Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ”

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

РАЗРАБОТКА СИСТЕМЫ АВТОМАТИЧЕСКОЙ КЛАССИФИКАЦИИ ИЗОБРАЖЕНИЙ ФОТОБАНКОВ «SHUTTERSTOCK» И «FOTOLIA» ПО КЛЮЧЕВЫМ СЛОВАМ

Автор Деменчук Влада Анатольевна _____
(Фамилия, Имя, Отчество) (Подпись)

Направление подготовки (специальность) _____

09.04.04 – Программная инженерия
(код, наименование)

Квалификация магистр
(бакалавр, магистр)

Руководитель Радченко И.А., доцент, к.т.н. _____
(Фамилия, И., О., ученое звание, степень) (Подпись)

К защите допустить

Зав. кафедрой Муромцев Д.И., доцент, к.т.н. _____
(Фамилия, И., О., ученое звание, степень) (Подпись)

“____” _____ 20 ____ г.

Санкт-Петербург, 2018 г.

Студент Деменчук В.А. Группа Р4217 Кафедра ИПМ Факультет ПИ и КТ
(Фамилия, И, О.)

Направленность (профиль), специализация

09.04.04 – Разработка программно-информационных систем

Консультант (ы):

а) _____ _____
(Фамилия, И., О., ученое звание, степень) (Подпись)

б) _____ _____
(Фамилия, И., О., ученое звание, степень) (Подпись)

ВКР принята “ _____ ” _____ 20 _____ г.

Оригинальность ВКР _____ %

ВКР выполнена с оценкой _____

Дата защиты “ _____ ” _____ 20 _____ г.

Секретарь ГЭК _____ _____
(Фамилия, И., О.) (Подпись)

Листов хранения _____

Демонстрационных материалов / чертежей хранения _____

Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ”

УТВЕРЖДАЮ

Зав. кафедрой_____

_____

(ФИО) (подпись)

«____» «_____» 20___ г.

З А Д А Н И Е
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

Студенту Деменчук В.А. **Группа** P4217 **Кафедра** ИПМ **Факультет** ПИ и КТ

Руководитель Радченко И.А., доцент, к.т.н., кафедра ИПМ, доцент

(ФИО, ученое звание, степень, место работы, должность)

1 Наименование темы:

Разработка системы автоматической классификации изображений фотобанков «Shutterstock» и «Fotolia» по ключевым словам

Направление подготовки (специальность) 09.04.04 – Программная инженерия

Направленность (профиль) 09.04.04 – Разработка программно-информационных систем

Квалификация магистр

2 Срок сдачи студентом законченной работы « _____ » « _____ » 20____ г.

3 Техническое задание и исходные данные к работе

Необходимо разработать систему автоматической классификации изображений фотобанков «Shutterstock» и «Fotolia» по категориям на основе ключевых слов, которая осуществляет сбор, хранение, обработку, классификацию данных изображений, а также автоматический выбор категорий для загруженных автором изображений в веб-интерфейсах фотобанков.

4 Содержание выпускной работы (перечень подлежащих разработке вопросов)

Обзор предметной области. Постановка задачи. Обзор основных этапов классификации. Проведение экспериментов для выбора алгоритма классификации. Реализация системы.

5 Перечень графического материала (с указанием обязательного материала)

Презентация в электронном виде.

6 Исходные материалы и пособия

1. Silla Jr, Carlos N. and Freitas, Alex A. (2011) A survey of hierarchical classification across different application domains // Data Mining and Knowledge Discovery – January 2011, Volume 22, Issue 1–2, pp 31–72
2. Andreas C Müller; Sarah Guido, Introduction to machine learning with Python: a guide for data scientists – Sebastopol, CA: O'Reilly Media, Inc, 2017
3. Learn Extension Basics [Электронный ресурс] // Chrome. – Режим доступа: https://developer.chrome.com/extensions/getstarted
4. Supervised learning [Электронный ресурс] // Scikit-learn. – Режим доступа: http://scikit-learn.org/stable/supervised_learning.html

7 Дата выдачи задания « _____ » « _____ » 20 _____ г.

Руководитель ВКР _____
(подпись)

Задание принял к исполнению _____ « _____ » « _____ » 20 _____ г.
(подпись)

Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ”

АННОТАЦИЯ

ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

Студент Деменчук Влада Анатольевна
(ФИО)

Наименование темы ВКР: Разработка системы автоматической классификации изображений фотобанков «Shutterstock» и «Fotolia» по ключевым словам

Наименование организации, где выполнена ВКР Университет ИТМО

ХАРАКТЕРИСТИКА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

1 Цель исследования: разработать систему автоматической классификации изображений по категориям на основе ключевых слов

2 Задачи, решаемые в ВКР: Обзор предметной области. Выбор фотобанков. Обзор основных этапов классификации. Проведение экспериментов и выбор алгоритма классификации. Реализация системы.

3 Число источников, использованных при составлении обзора: 11

4 Полное число источников, использованных в работе: 23

5 В том числе источников по годам:

Отечественных			**Иностранных**		
Последние 5 лет	От 5 до 10 лет	Более 10 лет	Последние 5 лет	От 5 до 10 лет	Более 10 лет
			6	1	

6 Использование информационных ресурсов Internet да, 18 ссылок
(Да/нет, число ссылок в списке литературы)

7 Использование современных пакетов компьютерных программ и технологий
(Указать, какие именно, и в каком разделе работы)

Пакеты компьютерных программ и технологий	**Параграф работы**
Java, Spring Boot, PostgreSQL,	Глава 3.1
Python 3, scikit-learn, Pandas, NumPy	Глава 3.2
Flask	Глава 3.3
HTML, CSS, JavaScript, jQuery	Глава 3.4

8. Краткая характеристика полученных результатов Выполнен обзор предметной области, выбраны два фотобанка «Shutterstock» и «Fotolia», проведен обзор основных этапов классификации, проведены эксперименты и выбран алгоритм классификации, реализована система.

9. Полученные гранты, при выполнении работы нет
(Название гранта)

10. Наличие публикаций и выступлений на конференциях по теме выпускной работы да
(Да, нет)

а) 1 Деменчук В.А. Проблема программного взаимодействия со сторонними веб-интерфейсами и ее решение с помощью создания расширения для Google Chrome // Альманах научных работ молодых ученых Университета ИТМО. СПб.: Университет ИТМО, 2018 (в печати).
2 Деменчук В.А. Классификация изображений микростока Shutterstock по ключевым словам // Сборник тезисов докладов конгресса молодых ученых. Электронное издание. – СПб: Университет ИТМО, 2018.

б) 1 Деменчук В.А. Доклад на тему «Проблема программного взаимодействия со сторонними веб-интерфейсами
(Библиографическое описание выступлений на конференциях)
и ее решение с помощью создания расширения для Google Chrome» на XLVII научной и учебно-методической конференции Университета ИТМО
2 Деменчук В.А. Доклад на тему «Классификация изображений микростока Shutterstock по ключевым словам» на VII Конгрессе молодых ученых

Выпускник Деменчук В.А. _____
(ФИО) (подпись)

Руководитель Радченко И. А. _____
(ФИО) (подпись)

«_____» _____ 2018 г.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Фотобанк – онлайн поставщик (торговая площадка, двусторонний рынок) качественных лицензионных изображений. С помощью фотобанков в настоящее время дизайнеры, фотографы и иллюстраторы продают свои работы большой аудитории, которая в свою очередь может использовать купленные изображения в личных и коммерческих целях.

Перед тем, как изображения будут доступны для продажи авторам необходимо подготовить свои работы, то есть загрузить их, и для каждого изображения внести описание, ключевые слова (теги), выбрать категории и другие параметры. Пропорционально росту числа загружаемых изображений и количеству фотобанков, с которыми работает автор, увеличивается время, затрачиваемое на заполнение требуемых данных.

Описание и ключевые слова вносятся автором в метаданные изображений один раз и автоматически извлекаются фотобанками во время загрузки. Категорию возможно указать только с помощью веб-интерфейса из списка, уникального для каждого фотобанка.

Целью данной работы является разработка системы автоматической классификации изображений по категориям на основе ключевых слов.

Для достижения поставленной цели необходимо решить следующие задачи:

- Обзор предметной области.
- Выбор фотобанков.
- Обзор основных этапов классификации.
- Реализация системы.
  - Разработка модуля сбора и хранения данных изображений.
  - Разработка модуля обработки данных изображений и построения моделей.

- Проведение экспериментов для выбора алгоритма классификации.
- Разработка модуля классификации изображений.
- Разработка модуля автоматического выбора категории в веб-интерфейсах фотобанков.

1 Предметная область и постановка задачи

1.1 Основные понятия

Фотобанк – это сервис, который предоставляет богатую коллекцию изображений для покупки. Каждое изображение содержит ключевые слова, которые точно его описывают и необходимы для осуществления поиска по введенному пользователем запросу. Ключевые слова отображают не только объекты, присутствующие на изображении, но и смысл, идею место, действия на изображении. Также каждое изображение имеет одну или несколько категорий, что позволяет искать изображения в конкретных областях.

Интеллектуальный анализ данных (Data Mining) – это процесс обнаружения в больших объемах данных скрытых закономерностей [1].

Машинное обучение – раздел искусственного интеллекта, который фокусируется на методах построения алгоритмов, которые способны обучаться.

Классификация является одной из основных задачей Data Mining [1]. Ее целью является отнесение объекта к одному из заранее известных классов. Задача классификации относится к обучению с учителем.

Обучение с учителем подразумевает обучение модели на размеченных данных, которые представляют собой объекты и соответствующие им классы. Обучение с учителем подразумевает деление исходных данных на обучающую и тестовую выборки. Соответственно обучающая выборка используется для обучения модели, тестовая – для тестирования обученной модели с целью оценки ее качества.

Модель – это результат обучения классификатора.

Под классификатором подразумевается алгоритм машинного обучения.

1.2 Выбор фотобанка

Основным и единственным объективным критерием выбора фотобанков является размер коллекции изображений, что обуславливает большое количество пользователей. В таблице 1 представлены фотобанки в порядке убывания размеров коллекции изображений.

Таблица 1 – Размеры коллекций изображений фотобанков

Фотобанк	**Размер коллекции изображений**
Shutterstock	более 199 млн. [2]
Fotolia	более 121 млн. [3]
Dreamstime	более 79 млн. [4]
Deposit Photos	более 75 млн. [5]
Big Stock Photo	более 66 млн. [6]
123RF	более 65 млн. [7]
Phototimes	более 55 млн. [8]

В рамках данной работы были выбраны фотобанки «Shutterstock» и «Fotolia». Они являются самыми популярными, а также лидерами по размеру коллекции изображений. Размер их коллекций составляет более 199 млн. и 121 млн. изображений соответственно.

Shutterstock на данный момент предоставляет возможность выбрать для каждого изображения только одну категорию из 26. У Fotolia доступны 1097 категорий, которые представляют собой иерархию с максимальной глубиной равной 3. Для каждого изображения есть возможность выбрать одну категорию с одной или двумя подкатегориями.

1.3 Постановка задачи

Формальная постановка задачи классификации

Дано:

X– множество объектов

Y– множество классов

$y: X \to \mathrm{Y}$ − неизвестная целевая функция, значения которой известны только на объектах конечной обучающей выборки $X^m \to \{(x_{1,}y_{1,}) \dots, (x_{m,}y_{m,})\}$.

Необходимо:

Построить алгоритм a: $X \to \mathrm{Y}$, который способен классифицировать произвольный объект на всем множестве X.

В нашем случае объектами являются изображения, классами – категории, признаками – ключевые слова изображений.

Так как международным языком является английский, изображения на фотобанках «Shutterstock» и «Fotolia» описываются ключевыми словами на английском языке.

Данная задача сводится к многоклассовой классификации, потому что количество категорий, на которые будет производиться классификация изображений Shutterstock и Fotolia больше двух.

С учетом иерархичности категорий Fotolia, задача классификации для данного фотобанка является не только многоклассовой, но и иерархической. Она состоит в том, чтобы определить для объекта один или несколько классов в иерархии. При этом классы являются непересекающимися, так объект может одновременно относиться только к одному классу или к одной ветви иерархии классов.

Для того, чтобы решить задачу многоклассовой классификации, можно свести ее к решению нескольких задач бинарной классификации. Большинство методов уже

основаны на бинарных классификаторах или их можно к ним свести. Рассмотрим два подхода сведения многоклассовой классификации к бинарной: «One-vs-all» и «One-vs-One».

Целью подхода «One-vs-all» («One-vs-Rest») является построение бинарных классификаторов, количество которых равно количеству классов. Построенные классификаторы будут отделять каждый класс от остальных [9]. В качестве результата классификации, будет получен тот класс, которому соответствует наиболее высокая оценка принадлежности объекта к одному из классов, вычисленная алгоритмом.

Подход «All-vs-all» предполагает построение бинарных классификаторов для каждой пары классов [9]. Количество классификаторов будет равно квадрату количества классов. Для каждого класса классификатором вычисляется суммарная оценка принадлежности к нему и в качестве результата будет возвращена максимальная.

Для решения задачи иерархической классификации также существует несколько подходов: «плоский» («flat»), «локальный» («local») и «глобальный» («global»).

«Плоский» подход предполагает, что иерархическая классификация может быть преобразована в плоскую без учета информации о связях классов предок-потомок, которая присутствует в иерархии классов [11]. Т.е. в нем полностью игнорируется иерархия классов, и предсказывается принадлежность объекта только к листовому классу. При этом объекту будут присвоены все родительские классы предсказанного листового. Недостаток данного подхода заключается в необходимости создания классификатора для распознавания большого количества классов, а именно всех листовых.

«Локальный» подход предполагает последовательное предсказание сначала одного из классов первого уровня иерархии, затем одного из дочерних классов,

соответствующих предсказанному родительскому классу, и так далее. Процесс повторяется до тех пор, пока не будет достигнут листовой класс.

Существует три способа использования локального классификатора:

- **Для каждого узла иерархии**: используются бинарные независимые классификаторы для предсказания каждого класса иерархии. У данного подхода есть несколько проблем. Первая из них - количество обучаемых классификаторов растет пропорционально количеству классов в иерархии. Вторая – результаты могут быть непоследовательными, ввиду отсутствии гарантии соблюдения иерархии классов. [11]
- **Для каждого родительского узла**. Этот подход состоит в обучении каждого родительского узла иерархии классов как многоклассового классификатора. Преимуществом данного способа является ограниченность классов на заданном уровне по принципу соответствия предсказанному родителю на предыдущем уровне.
- **Для каждого уровня**. Этот подход состоит в обучении многоклассового классификатора для каждого уровня иерархии [10].

Преимуществом локального подхода является возможность использовать различные алгоритмы классификации на разных уровнях иерархии, в зависимости от их эффективности. Однако имеются и недостатки. Во-первых, если класс объекта на верхнем уровне был определен неправильно, то ошибка будет переходить на следующие уровни иерархии. Во-вторых, необходимо использовать большое количество локальных моделей.

«Глобальный» подход при классификации объекта рассматривает иерархию как единое целое. В данном случае предсказание может происходить на любом уровне иерархии. При этом используется одна модель, а не множество - как в предыдущих двух подходах. Однако глобальный классификатор имеет высокую сложность разработки [11].

Был выбран локальный подход для каждого родительского узла, ввиду его простоты реализации и гибкости. Также использование данного подхода позволяет сократить количество моделей, так как в нашем случае количество листовых классов достаточно большое, что влечет за собой создания большого количества моделей при «плоском» и других способах «локального» подхода.

2 Основные этапы классификации данных

В данном разделе рассмотрены основные этапы классификации данных: сбор и обработка данных, обучение и тестирование, оценка качества классификации; представлены краткие описания популярных алгоритмов классификации.

2.1 Сбор данных

Для загрузки данных с сайтов можно использовать готовое API. Но не все сайты его предоставляют. В таких случаях приходится извлекать необходимую информацию самостоятельно.

Так как необходимо извлечь большой объем данных, то следует автоматизировать данный процесс, т.е. выполнить синтаксический анализ html-разметки [12]. Для получения большей гибкости, необходимо создать собственный анализатор для каждого сайта, чтобы учитывать особенности каждого.

Для того извлечь необходимые данные со сторонних сайтов необходимо эмулировать работу браузера, т.е. посылать http-запросы стороннему веб-серверу и получать ответы, в рассматриваемом случае ответом является html-код. После чего из полученного html-кода извлечь необходимые данные.

2.2 Обработка данных

После того, как были получены текстовые данные их необходимо подготовить для классификации, т.е. преобразовать в числовой формат.

Наиболее популярным и простым векторным представлением является «мешок слов», суть которого состоит в подсчете количества вхождений каждого слова(токена) в объект, но при этом порядок слов не учитывается [13]. Следовательно «мешок слов» представляет объект в виде вектора частот слов.

Соответственно, размер вектора будет равен количеству слов в каждом объекте, среди которых часто встречаются неинформативные слова, такие как, например, союзы, предлоги – стоп слова. Избавиться от таких слов можно с помощью их удаления. Обычно можно использовать готовые списки стоп-слов на основе соответствующего языка. Как правило, фиксированные списки могут быть полезны при работе с небольшими наборами данных.

Еще одним способом борьбы с неинформативными словами является мера TF-IDF, которая используется для оценки важности слова в контексте документа (текстового объекта), входящего в коллекцию документов или корпус [14]. То есть больший вес имеет то слово, которое часто встречается в конкретном документе, но при этом редко встречается в остальных документах корпуса. Значит это слово имеет большую значимость для этого конкретного документа. Вес слова пропорционален количеству его вхождений в документ, и обратно пропорционален частоте вхождений этого слова в остальных документах коллекции.

2.3 Обучение и тестирование

Для проведения обучения и тестирования полученный набор данных необходимо разделить на обучающую и тестовую выборки. Это можно выполнить путем случайного разбиения в определенном соотношении – обычно 70% на 30%. Однако данный способ имеет недостаток в том, что он не гарантирует надежных результатов, т.к. такое распределение данных может оказаться неудачным и в тестовую выборку попадут нетипичные примеры, что покажет ложные высокие или низкие результаты классификации. Для решения данной проблемы применяется k-блочная перекрестная валидация (k-fold cross-validation) [13].

Этот способ предполагает деление данных на k частей одинакового размера – обычно k=5 или k=10. Обучение и тестирование происходит последовательно. На первом шаге используется первый блок для тестирования, а остальные четыре блока

– для обучения. Таким образом номер блока, который используется для тестирования соответствует номеру шага. Аналогичная процедура повторяется еще k-1 раз.

При этом данные должны быть разбиты таким образом, чтобы для каждого блока соотношение количества объектов, относящихся к разным классам, соответствовало этому соотношению для исходного набора данных. Этого можно добиться обычным перемешиванием данных.

2.4 Обзор алгоритмов классификации данных

Наивный Байес

Данный алгоритм является одним из самых известных и простых алгоритмов машинного обучения. Он является вероятностным алгоритмом, в основе которого лежит предположение, что все признаки являются независимыми случайными величинами. Преимуществом алгоритма является высокая скорость обучения.

Деревья решений

Дерево решений представляет собой логистический алгоритм машинного обучения, в основе которого лежит дерево. Дерево можно представить в виде иерархии условий. В таком дереве каждый узел является признаком, каждая ветка правилом, а каждый листовой узел является классом. Деревья принятия решений подразделяют пространство признаков на области с одинаковыми значениями.

Случайный лес

Случайный лес представляет собой набор деревьев решений. Практически всегда показывает лучшие результаты, чем одно дерево решений.

Логистическая регрессия

Предсказывает вероятность появления события с помощью логической функции. Используется для предсказания бинарных дискретных значений (True/False, 1/0). Может использоваться в многоклассовой классификации только при использовании подхода one-vs-rest.

Метод опорных векторов

Классифицирует данные на основе векторов, построенных в N-мерном пространстве. N – количество признаков, координаты объектов в этом пространстве определяются значениями признаков.

2.5 Метрики оценки качества классификации

Для оценки качества классификации существуют различные метрики.

Самой простой является метрика accuracy – доля правильных ответов алгоритма. Однако эта метрика бесполезна в задачах с неравными классами.

Этим недостатком не обладают следующие метрики.

Precision (точность) и recall (полнота) используются для оценки качества работы алгоритма на каждом из классов [15].

Precision показывает долю объектов, предсказанных классификатором как положительные, которые в действительности являются положительными.

$$\text{Prec} = \frac{TP}{TP + FP}, (2.4.1)$$

где TP – истино-положительные результаты,

FP – ложно-положительные результаты

Recall показывает долю действительно предсказанных положительных объектов из всех объектов положительного класса.

$$\text{Rec} = \frac{TP}{TP + FN} \quad (2.4.2),$$

FN – ложно-отрицательные результаты

Также существует метрика, которая объединяет метрики precision и recall - F-measure (F-мера). Она вычисляется как гармоническое среднее между приведенными выше точностью и полнотой.

$$F_1 = \frac{2 \times Prec \times Rec}{Prec + Rec} \quad (2.4.3)$$

В задачах многоклассовой классификации оценка качества сводится к вычислению вышеуказанных метрик для каждого класса с последующим усреднением.

Для задачи иерархической классификации полнота, точность и F-мера модифицируются, так как общепринятые метрики не позволяют учесть информацию об иерархии классов [15].

Точность рассчитывается как отношение количества правильных предсказанных ко всем предсказанным категориям.

$$hPre_i = \frac{\left|\hat{P}_i \cap \hat{T}_i\right|}{\left|\hat{P}_i\right|} \quad (2.4.4),$$

где, для тестового объекта i, $\hat{P}_i$ – все предсказанные результаты и все их родительские классы

$\hat{T}_i$ – все истинные результаты и все их родительские классы

Полнота рассчитывается как отношение количества категорий, правильно предсказанных классификатором к общему количеству правильных категорий.

$$hRec_i = \frac{\left|\hat{P}_i \cap \hat{T}_i\right|}{\left|\hat{T}_i\right|} \quad (2.4.5)$$

F-мера рассчитывается как гармоническое среднее между иерархической точностью и полнотой.

$$hF_{1,i} = \frac{2 \times hPre_i \times hRec_i}{hPre + hRec} \quad (2.4.6)$$

Данные метрики учитывают специфику иерархических классификаторов, что позволяет получить не нулевую оценку для ответов, совпадающих на некоторых уровнях иерархии с правильными.

3 Реализация системы

Разработанная система состоит из следующих модулей:

- модуль сбора и хранения данных изображений;
- модуль обработки данных изображений и построения моделей;
- модуль классификации изображений;
- модуль автоматического выбора категории в веб-интерфейсах фотобанков.

На рисунке 1 представлена схема модулей разработанной системы и их взаимодействие.

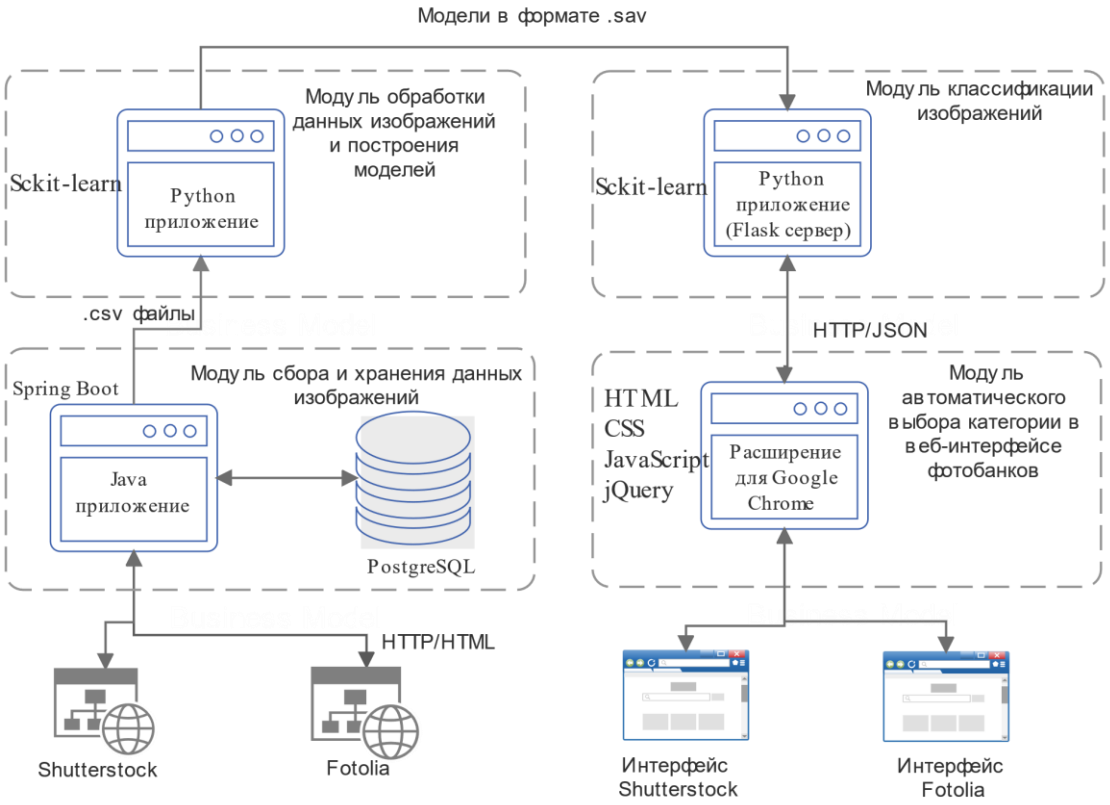

Рисунок 1 – Схема модулей разработанной системы и их взаимодействие

3.1 Разработка модуля сбора и хранения данных изображений

Модуль для сбора и хранения данных разработан с использованием языка Java, фреймворка Spring Boot. Главными преимуществами языка Java являются скорость работы [16] и хорошая поддержка многопоточного кода, что позволило эффективно загрузить данные с фотобанков. Spring Boot использовался для упрощения процесса разработки, так как этот фреймворк предоставляет готовую реализацию существенной части необходимого функционала [17], в том числе взаимодействие с СУБД.

Для работы с базами данных выбрана СУБД PostgreSQL, так как она обладает высокой надежностью и стабильностью [18]. Также используется модуль Spring Boot Data JPA, и входящий в его состав JPA провайдер Hibernate.

Shutterstock

Для того, чтобы загрузить необходимые данные изображений с фотобанка «Shutterstock» были использованы методы Shutterstock API [19], представленные в таблице 2.

Таблица 2 – Описание методов API фотобанка «Shutterstock»

Метод	**Параметры**	**Описание**
[GET] /images/categories	language – язык запроса и ответа	возвращает список всех категорий
[GET] /images/search	*language* – язык запроса и ответа *view* – преставление изображения *category* – имя или id категории, к которой должны принадлежать найденные изображения *sort* – определяет порядок сортировки *page* – номер страницы результатов поиска *per_page* – количество изображений на странице результатов поиска	позволяет найти изображения по параметрам

Shutterstock поддерживает 20 языков, поэтому необходимо указывать требуемый язык во всех перечисленных запросах. В нашем случае следует использовать language=en.

Для того, чтобы выполнить поиск изображений по категориям, нужно использовать параметр category. А чтобы среди данных найденных изображений сразу были ключевые слова необходимо использовать параметр view=full.

Данные, полученные от Shutterstock API, представлены в формате JSON. Пример части ответа со списком категорий:

```
{
    "data": [
        {
            "id": "26",
            "name": "Abstract"
        },
        {
            "id": "1",
            "name": "Animals/Wildlife"
        },
        {
            "id": "11",
            "name": "The Arts"
        },
        {
            "id": "3",
            "name": "Backgrounds/Textures"
        },
        {
            "id": "27",
            "name": "Beauty/Fashion"
        },
        {
            "id": "2",
            "name": "Buildings/Landmarks"
        }, <...>, {
            "id": "24",
            "name": "Vintage"
        }
    ]
}
```

Пример части ответа с информацией об изображении:

```
{
    "page": 1,
    "per_page": 1,
    "total_count": 81995193,
    "data": [
        {
            "id": "550139338",
            "added_date": "2017-01-06",
            "aspect": 1,
            "categories": [
                {
                    "id": "7",
                    "name": "Healthcare/Medical"
                },
                {
                    "id": "17",
                    "name": "Signs/Symbols"
                }
            ],
            "contributor": {
                "id": "762925"
            },
            "description": "Hand drawn blue ribbon with a mustache to
Prostate Cancer Awareness month.\rMedical sign for men's health.",
            "image_type": "vector",
            "is_adult": false,
            "is_illustration": true,
            "has_property_release": true,
            "keywords": ["aid", "association", "attribute",
"awareness", "background", "badge", "band", "blue", "bow", "campaign",
"cancer", "care", "charity", "colon", "concept", "cross", "disease",
"element", "emblem", "fabric", "global", "health", "healthy", "help",
"hipster", "hope", "icon", "illness", "illustration", "isolated",
"life", "loop", "male", "man", "medical", "medicine", "mustaches",
"pin", "prostate", "ribbon", "sick", "sickness", "sign", "silk",
"single", "solidarity", "support", "symbol", "tolerance", "vector"
],
            "media_type": "image",
            "models": [
                {
                    "id": "20515978"
                }
            ]
        }
    ]
}
```

Одним из удобных способов работы с JSON форматом в Java является использование библиотеки Jackson. Преобразование выполняется путем вызова метода readValue() класса ObjectMapper, в качестве параметра которому передается ответ от Shuttertock'а в JSON формате. Метод возвращает объекты SearchCategoryResult и SearchResult, представленные на рисунке 2.

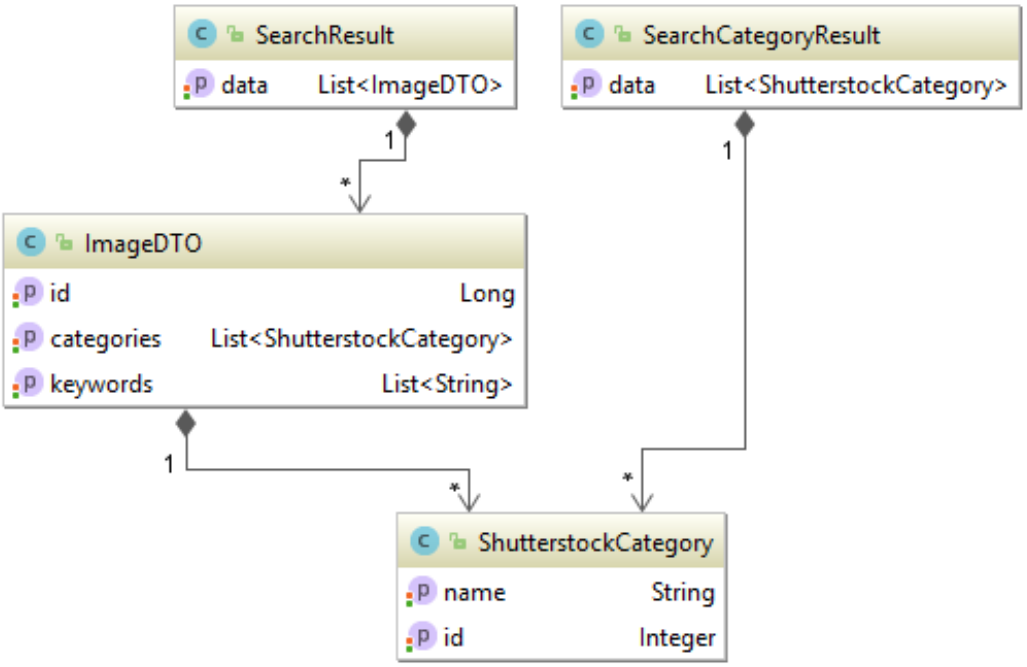

Рисунок 2 – Диаграмма классов ответов от Shutterstock API

Для уменьшения количества времени, затраченного за сбор данных изображений фотобанка Shutterstock, загрузка производилась в несколько потоков. Для реализации многопоточного кода был использован интерфейс ExecutorService, доступный в Java начиная с версии 1.5. Он позволяет добавлять в очередь задачи, которые могут быть выполнены параллельно, и выполняет эти задачи в разных потоках, в зависимости от параметров. В данной работе использовалась реализация интерфейса с фиксированным числом потоков, равным 5 (доступ к данной реализации можно получить, вызвав метод Executors.newFixedThreadPool).

Максимальное количество изображений, которое может быть получено за один запрос, составляет 500 штук. Для получения большего количества изображений

необходимо осуществить несколько запросов, последовательно увеличивая параметр page. Однако, с помощью этого способа все равно нельзя получить более 2000 изображений. Так, например, при выставлении параметру per_page значения по умолчанию (20), при попытке загрузить 101-ую страницу Shutterstock не возвращает данные изображений. Вместо этого в ответе содержится ошибка «Too Many Results: Please provide more query parameters to narrow your search.», то есть предлагается указать дополнительные параметры поиска. Однако никаких других параметров указать невозможно, ведь задачей является получение большого объема изображений из одной категории.

Для решения описанной выше проблемы был использован другой подход поиска и загрузки изображений. Вместо сортировки по умолчанию (по популярности), была использована случайная сортировка. Это позволило выполнять множество одинаковых запросов, возвращающих данные разных изображений, с указанием первой страницы и не получать ошибку. Возможные дубликаты отсеивались с использованием множества (HashSet) id уже загруженных изображений.

С помощью данного API были получены список из 26 категорий и данные 0,5% изображений каждой категории, то есть 1,5 млн изображений. По данными подразумеваются ключевые слова и id соответствующей категории.

Fotolia

От фотобанка «Fotolia» был получен отказ в использовании API, поэтому сбор данных осуществлялся путем загрузки и синтаксического анализа html-страниц сайта.

Рассмотрим структуру элементов html разметки, содержащую необходимые для классификации данные.

Страница поиска. Список категорий. Из этой разметки извлекаются названия и Id категорий.

```
<li><a
href="/search?k=sun&amp;filters%5Bcontent_type%3Aall%5D=1&amp;search-
submit=Search&amp;nca=596" class="dropdown-select-option"
rel="nofollow">Landscapes<span class="description left-
s">(~ 1,500,000)</span></a></li>
<li><a
href="/search?k=sun&amp;filters%5Bcontent_type%3Aall%5D=1&amp;search-
submit=Search&amp;nca=695" class="dropdown-select-option"
rel="nofollow">People<span class="description left-
s">(~ 476,000)</span></a></li>
<li><a
href="/search?k=sun&amp;filters%5Bcontent_type%3Aall%5D=1&amp;search-
submit=Search&amp;nca=498" class="dropdown-select-option"
rel="nofollow">Hobbies and Leisure<span class="description left-
s">(~ 402,000)</span></a></li>
```

Страница поиска. Информация о изображении. Из этой разметки извлекаются Id изображений для последующего получения их данных.

```
<a href="/id/133567224" class="thumb-frame ftl-tooltip-content"><span
class="thumb-inner"><img alt="Vector modern sun set sunshine"
title="Vector: Vector modern sun set sunshine" width="240" height="240"
src="https://t3.ftcdn.net/jpg/01/33/56/72/240_F_133567224_VyuvF0d3uoXqQ
7SE4fxdXkVAHdnAFzUb.jpg" /></span></a>
<a href="/id/114244324" class="thumb-frame ftl-tooltip-content"><span
class="thumb-inner"><img alt="yellow sun icon" title="Vector: yellow
sun icon" width="240" height="240"
src="https://t4.ftcdn.net/jpg/01/14/24/43/240_F_114244324_HkVNU7BbaqHal
aOfSUM4DjvwGygDEMRL.jpg" /></span></a>
<a href="/id/207785952" class="thumb-frame ftl-tooltip-content"><span
class="thumb-inner"><img alt="Close-up image of unrecognizable female
person applying sunscreen creme." title="Photo: Close-up image of
unrecognizable female person applying sunscreen creme." width="360"
height="240"
src="https://t4.ftcdn.net/jpg/02/07/78/59/240_F_207785952_AIzoAOFaLmXna
2qzPe9pY8G0cvXG0i50.jpg" /></span></a>
```

Страница информации об одном изображении. Категории изображения.

```
<dt>Category:</dt>
<dd><ol class="breadcrumb breadcrumb-small">
<li><a class="breadcrumb-item" href="/Search/Category/444">Graphic
Resources</a>
<span class="separator">&gt;</span></li>
<li><a class="breadcrumb-item"
href="/Search/Category/454">Backgrounds</a>
</li>
</ol>
```

Страница информации об одном изображении. Категории изображения.

```
<dt>Keywords:</dt>
<div id="keywords_list" class="display-none">
["apple","aquarelle","art","artwork","background","color","design","dra
w","drawn","fabric","food","fresh","fruit","garden","green","hand","har
vest","healthy","illustration","isolated","leaf","leaves","natural","na
ture","painting","pattern","plant","print","repeated","ripe","seamless"
,"season","slice","summer","textile","texture","vegetarian","wallpaper"
,"watercolor","watercolour","white"]</div>
<dd id="content-keywords-list" class="tag-list">
```

Для получения категорий необходимо выполнить следующие шаги:

Рисунок 3 – Блок схема алгоритма получения категорий

Для получения данных изображений необходимо сначала загрузить страницу результатов поиска и найти на ней id изображений:

Рисунок 4 – Блок схема алгоритма получения id категорий

Далее необходимо загрузить страницы с описанием каждого найденного изображения. Id изображений используется при составлении URL таких страниц.

Рисунок 5 – Блок схема алгоритма получения данных изображений

Автоматические запросы являются проблемой для владельцев сайтов. Для защиты от данного типа взаимодействия сотрудники Fotolia отслеживают подозрительную активность пользователей, и блокируют их IP-адреса, предотвращая доступ. Для возобновления процесса сбора данных с Fotolia были использованы proxy-серверы.

Таким образом, было получено 1097 категорий, которые содержат 21 категорию первого уровня, 153 категорий второго уровня, 923 категорий третьего уровня и данные 0,5% изображений каждой категории, то есть 500 000 изображений.

Структура базы данных была создана Hibernate’ом автоматически, на основе ORM классов, представленных на рисунке 6.

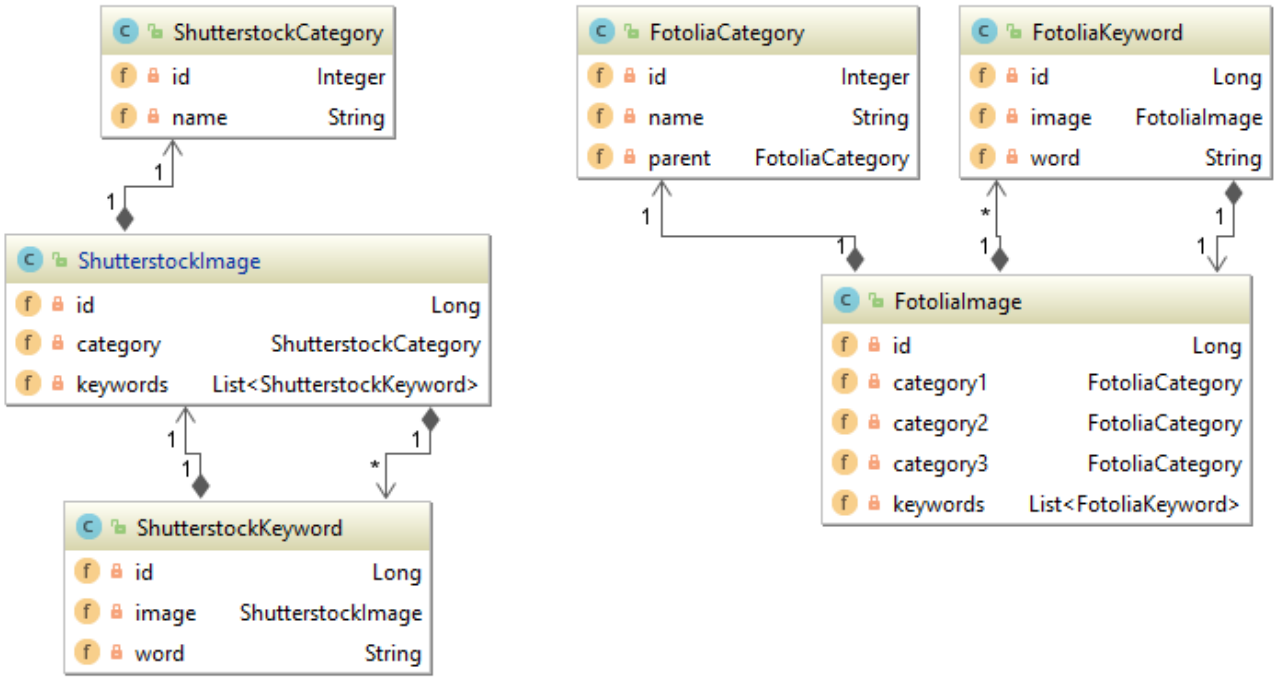

Рисунок 6 – Диаграмма классов ORM

На диаграмме показаны классы ORM, используемые для простого взаимодействия с базой данных:

– ShutterstockImage – описывает изображение фотобанка «Shutterstock». Содержит id изображения, категорию и список ключевых слов.
– ShutterstockCategory – описывает категорию фотобанка «Shutterstock». Содержит id категории и её название.
– FotoliaImage – описывает изображение фотобанка «Fotolia». Содержит id изображения, категории первого, второго и третьего уровней и список ключевых слов.

– FotoliaCategory – описывает категорию фотобанка «Fotolia». Содержит id категории, её название, и ссылку на родительскую категорию для категорий второго и третьего уровней.
– ShutterstockKeyword, FotoliaKeyword – необходимы для хранения ключевых слов изображения. Содержат само ключевое слово и ссылку на объект изображения. Id необходим для упрощения работы со Spring Data.

Классы ORM повторяют структуру базы данных, представленную на рисунке 7.

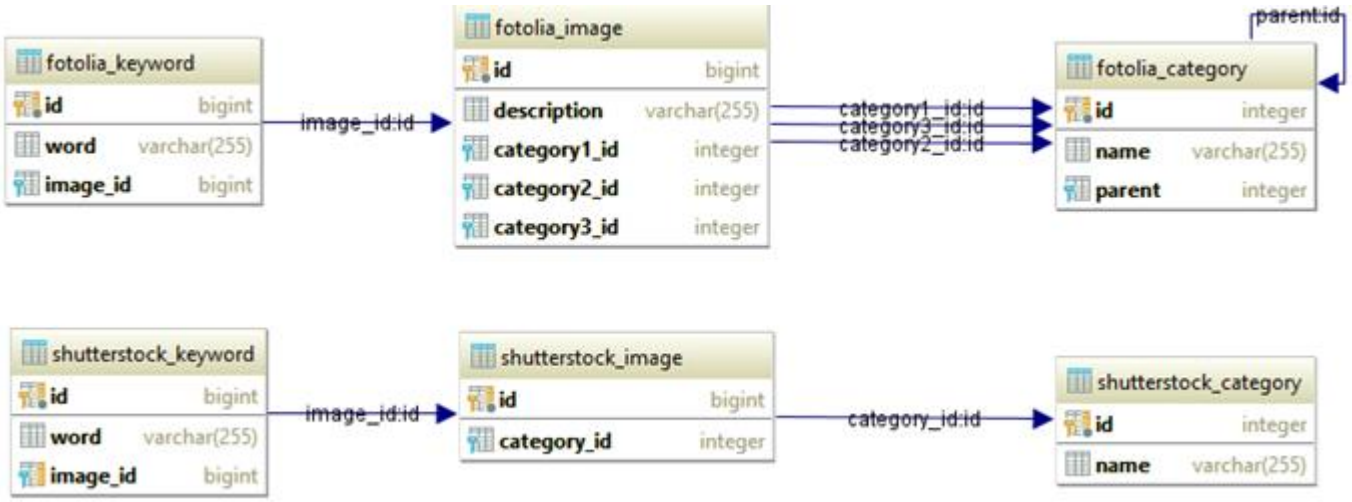

Рисунок 7 – Схема базы данных

Экспорт данных в csv файлы

Несмотря на удобство и скорость разработки с использованием ORM, этот подход показывает среднюю производительность при работе с большим объемом данных. Например, при чтении данных из базы фреймворку Hibernate необходимо «собирать» объекты из нескольких запросов, а именно: загрузить информацию о самом изображении из таблицы shutterstock_image или fotolia_image, затем загрузить ключевые слова из таблицы shutterstock_keyword или fotolia_keyword, загрузить категории из таблиц shutterstock_category или fotolia_category. Это оправдано при

работе с единичными объектами, например, при загрузке данных, но при выгрузке информации из базы данных в файлы можно воспользоваться более эффективным способом. Для этого необходимо выполнить следующие действия:

1. Получить доступ к фабрике сессий работы с базой данных.
2. Открыть новую сессию, не хранящую состояние (Stateless Session; отличается отсутствием кэширования первого и второго уровней, отсутствием взаимодействия с моделью данных Hibernate и перехватчиками событий).
3. Начать новую транзакцию.
4. Выполнить запрос на получение id изображения и id категории (или трех категорий для Fotolia) из таблицы с информацией об изображениях.
5. Результаты запроса сохранить в словаре (HashMap), ключом которого является id изображения, а в значениях содержатся id категорий.
6. Выполнить запрос на получение id изображения и ключевых слов из таблицы с информацией об ключевых словах.
7. Результаты запроса сохранить в словаре (HashMap), ключом которого является id изображения, а в значениях содержатся списки ключевых слов (ArrayList).
8. Завершить транзакцию.
9. Для каждой записи в словаре изображений-категорий получить соответствующую запись из словаря ключевых слов, полученные данные сохранить в csv файл.

Данный способ позволяет существенно сократить время выгрузки данных из базы.

3.2 Разработка модуля обработки данных и построения моделей

Модуль для обучения модели разработан с использованием языка Python 3, который является общепринятым языком для работы с данными. Он предлагает широкий набор инструментов, которые позволяют довольно просто и удобно работать.

Для машинного обучения использовалась популярная Python библиотека scikit-learn с открытым исходным кодом. В ней реализовано множество алгоритмов машинного обучения. Данная библиотека имеет подробную качественную документацию, большое количество обучающих материалов, демонстрирует высокую производительность, а также у scikit-learn [20] есть активное сообщество пользователей.

Для работы с табличными данными были использованы возможности библиотеки Python – Pandas. Для выполнения операций над массивами использовался пакет NumPy.

На выходе модуля сбора и хранения получаются данные изображений, записанные в CSV-файлах. Эти файлы были импортированы с помощью библиотеки pandas.

Перед тем как предсказывать категорию для загруженных автором изображений, необходимо представить текстовые данные изображений – ключевые слова – в числовом формате, построить и обучить модели, на основе которых будет работать выбранный классификатор.

Для того, чтобы текстовые данные представить в числовом виде, была использована модель «мешок слов», которая реализована в классе CountVectorizer библиотеки scikit-learn.

Для избавления от неинформативных слов был использован встроенный список английских стоп-слов, реализованный в модуле feature_extraction.text.

Также был использован метод tf-idf, который scikit-learn реализует в двух классах: TfidfTransformer, который преобразует матрицу частот слов для каждого объекта, полученную с помощью CountVectorizer, и TfidfVectorizer, который реализует и «мешок слов», и преобразование tf-idf. Был использован первый вариант.

После всех преобразований необходимо выполнить разбиение полученных данных на тестовую и обучающую выборки. Такое разбиение можно реализовать с помощью встроенной функцией scikit-learn – KFold(), где в качестве аргумента нужно указать количество блоков - n_splits=5. Для перемешивания наборов данных был использован генератор псевдослучайных чисел, в котором можно указать начальное значение с помощью параметра random_state. Это позволит сделать результат воспроизводимым.

Следующим шагом является построение модели машинного обучения. Эксперименты проводились с использованием следующих алгоритмов машинного обучения, реализованных в scikit-learn:

- Случайный лес (RandomForestClassifier)
- Наивный Байес (MultinomialNB)
- Логистическая регрессия (LogisticRegression)
- Дерево решений (DecisionTreeClassifier)
- Метод опорных векторов (LinearSVC)

Shutterstock

Для оценки эффективности каждого алгоритма были использованы функции precision_score, recall_score, recall_score, которые предоставляют отчеты о трех метриках: точности, полноте и F1-мере.

Подсчет метрик производился для каждого класса согласно формулам (2.4.1 – 2.4.3), после чего выполнялось их усреднение.

Для того чтобы выбрать алгоритм, был проведен ряд экспериментов.

В таблице 3 представлены оценки для каждого алгоритма.

Таблица 3 – Результаты классификации Shutterstock

Алгоритмы	**Точность (%)**	**Полнота (%)**	**F1-мера (%)**
Случайный лес	*75,0*	*75,0*	*74,1*
Наивный Байес	63,3	63,0	61,0
Логистическая регрессия	69,0	69,2	68,3
Дерево решений	69,1	69,1	69,1
Метод опорных векторов	70,0	70,0	70,0

В результате проведенных экспериментов для Shutterstock, лучшие результаты показал «Случайный лес».

Fotolia

Отличительной чертой «Fotolia» является иерархичность категорий, что требует иного подхода к формированию блоков для перекрестной проверки, обучению и тестированию, а также к подсчету метрик для оценки качества полученных классификаторов.

Основной проблемой при формировании обучающей и тестовой выборки является дублирование данных из тестовой в обучающей.

Для решения данной проблемы были выполнены следующие шаги:

1. Для каждой категории третьего уровня данные изображений были отсортированы случайным образом и поделены на 5 равных частей.
2. Данные полученных выборок использовались для обучения и перекрестной проверки классификаторов третьего уровня.

3. Для каждой категории второго уровня, у которых нет третьего уровня данные изображений были отсортированы случайным образом и поделены на 5 равных частей.
4. Для обучения и перекрестной проверки классификаторов второго уровня использовалась обучающая выборка, которая состоит из объединения частей двух обучающих выборок, полученных на предыдущих шагах.
5. Для обучения и перекрестной проверки классификаторов первого уровня использовалась та же обучающая выборка, что и на предыдущем шаге.
6. При тестировании классификаторов трех уровней использовались тестовые данные, которые состоят из объединения тестовых выборок, полученных на первом и третьем шаге.

Обучение выполняется следующим образом:

- Производится обучение классификатора первого уровня, для которого обучающей выборкой являются данные всех изображений, где классами являются id категорий первого уровня.
- Для каждой категории первого уровня производится обучения классификатора, для которого обучающей выборкой являются данные всех изображений, категория первого уровня которых совпадает с текущей.
- Для каждой категории второго уровня производится обучения классификатора, для которого обучающей выборкой являются данные всех изображений, категория второго уровня которых совпадает с текущей.

Тестирование выполняется следующим образом:

- Для каждого изображения определяется категория первого уровня.

- По полученной первой категории находится соответствующий классификатор второго уровня.
- Определяется категория второго уровня.
- Если у полученной категории есть третий уровень, то находится соответствующий классификатор третьего уровня.
- Определяется категория третьего уровня.

Подсчет метрик производился для каждого объекта согласно формулам (2.4.4 – 2.4.6), после чего выполнялось их усреднение.

Для того чтобы выбрать алгоритм, был проведен ряд экспериментов.

В таблице 4 представлены оценки для каждого алгоритма.

Таблица 4 – Результаты классификации Fotolia

Алгоритмы	**Точность (%)**	**Полнота (%)**	**F1-мера (%)**
Случайный лес	*72,4*	*74,5*	*73,0*
Наивный Байес	53,6	55,0	54,2
Логистическая регрессия	65,2	67,0	65,9
Дерево решений	64,1	66,0	64,9
Метод опорных векторов	67,9	69,9	68,7

В результате проведенных экспериментов для Fotolia, лучшие результаты показал «Случайный лес».

Условия проведения экспериментов: Intel Core i7-6700K 4.00GHz, 16 GB RAM.

Для обучения модели на обучающей выборке X, y была использована функция fit(X, y). Для предсказания категории на тестовых данных была использована функция predict(X).

Далее необходимо сохранить модель для того, чтобы использовать ее для определения категорий для загруженных автором изображений. Для этого были использованы модуль pickle от sklearn, который позволяет сохранять и загружать сложные объекты, и его функция dumps, которая записывает сериализованный объект в файл.

3.3 Разработка модуля классификации изображений

Данный модуль(сервер) предназначен для взаимодействия с клиентской частью по протоколу HTTP. Клиент отправляет серверу запрос с ключевыми словами, сервер, используя заранее обученную модель, определяет категорию для полученных ключевых слов и возвращает ее в ответ клиенту.

Для разработки данного модуля был использован один из самых популярных, легковесный и простой в использовании фреймворк Flask [21], который предназначен для создания веб-приложений на Python. Использования языка Python обусловлено необходимостью работы моделями классификаторов, обученными с помощью библиотеки scikit-learn. API разработанного Flask-сервера представлено в таблице 5.

Таблица 5 – Описание методов API Flask-сервера

Метод	**Параметры**	**Описание**
[GET] /shutterstock/	keywords – ключевые слова изображения	возвращает название предсказанной категории для отправленных ключевых слов изображения фотобанка Shutterstock
[GET] /fotolia/	keywords – ключевые слова изображения	возвращает id предсказанных категорий первого, второго и третьего уровня для отправленных ключевых слов изображения фотобанка Fotolia

Описание работы модуля

1. Старт сервера

   Shutterstock

   1.1. Загрузка из файла модели для предсказания категории.

   Fotolia

   1.2. Загрузка из файла модели для предсказания категории первого уровня.

   1.3. Загрузка из файла моделей для предсказания категории второго уровня, добавление их в словарь по ключу, который является id категории первого уровня.

   1.4. Загрузка из файла моделей для предсказания категории третьего уровня, добавление их в словарь по ключу, который является id категории второго уровня.

2. Классификация изображений по ключевым словам.

   Shutterstock

2.1. Предсказание по полученным от клиента ключевым словам id категории с использованием модели с первого шага.

2.2. Получение названия категории по id.

Fotolia

2.3. Предсказание по полученным от клиента ключевым словам id категории первого уровня с использованием модели этого же уровня.

2.4. Получение модели для предсказания категории второго уровня в словаре по id категории первого уровня.

2.5. Предсказание категории второго уровня.

2.6. Получение модели для предсказания категории третьего уровня в словаре по id категории второго уровня.

2.7. Если такая модель найдена, то предсказание категории третьего уровня.

2.8. Предсказание названия категории по id.

3.4 Разработка модуля автоматического выбора категории в веб-интерфейсах фотобанков

Так как фотобанки «Fotolia» и «Shutterstock», являются коммерческими проектами, то их исходный код закрыт. Следовательно, необходимо создать отдельную программу, которая позволит взаимодействовать с интерфейсами рассматриваемых фотобанков.

Единственным возможным решением расширить функционал сторонних веб-сайтов является разработка расширения для браузера – программу, которая позволяет модифицировать веб-страницы, а также добавлять дополнительные функции в браузер и веб-сайты.

Было принято решение разработать расширение для самого популярного браузера Chrome [22], которое позволяет автоматически устанавливать категории изображениям, загружаемым автором для фотобанков «Fotolia» и «Shutterstock».

Рассмотрим основные шаги разработки расширения:

– создание файла манифеста;
– создание интерфейса расширения;
– реализации функциональности расширения [23].

Для разработки расширения использованы языки веб-разработки такие как HTML, CSS, JavaScript, а также библиотека jQuery для упрощения взаимодействия с веб-элементами.

Манифест является основным файлом расширения в формате JSON, в котором содержится основная информация о расширении (название, описание, права доступа, подключаемые скрипты, стили и т. д.);

Интерфейс расширения представляет собой иконку, по нажатию на которую появляется всплывающее окно, которое содержит одну стилизованную кнопку «Categorize».

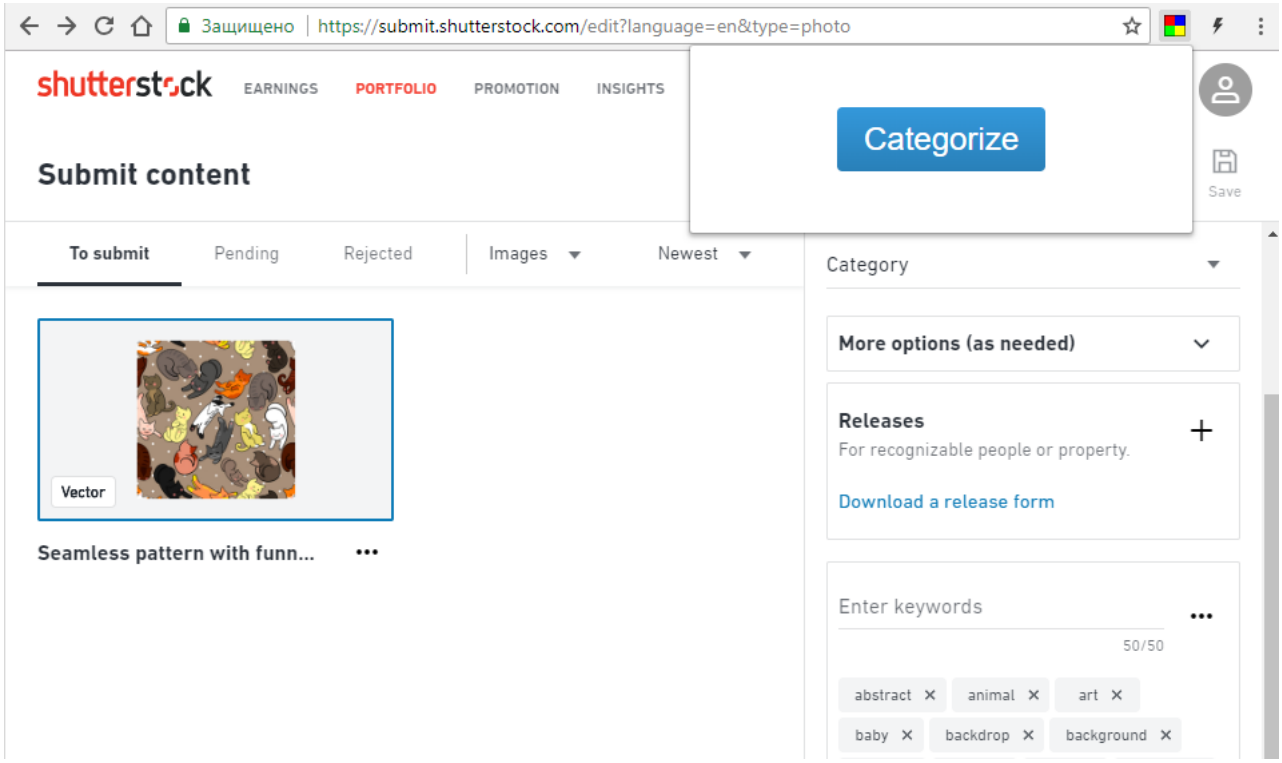

Рисунок 8 – Скриншот работы расширения на странице фотобанка «Shutterstock»

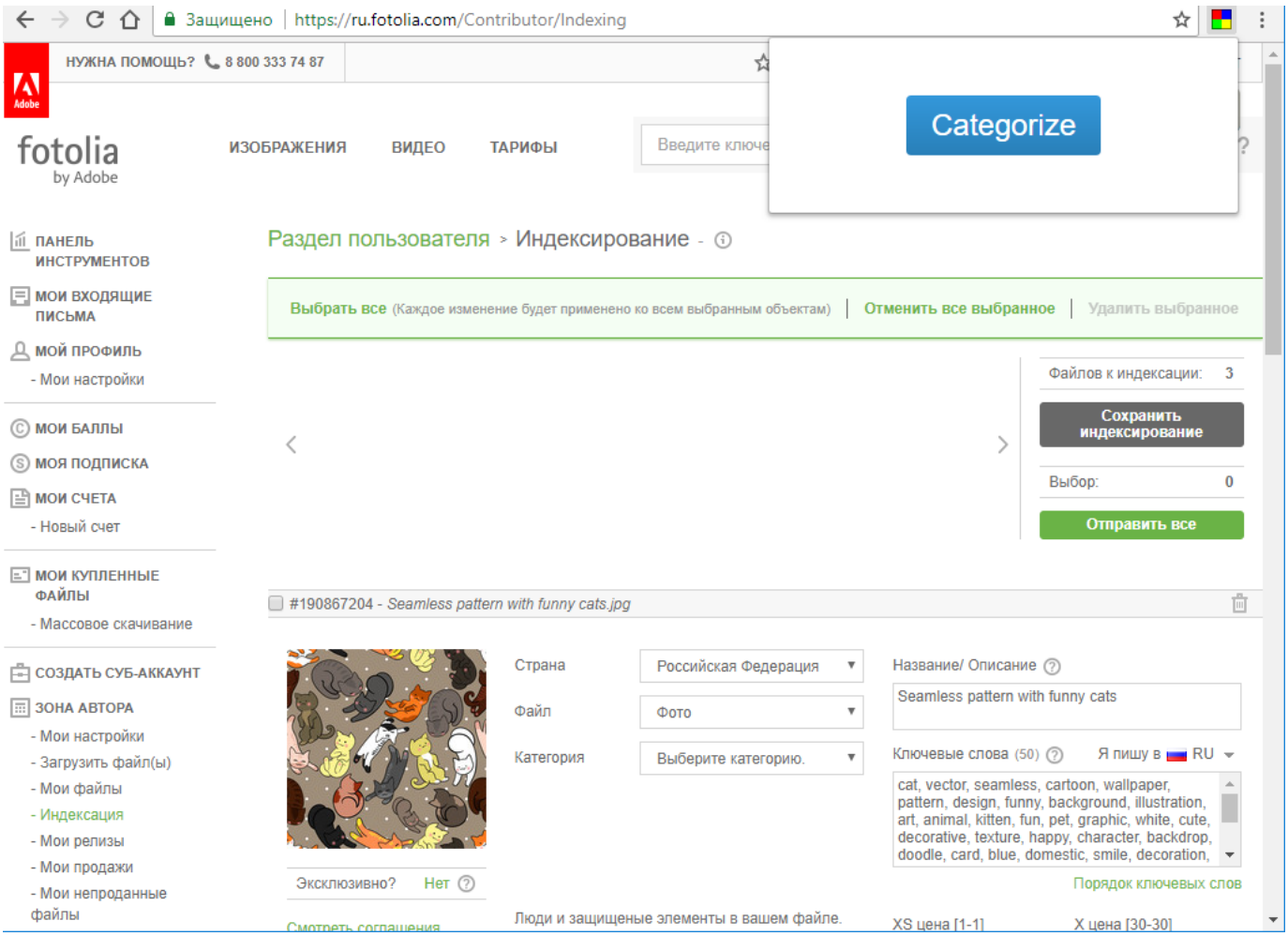

Рисунок 9 – Скриншот работы расширения на странице фотобанка «Fotolia»

Описание работы расширения

Как только открывается всплывающее окно, используя возможности Chrome Javascript API, получаем адрес активной вкладки.

В зависимости от соответствия полученного адреса одному из адресов рассматриваемых фотобанков, кнопке, которая находится внутри всплывающего окна, назначается соответствующий способ взаимодействия с ключевыми словами и категориями изображений.

У Shutterstock панель для ввода ключевых слов и выпадающий список категорий для каждого изображения появляется только после нажатия на него. Соответственно, при выделении нескольких изображений выбранная в панели

категория будет назначена для всех выделенных изображений. Поэтому для того, чтобы определить категорию для выбранных изображений, необходимо по очереди выделять по одному изображению, получать его ключевые слова, определять категорию, указывать её в выпадающем списке. То же самое касается и случая, когда выбраны все изображения. Определение категории осуществляется с помощью get-запроса к Flask-серверу. Параметрами запроса являются ключевые слова изображения, в ответе содержится название предсказанной категории на английском языке.

Fotolia предоставляет возможность доступа к подобным панелям без дополнительных действий, так как данные загруженных изображений всегда присутствуют на странице, вне зависимости от выделения, что облегчает процесс сбора ключевых слов и установки категории.

Однако, на Fotolia существует возможность массового редактирования изображений. Она выполнена следующим образом: пользователь выделяет несколько изображений. При выборе категории для одного из выделенных изображений, такая же категория назначается всем выделенным изображениям. Поэтому, перед началом работы плагина необходимо убрать все выделение, а после, для удобства пользователя, восстановить его.

Последовательность шагов работы расширения с Fotolia аналогична работе с Shutterstock. Для каждой картинки необходимо получить список ключевых слов, отправить get-запрос к Flask-серверу, в ответе получить id предсказанных категорий первого, второго и третьего уровня (при наличии третьего уровня), выбрать полученные категории в соответствующих выпадающих списках.

Более детально взаимодействие расширения с фотобанками «Shutterstock» и «Fotolia» представлено на диаграммах последовательности (рисунки 10, 11).

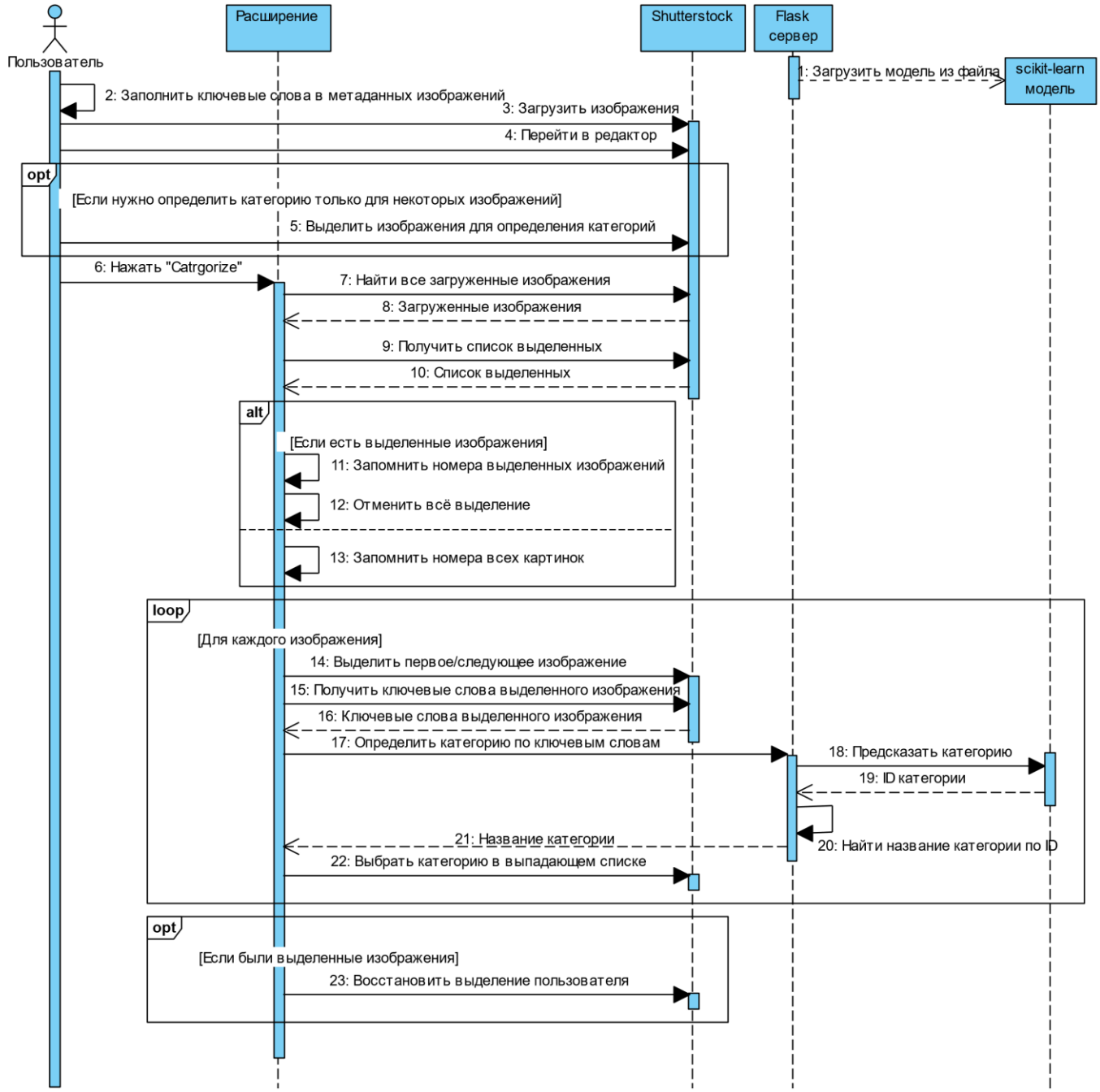

Рисунок 10 – Диаграмма взаимодействия расширения с фотобанком Shuttersrock и Flask-сервером

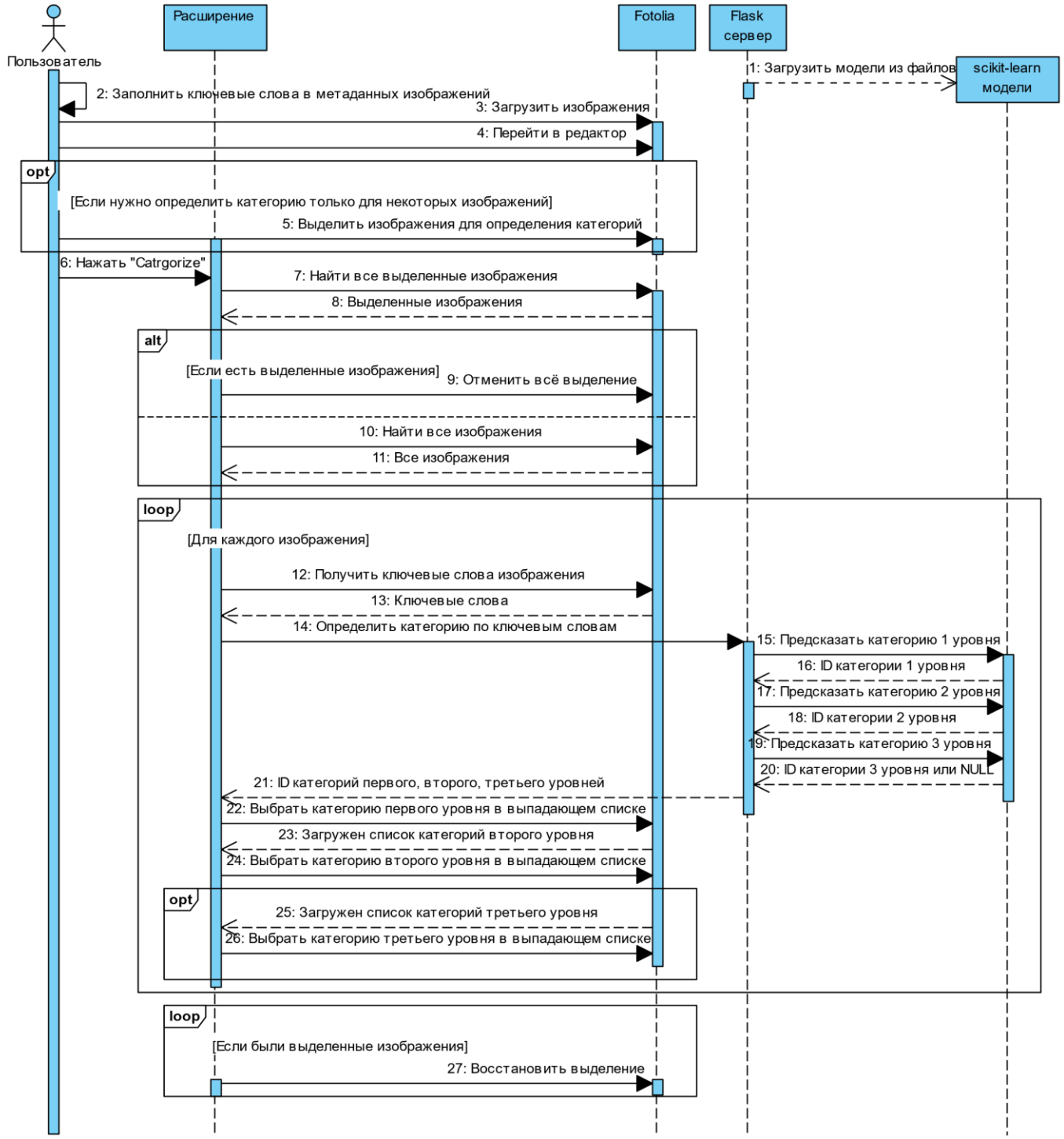

Рисунок 11 – Диаграмма взаимодействия расширения с фотобанком Fotolia и Flask-сервером

ЗАКЛЮЧЕНИЕ

В рамках выполнения данной работы были решены следующие задачи:

- Выполнен обзор предметной области.
- Выбраны фотобанки «Shutterstock» и «Fotolia».
- Выполнен обзор основных этапов классификации.
- Реализована система, которая включает в себя следующие модули:
  - Модуль сбора и хранения данных изображений.
  - Модуль обработки данных изображений и построения моделей.
    - Проведен ряд экспериментов и выбраны алгоритмы классификации.
  - Модуль классификации изображений.
  - Модуль автоматического выбора категории в веб-интерфейсах фотобанков.

В дальнейшем планируется расширить список фотобанков, для которых будет осуществляться автоматическая классификация изображений по категориям на основе ключевых слов.

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

Сокращение	Описание
API	Application Programming Interface
JSON	JavaScript Object Notation
Id	Identifier
TF-IDF	TF – term frequency, IDF – inverse document frequency
ORM	Object-Relational Mapping
URL	Uniform Resource Locator
СУБД	Система Управления Базами Данных

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Battula, Bhanu Prakash, and R. Satya Prasad. An Overview of Recent Machine Learning Strategies in Data Mining // International Journal of Advanced Computer Science and Applications(IJACSA), Volume 4 Issue 3, 2013.
2. Shutterstock [Электронный ресурс] – Режим доступа: URL: https://www.shutterstock.com/ru/subscribe (дата обращения: 10.01.2018)
3. О Fotolia [Электронный ресурс] // Fotolia. – Режим доступа: URL: https://ru.fotolia.com/Info/AboutUs# (дата обращения: 10.01.2018)
4. Depositphotos [Электронный ресурс] – Режим доступа: URL: https://ru.depositphotos.com/about.html (дата обращения: 10.01.2018)
5. Dreamstime [Электронный ресурс] – Режим доступа: URL: https://ru.dreamstime.com/ (дата обращения: 10.01.2018)
6. Bigstockphoto [Электронный ресурс] – Режим доступа: URL: https://www.bigstockphoto.com/ru/ (дата обращения: 10.01.2018)
7. Phototimes [Электронный ресурс] – Режим доступа: URL: https://phototimes.ru/ (дата обращения: 10.01.2018)
8. Наша История [Электронный ресурс] // 123rf – Режим доступа: URL: https://ru.123rf.com/ourstory.php (дата обращения: 10.01.2018)
9. Neha Mehra, Surendra Gupta. Survey on Multiclass Classification Methods // (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 4 (4), 2013, 572 - 576
10. Silla Jr, Carlos N. and Freitas, Alex A. (2011) A survey of hierarchical classification across different application domains // Data Mining and Knowledge Discovery – January 2011, Volume 22, Issue 1–2, pp 31–72.
11. Helyane Bronoski Borges, Carlos N. Silla, Jr., Júlio Cesar Nievola. An evaluation of global-model hierarchical classification algorithms for hierarchical

classification problems with single path of labels // Computers & Mathematics with Applications, Volume 66 Issue 10, December 2013, pages 1991-2002.

12. Роговой Н. Web parsing: задачи, проблемы, инструменты // Inostudio. – Режим доступа: URL: https://inostudio.com/ru/article/web-parsing.html (дата обращения: 20.01.2018)
13. Andreas C Müller; Sarah Guido, Introduction to machine learning with Python: a guide for data scientists – Sebastopol, CA: O'Reilly Media, Inc, 2017.
14. S.Brindha, Dr.K.Prabha, Dr.S.Sukumaran. THE COMPARISON OF TERM BASED METHODS USING TEXT MINING // IJCSMC, Vol. 5, Issue. 9, September 2016, pg.112 – 116.
15. Shou Feng, Ping Fu* and Wenbin Zheng. A Hierarchical Multi-Label Classification Algorithm for Gene Function Prediction // Algorithms 2017, 10(4), 138.
16. Python 3 programs versus Java [Электронный ресурс] // The Computer Language Benchmarks Game – Режим доступа: URL: https://benchmarksgame-team.pages.debian.net/benchmarksgame/faster/python.html (дата обращения: 05.01.2018)
17. Spring Boot [Электронный ресурс] // Spring by Pivotal – Режим доступа. URL: https://spring.io/projects/spring-boot (дата обращения: 11.01.2018)
18. Введение в PostgreSQL [Электронный ресурс] // Professional Postgres – Режим доступа: URL: https://edu.postgrespro.ru/dba1-9.4/dba1_01_introduction.pdf (дата обращения: 20.01.2018)
19. API Resources [Электронный ресурс] // Shutterstock. – Режим доступа: URL: https://developers.shutterstock.com/images/apis (дата обращения: 11.01.2018)
20. Supervised learning [Электронный ресурс] // Scikit-learn. – Режим доступа: URL: http://scikit-learn.org/stable/supervised_learning.html (дата обращения: 03.02.2018)

21. Flask [Электронный ресурс] – Режим доступа: URL: https://github.com/pallets/flask (дата обращения: 20.02.2018)
22. Browser Statistics [Электронный ресурс] // w3schools.com – Режим доступа: URL: https://www.w3schools.com/browsers/default.asp (дата обращения: 11.03.2018)
23. Learn Extension Basics [Электронный ресурс] // Chrome. – Режим доступа: URL: https://developer.chrome.com/extensions/getstarted (дата обращения: 12.03.2018)