

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра ИС

ОТЧЕТ
по учебной практике
Тема: Определение палитры растрового изображения

Студент гр. 1374	_____	Зырянов В.М.
Студент гр. 1374	_____	Колпачев В.Ю.
Студент гр. 1374	_____	Москвин С.О.
Куратор	_____	Прокопович Ю.В.

Санкт-Петербург
2023

ЗАДАНИЕ НА УЧЕБНУЮ ПРАКТИКУ

Студенты: Москвин Сергей, Колпачев Владислав,

Зырянов Виталий

Группа: 1374

Тема практики: Определение палитры растрового изображения

Исходные данные:

Растровое изображение произвольного размера в форматах: PNG, JPG, BMP.

Изображения в формате PNG может содержать области с прозрачностью: прозрачность должна корректно обрабатываться приложением.

Содержание пояснительной записки:

«Содержание», «Введение», «Основная часть», «Заключение», «Список использованных источников»

Предполагаемый объем пояснительной записки:

Не менее 11 страниц.

Сроки прохождения практики: 10.03.2023 – 01.05.2023

Дата сдачи отчета:

Дата защиты отчета:

Студент гр. 1374

Зырянов В.М.

Студент гр. 1374

Колпачев В.Ю.

Студент гр. 1374

Москвин С.О.

Куратор

Прокопович Ю.В.

АННОТАЦИЯ

Работа по теме "Определение палитры растрового изображения" представляет собой разработку программного продукта для определения цветовой палитры в растровых изображениях. Проект включает разработку алгоритмов и методик для автоматического определения популярных цветов изображения, а также создание графического интерфейса, позволяющего пользователям работать с изображениями. Для достижения поставленной цели будет применяться язык программирования C++ с использованием средств среды разработки QtCreator.

СОДЕРЖАНИЕ

	Введение	5
1.	Описание приложения	6
2.	Руководство оператора на приложение	7
3.	Описание алгоритмов	8
4.	Обоснование выбора алгоритмов	9
	Заключение	10
	Список использованных источников	11

ВВЕДЕНИЕ

Цветовая палитра, используемая в растровых изображениях, является одним из важных характеристик, определяющих внешний вид изображения. Часто возникает необходимость автоматического определения цветовой палитры, особенно при работе с большим количеством изображений.

Целью данной работы является определение палитры растрового изображения с использованием языка программирования C++ и фреймворка Qt.

Задачи необходимые для выполнения поставленной цели:

1. Изучить библиотеку Qt и возможности работы с изображениями.
2. Создать приложение на C++, которое определяет палитру изображения.
3. Осуществить вывод палитры в окно приложения с указанием шестнадцатеричных кодов цветов в формате RGB.

1. ОПИСАНИЕ ПРИЛОЖЕНИЯ

Приложение является программным инструментом для автоматического определения цветовой палитры в растровых изображениях формата PNG, JPG, BMP. Приложение имеет графический интерфейс пользователя, который позволяет легко загрузить изображение, проанализировать его цветовую палитру, а затем сохранить полученную палитру в файле формата PNG.

2. РУКОВОДСТВО ОПЕРАТОРА НА ПРИЛОЖЕНИЕ

Для использования приложения необходимо выполнить следующие действия:

Шаг 1. Запуск приложения

Шаг 2. Загрузка изображения

Первым шагом необходимо загрузить растровое изображение в приложение. Для этого нужно нажать кнопку "Загрузить изображение" и выбрать изображение из файловой системы.

Шаг 3. Определение палитры

После загрузки изображения его палитра будет автоматически определена приложением. На экране отобразятся цвета и их названия в hex.

Шаг 4. Экспорт палитры

После определения палитры, Вы можете сохранить ее результаты. Нажмите кнопку "Экспортировать палитру", выберите папку для сохранения и укажите имя файла. Результаты определения палитры будут сохранены в выбранном вами месте в формате PNG. описание алгоритмов, используемых в приложении

3.ОПИСАНИЕ АЛГОРИТМОВ

- Окно приложения представлено классом `MainWindow`, в котором реализованы слоты нажатия кнопок, функция нахождения и отображения палитры и поле `palette` типа `QList<QColor>`, хранящее найденную палитру.
- Для загрузки изображения в приложение используется функция `QFileDialog::getOpenFileName` из библиотеки `QFileDialog` и функция `load` из библиотек `QImage` и `QPixmap`.
- Для подсчета количества пикселей определенного цвета используется класс `QMap`. В роли ключей выступает имя цвета в шестнадцатеричном формате в `QString`, роль значений выполняет число пикселей данного цвета в формате `int`. Сам подсчет ведется попиксельно.
- Для нахождения самых часто встречающихся цветов используются функции `values` и `keys` из библиотеки `QMap`, а также функция `std::max_element`
- Вывод цветов осуществляется с предварительным сравнением уже выведенных на предмет визуальной близости к ним. Это достигается при помощи функции `getDistanceBetweenColors`, которая находит расстояние между цветами в трехмерном пространстве RGB с поправкой на неоднородность человеческого зрения. Так расстояние по оси Red умножается на 0.299, по оси Blue - на 0.114, по оси Green - на 0.587. В программе близкими цветами считаются цвета, евклидово расстояние между которыми меньше 35.
- Для сохранения палитры используется функция `QFileDialog::getSaveFileName` из библиотеки `QFileDialog` и функция `save` из библиотеки `QImage`.

4.ОБОСНОВАНИЕ ВЫБОРА АЛГОРИТМОВ

Алгоритм определения цветов, заключающийся в прохождении по каждому пикселю и выборе наиболее популярных, является наиболее простым и эффективным. Сложность алгоритма не может быть меньше чем $O(w*h)$, где w и h - ширина и высота изображения соответственно. Просмотреть цвета изображения быстрее $O(w*h)$ невозможно без потерь. В результате получается наиболее близкое к человеческом зрению определение палитры изображения. Это достигается также за счет выбранной метрики измерения расстояния между цветами.

Кроме того, данный алгоритм не требует сложных вычислений, и его можно легко реализовать на языке C++ с использованием фреймворка Qt.

ЗАКЛЮЧЕНИЕ

Проведена аналитическая работа, в ходе которой были изучены существующие алгоритмы и методы для нахождения палитры растрового изображения. Создано приложение, содержащее выбранный нами алгоритм и удовлетворяющее всем предварительным требованиям.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Официальная документация фреймворка Qt.

URL:<https://doc.qt.io>. (Дата обращения 16.04.23)