

Sudoku-Löser 2.0 mit Sudoku-Generator und OCR-Sudoku-Erkennung

Ein Computerprogramm in der Programmiersprache Python 3.6 von **Demian Rothammel** (14 Jahre), Graf-Münster-Gymnasium Bayreuth

Kurzfassung der Arbeit

Ich habe mit Python 3.6 einen **Sudoku-Löser mit Sudoku-Generator und OCR Sudoku-Erkennung** programmiert. Dieses Programm löst z.B. ein extrem schweres Sudoku, das viele Lösungsalternativen hat, in **0,6 Sekunden**. Um **10.000 Lösungen** für dieses Sudoku zu berechnen, benötigt es **3 Minuten und 12 Sekunden**.

Basis hierfür war das einfache Sudoku-Lösungsprogramm, das ich 2017 im Wettbewerb Schüler Experimentieren eingereicht und 2018 wesentlich weiter entwickelt habe.

Die wichtigsten Programm-Merkmale

- Das Programm kann **alle Sudokus lösen**, selbst solche, bei denen nur ein einziger Wert vorgegeben ist.
- Falls **mehrere Lösungen** existieren, werden diese berechnet und ausgegeben, wobei ich die Ausgabe auf maximal 10.000 Lösungsvarianten beschränkt habe, um die Laufzeit einzugrenzen.
- Bei der Dateneingabe ermöglicht das Programm alternativ zur manuellen Eingabe das **Einlesen aus einer Bilddatei** des zu lösenden Sudokus. Hierfür habe ich eine Webcam und Open-CV verwendet.
- Darüber hinaus verfügt das Programm über einen **Sudoku-Generator**, der Sudokus mit verschiedenen, vom Benutzer gewählten Schwierigkeitsgraden erzeugen kann.
- Nachdem ein Sudoku gelöst oder erstellt wurde, werden ein **Richtigkeitscheck** durchgeführt und das Ergebnis sowie die einzelnen Zwischenlösungsschritte, die hierfür jeweils verwendeten Lösungsstrategien und der **Schwierigkeitsgrad** ausgegeben.

Lösungsstrategien und Algorithmen

Die Programmlogik greift auf drei Lösungsstrategien zurück:

1. **Erstellen von Kandidatenlisten**,
2. **Scannen nach Überschneidungen** und
3. **Ausprobieren**. Hierfür habe ich einen **Backtracking-Algorithmus** entwickelt, der eine vielfache Verschachtelung von Ausprobierschritten und damit das Lösen extrem schwieriger Sudokus sowie das Finden mehrerer Lösungen ermöglicht.