



Sudoku checker



Piotr Gdowski
Michał Martusewicz



The purpose of the project: checking if sudoku is correctly solved

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



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

Hand-developed sudoku dataset

Handwritten numbers above the grid: 361, 56, 36.

Arrow labeled **B** points to the top row.

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

Handwritten checkmark below the grid.

Handwritten numbers above the grid: 30, 20, 20, 18.

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

Handwritten numbers below the grid: 25, 9, 14, 20, 27, 11, 7.

Handwritten number above the grid: 123.

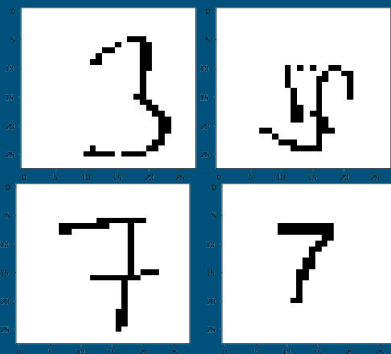
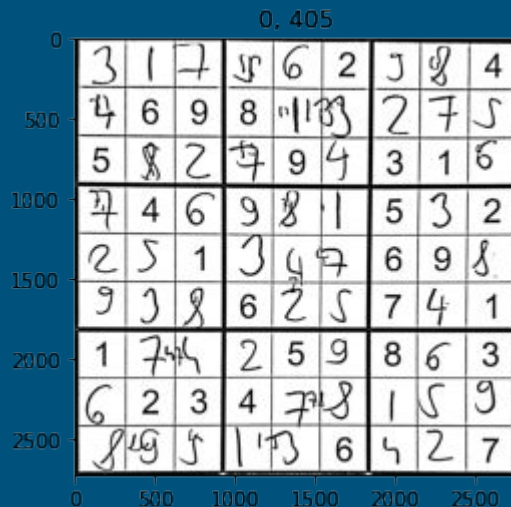
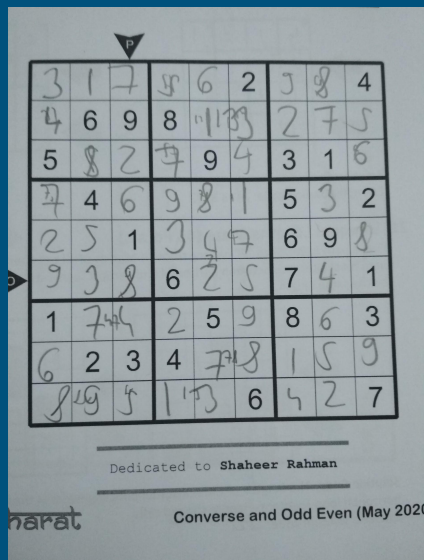
Arrow labeled **B** points to the top row.

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

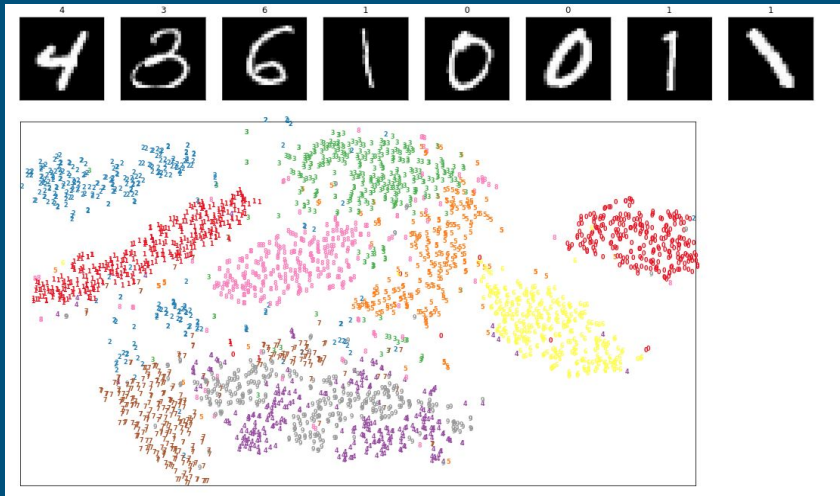
Pipeline

Preprocessing

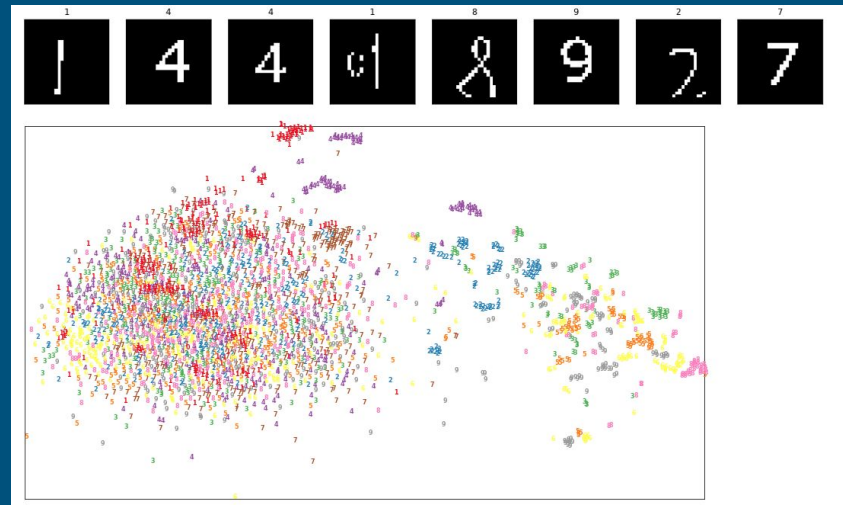
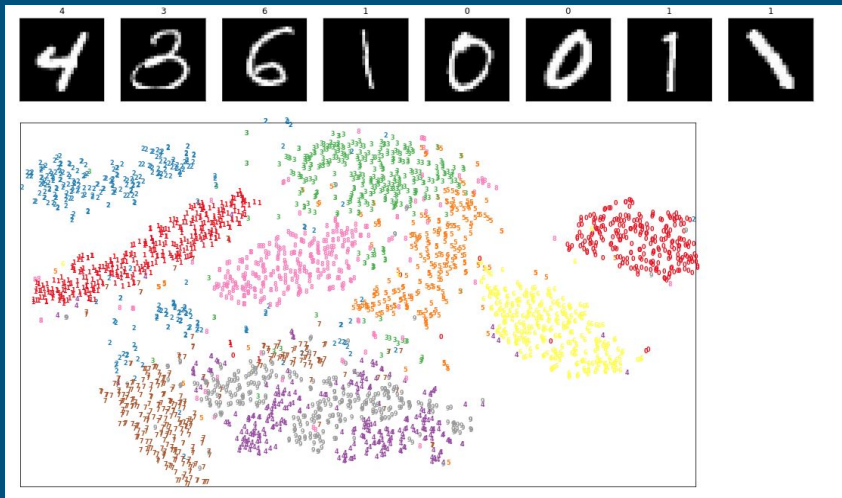
1. Deskewing
2. Thresholding
3. Dividing into 81 digits
4. Removing border



MNIST dataset (T-SNE decomposition)



Sudoku dataset (T-SNE decomposition)



Digits recognition

Baseline recognition - KNN

- KNN trained on MNIST, tested on sudoku_test - 25% recognised digits
- KNN trained on sudoku_train, tested on sudoku_test - 74%

Neural net

- Basic net: 74%

We tested many different nets and parameters and got:

- Final net: 92%

Final net

1. 4 layers of conv2d and relu after each
2. maxPool2d
3. dropout (25%)
4. Linear1, relu
5. dropout (50%)
6. Linear2
7. log softmax

Results

Sudoku checker accuracy

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

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

Sudoku checker accuracy

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

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

Sudoku checker accuracy

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

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

What have we learnt

- Colab sucks
- MNIST didn't help
- Graphic card has limited memory

Further work

- collect more labeled data -> different handwriting styles
- improve data preprocessing
- better net architecture
- release as mobile app
- different approach?
 - Train neural net with the whole sudoku diagram