Sudoku Reader

Justine Bruns, Dennis Kempf

Gruppe E

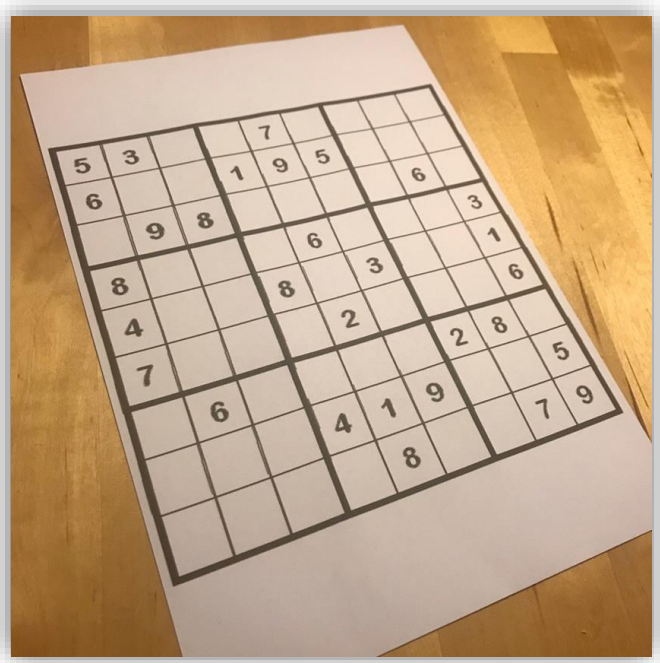
Abschlusspräsentation - 28.01.2020

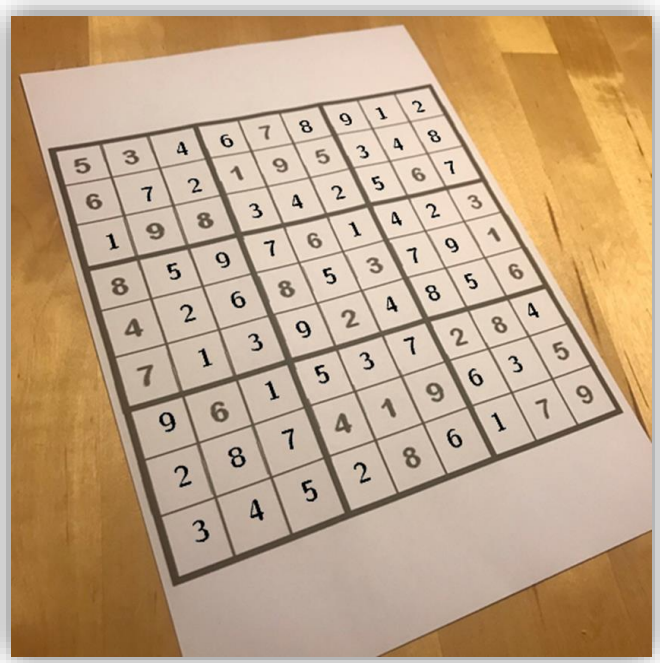
Gliederung

- Problemstellung
- Vorgehen
- Demo
- Fazit
- Ausblick

Problemstellung

Ziel





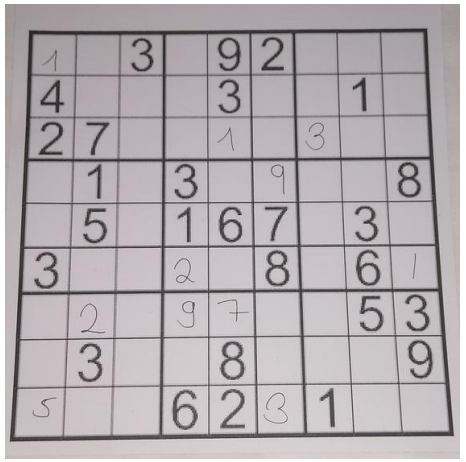
Problemstellung

Szenarien

Problemstellung

Szenarien

- Nutzer benötigt Hilfestellung beim Lösen eines Rätsels
  - Noch keine eigenen Einträge
  - Eigene Einträge vorhanden



Problemstellung

Szenarien

- Nutzer benötigt Hilfestellung beim Lösen eines Rätsels
  - Noch keine eigenen Einträge
  - Eigene Einträge vorhanden
- Verschiedene richtige Lösungen

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

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

Problemstellung

Szenarien

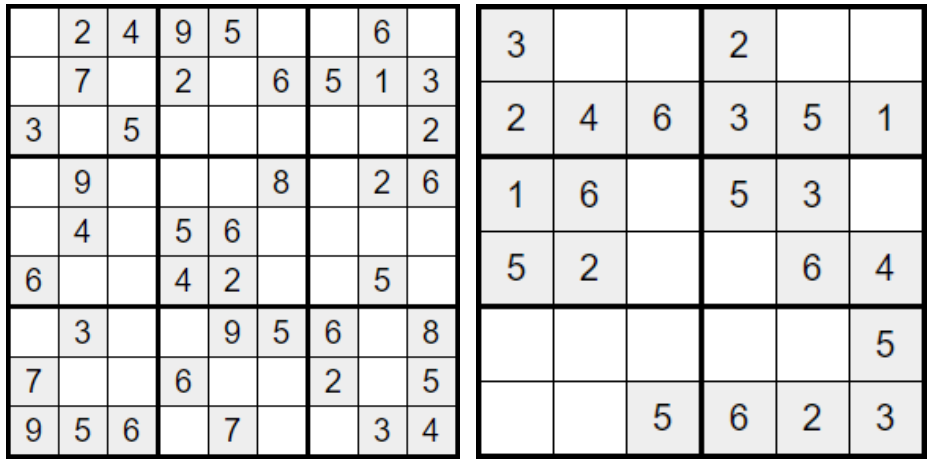
- Nutzer benötigt Hilfestellung beim Lösen eines Rätsels
  - Noch keine eigenen Einträge
  - Eigene Einträge vorhanden
- Verschiedene richtige Lösungen
- Mehrfarbige Rätsel

Problemstellung
Szenarien

- Nutzer benötigt Hilfestellung beim Lösen eines Rätsels
  - Noch keine eigenen Einträge
  - Eigene Einträge vorhanden
- Verschiedene richtige Lösungen
- Mehrfarbige Rätsel
- Unterschiedliche Perspektiven

Problemstellung
Szenarien

- Nutzer benötigt Hilfestellung beim Lösen eines Rätsels
  - Noch keine eigenen Einträge
  - Eigene Einträge vorhanden
- Verschiedene richtige Lösungen
- Mehrfarbige Rätsel
- Unterschiedliche Perspektiven
- Verschiedene Anzahl an Feldern



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

3			2		
2	4	6	3	5	1
1	6		5	3	
5	2			6	4
					5
		5	6	2	3

Problemstellung
Einschränkungen

- Nutzer benötigt Hilfestellung beim Lösen eines Rätsels
  - Noch keine eigenen Einträge
  - Eigene Einträge vorhanden
- Verschiedene richtige Lösungen
- Mehrfarbige Rätsel
- Unterschiedliche Perspektiven
- Verschiedene Anzahl an Feldern

Wir unterscheiden nicht zwischen hangeschriebenen und computergenerierten Ziffern.

Problemstellung
Einschränkungen

- Nutzer benötigt Hilfestellung beim Lösen eines Rätsels
  - Noch keine eigenen Einträge
  - Eigene Einträge vorhanden
- Verschiedene richtige Lösungen
- Mehrfarbige Rätsel
- Unterschiedliche Perspektiven
- Verschiedene Anzahl an Feldern

Wir betrachten nur die erste gefundene Lösung des Sudoku Solvers.

Problemstellung
Einschränkungen

- Nutzer benötigt Hilfestellung beim Lösen eines Rätsels
  - Noch keine eigenen Einträge
  - Eigene Einträge vorhanden
- Verschiedene richtige Lösungen
- Mehrfarbige Rätsel
- Unterschiedliche Perspektiven
- Verschiedene Anzahl an Feldern

Wir nehmen an, dass Sudoku-Rätsel immer schwarz auf weiß (oder zumindest dunkel auf hell) dargestellt werden.

Problemstellung
Einschränkungen

- Nutzer benötigt Hilfestellung beim Lösen eines Rätsels
  - Noch keine eigenen Einträge
  - Eigene Einträge vorhanden
- Verschiedene richtige Lösungen
- Mehrfarbige Rätsel
- Unterschiedliche Perspektiven
- Verschiedene Anzahl an Feldern

Sudoku-Rätsel dürfen nicht kopfüber oder um mehr als 45° gedreht fotografiert werden.

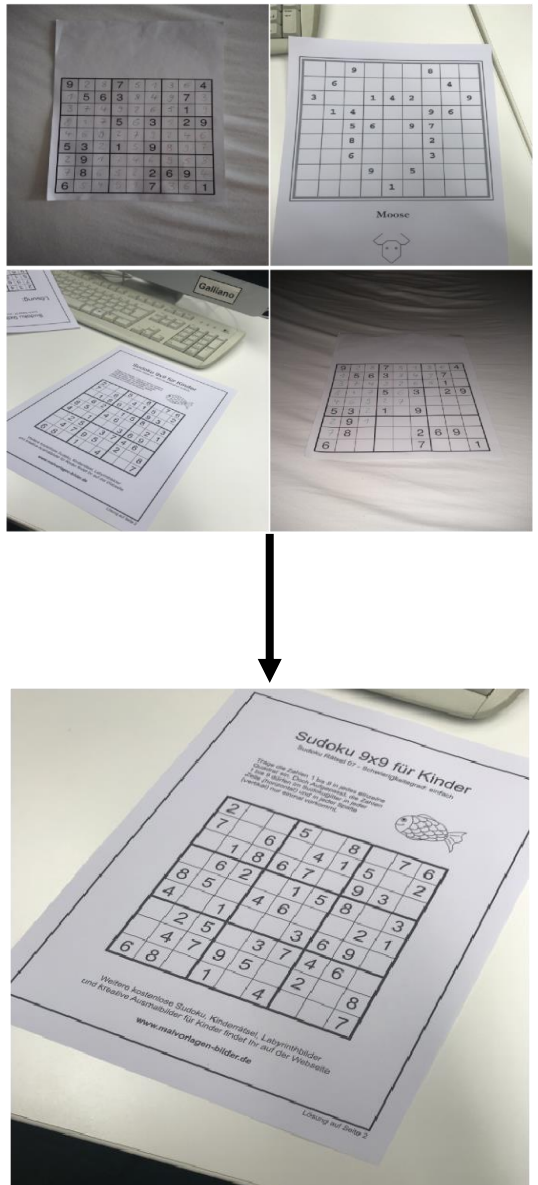
Problemstellung
Einschränkungen

- Nutzer benötigt Hilfestellung beim Lösen eines Rätsels
  - Noch keine eigenen Einträge
  - Eigene Einträge vorhanden
- Verschiedene richtige Lösungen
- Mehrfarbige Rätsel
- Unterschiedliche Perspektiven
- Verschiedene Anzahl an Feldern

Wir nehmen an, dass jedes Sudoku-Rätsel eine Größe von 9x9 besitzt.

Vorgehen

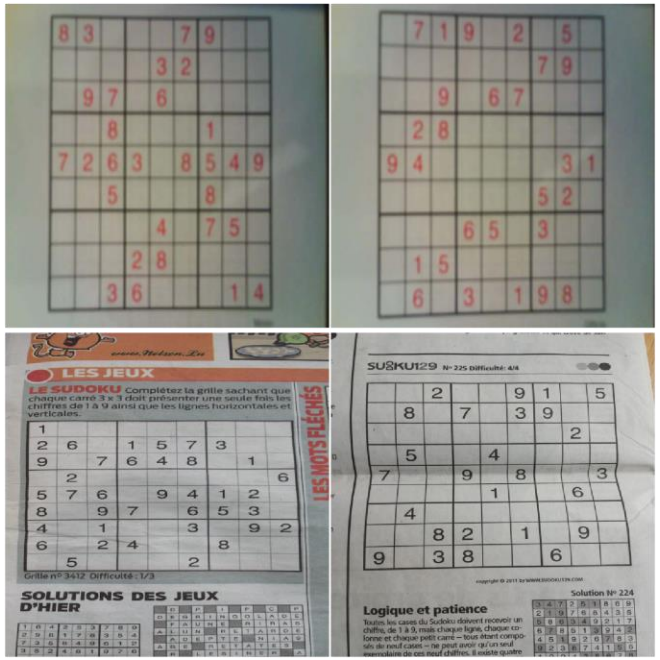
0. Akquisition des Datensatzes
1. Erkennung des Sudoku-Rätsels
2. Korrektur der Kameraperspektive
3. Extraktion der Sudoku-Zellen
4. Erkennung der Ziffern
5. Lösen der Rätsels
6. Darstellung der Lösung



Vorgehen

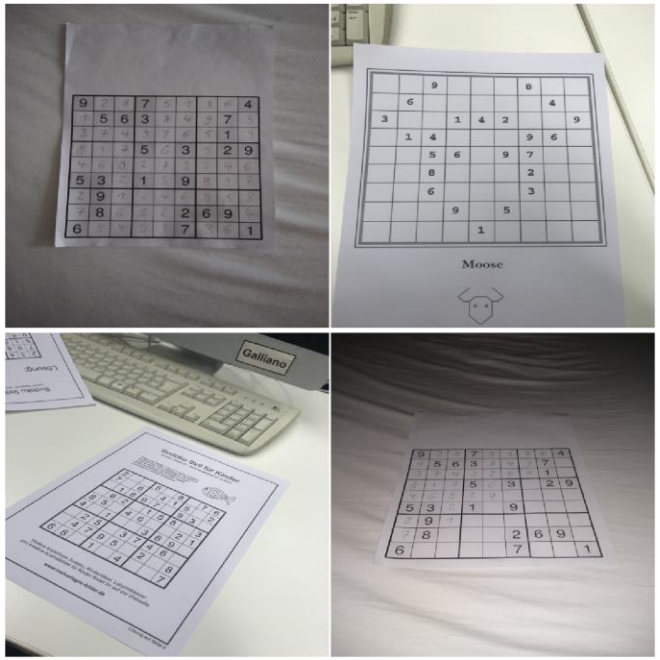
Akquisition des Datensatzes

Datensatz aus GitHub



200 Bilder mit Labels [1]

Eigener Datensatz



292 Bilder mit Labels

Vorgehen

Akquisition des Datensatzes

- Bilddateien
- Abgebildete Ziffern (0 bis 9)
- Art der Ziffern (handgeschrieben oder computergeneriert)

0: Leeres Feld
1-9: Ziffer

digits.txt

```
1 0 3 0 9 2 0 0 0
4 0 0 0 3 0 0 1 0
2 7 0 0 1 0 3 0 0
0 1 0 3 0 9 0 0 8
0 5 0 1 6 7 0 3 0
3 0 0 2 0 8 0 6 1
0 2 0 9 7 0 0 5 3
0 3 0 0 8 0 0 0 9
5 0 0 6 2 3 1 0 0
```

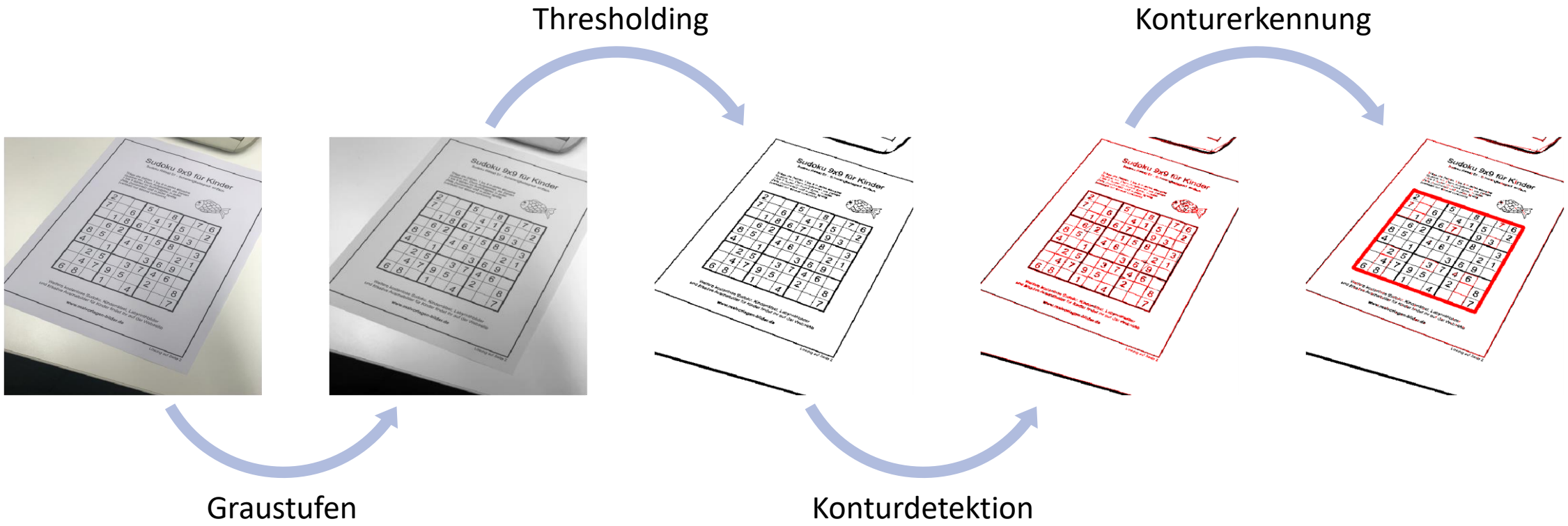
classes.txt

```
H E C E C C E E E
C E E E C E E C E
C C E E H E H E E
E C E C E H E E C
E C E C C C E C E
C E E H E C E C H
E H E H H E E C C
E C E E C E E E C
H E E C C H C E E
```

E: Empty
C: Computer-generated
H: Handwritten

Vorgehen

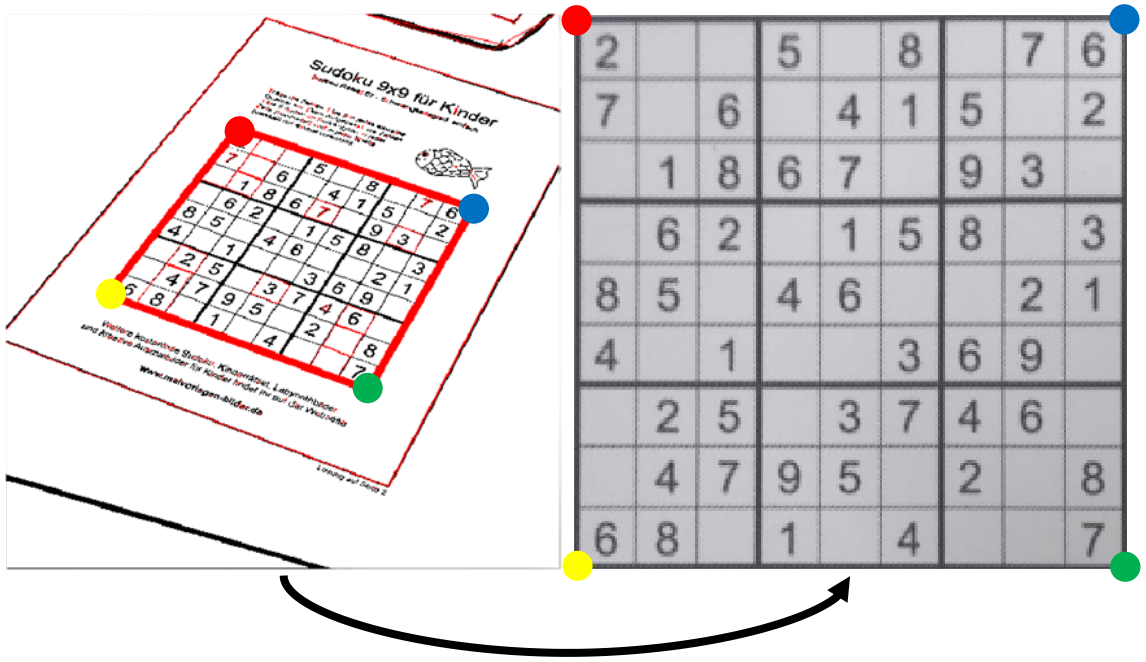
Erkennung des Sudoku-Rätsels



Vorgehen

Korrektur der Kameraperspektive

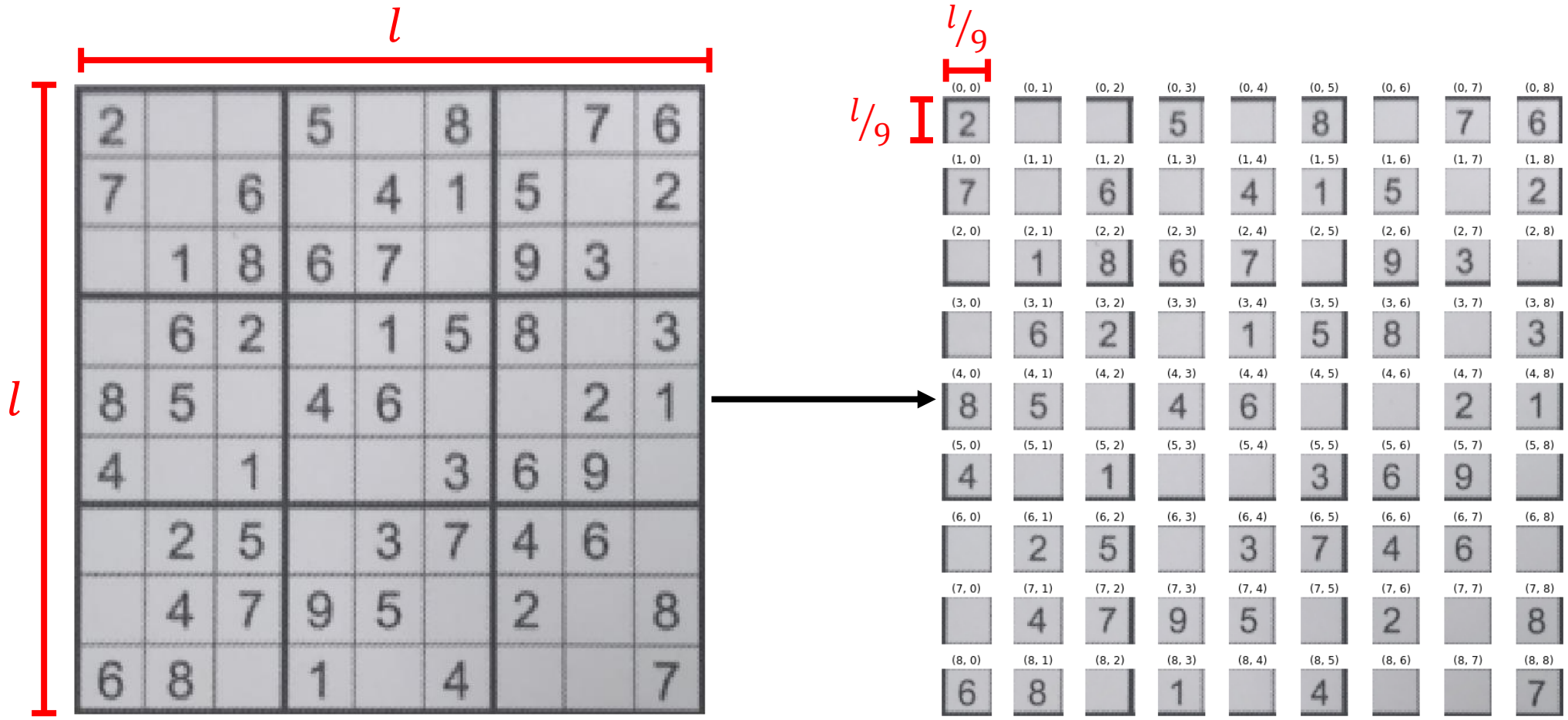
1. Eckpunkte finden
2. Zielpunkte definieren
3. Transformationsmatrix über Eck- und Zielpunkt berechnen
4. Transformationsmatrix auf Quellbild anwenden



Perspektivische Transformation

Vorgehen

Extraktion der Sudoku-Zellen

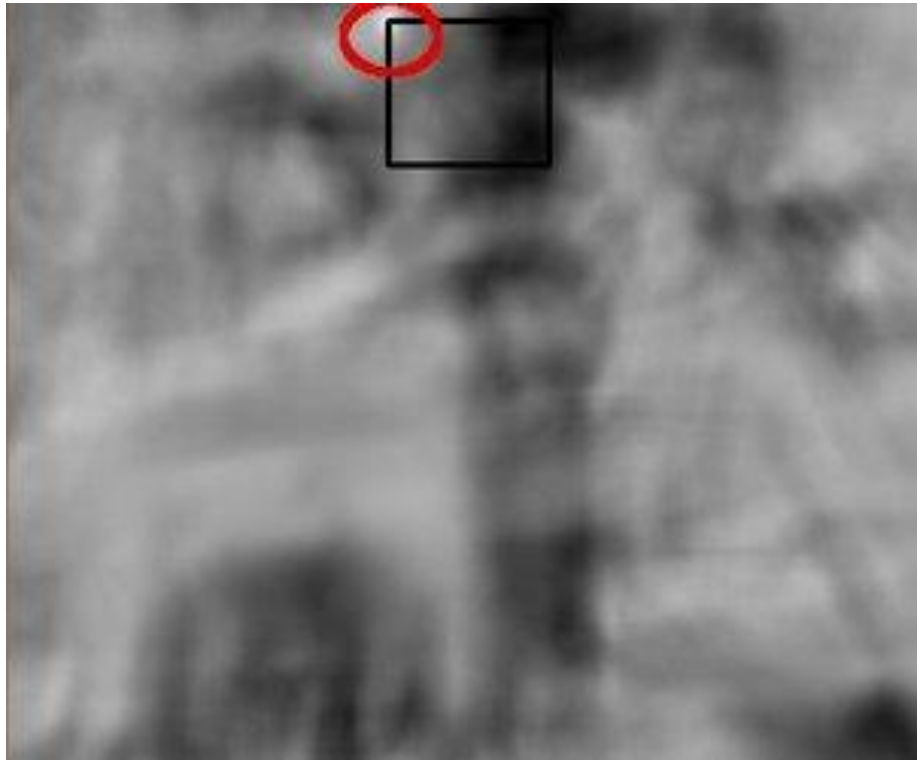


Vorgehen

Erkennung der Ziffern

Erster Ansatz: *Template Matching*

[2]



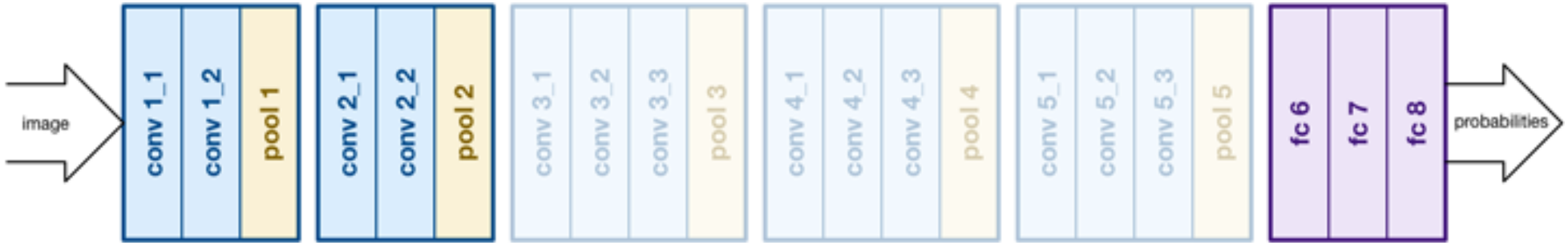
[2]

Vorgehen
Erkennung der Ziffern

- Probleme beim Template Matching:
  - Ergebnis hängt stark von den verwendeten Templates ab
  - Handgeschriebene Ziffern können nur schwer klassifiziert werden
  - Grundsätzlich nicht rotations/skalierungsinvariant

Vorgehen
Erkennung der Ziffern

- Einsatz von Convolutional Neural Networks
- Architektur inspiriert von VGGNet (2014)



[3]

Vorgehen
Erkennung der Ziffern

- Verschiedene Trainingsansätze
  - MNIST
  - MNIST (augmentiert)
  - Unser Sudoku Datensatz
  - Unser Sudoku Datensatz (augmentiert)
  - Transfer Learning
    - Netzwerk auf MNIST vortrainiert
    - Letzte Schicht auf unseren Datensatz optimiert
  - Ensemble Learning
    - Durchschnitt von 10 relativ „schwachen“ Netzwerken

Vorgehen
Erkennung der Ziffern

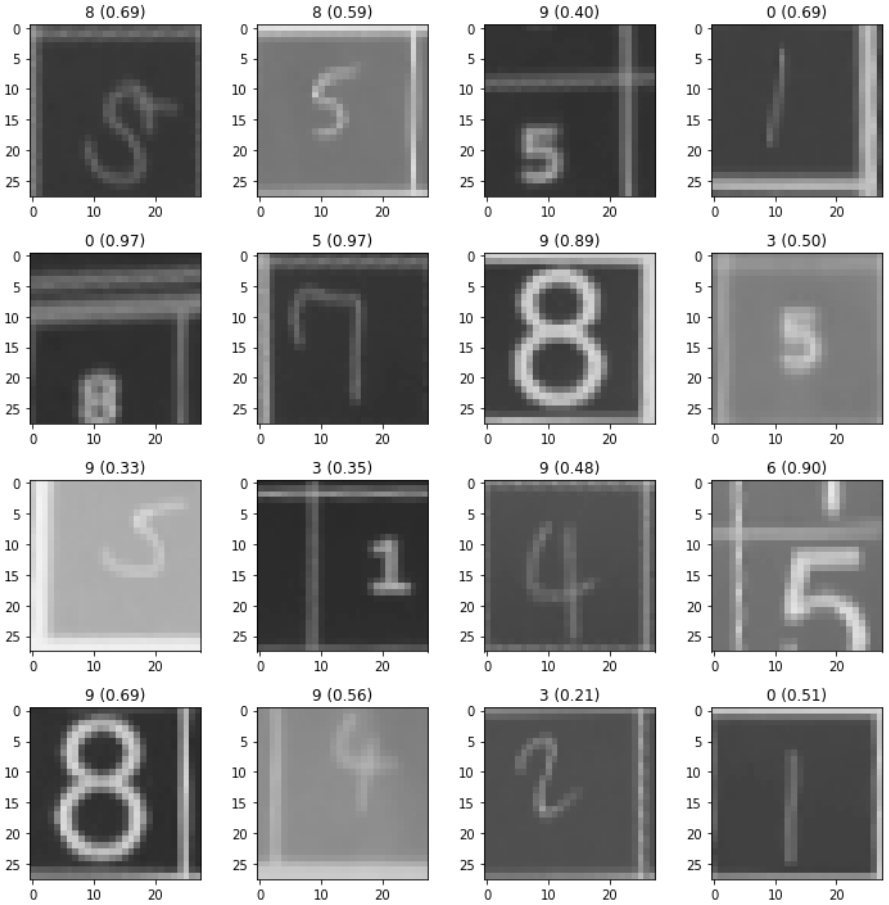
- Verschiedene Trainingsansätze
  - MNIST — 12% Genauigkeit
  - MNIST (augmentiert) — 71% Genauigkeit
  - Unser Sudoku Datensatz — 94% Genauigkeit
  - Unser Sudoku Datensatz (augmentiert) — 95% Genauigkeit
  - Transfer Learning — 91% Genauigkeit
    - Netzwerk auf MNIST vortrainiert
    - Letzte Schicht auf unseren Datensatz optimiert
  - Ensemble Learning — 92% Genauigkeit
    - Durchschnitt von 10 relativ „schwachen“ Netzwerken

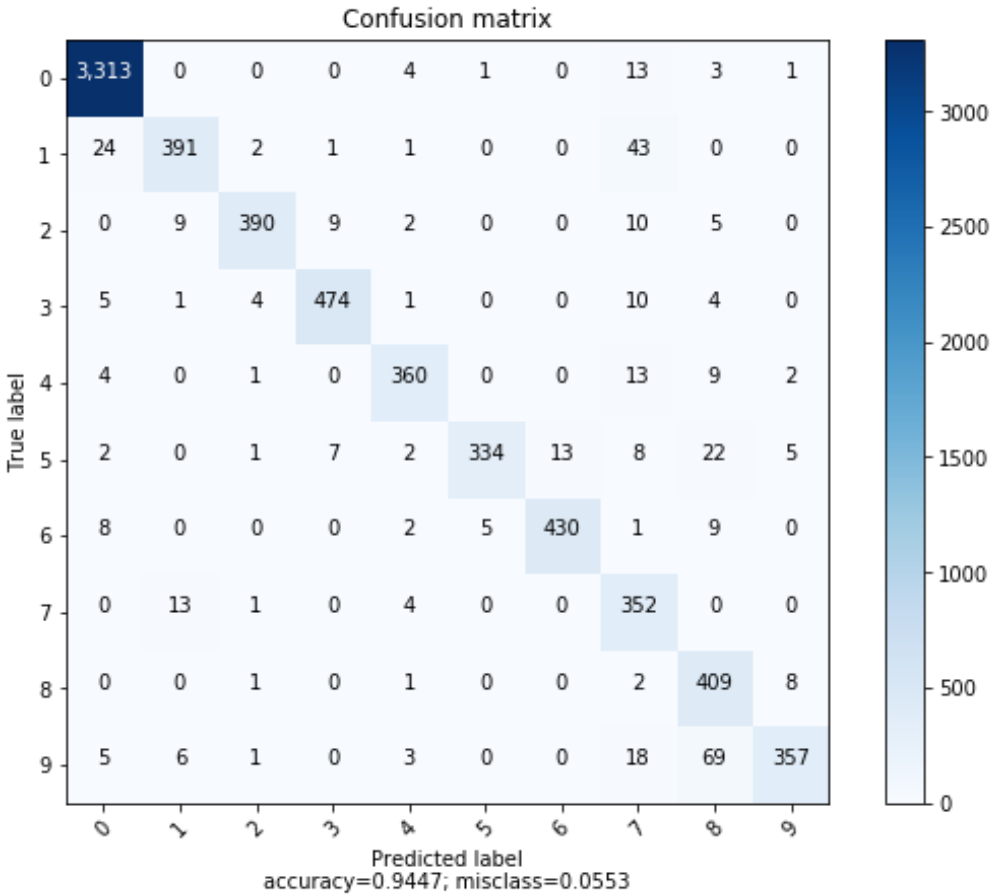
Vorgehen
Erkennung der Ziffern

- Verschiedene Trainingsansätze
  - MNIST — 12% Genauigkeit
  - MNIST (augmentiert) — 71% Genauigkeit
  - Unser Sudoku Datensatz — 94% Genauigkeit
  - Unser Sudoku Datensatz (augmentiert) — 95% Genauigkeit
  - Transfer Learning — 91% Genauigkeit
    - Netzwerk auf MNIST vortrainiert
    - Letzte Schicht auf unseren Datensatz optimiert
  - Ensemble Learning — 92% Genauigkeit
    - Durchschnitt von 10 relativ „schwachen“ Netzwerken

Vorgehen

Erkennung der Ziffern

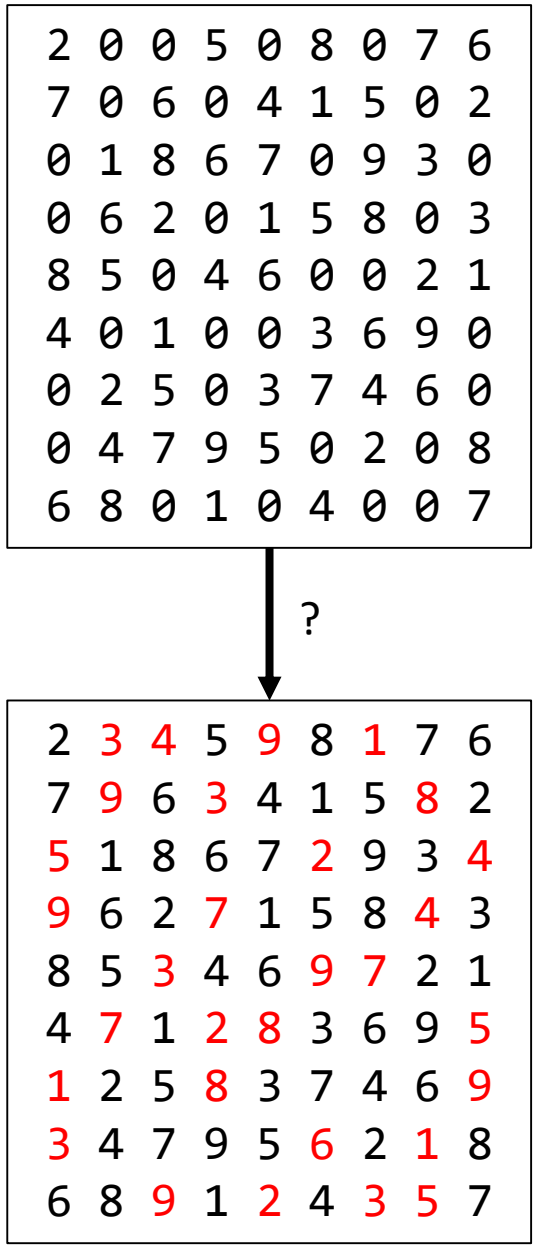




Vorgehen
Lösen des Rätsels

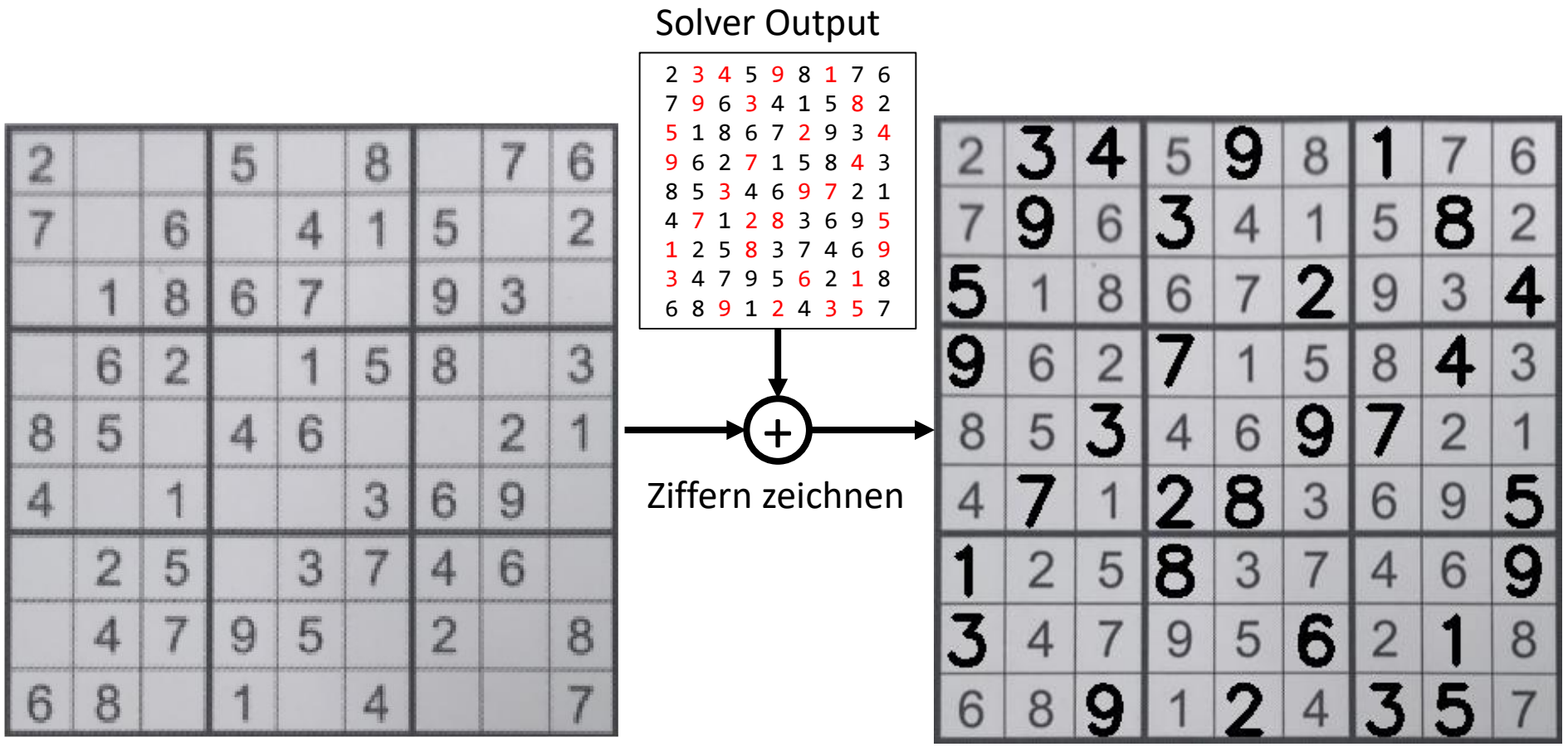
- Erster Ansatz: Bruteforce mit Backtracking
  - Versucht alle möglichen Zahlenkombinationen
  - Problem: Sehr zeitaufwändig (nicht echtzeitfähig)

- Zweiter Ansatz: Erweiterter Algorithmus
  - https://github.com/jjallan/sudoku
  - Formuliert Sudoku Rätsel als *Exact Cover Problem*
  - Basiert auf Donald Knuths Algorithm X
  - Deutlich schneller



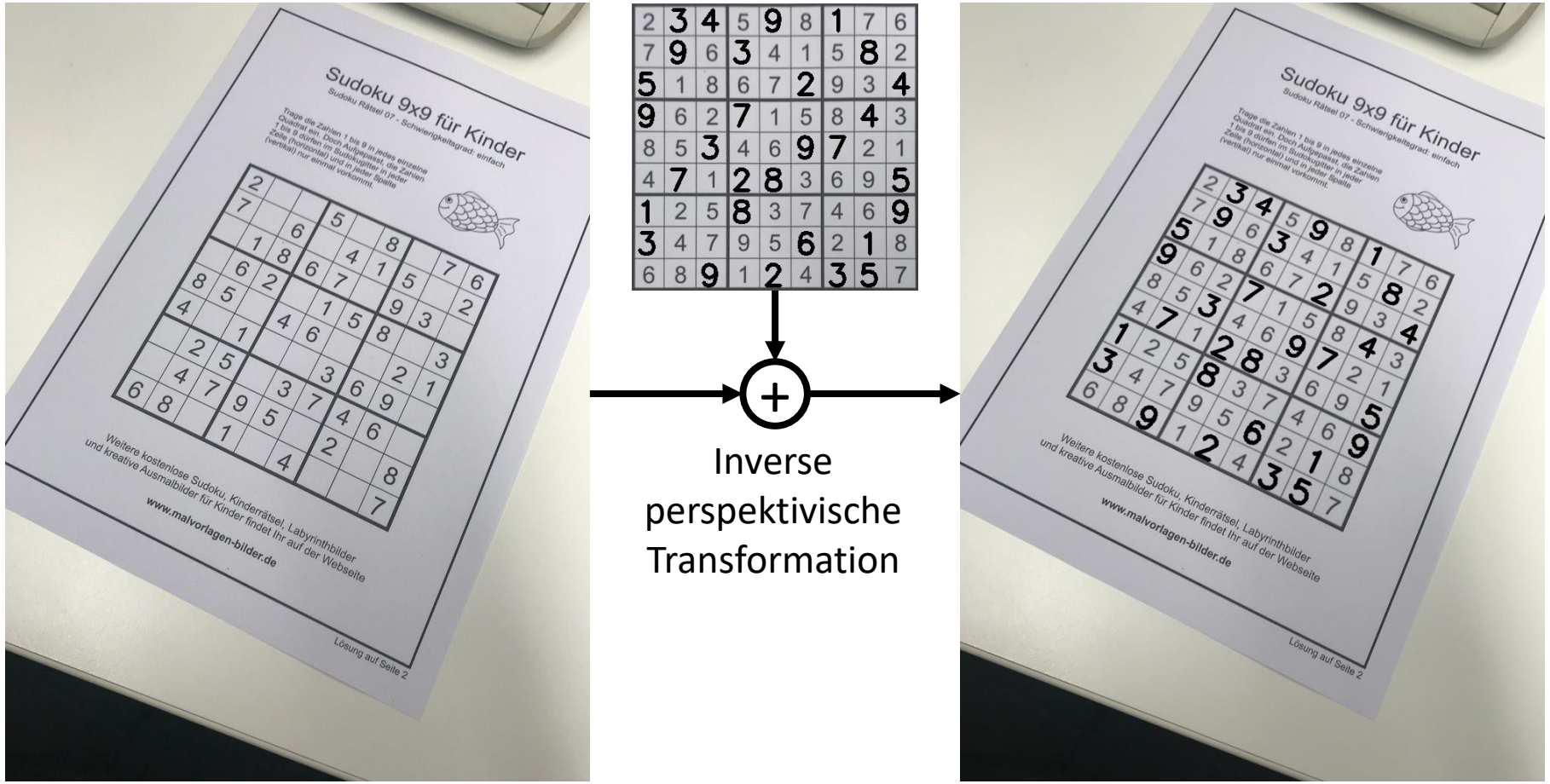
Vorgehen

Darstellung der Lösung

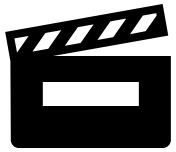


Vorgehen

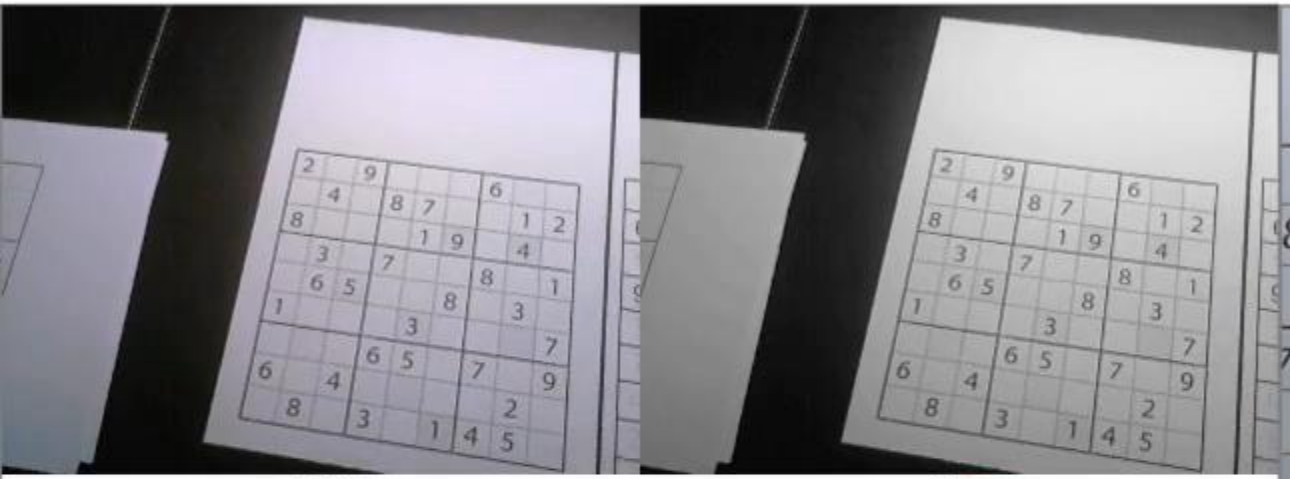
Darstellung der Lösung

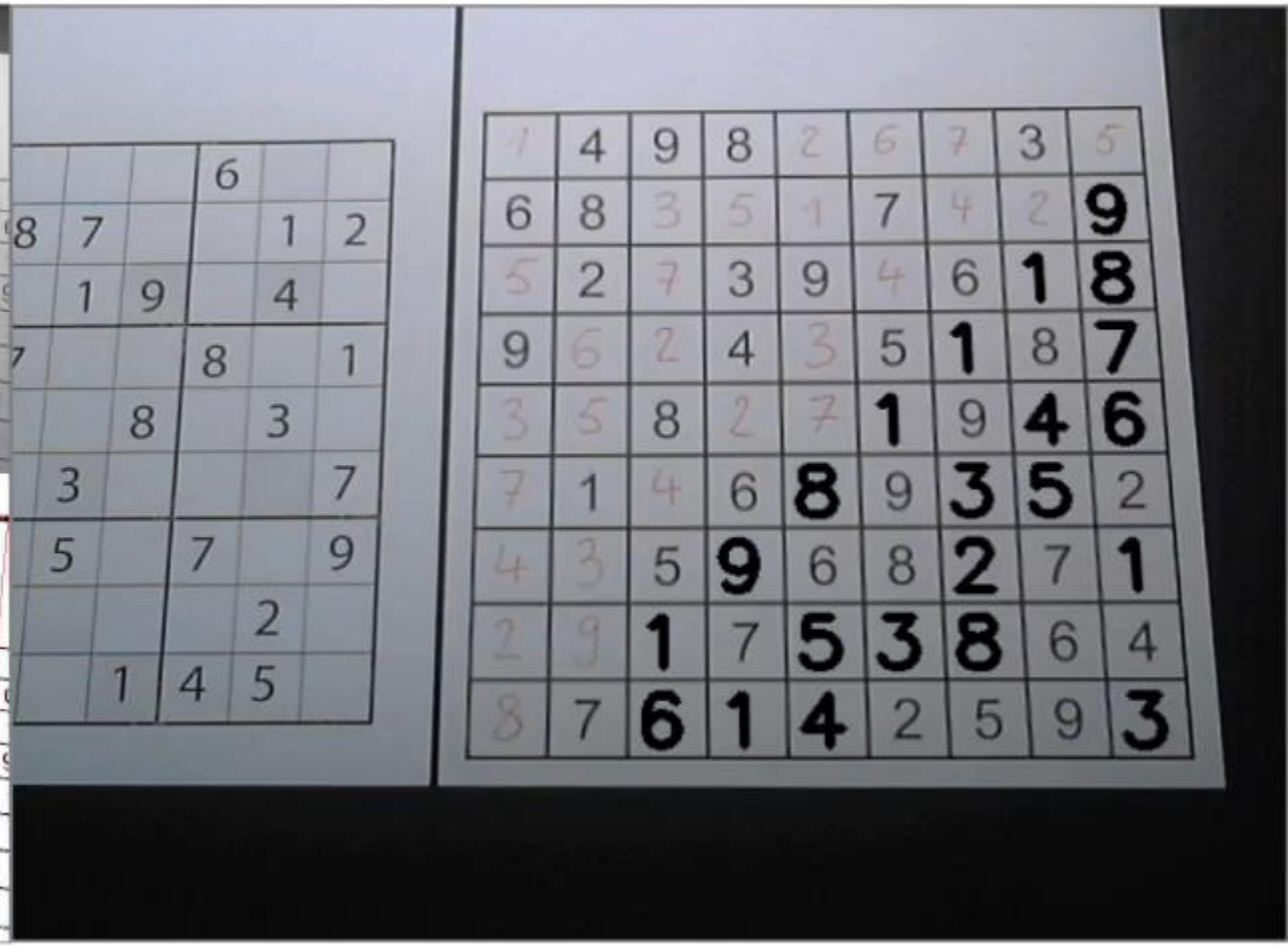


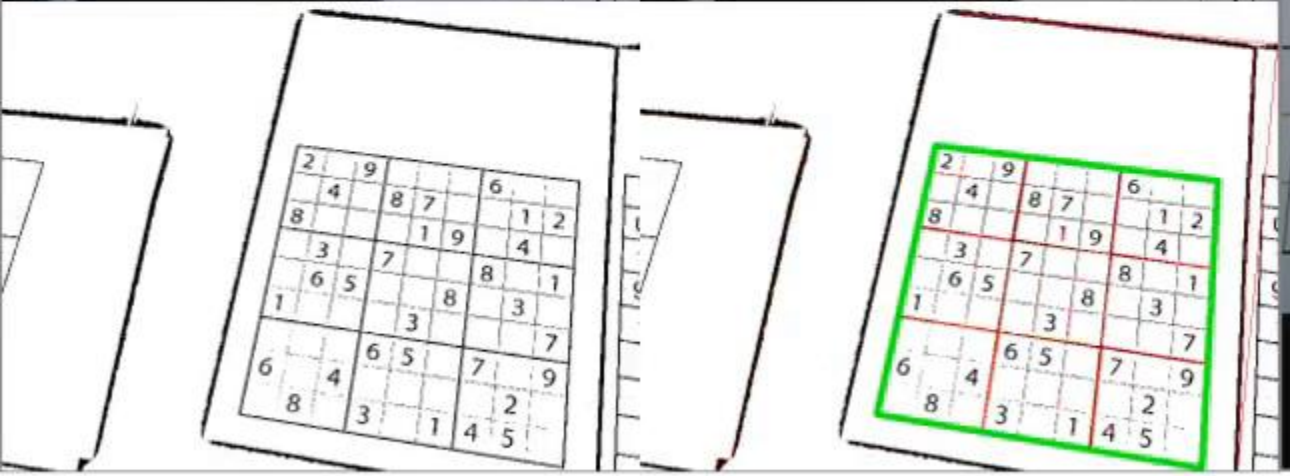
Demo



Input **Grayscale** **AR Overlay**







Contours **Detection**

Fazit

- Viele Schritte mit klassischen Methoden sehr gut realisierbar
- Einsatz von Template Matching für die zuverlässige Erkennung von handgeschriebenen Ziffern sehr aufwändig
- Auch kleine CNNs können gute Ergebnisse in der Klassifikation von handgeschriebenen und computergenerierten Zahlen liefern

Ausblick

- Zwischen handgeschriebenen und computergenerierten Zahlen unterscheiden
  - $\rightarrow$ Prüfen ob eigene *handgeschriebene* Zahlen richtig sind
  - $\rightarrow$ Eigene Lösung ggf. vom Solver korrigieren lassen
- Erkennung ist robust gegenüber farbigen Rätseln
- Auch gedrehte Rätsel werden korrekt erkannt und gelöst
- Automatische Erkennung der Felderanzahl (z. B. 7x7 oder 8x8)

Danke für eure Aufmerksamkeit

Quellen

[1] https://github.com/wichtounet/sudoku_dataset (letzter Aufruf: 19.01.2020)

[2] https://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template_matching/template_matching.html (letzter Aufruf: 21.01.2020)

[3] https://machinethink.net/blog/convolutional-neural-networks-on-the-iphone-with-vggnet/ (letzter Aufruf: 21.01.2020)