# Руководство пользователя

Для запуска процедуры получения координат воспользуйтесь скриптом main.py:

python main.py --crop\_name "D:\satellite\_hackathon\18. Sitronics\1\_20\crop\_0\_0\_0000.tif" --layout\_name "D:\satellite\_hackathon\18. Sitronics\layouts\layout\_2021-08-16.tif"

В папке result, рядом с файлом main.py, создастся файл coords.csv в требуемом формате.

## Запуск с помощью docker

Сначала необходимо собрать образ с помощью команды:

docker build -t satt\_app .

Затем, необходимые подложки поместить в папку layouts и необходимые сцены поместить в папку crops. После этого необходимо запустить контейнер следующим образом:

Для докера в Windows:

docker run --shm-size 8G --name satt -v .\:/app -t satt\_app:latest --crop\_name crops/crop\_0\_0\_0000.tif --layout\_name layouts/layout\_2021-08-16.tif

Для докера в Linux:

docker run --shm-size 16G --name satt -v ./:/app -t satt\_app:latest --crop\_name crops/crop\_0\_0\_0000.tif --layout\_name layouts/layout\_2021-08-16.tif

## Запуск процедуры поиска испорченых пикселей с помощью скрипта pixels\_detection.py

Для запуска процедуры получения координат испорченных пикселей и исправленного изображения воспользуйтесь скриптом pixels\_detection.py:

python pixels\_detection.py --crop\_name "D:\satellite\_hackathon\18. Sitronics\1\_20\crop\_0\_0\_0000.tif" [--output\_tif output\_0\_0.tiff --output\_txt output\_0\_0.txt]

# Руководство программиста

Алгоритм геопривязки main.py содержит следующие функции:

**calculate\_satellite\_matching**

Данная функция находит матчинги для патча и подложки с помощью алгоритма SIFT

**calc\_output\_coordinates**

Данная функция на основе вычисленного преобразования гомографии находит координаты в системе координат ESPG:32637

**resizing**

Данная функция ресайзит подложку

**get\_crop\_image**

Данная функция нормирует tiff изображение

**main**

Данная функция читает подложку, читает патч, нормирует патч и подложку, решейпит подложку, запускает матчинг, вычисляет преобразование координат, записывает в файл.

Алгоритм поиска пискелей в файле pixels\_detection.py содержит следующие функции:

**search\_error\_pixels\_in\_slice**

Создает размытое изображение. Пробегается по каждому пикселю изображения и текщий пиксель сильно больше или меньше пикселя на размытом изображении, заменяет его на пиксель с размытого изображения

**save\_slices\_list\_to\_tiff**

Сохраняет tiff в файл

**save\_error\_pixels\_information**

Сохраняет информацию о пикселях в txt файл

**search\_error\_pixels\_in\_image**

Разбивает исходный tiff на слои, на каждом слое запускает его обработку

**Main**

Читает файл, запускает поиск, запускает сохранение