



Модуль выравнивания цвета

Техническое описание

Листов 18

СОДЕРЖАНИЕ

1	Общее описание приложения	6
1.1	Назначение ПО	6
2	Функциональные возможности приложения	7
2.1	Описание алгоритма	7
3	Архитектура приложения	10
3.1	Схема архитектуры	10
3.2	Основные компоненты приложения	10
4	Рекомендуемое оборудование и программное обеспечение	12
4.1	Рекомендуемое оборудование	12
4.2	Программное обеспечение	12
5	Входные и выходные данные	14
5.1	Входные данные	14
5.2	Выходные данные	14
6	Описание технических средств хранения исходного текста и объектного кода программного обеспечения, а также технических средств компиляции исходного текста в объектный код программного обеспечения	15
6.1	Хранение исходного кода	15
6.2	Компиляция исходного кода	15
6.3	Адрес нахождения средств хранения кода	16
7	Перечень используемых технологий	17

ВВЕДЕНИЕ

Настоящий документ представляет собой техническое описание программного обеспечения «Модуль выравнивания цвета» ООО «Пиксельпро». Документ содержит общее описание модуля, описание функциональных возможностей и алгоритма работы, архитектуры, входных и выходных данных. В документе указано рекомендуемое оборудование и программное обеспечение, необходимое для эксплуатации модуля, а также приведен перечень используемых технологий. Кроме того, в документе приведено описание технических средств хранения и компиляции исходного кода программного обеспечения.

ПРИНЯТЫЕ ТЕРМИНЫ И СОКРАЩЕНИЯ

Термины и определения, используемые в настоящем документе, представлены в следующей таблице (Таблица 1).

Таблица 1 — Термины и определения, используемые в настоящем документе

Термин	Определение
Архитектура «клиент-сервер»	Вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами
Искусственный интеллект	Свойство искусственных интеллектуальных систем выполнять творческие функции, которые традиционно считаются прерогативой человека; также это наука и технология создания интеллектуальных машин, особенно интеллектуальных компьютерных программ
Компьютерное зрение	Область искусственного интеллекта, связанная с анализом изображений и видео
Машинное обучение	Класс методов искусственного интеллекта, характерной чертой которых является не прямое решение задачи, а обучение в процессе применения решений множества сходных задач
ПО	Программное обеспечение
Фотограмметрия	Техника определения размеров, положения, форм и других визуальных характеристик объектов по фотографическим снимкам
CPU (англ. Central Processing Unit)	Центральный процессор компьютера
GPU (англ. Graphics Processing Unit)	Графический процессор, предназначенный для обработки и отображения компьютерной графики
Docker	Программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации
Docker-compose	Компонент экосистемы Docker, который позволяет запускать множество контейнеров одновременно и маршрутизировать потоки данных между ними
Docker-образ	Неизменяемый файл, который содержит всё необходимое для запуска приложения: исходный код, среду выполнения, библиотеки, зависимости, переменные окружения и конфигурационные файлы
SMB	Сетевой протокол для общего доступа к файлам, который позволяет приложениям компьютера читать и за-

	<p>писывать файлы, а также запрашивать службы серверных программ в компьютерной сети. Протокол SMB может использоваться поверх протокола TCP/IP или других сетевых протоколов. С помощью протокола SMB приложение (или использующий его пользователь) может получать доступ к файлам и другим ресурсам удаленного сервера. Это позволяет приложениям читать, создавать и обновлять файлы на удаленном сервере.</p>
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

1 Общее описание приложения

Модуль выравнивания цвета (далее – модуль) представляет собой веб-приложение, предназначенное для выравнивания цветовых оттенков между группой снимков.

Модуль обрабатывает исходные фотографии, находит области их пересечения и редактирует каждую фотографию так, чтобы при наложении соседних фотографий место их примыкания стало менее заметным.

Работа приложения основана на таких методах искусственного интеллекта, как машинное обучение и компьютерное зрение.

1.1 Назначение ПО

Модуль выравнивания цвета может применяться в комплексе с другими средствами или мероприятиями, целью которых является решение задач построения топографических карт или 3D моделей местности. Модуль используется как отдельный метод в составе комплексной обработки снимков – материалов топографической съёмки. Кроме того, модуль может использоваться как самостоятельное ПО для решения различных задач, где нужно осуществлять обработку группы изображений.

2 Функциональные возможности приложения

Основной функциональной возможностью ПО «Модуль выравнивания цвета» является выравнивание цветовых оттенков для группы фотографий, которые указал пользователь. Модуль меняет тон и яркость цвета на каждом снимке так, чтобы сгладить места примыкания этого снимка к соседним фотографиям.

Для реализации этой возможности модуль обеспечивает выполнение следующих функций:

1. Получение входных данных для задачи обработки изображений от пользователя, включающих:
 - директорию с исходными снимками, которые нужно обработать, в определенном SMB хранилище изображений;
 - директорию для обработанных снимков — куда будет сохранен результат обработки изображений.
2. Выравнивание цвета для группы фотографий из директории, которую указал пользователь.
3. Сохранение обработанных изображений в указанную директорию для обработанных снимков.

2.1 Описание алгоритма

Алгоритм работы модуля выравнивания цвета состоит из следующих шагов.

1. Подготовка к работе
Подготовка подразумевает сканирование входных данных и извлечение GPS-данных, если они предоставлены.
2. Поиск ключевых точек
Поиск ключевых точек осуществляется с помощью алгоритма SIFT на платформе COLMAP.
Примечание 1. SIFT (от англ. scale-invariant feature transform, масштабно-инвариантная трансформация признаков) — это алгоритм выявления признаков в компьютерном зрении для определения и описания локальных признаков изображений.
Примечание 2. COLMAP — это универсальное программное обеспечение для фотограмметрии, которое совмещает в себе алгоритмы построения 3D структур

по набору изображений (Structure From Motion) и алгоритмы многоракурсного стерео-отождествления (Multi-View Stereo).

3. Предварительный отбор пар изображений по GPS

Данный шаг нужен для оптимизации алгоритма в целях ускорения поиска соответствий между изображениями. Этот шаг пропускается, если GPS-данные не предоставлены.

4. Поиск пересекающихся изображений

Поиск пересекающихся изображений производится с помощью платформы COLMAP. Алгоритм проверяет найденные по GPS пары изображений и оставляет только те из них, которые можно наложить друг на друга для сравнения цвета.

5. Создание набора данных для выравнивания цветовых оттенков

На данном шаге алгоритм использует найденные ранее пары пересекающихся фотографий и создает набор цветовых оттенков с координатами совпадающих ключевых точек.

- Сбор цветовых оттенков ключевых точек на каждом изображении

Это вспомогательный шаг, чтобы в дальнейшем уменьшить количество операций чтения с диска.

- Создание набора данных для оптимизации (выравнивания цвета).

6. Оптимизация (выравнивание цвета)

На этом шаге каждому изображению назначается набор коэффициентов, которые оптимизируются методом градиентного спуска. С помощью этих коэффициентов изображения освещаются/затемняются по каждому каналу так, чтобы минимизировать разницу тона и яркости на пересечении изображений. В качестве библиотеки для решения задачи оптимизации используется фреймворк машинного обучения PyTorch, который в составе своих методов содержит метод градиентного спуска.

Примечание 1. Метод градиентного спуска — это численный метод нахождения локального минимума или максимума функции с помощью движения вдоль градиента (градиент — это градуированный переход между двумя или более цветами или двумя оттенками одного цвета). Метод градиентного спуска является одним из основных численных методов современной оптимизации.

Примечание 2. PyTorch — это фреймворк машинного обучения для языка Python с открытым исходным кодом. PyTorch выделяется своей гибкостью, удобством использования и большими возможностями ускорения GPU, что делает его популярным инструментом для решения широкого спектра задач машинного обучения и, в особенности, компьютерного зрения.

7. Обработка изображений

На данном шаге применяются все преобразования, полученные на предыдущем шаге. После этого модуль сохраняет на диск обработанные изображения.

Ниже приведены примеры наложения фотографий, сделанные до и после применения модуля выравнивания цвета.

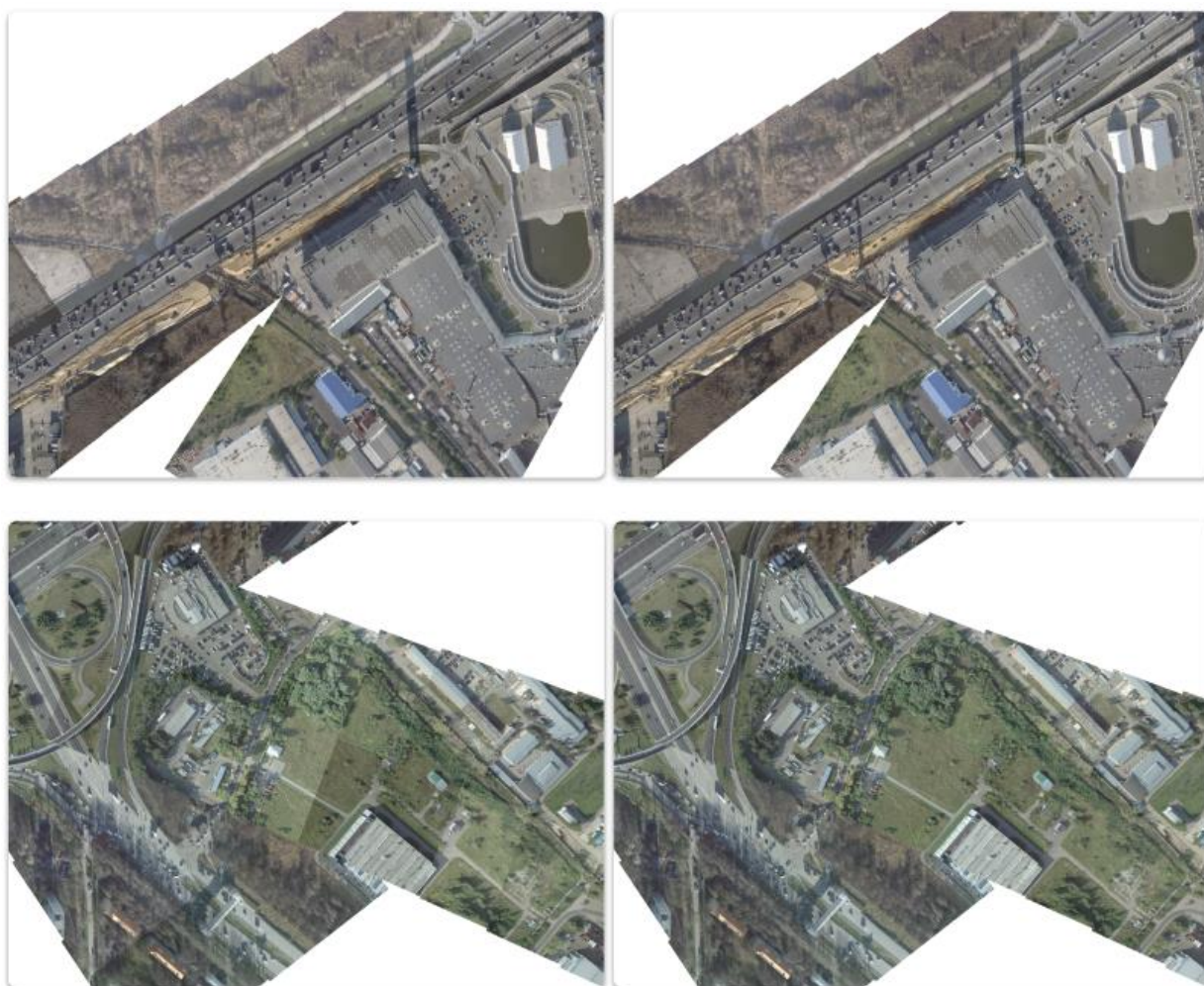


Рисунок 1 — Примеры наложения фотографий до применения модуля выравнивания цвета (слева) и после применения модуля (справа)

3 Архитектура приложения

Приложение разработано на базе клиент-серверной архитектуры, где клиентом является Frontend (веб-интерфейс, с которым взаимодействует пользователь), а сервер — это основная часть ПО, реализующая логику продукта. Взаимодействие с серверной частью происходит посредством HTTP запросов.

Логическая архитектура приложения соответствует схеме, приведенной на рисунке ниже (Рисунок 2).

3.1 Схема архитектуры

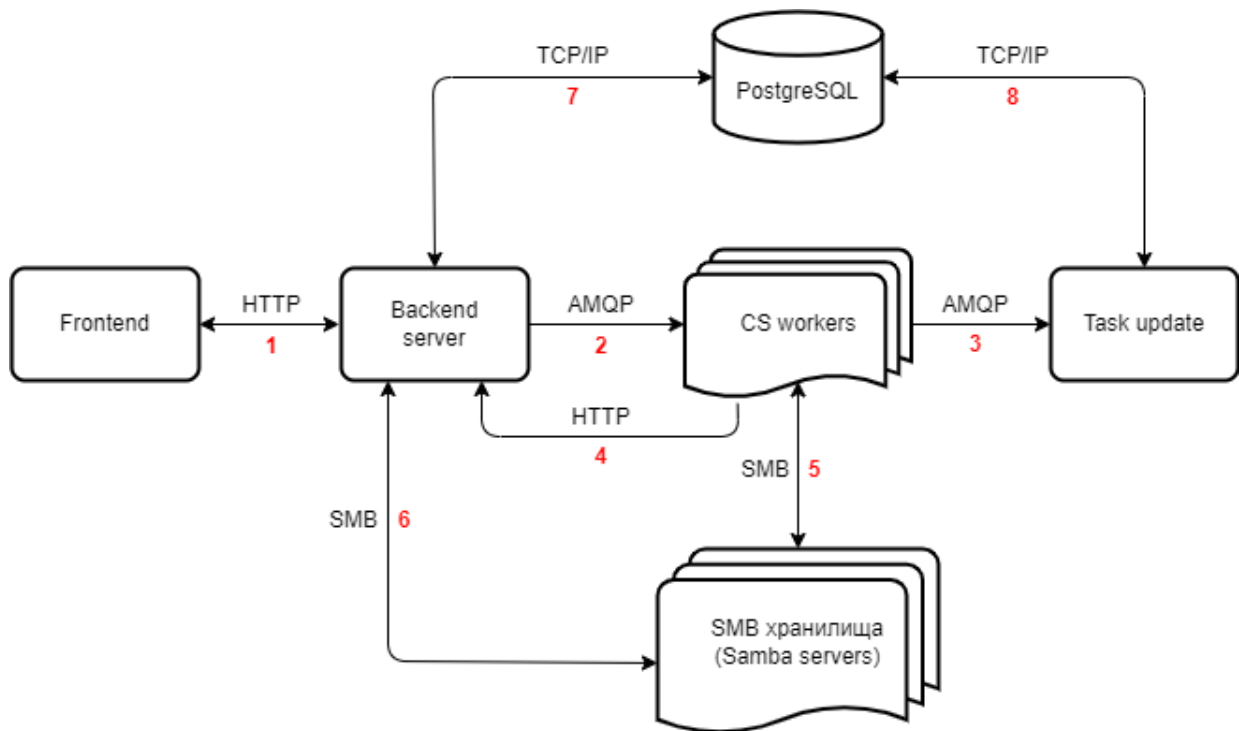


Рисунок 2 — Схема архитектуры

Обработчики для задачи выравнивания цвета обозначены на данной диаграмме как «CS workers».

3.2 Основные компоненты приложения

Основные компоненты приложения и их описание представлены в следующей таблице (Таблица 2).

Таблица 2 — Основные компоненты приложения

Компоненты приложения	Назначение	Продукт / Технология
Хранилище данных	Осуществляет долговременное централизованное хранение данных: задач на обработку изображений, созданных пользователем.	PostgreSQL
SMB хранилище изображений	Файловое хранилище изображений в формате .jpg. Осуществляет хранение исходных изображений пользователя и обработанных изображений (после процедуры выравнивания цвета).	Файловая система Server Message Block (SMBFS), Samba Server
Frontend	Клиентская часть продукта (веб-интерфейс, с которым взаимодействует пользователь).	JavaScript, React, Redux
Сервер	Основная часть ПО, реализующая логику продукта. Состоит из следующих компонентов: - Backend server, - обработчиков CS workers (обработчиков для задачи выравнивания цвета), - обработчиков Task update (которые следят за прогрессом задач и обновляют их статус ¹).	AIОHTTP, Python, RabbitMQ, библиотеки Python (см. п. 6)

¹ Статус задачи может быть: running (в процессе), completed (завершена), error (завершена с ошибкой) или canceled (отменена)

4 Рекомендуемое оборудование и программное обеспечение

4.1 Рекомендуемое оборудование

В данном разделе приводятся ориентировочные характеристики оборудования, при использовании которого можно гарантировать работоспособность модуля выравнивания цвета со всеми его функциональными возможностями.

Требования к оборудованию зависят от желаемой производительности. Оптимальная конфигурация сервера обработки изображений должна иметь следующие характеристики:

- процессор: Intel(R) Core(TM) i7-7700 CPU @ 3.60 ГГц;
- оперативное запоминающее устройство: 32 ГБ DDR4 DIMM 2133 МГц;
- видеокарта NVIDIA GeForce GTX TITAN X, с поддержкой cuda 11.0, cudnn 8;
- подсистема хранения: HDD SATA-3 250 ГБ;
- интернет-соединение со скоростью 1 Мбит/с.

Рекомендуемые типы камер, которые производят съемку:

- A3,
- CityMapper,
- UrbanMapper,
- DJI Hasselblad.

Примечания:

1. ПО сохраняет свою работоспособность при использовании менее мощного оборудования. Однако в этом случае рекомендуется сначала провести тестирование приложения на этом оборудовании, либо направить запрос на возможность его применения производителю ПО.
2. Требования к оборудованию, приведенные в данном разделе, являются ориентировочными. Характеристики оборудования оговариваются отдельно для каждой поставки.

4.2 Программное обеспечение

Модуль выравнивания цвета обеспечивает работоспособность всех своих функциональных возможностей в среде операционной системы Ubuntu версии 22.04.

Кроме того, на сервере обработки изображений должно быть установлено следующее общесистемное ПО:

1. Docker: версия 19.03 и выше

2. Docker-compose: версия 1.28.* и выше
3. Драйверы NVIDIA Linux: версия 418.81.07 и выше
4. NVIDIA Container Toolkit: версия 12.2 и выше

5 Входные и выходные данные

5.1 Входные данные

Входными данными для приложения являются изображения в следующих форматах:

- jpg/jpeg
- png
- tif/tiff

Изображения должны быть помещены в специальную папку, которую укажет пользователь в качестве директории с исходными снимками.

5.2 Выходные данные

Выходные данные приложения — это обработанные снимки. Обработанные снимки сохраняются в формате jpg. Они сохраняются в специальную папку, которую укажет пользователь в качестве директории для обработанных снимков.

6 Описание технических средств хранения исходного текста и объектного кода программного обеспечения, а также технических средств компиляции исходного текста в объектный код программного обеспечения

6.1 Хранение исходного кода

Для разработки ПО «Модуль выравнивания цвета» используется инструмент GitLab Community Edition, размещенный на локальном сервере ООО «Пиксельпро»

GitLab — это веб-инструмент жизненного цикла DevOps с открытым исходным кодом, представляющий систему управления репозиториями кода для Git (распределённой системы управления версиями) с собственной справочной системой «вики», системой отслеживания ошибок, CI/CD пайплайном и другими функциями. (CI/CD пайплайн — это поток автоматической интеграции и доставки/развёртывания приложений.)

Исходный код программного обеспечения «Модуль выравнивания цвета» хранится в корпоративном репозитории GitLab.

Обновление модуля осуществляется с помощью инструмента Docker-compose.

6.2 Компиляция исходного кода

Для сборки (компиляции) исходного кода в объектный код ПО используется платформа Docker. В процессе сборки формируются Docker-образы, которые содержат всё необходимое для запуска приложения: исходный код, среду выполнения, библиотеки, зависимости, переменные окружения и конфигурационные файлы.

Docker-образы являются дистрибутивом прикладного ПО — Модуля выравнивания цвета — и содержатся на поставляемом носителе данных.

Запуск Docker-контейнеров из Docker-образов и развёртывание прикладного ПО осуществляется средствами Docker и Docker-compose.

Примечание 1. Docker-образ — это неизменяемый файл, из которого разворачивается контейнер. Docker-контейнер — это работающий в изолированной среде экземпляр образа. В один контейнер «упаковывается» один работающий серверный процесс.

Примечание 2. Docker — это программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации. Docker позволяет «упаковать» приложение со всем его окружением и зависимостями в контейнер, который может быть развёрнут на любой Unix-подобной системе. Docker позволяет настраивать взаимодействие контейнеров с внешней средой и другими контейнерами, а также регулировать потребление ресурсов.

Примечание 3. Docker-compose — это компонент экосистемы Docker, который позволяет запускать множество контейнеров одновременно и маршрутизировать потоки данных между ними.

6.3 Адрес нахождения средств хранения кода

Для хранения исходного кода программного обеспечения «Модуль выравнивания цвета» и его версий используется локальное файловое хранилище на выделенном компьютере, находящемся по адресу: г. Москва, ул. Тайнинская, д.7.

7 Перечень используемых технологий

Для реализации ПО «Модуль выравнивания цвета» использовались следующие инструменты и технологии:

- Docker (версия 19.03 и выше)
- Docker-compose (версия 1.28.* и выше)
- PostgreSQL 9.6
- RabbitMQ 3.8.*
- JavaScript и JavaScript-библиотеки:
 - React
 - Redux
- АИОНТТР 3.8.4 (Asynchronous HTTP Client/Server for Python)
- Python 3.7
- Библиотеки:
 - Colmap (версия 3.7)
 - Eigen (версия 3.4)
 - Ceres Solver (версия 2.1.0)
 - FFmpeg (версия 4.2.7-0ubuntu0.1)
 - libsm6 (версия amd64-2:1.2.3-1)
 - libxext6 (версия amd64-2:1.3.4-0ubuntu1)
 - libglib2.0-0
 - libxrender1
 - libgl1-mesa-glx
- Библиотеки Python:
 - Cameratransform (версия 1.2)
 - Exiv2 (версия 0.14.0)
 - NumPy (версия 1.22)
 - OpenCV (версия 4.5.5.64)
 - Pandas (версия 1.4)
 - Pillow (версия 9.0)
 - Psutil (версия 5.9.0)
 - Puroj (версия 3.4.1)
 - PyTorch (версия 1.13)
 - Pyluturbojpeg (версия 1.7.0)

- SciPy (версия 1.8)
- Shapely (версия 2.0.1)
- Tqdm (версия 4.63)
- Vector3d (версия 1.1.1)