Решение судоку при помощи компьютерного зрения

Бызова Анна, Дымова Диана

Описание

Проект реализует автоматическое распознавание чисел на игровом поле судоку с помощью компьютерного зрения, а затем применяет алгоритм решения судоку для поиска ответа. Это позволяет не только решать стандартные задачи, но и изучать теорию чисел и алгоритмы.

			За	да	ча					
8	4	5	1	2		7	3	9		
7		9	5		4		6	2		
3		2	9	8	7		5	4		
	3	6			5	2	8			
4	8	1	6	9	2		7	3		
	5	7	3	1	8	4	9	6		
1	7		2	6		3	4	5		
5	9	3	7	4	1	6		8		
6		4	8	5		9	1			

			Peı	ше⊦	ие			
8	4	5	1	2	6	7	3	9
7	1	9	5	3	4	8	6	2
3	6	2	9	8	7	1	5	4
9	3	6	4	7	5	2	8	1
4	8	1	6	9	2	5	7	3
2	5	7	3	1	8	4	9	6
1	7	8	2	6	9	3	4	5
5	9	3	7	4	1	6	2	8
6	2	4	8	5	3	9	1	7

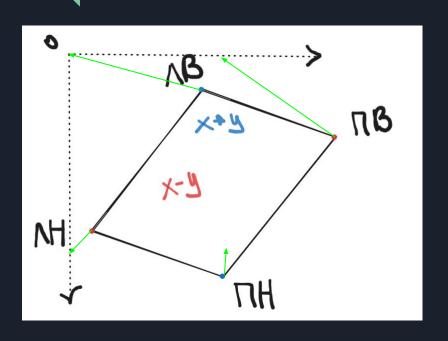
		За	да	ча								Рег	шен	ние			
	8		6	5			1		4	3	8	9	6	5	2	7	1
6	9			2	4	8	3		5	6	9	7	1	2	4	8	3
	1			8	6		5		7	2	1	3	4	8	6	9	5
8		2	7	4		Г	6		9	8	3	2	7	4	1	5	6
4			3	9	8		7		1	4	6	5	3	9	8	2	7
7	5			1		3	4		2	7	5	6	8	1	9	3	4
9	2		5	3	7	Г	8		6	9	2	1	5	3	7	4	8
1		8	9	7	5		2		3	1	4	8	9	7	5	6	2
		4	2		3	1	9		8	5	7	4	2	6	3	1	9
	8 4 7	6 9 1 8 4 7 5	8 9 1 8 2 4 7 5 9 2 1 8	8 6 9 1 8 2 7 4 3 7 5 1 9 2 5 1 8 9	6 9 2 1 8 8 2 7 4 4 3 9 7 5 1 9 2 5 3 1 8 9 7	8 6 5 6 9 2 4 1 8 6 8 2 7 4 4 3 9 8 7 5 1 9 9 2 5 3 7 1 8 9 7 5	8 6 5 4 8 6 9 2 4 8 1 8 6 6 8 2 7 4 8 4 3 9 8 7 5 1 3 9 2 5 3 7 1 8 9 7 5	8 6 5 1 6 9 2 4 8 3 1 8 6 5 8 2 7 4 6 4 3 9 8 7 7 5 1 3 4 9 2 5 3 7 8 1 8 9 7 5 2	8 6 5 1 6 9 2 4 8 3 1 8 6 5 8 2 7 4 6 4 3 9 8 7 7 5 1 3 4 9 2 5 3 7 8 1 8 9 7 5 2	8 6 5 1 4 6 9 2 4 8 3 1 8 6 5 7 8 2 7 4 6 4 3 9 8 7 7 5 1 3 4 9 2 5 3 7 8 1 8 9 7 5 2	8 6 5 1 4 3 6 9 2 4 8 3 5 6 1 8 6 5 7 2 8 2 7 4 6 9 8 4 3 9 8 7 1 4 7 5 1 3 4 2 7 9 2 5 3 7 8 6 9 1 8 9 7 5 2 3 1	8 6 5 1 1 4 3 8 6 9 2 4 8 3 5 6 9 7 1 8 6 5 6 9 7 2 1 8 2 7 4 6 6 9 8 3 1 4 6 9 8 7 7 5 9 8 3 7 5 1 3 4 2 7 5 9 2 7 5 9 2 1 8 9 7 5 2 3 1 4	8 6 5 1 1 4 3 8 9 6 9 2 4 8 3 5 6 9 7 1 8 6 5 9 8 3 2 4 3 9 8 7 1 4 6 5 4 3 9 8 7 2 1 4 6 5 7 5 1 3 4 2 7 5 6 9 2 5 3 7 8 6 9 2 1 1 8 9 7 5 2 3 1 4 8	8 6 5 8 9 6 9 7 1 1 8 6 5 6 9 7 1 8 2 7 4 6 9 8 3 2 7 4 3 9 8 7 1 4 6 5 3 7 5 1 3 4 2 7 5 6 8 9 2 5 3 7 8 6 9 2 1 5 1 8 9 7 5 2 3 1 4 8 9	8 6 5 7 1 6 9 2 4 8 3 1 8 6 5 9 7 1 2 8 2 7 4 6 9 8 3 2 7 4 4 3 9 8 7 1 4 6 5 3 9 7 5 1 3 4 2 7 6 8 1 9 2 5 3 7 8 6 9 2 1 5 3 1 8 9 7 5 2 3 1 4 8 9 7	8 6 5 7 1 2 4 8 3 5 6 9 7 1 2 4 8 6 5 6 9 7 1 2 4 8 6 5 6 9 7 1 2 4 8 6 8 2 7 4 8 6 9 8 3 2 7 4 1 4 3 9 8 7 7 2 1 3 4 8 6 9 8 3 2 7 4 1 1 4 6 5 3 9 8 8 2 7 5 6 8 1 9 8 9 7 5 2 1 5 3 7 3 1 4 8 9 7 5 5	8 6 5 7 1 2 4 8 3 1 8 6 5 6 9 7 1 2 4 8 6 9 8 2 7 4 6 9 8 3 2 7 4 1 5 4 3 9 8 7 1 2 4 8 6 9 8 2 7 4 1 5 3 9 8 2 7 5 1 3 4 6 5 3 9 8 2 7 5 1 3 4 6 6 9 2 7 4 1 5 8 2 7 5 6 8 1 9 3 9 2 5 3 7 8 6 9 2 1 5 3 7 4 1 8 9 7 5 6 9 2 1 5 3 7 4

			38	ада	ча							Pe	шеі	ние			
	1		-	6	18	7	5		2	1	7	3	6	4	8	5	9
		6	1	8	-	2		3	5	9	6	7	8	1	2	4	3
	The same	4		5			1	8	8	3	4	2	5	9	7	1	6
	7			7		3			1	4	5	9	7	6	3	8	2
	B	3		2	8	4	, 33	5	7	6	3	1	2	8	4	9	5
9		2	4						9	8	2	4	3	5	6	7	1
3	7		8	9		5	6	3	3	7	1	8	9	2	5	6	4
		8				.00	8	7	4	5	8	6	1	3	9	2	7
		9	5			B	2		6	2	9	5	4	7	1	3	8

Процесс решения мы разбили на несколько этапов:

- Определение границ судоку и выравнивание поля (OpenCV)
- Pаспознание исходных данных (EasyOCR)
- Решение задачи (PuLP)
- Отображение ответа на исходном изображении

Этап 1. Ориентирование изображения

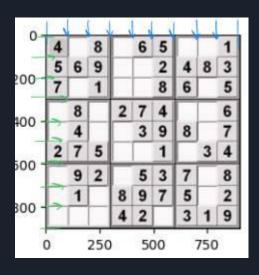


Определим вершины квадрата судоку.

верхний левый угол - наименьшая сумма координат нижний правый - наибольшая сумма координат

верхний правый угол - наибольшая разность нижний левый - наименьшая разность

Этап 2. Подготовка к распознаванию



Разделяем изображение на 9 частей по строкам и на 9 по столбцам, чтобы в дальнейшем распознавать цифры строго внутри ячеек
Для этого используем функцию np.split

```
# Разделяем наш судоку на 9 строк и 9 столбцов и распознаём каждое значение split = np.split(board, 9, axis=1) for col, j in enumerate(split): # digs = np.split(j, 9) # тут разделяем горизонтальные блоки на 9 ячеек
```

Этап 3. Распознавание

Для распознавания чисел используем модель OCR

```
# Загружаем модель OCR (оптическое распознавание символов)
import easyocr
reader = easyocr.Reader(['en']) - создаем функцию чтения изображения
```

С помощью функции распознаем изображение внутри каждой ячейке (циклом) и отображаем его в датафрейм

```
# Распознаём число в ячейке и записываем его в датафрейм и список с координатами text = reader.readtext(d, allowlist='0123456789', detail=0)
# распознавание цифр из списка 0123456789 без дополнительной информации if len(text) > 0: # проверяем был ли распознан текст

df.iloc[row, col] = text[0] # если да - записываем его в датафрейм sudoku_map.append([text[0], str(row+1), str(col+1)])
```

Этап 4. Решение судоку

Решаем задачу с помощью перебора
По правилам решения судоку
1/ Каждая цифра (от 1 до 9) в строке должна быть представлена 1 раз
2/ Каждая цифра в столбце должна быть представлена 1 раз
3/ Каждая цифра в квадрате 3 на 3 должна быть представлена один раз

Используем эти ограничения для определения подходящего варианта

```
prob = LpProblem("Судоку", LpMaximize)

# Создаём Pulp словарь с переменными возможных ответов
choices = LpVariable.dicts("Choice", (vals, rows, cols), 0, 1,
LpInteger)
```

Этап 5. Отрисовываем конечный результат

Пишем решение поверх изображения внутри ячеек С помощью функции imshow()

			38	ада	ча			
	1			6	0	7	5	
		6	- 50	E	-	2		3
		4		5			1	-
	1			7				S
	B	3		2	8	4		5
9	K	2	4		193			
3	7		8	9		5	6	3
		8						7
		9	5			8	(3)	

			Pe	шен	ние			
2	1	7	3	6	4	8	5	9
5	9	6	7	8	1	2	4	3
8	3	4	2	5	9	7	1	6
1	4	5	9	7	6	3	8	2
7	6	3	1	2	8	4	9	5
9	8	2	4	3	5	6	7	1
3	7	1	8	9	2	5	6	4
4	5	8	6	1	3	9	2	7
6	2	9	5	4	7	1	3	8