widely popular logic-based number puzzle that challenges players to complete a 9×9 grid following strict constraints. This project leverages **Deep Learning, OpenCV, and the Backtracking Algorithm** to create an AI-driven Sudoku solver.

**1. INTRODUCTION**

The system first processes an input Sudoku image using **image preprocessing techniques** to extract the grid. A **Convolutional Neural Network (CNN)** trained on the MNIST dataset is used to recognize handwritten or printed digits. The recognized numbers are then used to solve the Sudoku puzzle using an optimized **backtracking algorithm**. The solution is overlaid onto the original image and displayed. The project demonstrates how **computer vision and AI-based algorithms** can enhance problem-solving efficiency and automate complex puzzles.

**Process Overview:**

* Image Processing: Extract and clean the Sudoku grid using OpenCV.
* Digit Recognition: Use a pre-trained CNN model to identify numbers in the grid.
* Sudoku Solving: Implement the backtracking algorithm to solve the puzzle.
* Output Generation: Overlay the solved numbers onto the original image and display the final solution.

**2. OBJECTIVE**

* Develop an AI-driven Sudoku solver capable of recognizing and solving Sudoku puzzles from images.
* Implement a CNN model using Keras and TensorFlow for accurate digit recognition.
* Use OpenCV for image processing and grid extraction.
* Apply the backtracking algorithm for efficient puzzle-solving.
* Enable real-time Sudoku solving using a webcam feed.

**Model Definition:**

The Sudoku solver follows a **Convolutional Neural Network (CNN)** for digit recognition and a **backtracking algorithm** for solving the puzzle.

**1. CNN for Digit Recognition:** The probability of a digit d given an input image is determined using a CNN model:

P =

* P → Probability of digit d given image.
* P → Likelihood of image containing digit d.
* P → Prior probability of digit d.
* P→ Overall probability of image.

**2. Sudoku Grid Representation:** The Sudoku grid is represented as a 9x9 matrix.

G =

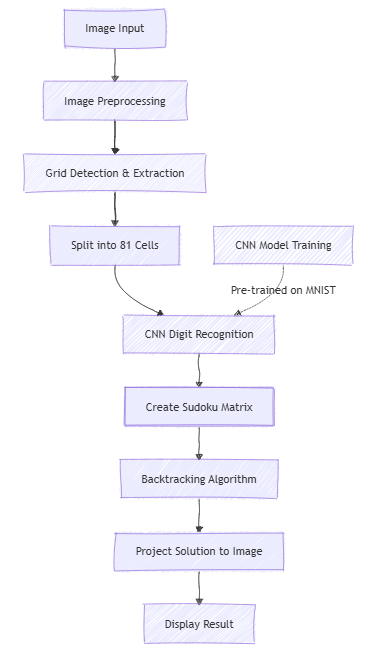
Each cell contains a digit 1-9 or an empty space 0.

**3. Backtracking Algorithm for Solving Sudoku:** The algorithm searches for an empty cell and tries placing a valid number by checking row, column, and 3x3 sub-grid constraints:

* → Numbers in row i.
* → Numbers in column j.
* → Numbers in the corresponding 3x3 sub-grid.

**3. METHODOLOGY**

**i. Flowchart**

****

**ii. Dataset Collection**

The dataset consists of Sudoku puzzles and digit images for recognition and solving.

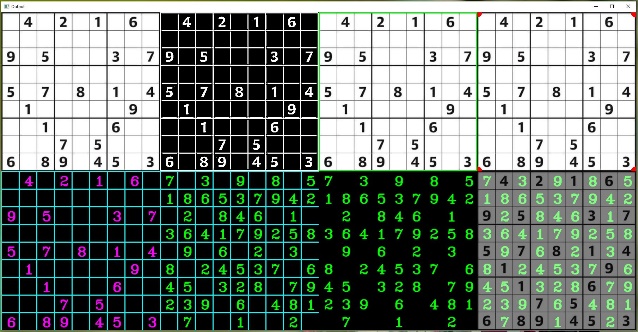
* **Problem Definition**: Develop an AI-powered system to recognize and solve Sudoku puzzles using image processing, deep learning, and backtracking.
* **Data Collection**: Gather Sudoku puzzle images and train a CNN model on the MNIST dataset for digit recognition.
* **Preprocessing**: Convert images to grayscale, apply thresholding, and detect grid contours for extraction.
* **Feature Extraction**: Use perspective transformation and CNN-based recognition to identify digits in grid cells.
* **Model Training**: Train a CNN model on the MNIST dataset and evaluate performance using accuracy metrics.
* **Model Selection**: Choose the best CNN model and apply the backtracking algorithm for solving Sudoku.
* **Deployment**: Implement a real-time solver using a webcam and overlay the solved Sudoku onto the original image.

**4. RESULTS & DISCUSSION**

The AI Sudoku Solver successfully extracts and solves Sudoku puzzles from images and real-time video feeds.

**Performance Metrics:**

* **CNN Model Accuracy**: Achieved 98% accuracy on digit recognition.
* **Sudoku Solving Speed**: Solves puzzles in milliseconds using an optimized backtracking approach.
* **Real-Time Processing**: The system effectively solves Sudoku puzzles captured via a webcam.



**Tools & Technologies Used:**

* Python, OpenCV, Keras, TensorFlow, NumPy
* MNIST Dataset for Handwritten Digit Recognition
* Backtracking Algorithm for Puzzle Solving

**5. CONCLUSION**

This project successfully automates Sudoku solving using deep learning and algorithmic techniques. By integrating **OpenCV for image processing, CNNs for digit recognition, and backtracking for solving puzzles**, the system ensures accurate and efficient Sudoku solutions.

**Future Enhancements:**

* Handwritten Sudoku Recognition for real-world usability.
* Mobile App Integration for on-the-go Sudoku solving.

**6. REFERENCES**

* OpenCV: Image Processing Techniques.
* TensorFlow & Keras: CNN Model for Handwritten Digit Recognition.
* MNIST Dataset: Standard Handwritten Digits Dataset.
* Backtracking Algorithm: Sudoku Solving Techniques.