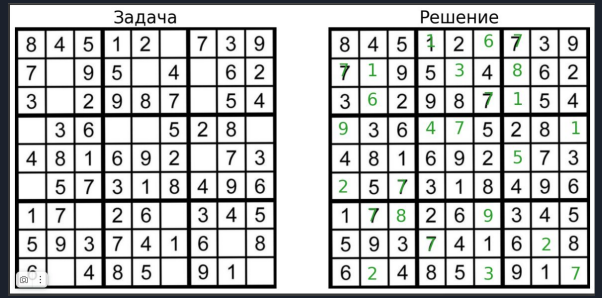
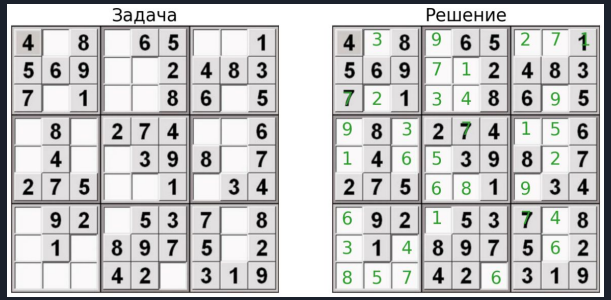
Решение судоку при помощи компьютерного зрения

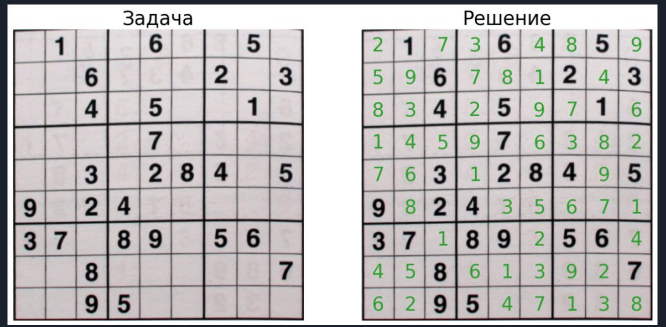
Бызова Анна, Дымова Диана

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.



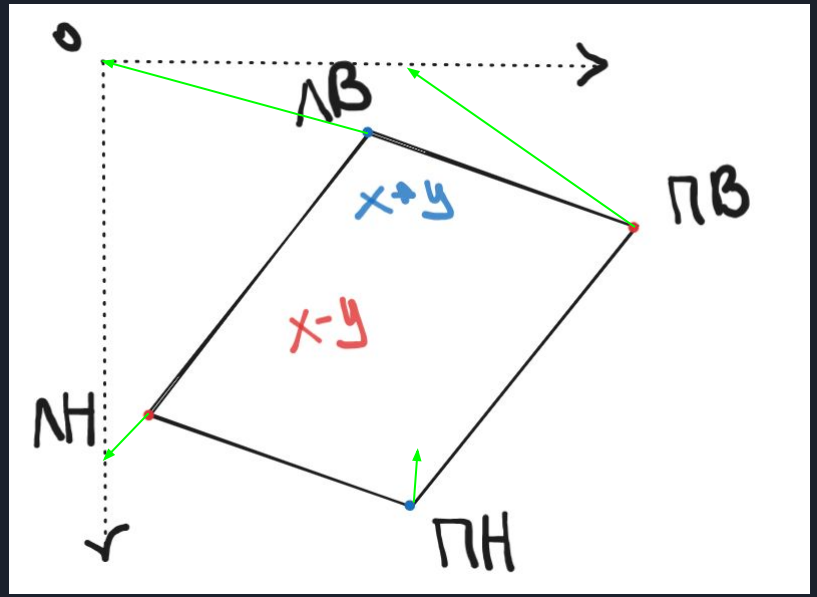




Процесс решения мы разбили на несколько этапов:

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

Этап 1. Ориентирование изображения

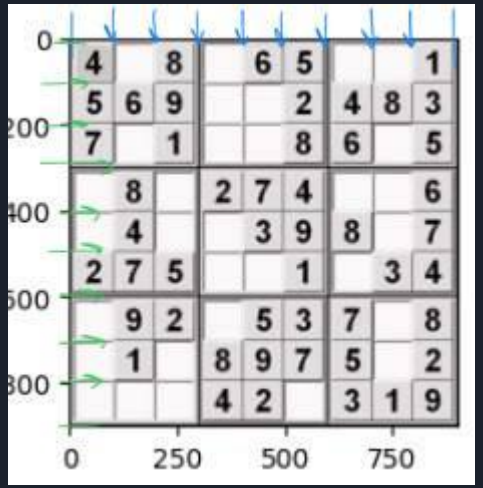


Определим вершины квадрата судоку.

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()