

Решатель sudoku по изображению с камеры

А. А. Муравцев¹

¹Высшая школа теоретической механики
Санкт-Петербургский Политехнический университет Петра Великого

1 мая 2022 г.

Структура проекта

Обработка изображения и распознавание

- Общая обработка полученного изображения

- Определение положения поля sudoku на обработанном изображении

- Разделение на ячейки

Решатель sudoku

Графическое приложение на PyQt

Распознавание поля sudoku

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

Разделение на ячейки

Преобразование в чёрно-белое изображение

(1) В оттенках серого

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

(2) Размытие по Гауссу

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

(3) Бинарное изображение
(только чёрный и белый)

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

(4) Инвертированное чёрно-белое изображение

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

(5) Удалён шум

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

(6) Увеличено количество белого

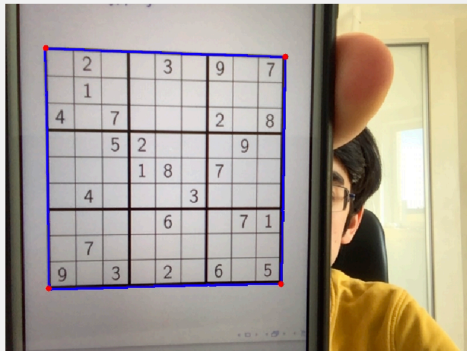
					9	4	7
		2		3		9	8
	6				2		1
						5	7
	7						6
8		3					
6			1				2
7	4			6		9	
	1	9	4				

Интерфейс GUI-приложения

Sudoku Solver



Camera image



Current scanned solvable sudoku field

```
[[0. 2. 0. 0. 3. 0. 9. 0. 7.]  
[0. 1. 0. 0. 0. 0. 0. 0. 0.]  
[4. 0. 7. 0. 0. 0. 2. 0. 8.]  
[0. 0. 5. 2. 0. 0. 0. 9. 0.]  
[0. 0. 0. 1. 8. 0. 7. 0. 0.]  
[0. 4. 0. 0. 0. 3. 0. 0. 0.]  
[0. 0. 0. 0. 6. 0. 0. 7. 1.]  
[0. 7. 0. 0. 0. 0. 0. 0. 0.]  
[9. 0. 3. 0. 2. 0. 6. 0. 5.]]
```

Run Sudoku Solver

Select image with sudoku field manually

Слот преобразования openCV изображения в PyQt изображение:

```
@pyqtSlot(np.ndarray)
def update_image(self, cv_img):
    """Updates the image_label with a new openCV image"""
    qt_img = self.convert_cv_to_qt(cv_img)
    self.frame_field.setPixmap(qt_img)
```

Слот обновления текста внутри QTextEdit:

```
@pyqtSlot(np.ndarray)
def update_text(self, sudoku_to_solve):
    """Updates the image_label with a new openCV image"""
    self.scanned_sudoku.setText(np.array2string(sudoku_to_solve))
```

Поток с результатом

```
class ThreadWithResult(threading.Thread):  
    def __init__(self, group=None, target=None, name=None,  
                 args=(), kwargs={}, *, daemon=None):  
        self.result = None  
  
    def function():  
        self.result = target(*args, **kwargs)  
    super().__init__(group=group, target=function,  
                    name=name, daemon=daemon)
```