

Санкт-Петербургский Государственный Университет
Математико-механический факультет

Кафедра системного программирования

Илья Игоревич Келим

Архитектура и системы для
комбинированных рекомендаций и поиска
эстетичных изображений

Промежуточный отчет по курсовой работе

Научный руководитель:
к. т. н., доцент Брыксин Т. А.

Консультант:
Основатель “graphica.ai” Брыксин М. А.

Санкт-Петербург
2019

Оглавление

Введение	3
1. Обзор	5
1.1. Обзор конкурентов	5
1.2. Обзор предметной области	6
2. Описание подхода	9
2.1. Условия решаемой задачи	9
2.2. Рекомендательная система	9
2.3. Поиск	10
3. Реализация	11
3.1. Архитектура	11
3.2. Модуль поиска по цвету	11
3.3. Модуль поиска по текстовой информации	13
3.4. Модуль комбинированного поиска	14
3.5. Рекомендации	14
3.6. Модуль рекомендаций по скрытым признакам	15
4. Апробация	17
4.1. Демонстрация результатов	17
4.2. Метрики	21
Заключение	24
Список литературы	25

Введение

В современном мире процесс поиска вдохновения дизайнерами частично перенесен в онлайн пространство. Люди ищут картинки, способные подарить им новые идеи или позволить взглянуть на какую-то концепцию с другой стороны. Но проблема поиска подходящих изображений с каждым днем становится все серьезнее, ведь в медиа пространстве ежедневно появляются миллионы новых изображений. И хотя значительная их часть может быть полезна, стремительные темпы роста количества доступной информации уже не позволяют лично просматривать все картинки и выбирать подходящие, так как на это требуется слишком много времени. По сути, эта проблема раскладывается на две части:

- Обилие визуального мусора — картинок в плохом разрешении, дублей, изображений, не имеющих ценности в рамках конкретной задачи, вынуждающее дизайнера тратить значительно больше сил и времени на поиск подходящего контента.
- Растущее множество разнообразных визуальных стилей, количество которых уже на данный момент столь велико, что человек не способен в нем ориентироваться. Более того, их список постоянно пополняется, ведь в каждом новом изображении может содержаться новый визуальный стиль.

Важно понимать, что, говоря об изображениях, используемых дизайнерами для поиска вдохновения, мы подразумеваем довольно узкий круг изображений — презентации и примеры интерфейсов, шрифты, иллюстрации дизайн концепций и т.д.

Чтобы снизить количество и увеличить качество информации, пригодной для просмотра, люди перекладывают задачу фильтрации и отбора подходящих изображений на компьютеры. На данный момент на рынке представлены продукты, предоставляющие следующие возможности для поиска вдохновляющих изображений:

- Поиск по тегам или ключевым словам.

- Поиск по категориям.
- Поиск визуально похожих изображений.
- Выдача рекомендаций по картинке.

Мы выделяем последний пункт как самый значимый, ведь именно он представляет собой суть решения проблемы переизбытка контента. Идеально работающая рекомендательная система должна выдавать пользователю именно те изображения, которые он отобрал бы для себя, будь у него возможность посмотреть все.

Постановка задачи

Целью данной работы является создание системы рекомендаций и поиска для дизайнерского контента. Для достижения этой цели поставлены следующие задачи:

- Проанализировать типы рекомендательных систем.
- Реализовать модуль поиска изображений, похожих по доминантным цветам.
- Реализовать модуль поиска изображений, похожих по текстовой информации (автор, ключевые слова в описании, название стиля).
- Реализовать модуль поиска изображений, похожих по визуальному стилю.
- Спроектировать общую архитектуру системы и интегрировать модули друг с другом.
- Реализовать прототип системы и провести апробацию в проекте *graphica.ai* graphica.ai.

1. Обзор

1.1. Обзор конкурентов

Область дизайнерского вдохновения на данный момент еще не до конца сформирована. Сервисы, предлагающие в ней свои услуги сейчас, практически не решают проблему переизбытка контента:

- Google Images¹ дает возможность поиска по ключевым словам, однако большая часть предоставляемых изображений не имеет отношения к дизайну, поэтому поиск подходящих изображений затруднен.
- Tumblr² и Pinterest³ позволяют искать по категориям, ключевым словам, а так же предоставляют рекомендации. Однако эти сервисы являются скорее социальными сетями для дизайнеров, чем платформами для поиска вдохновения. Переизбыток визуального мусора и смещение акцентов в сторону общения делают их также не оптимальными для решения поставленной задачи.
- Savee⁴ предлагает поиск по ключевым словам и тегам, однако предоставляемая рекомендательная система не позволяет обследовать область, не задавая ее словами, что невозможно в общем случае, из-за растущего количества визуальных стилей.
- Behance⁵ и Dribbble⁶ не модерятся, поэтому значительная часть предлагаемых изображений — визуальный мусор, а большую часть выдачи рекомендательной системы составляют не эстетичные работы. Поиск вдохновения на этих ресурсах требует значительных усилий и большого количества времени. Соответственно, не решает поставленную задачу в полной мере.

¹google.com

²tumblr.com

³pinterest.com

⁴savee.it

⁵behance.net

⁶dribbble.com

1.2. Обзор предметной области

В современном мире существует множество подходов к созданию рекомендательных систем [1, 2].

Неперсонализированные рекомендации (Summary based)

Подход неперсонализированных рекомендаций — это расчет рекомендаций на основании глобального рейтинга. В качестве рейтинга может использоваться средняя оценка товара, отношение лайков к просмотрам, количество скачиваний, или другая метрика, специфичная для системы. Под товаром здесь и далее будем подразумевать любую сущность, с которой работает система рекомендаций. Его очевидная проблема — игнорирование личных предпочтений пользователя. Также серьезным недостатком этого подхода является консерватизм, новые товары с трудом попадают в топ рейтинга, так как не предлагаются вообще никаким пользователям, следовательно их оценка растет медленнее. Реализация этого подхода очень проста, для каждого товара хранится его рейтинг, обновляющийся с каждой новой оценкой. При отсутствии или низком числе оценок начальный рейтинг может задаваться владельцами сервиса.

Коллаборативная фильтрация (Collaborative filtering)

Подход коллаборативной фильтрации [3, 4, 5, 6] разделяется на два похожих подхода:

- Кластеризация пользователей.
- Кластеризация товаров.

Кластеризация пользователей представляет собой объединение пользователей в группы по предпочтениям, и выдачу рекомендаций на основе того, что из еще не увиденного конкретным пользователем контента понравилось другим членам группы, в которую он помещен. Обычно предпочтения пользователей хранятся с помощью матрицы векторов

оценок каждого товара для каждого пользователя. Для распределения по группам используется поиск ближайших соседей по этим векторам.

Кластеризация товаров отличается от кластеризации пользователей лишь тем, что матрица отношений транспонирована. И для рекомендаций, с помощью поиска n ближайших соседей, ищутся товары, ближайшие к понравившимся пользователю ранее.

При росте количества пользователей и товаров матрицы становятся разреженными, и соответственно растут затраты на их хранение и распределение по группам.

Также существует проблема “холодного старта” — ситуация, когда активных пользователей слишком мало, и большая часть распределяется в группу, не оценившую ничего. В таком случае, рекомендации на основе коллаборативной фильтрации не работают.

Для решения проблемы холодного старта часто используются персонализированные рекомендации или рекомендации по содержанию. Для решения проблемы разреженности матриц используются подход приближения матрицами меньшего ранга [7, 8]. Он позволяет сократить затраты на хранение ненужной информации, однако получившиеся матрицы неинтерпретируемы.

Рекомендации по содержанию (Content Based)

Подход рекомендаций по содержанию [9, 10, 11] заключается в выделении единых ключевых характеристик всех товаров, по которым их можно сравнивать векторизации этих признаков, и выдаче товаров, вектора которых максимально близки к векторам товаров, уже просмотренных пользователем, в качестве рекомендаций. Проблема этого подхода заключается в сложности выделения ключевых характеристик, а также в слабой корреляции с глобальными предпочтениями пользователя. Преимуществом этого подхода можно считать решение проблемы холодного старта и независимость качества рекомендаций от действий пользователя. Также для нас важно, что при использовании этого подхода мы можем сами задавать, какие факторы имеют значение, полагаясь на мнение экспертов, а не на машину.

Смешанный подход

Идея смешанного подхода [12, 13] заключается в комбинации подходов, рассмотренных выше. Популярны следующие виды комбинаций:

- Использование нескольких подходов параллельно, присвоение каждой рекомендации некоторого веса и вывод товаров с максимальным весом.
- Передача вывода одного подхода на вход другому.
- Использование разных подходов для разных товаров.

Плюсом этого подхода является возможность работать с большим количеством разноплановой информации, получая таким образом возможность увеличить качество рекомендаций. Минусом же является сложность реализации и повышенные затраты на поддержку всей информации, необходимой для корректной работы нескольких подходов.

В этой работе мы рассмотрим смешанный подход к созданию рекомендательных систем подробнее, и выделим оптимальную комбинацию для решения поставленной задачи.

2. Описание подхода

2.1. Условия решаемой задачи

Пользователь заходит на сайт, чтобы найти вдохновение для своего нового проекта. Задача сервиса — упростить и ускорить этот процесс. Предлагаемый подход заключается в предоставлении рекомендательной системы для помощи поиска изображений, похожих на интересные, и поисковой системы для поиска изображений по текстовому описанию и цветовой гамме.

2.2. Рекомендательная система

Задача рекомендательной системы — предоставлять изображения, наиболее подходящие к данному. Предлагаемый подход заключается в формализации понятия “подходящее изображение” как изображения:

1. похожего по доминантным цветам;
2. выполненного в том же визуальном стиле;
3. использующего тот же шрифт;
4. созданного тем же автором;
5. относящегося к той же сфере,

и создании системы, работающей на основе указанных понятий. Характеристики пользователя в данной рекомендательной системе не учитываются.

Для решения этой задачи предлагается реализовать два модуля:

1. Модуль рекомендаций по цвету.
2. Модуль рекомендаций по доступной текстовой информации, включающей перечисленные выше понятия.

И интегрировать их друг с другом, реализуя смешанный подход к созданию рекомендательных систем.

2.3. Поиск

Задача поисковой системы — предоставлять изображения, подходящие под пользовательский запрос. Предлагаемый подход заключается в разделении пользовательского запроса на цвета и текстовое описание, состоящее из следующих понятий:

1. Имя автора.
2. Используемый шрифт.
3. Используемый стиль.
4. Тип работы.
5. Ключевые слова, описывающие работу.

Ответом на запрос предлагается считать пересечение изображений, содержащих данные цвета в качестве доминантных, и изображений, удовлетворяющих текстовому описанию. Для решения этой задачи реализуем модули поиска по цветам и по тексту и интегрируем их друг с другом, получив модуль поиска по визуальному стилю.

3. Реализация

3.1. Архитектура

Архитектура разработанной системы состоит из двух модулей — модуля рекомендаций, который состоит из модулей рекомендаций по цвету и рекомендаций по текстовой информации, и модуля поиска, который состоит из модулей поиска по текстовой информации и по цветам. Все модули получают запросы от клиента с помощью системы GraphQL⁷ и обращаются в базу данных PostgreSQL⁸, как указано на рисунке 1.

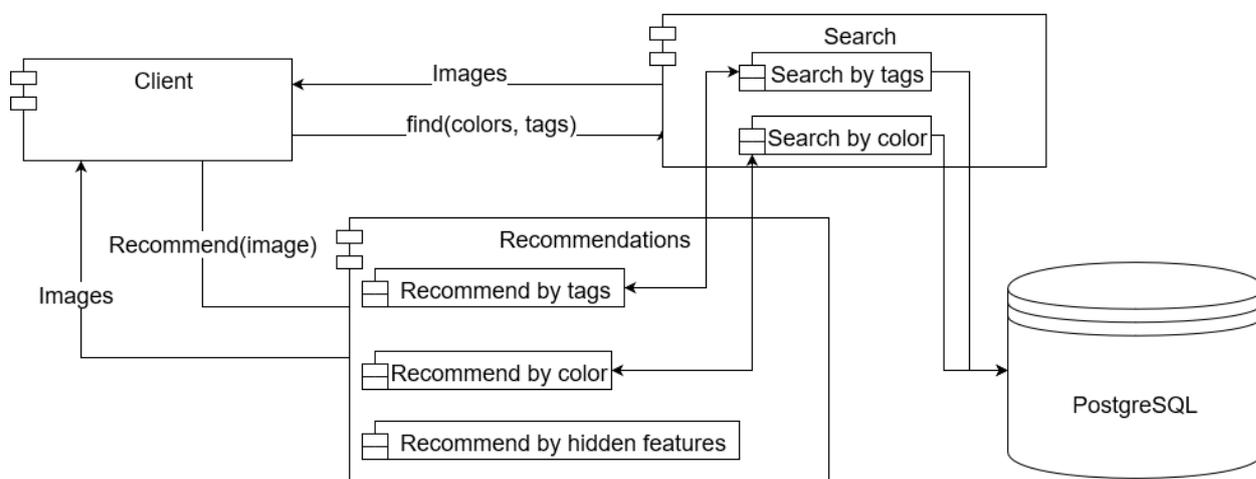


Рис. 1: Архитектура системы.

3.2. Модуль поиска по цвету

Работа модуля поиска по цвету логически разбивается на несколько этапов:

1. Вычисление доминантных цветов всех изображений.
2. Создание и сохранение структуры, предоставляющей поиск.
3. Обработка поискового запроса и выдача результатов.

⁷graphql.org

⁸www.postgresql.org

Вычисление доминантных цветов

Для вычисления доминантных цветов используются алгоритмы кластеризации из библиотеки `scikit-learn`⁹. Были опробованы алгоритмы `k-means`¹⁰ и `meanshift`¹¹. Приглашенные эксперты дизайнеры выделили алгоритм `meanshift` как более точный, поэтому мы выбрали его.

Выделенные доминантные цвета переводятся в цветовое пространство LAB¹². Это цветовое пространство было выбрано, так как расстояния между цветами в нем наиболее близки к человеческому восприятию[14]. Были реализованы системы, работающие с цветовыми пространствами RGB и HSL, и система, работающая с цветами, закодированными в LAB, была выбрана экспертами дизайнерами как более точная.

Полученные цвета сохраняются в базу данных.

Создание и хранение структуры, предоставляющей поиск

Для каждого изображения доминантные цвета объединяются в одномерный вектор, содержащий доминантные цвета в порядке увеличения процента площади, которую они занимают на изображении.

Получившийся вектор балансируется экспериментально подобранными весами для увеличения точности поиска. Затем вектора добавляются в единый индекс, предоставляемый библиотекой `annoy`¹³, который сохраняется на диск. Эта задача выполняется в отдельном потоке раз в сутки, чтобы поддерживать актуальность рекомендаций, не занимая ресурсы основного сервера.

Обработка поискового запроса и выдача результатов

Поисковый запрос состоит из одного или нескольких цветов, по которым требуется найти подходящие изображения. По этим цветам стро-

⁹scikit-learn.org

¹⁰scikit-learn.org/stable/modules/generated/sklearn.cluster.k_means.html?highlight=kmeans#sklearn.cluster.k_means

¹¹scikit-learn.org/stable/modules/generated/sklearn.cluster.MeanShift.html?highlight=meanshift#sklearn.cluster.MeanShift

¹²en.wikipedia.org/wiki/CIELAB_color_space

¹³<https://github.com/spotify/annoy>

ится вектор того же размера и формата, что и сохраненные в индексе. Для этого вектора с помощью метода Approximate Nearest Neighbors, предоставляемого библиотекой annou, в сохраненном индексе ищется требуемое количество наиболее похожих векторов. Изображения, вектора которых были выбраны, возвращаются в качестве ответа. Так же в поисковом запросе могут быть переданы изображения, на которых следуют осуществлять поиск. В таком случае возвращается пересечения переданных изображений с найденными ближайшими векторами.

Библиотека annou была выбрана на основе исследования [15], так как она оптимальна с точки зрения времени построения индекса и скорости поиска по нему. Были также опробованы библиотеки faiss¹⁴, flann¹⁵ и NGT¹⁶, но они, как и указано в исследовании, показали худшие результаты для нашей задачи.

3.3. Модуль поиска по текстовой информации

Новые изображения мы либо собираем из открытых источников, либо принимаем от пользователей. Для каждого изображения мы дополнительно сохраняем доступную текстовую информацию, окружающую его, разделяя ее на соответствующие категории:

1. Имя автора.
2. Описание работы.
3. Используемый шрифт.
4. Используемый стиль.
5. Тип работы.
6. Дополнительные теги.

¹⁴github.com/facebookresearch/faiss

¹⁵github.com/mariusmuja/flann

¹⁶github.com/yahoojapan/NGT

Поисковый запрос к этому модулю состоит из набора ключевых слов и фраз. Для ответа на запрос используется движок полнотекстового поиска¹⁷, встроенный в используемую нами базу данных PostgreSQL¹⁸, с его помощью мы находим вхождения фраз и слов из запроса в перечисленных выше категориях. Вывод сортируется по точности совпадения, и запрошенное количество лучших возвращается в качестве ответа на запрос. Все запросы сохраняются и переиспользуются для повышения скорости работы системы. В поисковом запросе могут быть переданы изображения, на которых следует осуществлять поиск. В таком случае создается комплексный запрос, первая часть которого отбирает в базе данных переданные изображения, а вторая осуществляет полнотекстовой поиск и сортировку уже только на них.

3.4. Модуль комбинированного поиска

Модуль комбинированного поиска объединяет в себе два описанных выше модуля и является модулем поиска по визуальному стилю. Поисковый запрос к нему состоит из нескольких цветов, ключевых слов и фраз. Часть запроса, содержащая цвета, передается на вход модулю поиска по цвету. Часть запроса, содержащая ключевые слова и фразы, передается на вход модулю поиска по тексту, так же в него передаются изображения, выданные модулем поиска по цвету, и поиск осуществляется только на них. Этот модуль основывается на идее смешанного подхода к созданию рекомендательных систем.

3.5. Рекомендации

Модули рекомендательной системы являются обертками над модулями поисковой системы, предоставляющими удобный интерфейс.

¹⁷www.postgresql.org/docs/current/textsearch-psql.html

¹⁸www.postgresql.org

Модуль рекомендации по цвету

Запрос к модулю рекомендаций по цвету содержит идентификатор изображения. Из базы данных достаются доминантные цвета этого изображения и передаются на вход модулю поиска по цвету.

Модуль рекомендаций по тексту

Запрос к модулю рекомендаций по тексту содержит идентификатор изображения. Из базы данных достаются:

1. Имя автора.
2. Дополнительные теги.
3. Название шрифта.
4. Название стиля.
5. Тип работы.

и передаются на вход модулю поиска по цвету.

Модуль комбинированных рекомендаций

Запрос к модулю комбинированных рекомендаций содержит идентификатор изображения. Сначала запускается модуль рекомендаций по цвету, а его вывод передается в модуль рекомендаций по тексту. Этот модуль является модулем рекомендаций по визуальному стилю и, так же как модуль поиска по визуальному стилю, реализует смешанный подход к созданию рекомендательных систем.

3.6. Модуль рекомендаций по скрытым признакам

Кроме описанных выше модулей поиска и рекомендаций был разработан дополнительный модуль рекомендаций по скрытым признакам.

Он использует нейронную сеть NasNet¹⁹ из библиотеки Keras²⁰, обученную на данных aesthetic dataset²¹ из исследования[14]. NasNet была выбрана из-за ее структуры, это глубокая нейронная сеть, которая широко используется для выделения скрытых признаков из изображений.

Обученная нейронная сеть получает на вход изображение и выдает вектор из четырех чисел, содержащий скрытые признаки. Среднее время работы для одного изображения в нашей базе данных — 1 секунда. Все вектора добавляются в индекс и сохраняются на диск. Выдача рекомендаций происходит аналогично описанному ранее подходу с использованием библиотеки annou.

¹⁹www.tensorflow.org/api_docs/python/tf/keras/applications/nasnet

²⁰www.tensorflow.org/guide/keras

²¹ritendra.weebly.com/aesthetics-datasets.html

4. Апробация

Все описанные модули были интегрированы в проект “graphica.ai”. Они поддерживают запросы от клиента в формате, используемом в проекте, и работают с информацией из базы данных, существующей в проекте.

4.1. Демонстрация результатов

Для демонстрации работы модулей были выбраны изображения из базы данных “graphica.ai”, содержащие различные работы дизайнеров.

Рекомендации по цвету



Рис. 2: Исходное изображение

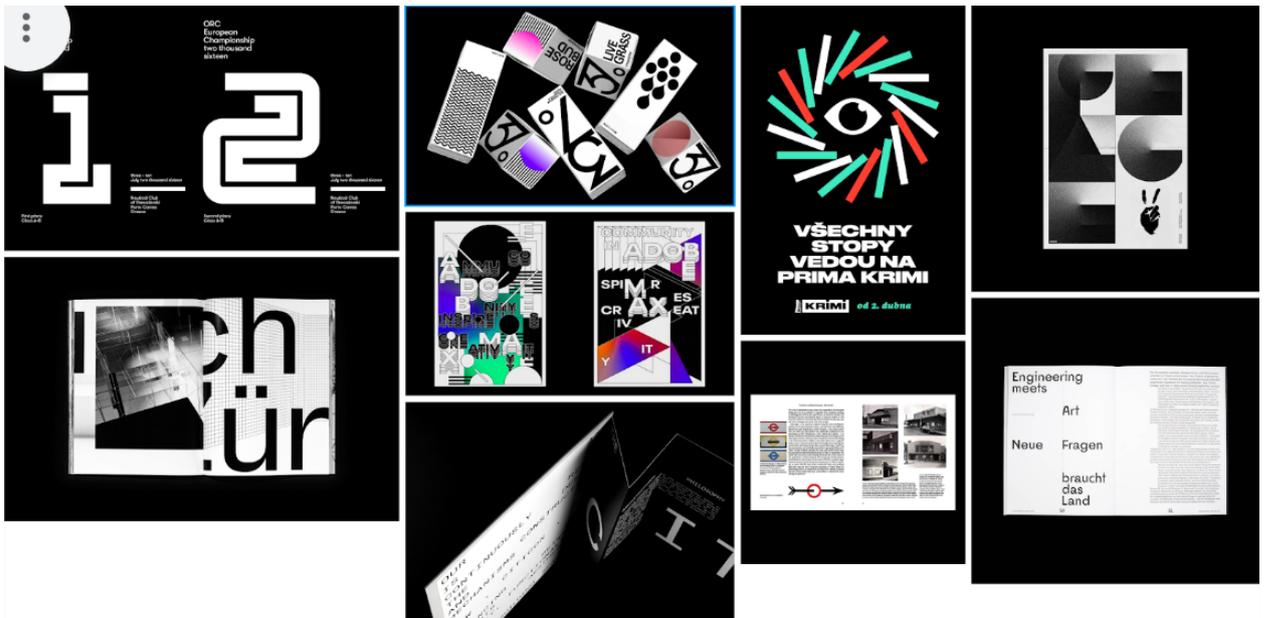


Рис. 3: Результат работы модуля рекомендаций по цвету для изображения (2)

Поиск по цвету

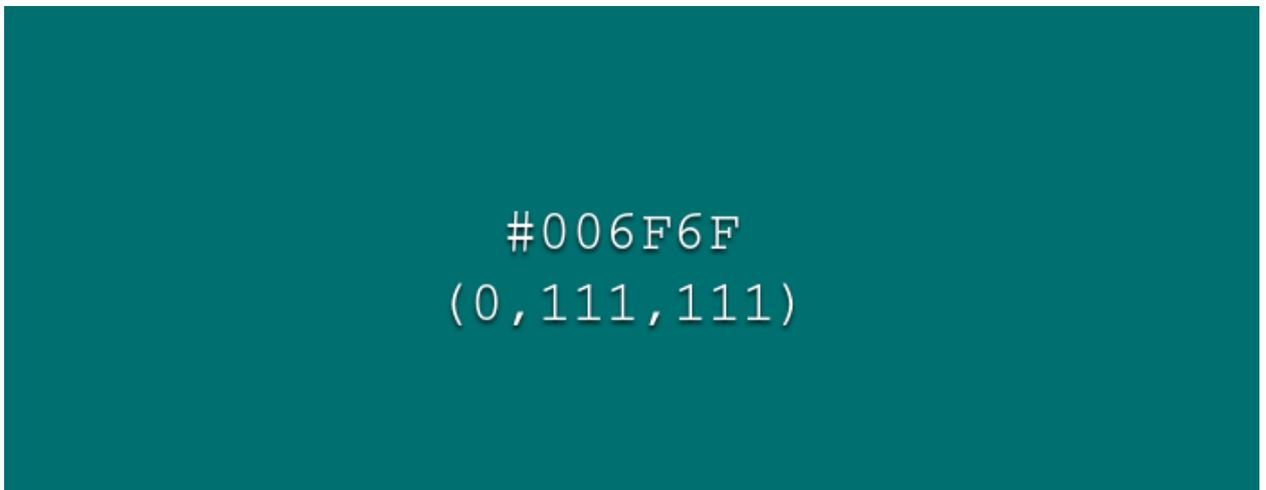


Рис. 4: Цвет 006F6F

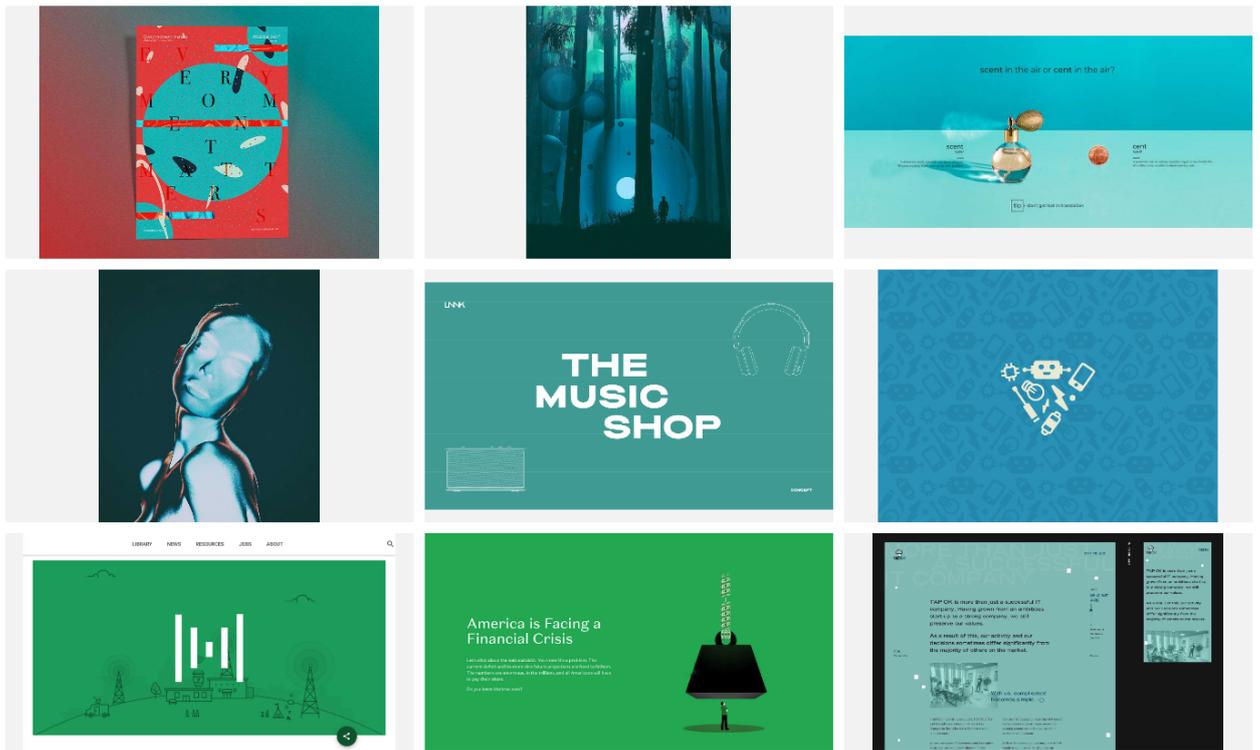


Рис. 5: Поиск по цвету 006F6F (4)

Поиск по тексту

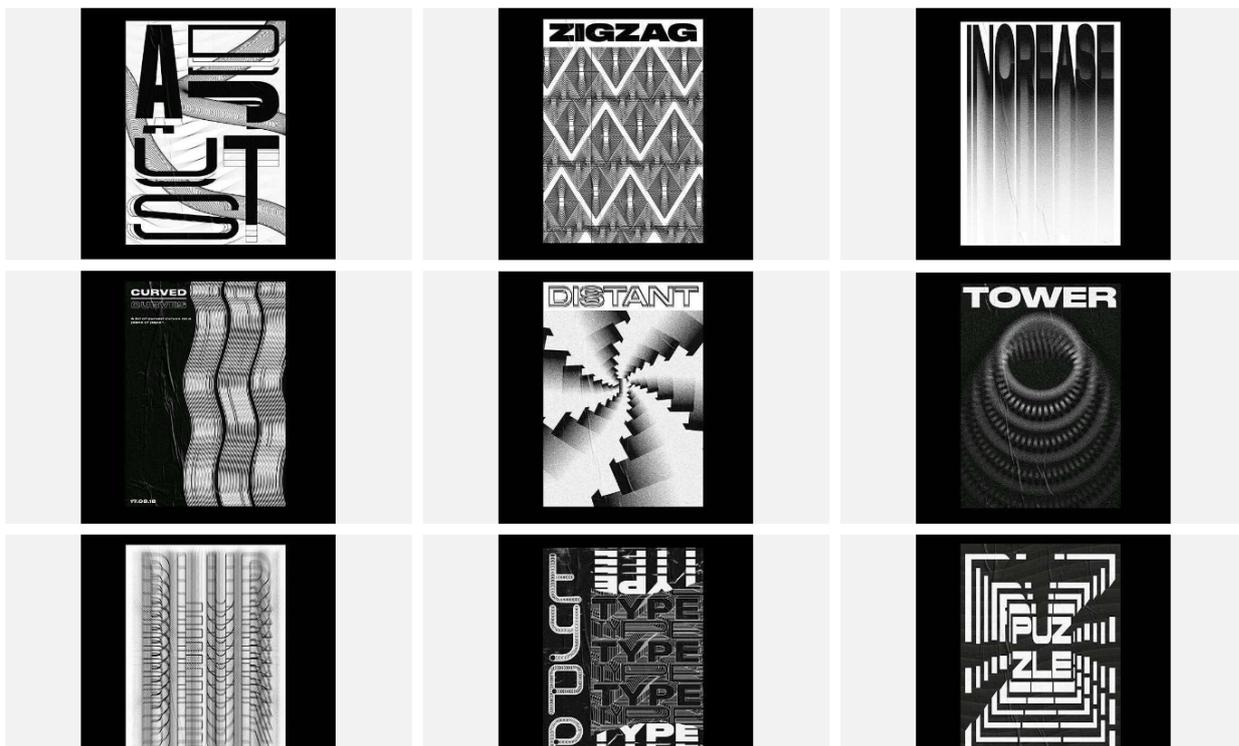


Рис. 6: Результат работы модуля поиска для запроса с ключевыми словами “poster” и “letters”

Комбинированный поиск

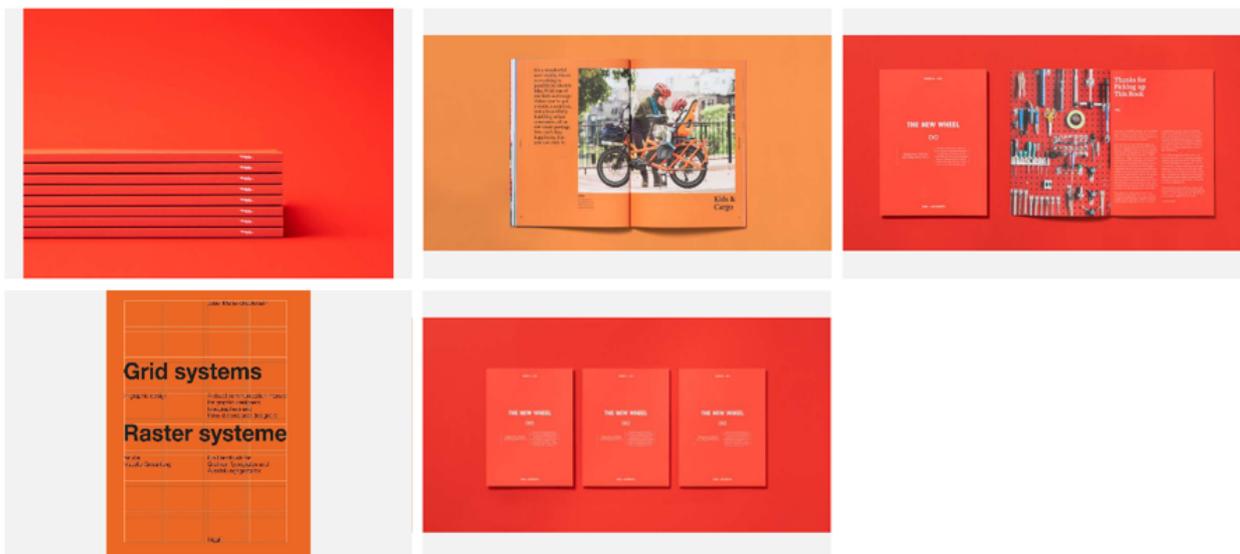


Рис. 7: Результат работы модуля поиска для запроса с красным цветом и ключевым словом “book”

4.2. Метрики

Все реализованные модули были интегрированы в проект “graphica.ai”. Для оценки проведенной работы мы использовали два типа метрик:

1. Время работы.
2. Удовлетворение пользователей.

Время работы

Рассмотрим таблицу, демонстрирующую среднее время ответа модулей на запросы. Каждому модулю было передано 1000 случайных запросов, измерялось время работы модулей, время передачи запроса и ответа не учитывалось. Тесты проводились на основном сервере проекта “graphica.ai”, на котором модули и будут использоваться.

	Поиск	Рекомендации
По цвету	0.005s	0.01s
По тексту	0.001s	0.01s
Комбинированный модуль	0.008s	0.03s

Рассмотрим таблицу, показывающую среднее время работы подготовительных шагов для поиска и рекомендаций, описанных выше.

Вычисление доминантных цветов	Создание индекса
10s	50s

Удовлетворение пользователей

Было проведено исследование в форме опроса, участие в котором приняло 60 человек. Мы предложили им сравнить выдачу двух рекомендательных систем — предыдущей системы “graphica.ai” и системы, разработанной в рамках курсовой работы. Опрос показал, что рекомендательная система, разработанная в рамках курсовой работы, нравится 80% пользователей сильнее (10).

Какой набор лучше?

60 ответов

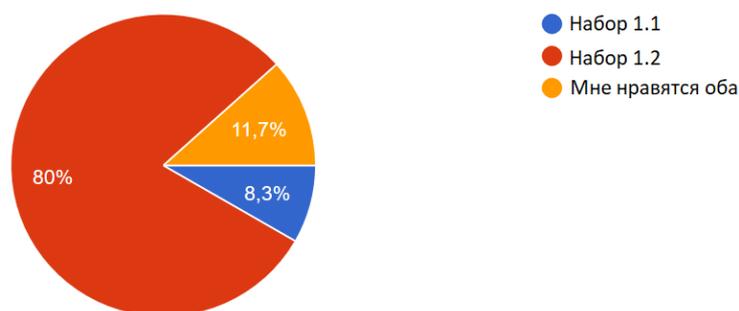


Рис. 8: Результаты сравнения двух рекомендательных систем

Так же мы спросили у пользователей, какие понятия они считают ключевыми для рекомендаций, и удостоверились, что они являются основными в нашей системе.

Что вы думаете о запросе, содержащем цвета и ключевые слова, представленном на картинке выше?



60 ответов



Рис. 9: Основные понятия схожести

Дополнительно мы выяснили, какие виды ключевых слов пользователи используют чаще всего, и проверили, что самые популярные доступны в нашей системе.

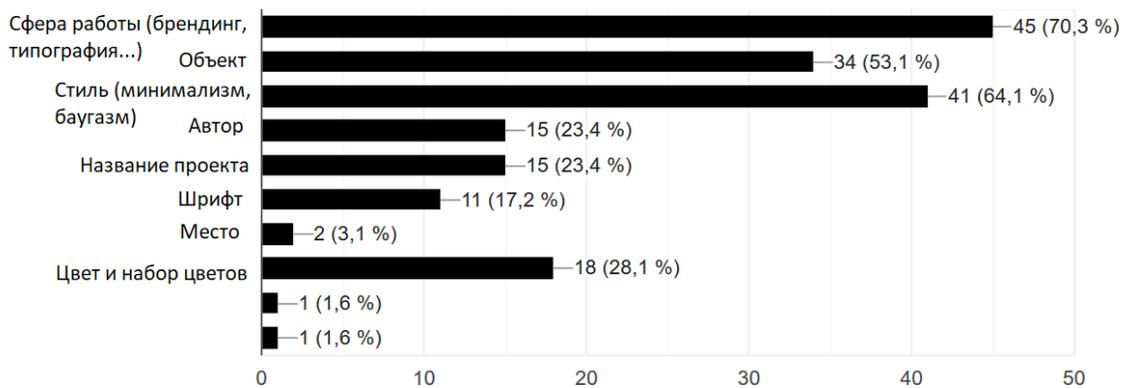


Рис. 10: Типы ключевых слов

Заключение

1. Произведен подробный обзор подходов к созданию рекомендательных систем.
2. Реализован и протестирован в проекте *graphica.ai* модуль поиска изображений, похожих по доминантным цветам.
3. Реализован и протестирован в проекте *graphica.ai* модуль поиска изображений, похожих по текстовой информации.
4. Реализован и протестирован в проекте *graphica.ai* модуль рекомендательной системы.
5. Реализованы и протестированы в проекте *graphica.ai* модули поиска и рекомендаций по визуальному стилю.
6. Реализован и протестирован в проекте *graphica.ai* дополнительный модуль рекомендаций по скрытым признакам.

Все цели курсовой работы выполнены. Поисковая и рекомендательные системы реализованы, апробированы и протестированы.

Список литературы

- [1] Liang Zhong Cui, Fu Liang Guo, and Ying Jie Liang. Research overview of educational recommender systems. In *ACM International Conference Proceeding Series*. Association for Computing Machinery, oct 2018.
- [2] S. Sarika and Sanjeev Dhawan. Comparison of Recommendation System Approaches. In *Proceedings of the International Conference on Machine Learning, Big Data, Cloud and Parallel Computing: Trends, Perspectives and Prospects, COMITCon 2019*, pages 76–78. Institute of Electrical and Electronics Engineers Inc., feb 2019.
- [3] Xiaowei Xu and Fudong Wang. Trust -based collaborative filtering algorithm. In *Proceedings - 2012 5th International Symposium on Computational Intelligence and Design, ISCID 2012*, volume 1, pages 321–324, 2012.
- [4] Sabanaz Sirajuddin Peerzade. Web service recommendation using PCC based collaborative filtering. In *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing, ICECDS 2017*, pages 2920–2924. Institute of Electrical and Electronics Engineers Inc., jun 2018.
- [5] Gaojun Liu and Xingyu Wu. Using collaborative filtering algorithms combined with Doc2Vec for movie recommendation. In *Proceedings of 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference, ITNEC 2019*, pages 1461–1464. Institute of Electrical and Electronics Engineers Inc., mar 2019.
- [6] Youngki Park, Sungchan Park, Sang Goo Lee, and Woosung Jung. Fast Collaborative Filtering with a k-nearest neighbor graph. In *2014 International Conference on Big Data and Smart Computing, BIGCOMP 2014*, pages 92–95. IEEE Computer Society, 2014.
- [7] Markus Weimer, Alexandros Karatzoglou, and Marcel Bruch.

- Maximum margin matrix factorization for code recommendation. In *RecSys'09 - Proceedings of the 3rd ACM Conference on Recommender Systems*, pages 309–312, 2009.
- [8] Thanh Tran and Kyumin Lee. Regularizing matrix factorization with user and item embeddings for recommendation. *International Conference on Information and Knowledge Management, Proceedings*, pages 687–696, 2018.
- [9] Usman Saeed. Comparative analysis of lip features for person identification. In *Proceedings of the 8th International Conference on Frontiers of Information Technology, FIT'10*, 2010.
- [10] Leandro Figueira Lessa and Wladimir Cardoso Brandão. Filtering graduate courses based on LinkedIn profiles. In *WebMedia 2018 - Proceedings of the 24th Brazilian Symposium on Multimedia and the Web*, pages 141–147. Association for Computing Machinery, Inc, oct 2018.
- [11] Tudor Barbu. Content-based image retrieval using Gabor filtering. In *Proceedings - International Workshop on Database and Expert Systems Applications, DEXA*, pages 236–240, 2009.
- [12] Ashish Pal, Prateek Parhi, and Manuj Aggarwal. An improved content based collaborative filtering algorithm for movie recommendations. In *2017 10th International Conference on Contemporary Computing, IC3 2017*, volume 2018-January, pages 1–3. Institute of Electrical and Electronics Engineers Inc., feb 2018.
- [13] Carlos A. Gomez-Uribe and Neil Hunt. The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems*, 6(4), dec 2015.
- [14] Ritendra Datta, Jia Li, and James Z. Wang. Algorithmic inferencing of aesthetics and emotion in natural images: An exposition. *Proceedings - International Conference on Image Processing, ICIP*, pages 105–108, 2008.

- [15] Martin Aumüller, Erik Bernhardsson, and Alexander Faithfull. Ann-benchmarks: A benchmarking tool for approximate nearest neighbor algorithms, 2018.