

# О ПОДХОДАХ К РАЗРАБОТКЕ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ СТИЛИСТИЧЕСКОГО АНАЛИЗА ИЗОБРАЖЕНИЙ

Денищик Р. И., Босько В. О., Рудикова Л. В.

Кафедра современных технологий программирования, Гродненский государственный университет имени Янки Купалы  
Гродно, Республика Беларусь  
E-mail: rudikowa@gmail.com

*В статье приводятся основные подходы к реализации мобильного приложения «Imagination» на платформе iOS, предназначенного для автоматического захвата изображений и проведения их стилистического анализа с последующей возможностью поиска схожих изображений по полученным метаданным. Кроме того, описываются аспекты реализации программного обеспечения и общие модели разрабатываемого программного продукта.*

## ВВЕДЕНИЕ

На сегодняшний момент широкое распространение и развитие получили различные алгоритмы и технологии, направленные на проведение эффективного анализа графической информации. На их основе реализовано приложение, которое предлагает пользователю произвести сканирование изображения и, в качестве результата, получить набор характеристик, которые невозможно выявить человеческим глазом. Данная разработка способна выступать в качестве дополнительного инструмента при проведении различного рода экспертных оценок изображений. Разрабатываемое приложение «Imagination» позволяет пользователю произвести анализ изображений с использованием различных математических алгоритмов. Выявить цветовые зависимости и используемые геометрические формы, загрузить результаты анализа на удаленное хранилище и создать общедоступные базы проанализированных объектов, произвести сравнительный анализ двух и более элементов.

### I. ОБ ОБЩЕЙ КОНЦЕПЦИИ МОБИЛЬНОГО ПРИЛОЖЕНИЯ

Основная концепция предлагаемого программного продукта связана с автоматическим захватом изображений, их последующим анализом и сохранением результатов. Поступающий через камеру мобильного устройства поток данных обрабатывается особым образом, результатом которого является набор контуров объектов. Полученные объекты выделяются на изображении и предлагаются пользователю на выбор для проведения анализа. После проведения анализа пользователь может составить так называемый паспорт изображения, который включает в себя цветовую и объектную карты. Цветовая карта содержит в себе различную информацию об использованных в изображении цветовых элементах, их распределении и представлении. Объектная карта представляет собой набор данных, которые описывают комбинации использо-

ванных элементарных объектных форм, частоту их применения и способы распределения. После проведения анализа изображения, пользователю предоставляется возможность сравнить полученные результаты, например, для определения уникальности изображения или поиска его аналогов. Основной функционал приложения «Imagination» включает в себя следующие возможности:

- регистрация и авторизация пользователя в приложении;
- произведение захвата изображения;
- составление цветовой карты изображения;
- составление объектной карты изображения;
- составление комплексного паспорта изображения, включающего в себя само изображение, цветовую, объектную карты и прочие характеристики;
- загрузка паспорта изображения на удаленное хранилище;
- просмотр паспортов изображений других пользователей;
- сравнение нескольких изображений;
- поиск схожих изображений по имеющимся метаданным.

При анализе схожих программных продуктов, можно сделать вывод, что их основным недостатком является работа исключительно с текстовыми данными. Как итог, можно установить, что решения, которые реализовывали бы заявленный нами функционал, отсутствуют, что открывает широкие возможности для продвижения предлагаемого программного продукта.

Работу основного функционала приложения, а именно захват и анализ изображений, можно разделить на три этапа: захват анализируемого изображения, анализ цветовой составляющей изображения и анализ объектной составляющей изображения. Рассмотрим каждый из этапов.

Захват анализируемого изображения происходит следующим образом. В первую очередь,

приложение перехватывает видеопоток с камеры устройства. После, на копию буфера данных поэтапно накладываются фильтры, которые в некоторой степени размывают изображение, что впоследствии позволяет алгоритмам лучше распознавать контуры различных объектов, не сталкиваясь со случайными разрывами из-за шумов на изображении. Затем, измененная копия изображения преобразуется к бинарному виду, состоящему только из черных и белых точек, это необходимый шаг для однозначного выделения контуров среди широкой палитры, представленной на изображении.

Итого, после завершения данного этапа, в памяти приложения имеется 2 варианта изображения: оригинал и бинарная черно-белая копия, для выделения контуров. Далее приходит время для 2-го этапа, а именно анализа цветовой составляющей. На этом этапе, алгоритм обрабатывает каждый пиксель оригинального изображения и составляет несколько специальных объектов, хранящий в себе информацию для различных цветовых схем. Один для RGB-схемы и один для HSV. В этих объектах находятся подструктуры, описывающие количество пикселей соответствующих некоторой компоненте цветовой схемы.

После этого, на основе данных объектов, строится графическое представление распределения цвета по изображению. Это делается исключительно для дополнительного ознакомления пользователя, потому как по факту, сравнение изображений производится не по графическому представлению, конечно, а по самим объектам.

Во время проведения объектного анализа, используется бинарное черно-белое изображение, на котором определяются все имеющиеся контуры. При этом используются конкретные значения специальных погрешностей, которые могут допустить некоторые разрывы, чтобы сохранить общую целостность контура. Далее, полученные контуры фильтруются и из них выделяются наборы контуров, соответствующих определенным критериям. Выделяются следующие фигуры: треугольники – контур, имеющий 3 угла; прямоугольники – выпуклые контуры, имеющие 4 угла, каждый из которых более 85, и менее 95 градусов; круги – выпуклые многоугольники, имеющие 8 и более углов, при этом площадь вписанной окружности должна быть меньше площади контура.

Составив специальные структуры данных, описывающие наборы контуров этих типов, мы представить их графически на оригинале изображения. Эта, опять же, производится исключительно для дополнительного информирования и вовлечения пользователя, потому как сравне-

ние изображений производится именно по структурам с данными. Для реализации приложения был выбран объектно-ориентированный язык программирования Objective-C. Objective-C – это компилируемый объектно-ориентированный язык программирования, используемый корпорацией Apple, построенный на основе языка C (Си) и парадигм Smalltalk [1-3].

Обработка взаимодействия между пользовательским интерфейсом и логикой следует архитектурному шаблону Model View Controller. Он предполагает разделение кода на классы, каждый из которых относится к одной из трех ролей. Классы моделей реализуют модель данных приложения, классы видов реализуют пользовательский интерфейс, а классы контроллеров координируют взаимодействие моделей и видов. В рамках этого паттерна модель и вид никогда не должны взаимодействовать друг с другом напрямую, минуя контроллер. С его помощью код структурируется на отдельные слои или зоны ответственности, что значительно облегчает повторное использование одних и тех же классов в разных местах приложения и перенос кода между приложениями. Внутреннее взаимодействие и строение мобильного приложения под управлением ОС iOS представлено на рисунке 1.

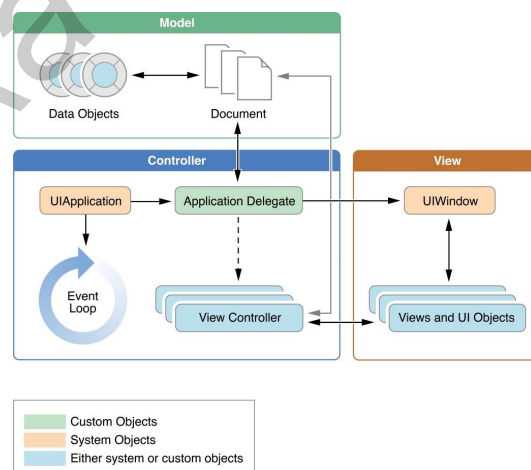


Рис. 1 – Строение мобильного приложения

При реализации механизмов захвата и анализа изображений нами была использована библиотека OpenCV. Библиотека OpenCV (Open Source Computer Vision).

## II. СПИСОК ЛИТЕРАТУРЫ

1. iOSDeveloperLibrary [Электронный ресурс] – Режим доступа: <https://developer.apple.com/library/ios/navigation>. – Дата доступа: 04.04.2017.
2. NSHipster [Электронный ресурс] – Режим доступа: <http://nshipster.com>. – Дата доступа: 04.04.2017.
3. Learning OpenCV. Computer vision with the OpenCV Library. (Gary Bradski, Adrian Kaehler), O'Reilly Media, Сентябрь 2008 г.