

# Руководство пользователя

## Введение

Данный проект разработан в рамках хакатона "Лидеры цифровой трансформации". Сервис предназначен для определения географической привязки сцен спутниковых снимков и корректировки битых пикселей. Сервис полностью автоматизирован и не требует промежуточной настройки в процессе работы.

## Возможности

- Высокоскоростной алгоритм обработки данных
- Автоматическое определение геопозиции
- Обнаружение и корректировка битых пикселей
- Поддержка вывода в нескольких форматах (CSV, GeoJSON, GeoTIFF)
- RESTful API для легкой интеграции

## Использование

### Использование с помощью API

Для обработки изображений через API сервис предоставляет эндпоинты, указанные ниже в разделе "Эндпоинты". Вы также можете использовать инструменты, такие как `curl` или Postman.

### Пример

1. Загрузите изображение для обработки:

```
curl -X POST "http://localhost:8000/process" -F "layout_name=<layout_filename>" -F "file=@<
```

2. Проверьте результаты:

```
curl -X GET "http://localhost:8000/coords?task_id=<task_id>"
```

# Использование с помощью скрипта

Чтобы обработать изображения с помощью скрипта, вы можете запустить его внутри Docker-контейнера или настроить локальную среду Python.

## Использование скрипта в Docker-контейнере

Запустите следующую команду:

```
docker run --rm -v ./app -v /layouts:<layouts_dir> nikolove18 python -m src.main --layout_name
```

## Использование скрипта локально

1. Настройте окружение Python и установите зависимости:

```
pip install -r requirements.txt
```

2. Запустите скрипт:

```
python main.py --crop_name <path_to_crop_image> --layout_name <path_to_layout_image>
```

# Эндпоинты

## POST /process

Запускает задачу обработки изображения.

- **Параметры:**
  - `layout_name` (string): Имя файла подложки.
  - `file` (file): Файл изображения для обработки.
- **Ответ:**
  - `task_id` (string): ID задачи обработки.

## GET /coords

Получает геопривязанные координаты обработанного изображения.

- **Параметры:**
  - `task_id` (string): ID задачи обработки.
- **Ответ:**

- JSON-объект с координатами и другой информацией об обработке.

## **GET /bug\_report**

Получает отчет об ошибках для коррекции битых пикселей.

- **Параметры:**
  - `task_id` (string): ID задачи обработки.
- **Ответ:**
  - JSON-объект с деталями отчета об ошибках.

## **GET /download/geojson**

Скачивает результат в виде файла GeoJSON.

- **Параметры:**
  - `task_id` (string): ID задачи обработки.
- **Ответ:**
  - Файл GeoJSON.

## **GET /download/geotiff**

Скачивает результат в виде файла GeoTIFF с геопривязкой.

- **Параметры:**
  - `task_id` (string): ID задачи обработки.
- **Ответ:**
  - Файл GeoTIFF.

## **GET /download/corrected\_pixels**

Скачивает откорректированные пиксели в исходной системе координат.

- **Параметры:**
  - `task_id` (string): ID задачи обработки.
- **Ответ:**
  - Файл GeoTIFF.

## **GET /download/bug\_report**

Скачивает отчет об ошибках в формате CSV.

- **Параметры:**

- `task_id` (string): ID задачи обработки.

- **Ответ:**

- CSV-файл.