

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт кибернетики
Направление подготовки 09.03.02 «Информационные системы и технологии»
Кафедра Информационных систем и технологий

БАКАЛАВРСКАЯ РАБОТА

Тема работы
Разработка веб-приложения для семантической разметки изображений

УДК: 004.93:004.738.1

Студенты

Группа	ФИО	Подпись	Дата
8ИЗБ	Иовлев Юлий Денисович		
8ИЗБ	Кузнецова Анастасия Сергеевна		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент каф. ИСТ	Хаустов П.А.	-		

КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент кафедры МЕН	Рахимов Т.Р.	к.э.н., доцент		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент кафедры ЭБЖ	Акулов П.А.	-		

ДОПУСТИТЬ К ЗАЩИТЕ:

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
ИСТ	Мальчуков А.Н.	к.т.н.		

Томск – 2017 г.

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт Кибернетики
Направление подготовки (специальность) Информационные системы и технологии
Кафедра Информационных систем и технологий

УТВЕРЖДАЮ:
Зав. кафедрой
_____ Мальчуков А.Н.
(Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

бакалаврской работы

Студенту:

Группа	ФИО
8ИЗБ	Иовлев Юлий Денисович
8ИЗБ	Кузнецова Анастасия Сергеевна

Тема работы:

Разработка веб-приложения для семантической разметки изображений

Утверждена приказом директора (дата, номер)

03.02.2017, № 664/с

Срок сдачи студентом выполненной работы:

01.06.2017

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе	Объектом исследования является компьютерное зрение и применение web-репозитория для решения задач кластеризации и классификации изображений
Перечень подлежащих исследованию, проектированию и разработке вопросов	Выполнение обзора существующих решений в исследуемой области Исследование технологий, библиотек, фреймворков и прочего программного обеспечения, которое может быть использовано в процессе разработки Проектирование архитектуры web-приложения Реализация web-репозитория для хранения изображений Создание и наполнение базы изображений

	Разработка программных средств для выполнения семантической сегментации изображений Автоматическое тестирование, составление протокола тестирования финансовый менеджмент ресурсоэффективность и ресурсосбережение социальная ответственность.
Перечень графического материала	Презентация в формате *.pptx на 16 слайдах

Консультанты по разделам выпускной квалификационной работы

Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Рахимов Т.Р.
Социальная ответственность	Акулов П.

Названия разделов, которые должны быть написаны на русском и иностранном языках:

Анализ предметной области
Методы и средства разработки
Проектирование веб-приложения
Реализация веб-приложения
Тестирование разработанных программных средств
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение
Социальная ответственность

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	10.01.2017
--	------------

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент кафедры ИСТ	Хаустов Павел Александрович	-		10.01.2017

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ИЗБ	Иовлев Юлий Денисович		
8ИЗБ	Кузнецова Анастасия Сергеевна		

Министерство образования и науки Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт Кибернетики

Направление подготовки (специальность) 09.03.02 Информационные системы и технологии

Уровень образования – бакалавриат

Кафедра информационных систем и технологий

Период выполнения: весенний семестр 2016/2017 учебного года

Форма представления работы:

Бакалаврская работа

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы

Срок сдачи студентами выполненной работы:	1.06.2017
---	-----------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
20.02.17	Анализ предметной области	15
01.03.17	Методы и средства разработки	5
03.04.17	Проектирование	20
05.05.17	Реализация	30
11.05.17	Тестирование	10
12.05.17	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	10
29.05.17	Социальная ответственность	10

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
ассистент кафедры ИСТ	Хаустов Павел Александрович	—		

СОГЛАСОВАНО:

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
ИСТ	Мальчуков Андрей Николаевич	к.т.н.		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСООБЪЕКТИВНОСТЬ И
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студентам:

Группа	ФИО
8ИЗБ	Иовлев Юлий Денисович
8ИЗБ	Кузнецова Анастасия Сергеевна

Институт	ИК	Кафедра	ИСТ
Уровень образования	Бакалавриат	Направление/специальность	Информационные системы и технологии

Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:

1. Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих	Человеческие ресурсы: 2 чел.
2. Нормы и нормативы расходования ресурсов	
3. Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования	

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения	Проведена оценка коммерческого потенциала: 1. Потенциальные потребители результатов исследования. 2. Анализ конкурентных технических решений 3. SWOT-анализ
2. Планирование и формирование бюджета научных исследований	Произведен расчет бюджета научных исследований.
3. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	Определена ресурсная, финансовая, бюджетная эффективность исследования посредством расчета интегрального финансового показателя, интегрального показателя ресурсоэффективности и эффективности.

Перечень графического материала (с точным указанием обязательных чертежей):

1. Оценка конкурентоспособности технических решений
2. Матрица SWOT
3. Альтернативы проведения НИ
4. График проведения и бюджет НИ
5. Оценка ресурсной, финансовой и экономической эффективности НИ

Дата выдачи задания для раздела по линейному графику	
---	--

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Рахимов Тимур Рустамович	к.э.н., доцент		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8ИЗБ	Иовлев Юлий Денисович		
8ИЗБ	Кузнецова Анастасия Сергеевна		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студентам:

Группа	ФИО
8ИЗБ	Иовлев Юлий Денисович
8ИЗБ	Кузнецова Анастасия Сергеевна

Институт	Институт Кибернетики	Кафедра	Информационных систем и технологий
Уровень образования	Бакалавриат	Направление/специальность	09.03.02 Информационные системы и технологии

Исходные данные к разделу «Социальная ответственность»:

1. Характеристика объекта исследования и области его применения	Данная работа посвящена разработке веб-приложения для семантической разметки изображений. Объектом исследования являются технологии машинного обучения и компьютерного зрения. Основным пользователем является программист.
---	---

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

<p>1. Профессиональная и социальная безопасность</p> <p>1.1. Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения.</p> <p>1.2. Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения.</p> <p>.</p>	<p>В данном пункте анализируются вредные и опасные факторы, которые могут возникать при разработке или эксплуатации проектируемого решения:</p> <p>К вредным относят:</p> <ul style="list-style-type: none"> - Отклонение показателей микроклимата в помещении. - Недостаточное освещение помещений. <p>К опасным относят:</p> <ul style="list-style-type: none"> - Электробезопасность. - Пожаровзрывобезопасность. <p>Мероприятия по минимизации опасности:</p> <ul style="list-style-type: none"> - профилактики возникновения аварий (диагностика оборудования, техническое обслуживание) - инженерная защита - физическая защита потенциально опасных объектов
<p>2. Экологическая безопасность</p>	<p>Непосредственно с выполнением данной работы, могут быть связаны негативно влияющие на экологию факторы, сопутствующие эксплуатации ПК. В частности, утилизация оргтехники.</p>
<p>3. Безопасность в чрезвычайных ситуациях</p>	<p>В данном подразделе рассматриваются вероятные чрезвычайные ситуации, которые могут возникнуть при разработке или эксплуатации проектируемого решения,</p>

	такие как: - Пожар - Кибертерроризм
4. Правовые и организационные вопросы обеспечения безопасности	- Специальные (характерные для рабочей зоны) правовые нормы трудового законодательства. В данном пункте приводятся особенности трудового законодательства применительно к конкретным условиям проекта. - Организационные мероприятия при компоновке рабочей зоны. В данном пункте приводятся эргономические требования к правильному расположению и компоновке рабочей зоны специалиста, рабочей зоны в производственных условиях для создания комфортной рабочей среды. Разработанный продукт позволит улучшить условия работы специалистов в области компьютерного зрения, снизить трудозатраты на подготовку тестовых данных и создать общую базу изображений.

Дата выдачи задания для раздела по линейному графику

Задание выдал консультант:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент	Акулов Петр Анатольевич			

Задание приняли к исполнению студенты:

Группа	ФИО	Подпись	Дата
8ИЗБ	Иовлев Юлий Денисович		
8ИЗБ	Кузнецова Анастасия Сергеевна		

РЕФЕРАТ

Выпускная квалификационная работа 98 с., 16 рис., 31 табл., 28 источников, 0 прил.

Ключевые слова: web-приложение, JavaScript, Node.js, HTML, компьютерное зрение, машинное обучение.

Объектом исследования является компьютерное зрение и применение web-репозитория для решения задач кластеризации и классификации изображений.

Цель работы – создание Web-приложения для семантической разметки изображений.

Область применения: компьютерное зрение, машинное обучение.

Экономическая значимость работы: снижение времени и трудозатрат на подготовку данных для решения задач машинного обучения.

СОКРАЩЕНИЯ

В данной работе применены следующие термины с соответствующими сокращениями:

hyperText markup language, version 5 (HTML5): Язык для структурирования и представления содержимого всемирной паутины.

node.js: Программная платформа, основанная на движке V8 (транслирующем JavaScript в машинный код), превращающая JavaScript из узкоспециализированного языка в язык общего назначения.

javaScript: Прототипно-ориентированный сценарный язык программирования.

application programming interface (API): Набор готовых классов, процедур, функций, структур и констант, предоставляемых приложением (библиотекой, сервисом) для использования во внешних программных продуктах.

internet information services (IIS): Представляет собой проприетарную связку серверов для ряда интернет-служб от Microsoft Corporation.

apache: Свободный web-сервер, представляющий собой кроссплатформенное программное обеспечение.

cascading style sheets (CSS3): Формальный язык описания внешнего вида документа, написанного с использованием языка разметки.

frame per seconds (FPS): Количество кадров, отображаемых на мониторе компьютера или ноутбука в течение одной секунды.

git: Распределённая система управления версиями.

github: Крупнейший web-сервис для хостинга IT-проектов и их совместной разработки. В его основе лежит система контроля версий Git. Сервис бесплатен для проектов с открытым исходным кодом.

asynchronous javascript and xml (AJAX): технология, которая позволяет обращаться к серверу без перезагрузки страницы, уменьшая время отклика и увеличивая быстродействие.

javascript object notation (JSON): текстовый формат передачи и обмена данными, базирующийся на языке JavaScript и в большей степени используемый именно с этим языком.

NoSQL: подход к созданию хранилищ баз данных, отличающихся от традиционных реляционных СУБД с доступом к данным средствами языка SQL.

Оглавление

ВВЕДЕНИЕ.....	14
1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ.....	16
1.1 Описание предметной области.....	16
1.2 Обзор аналогов разрабатываемого web-приложения	20
2 МЕТОДЫ И СРЕДСТВА РАЗРАБОТКИ.....	22
2.1 Выбор языков разработки.....	22
2.2 Выбор среды разработки.....	24
2.3 Выбор системы управления базами данных	24
2.4 Используемые средства и фреймворки	25
2.5 Выбор системы контроля версий.....	28
3 ПРОЕКТИРОВАНИЕ WEB-ПРИЛОЖЕНИЯ	30
3.1 Проектирование архитектуры web-приложения	30
3.2 Проектирование базы данных	34
4 РЕАЛИЗАЦИЯ WEB-ПРИЛОЖЕНИЯ	36
4.1 Используемые модули Node.js	36
4.2 Регистрация и авторизация.....	37
4.3 Каталог изображений	37
4.4 Семантический редактор	37
4.5 Система контроля версий	37
4.6 Разделение обязанностей участников командного проекта.....	38
5 ТЕСТИРОВАНИЕ РАЗРАБОТАННЫХ ПРОГРАММНЫХ СРЕДСТВ.....	41
5.1 Автоматическое тестирование	42
5.2 Ручное тестирование	43
6 РЕЗУЛЬТАТЫ РАБОТЫ	49

6.1	Функционал приложения.....	49
7	СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ	56
7.1	Производственная безопасность.....	56
7.1.1	Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения	58
7.1.1.1	Отклонение показателей микроклимата в помещении.....	58
7.1.1.2	Недостаточная освещенность помещений	59
7.1.2	Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения	59
7.1.2.1	Электробезопасность.....	59
7.1.2.2	Пожаровзрывобезопасность	61
7.2	Экологическая безопасность	61
7.3	Безопасность в чрезвычайных ситуациях	63
7.3.1	Инструкция безопасности внутри помещений.....	63
7.3.2	Обязанности работающих в помещениях.....	64
7.3.3	Необходимые действия при возникновении пожара в помещении ..	64
7.4	Правовые и организационные вопросы обеспечения безопасности.....	65
7.4.1	Правовые нормы трудового законодательства	65
7.4.2	Организационные мероприятия обеспечения безопасности	66
8	ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ.....	68
8.1	Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения.....	70
8.1.1	Потенциальные потребители результатов исследования	70
8.1.2	Анализ конкурентных технических решений	71
8.1.3	SWOT-анализ.....	73

8.2	Определение возможных альтернатив проведения научных исследований	75
8.3	Планирование научно-исследовательских работ	76
8.3.1	Структура работ в рамках научного исследования	76
8.3.2	Определение трудоемкости выполнения работ	77
8.3.3	Разработка графика проведения научного исследования	78
8.4	Бюджет научно-технического исследования.....	84
8.4.1	Расчет материальных затрат НТИ	84
8.4.2	Расчет основной заработной платы исполнителей системы	85
8.4.3	Дополнительная заработная плата исполнителей темы.....	89
8.4.4	Отчисления во внебюджетные фонды (страховые отчисления)	89
8.4.5	Расчет накладных расходов.....	90
8.4.6	Формирование бюджета затрат научно-исследовательского проекта 91	
8.4.7	Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования... 91	
	ЗАКЛЮЧЕНИЕ	96
	СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	97

ВВЕДЕНИЕ

Компьютерное зрение – это дисциплина, которая занимается извлечением информации из изображений. Методы компьютерного зрения обладают широкой областью применения: в микроскопии, оптической когерентной томографии, цифровой голографии, робототехнике, космической отрасли.

Например, компьютерное зрение может применяться при считывании штрихкодов с товаров. Но есть и более сложные задачи: считывание номеров автомобилей, анализ медицинских снимков, дефектоскопия на производстве, распознавание лиц и т.д. Активно развивается применение компьютерного зрения для создания систем дополненной реальности. Однако, необходимы тестовые данные, используемые для обучения классификаторов, для дальнейшего автоматического распознавания объектов на изображениях. Разработанные программные средства позволят размечать изображения, а также использовать и наполнять открытую базу данных изображений, что сократит время на подготовку тестовых данных, а также время поиска необходимых изображений.

Объектом исследования является компьютерное зрение и применение web-репозитория для решения задач кластеризации и классификации изображений.

Целью работы является создание web-приложения для семантической разметки изображений, используемых для решения задач компьютерного зрения.

Для достижения поставленной цели было необходимо решить следующие задачи:

- выполнить обзор существующих решений в исследуемой области;
- исследовать технологии, библиотеки, фреймворки и прочее программное обеспечение, которое может быть использовано в процессе разработки;

- реализовать web-репозиторий для хранения изображений;
- создать и наполнить базу изображений;
- разработать программные средства для выполнения семантической сегментации изображений;
- протестировать приложение, составить протокол тестирования.

Результатом работы должно являться работающее web-приложение, реализующее весь необходимый функционал web-репозитория изображений для подготовки изображений для дальнейшего анализа.

Выполнение задания в составе команды из двух человек позволит решить поставленные задачи в более полном объеме. Реализация клиент-серверной архитектуры подразумевает большой объем работ, включающий ознакомление с технической литературой, подготовку и настройку средств разработки, программного обеспечения, фреймворков, проектирование и реализацию непосредственно клиент-серверного приложения, а также тестирование как клиентской, так и серверной частей. В команде из двух человек обязанности можно разделить так, чтобы каждый из разработчиков реализовывал либо клиентскую, либо серверную часть, а тестировал оставшуюся. Кроме того, обсуждение проектных решений в составе из двух исполнителей позволяет избежать большинства ошибочных проектных решений еще на стадии обсуждения, что уменьшает риск непредвиденных временных затрат на выполнение поставленных задач.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Описание предметной области

Компьютерное зрение – это дисциплина (раздел искусственного интеллекта), которая занимается извлечением информации из изображений, причем изображения могут быть разного типа. Это могут быть фотографии, могут быть видео, могут быть наборы фотографий или медицинский снимок из магнитно-резонансного томографа. Например, компьютерное зрение может применяться при считывании штрихкодов с товаров. Штрихкоды были разработаны специально именно таким образом, чтобы максимально упростить компьютеру процесс считывания. Но есть и более сложные задачи: считывание номеров автомобилей, анализ медицинских снимков, дефектоскопия на производстве, распознавание лиц и т.д. Активно развивается применение компьютерного зрения для создания систем дополненной реальности.

Решение задач компьютерного зрения востребовано при создании приложений:

- выполняющих классификацию и сортировку изображений в домашних цифровых фотоальбомах;
- получающих изображения в геоинформационных системах;
- выполняющих поиск товаров по их изображениям в интернет-магазинах;
- реализующих системы биометрической идентификации;
- предназначенных для специального поиска изображений в различных социальных сетях (к примеру, поиска лиц людей, интересных пользователю) и т. д., в том числе для поиска изображений в глобальной сети «Интернет» [1].

Выполняемая работа направлена на подготовку изображений, в частности, для обучения нейронных сетей с целью распознавания образов на изображениях.

В связи с этим, к обработанным изображениям устанавливается ряд требований:

- На изображении должны быть выделены границы отдельных образов. Это необходимо для обучения сети и дальнейшего распознавания объектов на изображениях;
- Каждый выделенный объект должен принадлежать какому-либо классу. Такая информация позволит облегчить поиск изображений по образам определенного класса (машина, дерево и т.п.);
- Изображение должно описываться тегами, задающими классы образов на изображении. Они позволяют осуществлять поиск по изображениям репозитория.

В компьютерном зрении сегментация изображения представляет собой процесс разбиения цифрового изображения на некоторые сегменты (наборы пикселей, также называемые супер-пикселями) [2].

Цель сегментации – упростить или изменить изображение во что-то более содержательное и более простое для компьютерного анализа. Сегментация изображения обычно используется для нахождения объектов и их границ (линий, кривых и т.д.) на изображении. Точнее, сегментация изображения представляет собой процесс присвоения метки каждому пикселю изображения, так что пиксели с одинаковой меткой имеют определенные характеристики.

Результатом сегментации изображения является множество сегментов, которые в совокупности покрывают всё изображение или набор контуров, извлеченных из изображения. Каждый пиксель в области подобен по отношению к некоторому характеристическому или вычисленному свойству, такому как цвет, интенсивность или текстура. Прилегающие области значительно отличаются по отношению к одной и той же характеристике.

Сегментация изображений находит применение в следующих направлениях:

- Поиск изображений по содержанию;

- Машинное зрение;
- Медицинская визуализация;
 - 1) Нахождение опухолей и других патологий;
 - 2) Моделирование виртуальной хирургии;
 - 3) Измерение объемов тканей;
- Обнаружение объектов;
 - 1) Обнаружение пешеходов;
 - 2) Обнаружение лиц на фото;
 - 3) Обнаружение стоп-сигналов, дорожных знаков;
- Задачи распознавания;
 - 1) Распознавание лиц;
 - 2) Распознавание отпечатков пальцев;
 - 3) Аутентификация по радужной оболочке глаза;
- Видеонаблюдение.

Для сегментации изображений было разработано множество алгоритмов и методов общего назначения. Однако большинство методов сегментации основаны только на информации о цвете пикселей в изображении. Люди же используют гораздо больше знаний, при сегментации изображений, но внедрение этих знаний потребует значительных затрат времени и огромной базы знаний об изображениях.

В дополнение к традиционным методам сегментации существуют обучаемые методы сегментирования, которые могут моделировать некоторые из этих знаний.

Сегментация нейронной сетью основана на обработке небольших областей изображения с использованием искусственной нейронной сети или набора нейронных сетей. После такой обработки механизм принятия решений маркирует области изображения соответствующими категориями, распознаваемой нейронной сетью. Одной из таких сетей, разработанной специально для этого является карта Кохонена.

Подходы машинного обучения и нейронных сетей в частности создали новые алгоритмы для сегментирования изображений с превосходной точностью по сравнению с обычными алгоритмами. Сегментация с использованием машинного обучения является полуавтоматизированной, пока не удастся убрать человеческое вмешательство из процесса сегментации без потери точности. Но впервые за долгое время полуавтоматизированная сегментация становится быстрее, чем чисто ручная сегментация [3].

Это важная веха, демонстрирующая полезность компьютеризованной сегментации изображений. Однако важной задачей является еще больше сократить человеческие усилия, потребляемые полуавтоматизированной сегментацией.

В целом основные процессы, связанные с подготовкой обучающих выборок для искусственных нейронных сетей, представлены на рисунке 1.

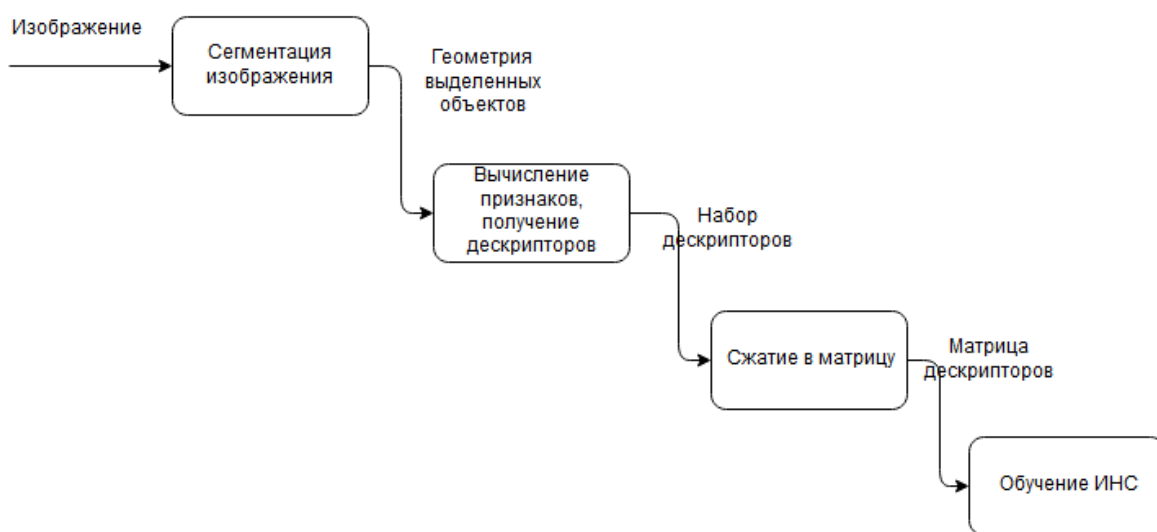


Рисунок 1 – Диаграмма основных процессов при подготовке данных

Из процессов, представленных выше, первый процесс (сегментация изображения) занимает наибольшие временные ресурсы, так как остальные процессы хорошо автоматизированы и выполняются достаточно быстро.

Соответственно, так как избежать ручной сегментации изображений пока не удастся, нужно стараться максимально сократить время на выполнение данного процесса. Во-первых, максимально автоматизировать процесс ручной

сегментации, во-вторых, постараться избежать повторной сегментации, за счет кэширования результатов проделанной работы.

1.2 Обзор аналогов разрабатываемого web-приложения

На данный момент в открытом доступе существует небольшое количество приложений, позволяющих в той или иной мере выполнить сегментацию изображений. Рассмотрим те из них, которые представляют собой web-приложения, независимые от операционной системы и дополнительного программного обеспечения.

Наиболее интересными, с точки зрения аналогий, представляются использование системы VGG Image Annotator, сервиса LabelMe и flash-приложения Szoter.

VGG Image Annotator – это проект с открытым исходным кодом, разработанный в группе Visual Geometry Group в Оксфордском университете. С помощью этого автономного приложения можно выделять регионы изображений и создавать текстовое описание этих регионов. Такие области изображения и их описания полезны для контролируемой подготовки алгоритмов машинного обучения [4].

LabelMe – это инструмент аннотации изображений на основе web-технологий, который позволяет пользователям размечать изображения и делиться аннотациями со всем миром [5].

Szoter – представляет собой web-сервис, реализованный с использованием Adobe Flash Player. Сервис позволяет загрузить изображение или сделать скриншот, добавить аннотацию, добавить текст, и затем сохранить полученное изображение на диск [6].

Результат сравнительного анализа представлен в таблице 1.

Таблица 1 – Сравнительная таблица web-сервисов

Наименование Web-сервиса	Сохранение и загрузка изображений	Наличие тегов	Выделение образов на изображении	Сохранение выделенных образов отдельно от изображения	Множественная загрузка	Готовая база изображений
VGG Image Annotator	Да	Нет	Да	Да	Нет	Нет
LabelMe	Да	Нет	Да	Да	Нет	Да
Szoter	Да	Нет	Да	Нет	Нет	Нет
Разрабатываемое приложение	Да	Да	Да	Да	Да	Да

По полученным данным можно сделать вывод, что существующее программное обеспечения для ручной сегментации изображений обладает некоторыми недостатками.

Большинство рассмотренных аналогов предоставляют лишь инструмент для сегментации изображений, у них имеется возможность только загрузить изображение, провести сегментацию и сохранить результат на персональный компьютер. Однако для максимальной автоматизации удобно хранить уже сегментированные изображения в самом приложении.

Исходя из полученных результатов, можно сделать вывод, что разработка приложения, позволяющего максимально автоматизировать процессы сегментации изображений, а также позволяющего создавать уже сегментированные коллекции изображений, на сегодняшний день является достаточно актуальной задачей.

2 МЕТОДЫ И СРЕДСТВА РАЗРАБОТКИ

Перед началом проектирования системы необходимо было определиться с инструментами, которые будут использоваться при разработке. При выборе инструментария проводился анализ современных web-технологий, СУБД и сред разработки. Основными критериями при выборе являлись: удобство использования, наличие развернутой документации, а также производительность.

2.1 Выбор языков разработки

В качестве языка программирования для серверной части web-приложения была выбрана программная платформа Node.js, основанная на движке V8 (транслирующем JavaScript в машинный код), превращающая JavaScript из узкоспециализированного языка в язык общего назначения.

Node.js добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой API (написанный на C++), подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода [7]. Node.js применяется преимущественно на серверной стороне. В основе Node.js лежит событийно-ориентированное и асинхронное программирование с неблокирующим вводом/выводом.

В отличие от IIS или Apache, Node.js не используют многопоточную модель. В Node.js есть только один рабочий поток, который обслуживает все запросы пользователей и отвечающие ресурсы.

Главным преимуществом Node.js является его производительность. Осуществляется это благодаря тому, что используется нестандартная архитектура общения клиента с сервером: для этого языка нет необходимости устанавливать и настраивать web-сервер.

Для разработки клиентской части web-приложения использовался язык HTML5, а также каскадные таблицы стилей – CSS.

HTML5 – это новая открытая платформа, предназначенная для создания web-приложений использующих аудио, видео, графику, анимацию и многое другое.

Несмотря на регулярное появление новых технологий, язык HTML остается самым популярным инструментом разметки. Современный HTML5 характеризуется массой преимуществ, основными из которых можно назвать:

- Возможность применения более простого кода, где div заменены усовершенствованными элементами;
- Утонченные дизайнерские формы, позволяющие создавать эксклюзивный облик web-сайта. В HTML5 удалось существенно улучшить пользовательский интерфейс, сделать его более функциональным, простым и интуитивно понятным. Валидация форм теперь доступна классическими средствами HTML;
- Возможность использования новых полей ввода для различных целей, включая поиск;
- Усовершенствованная семантика HTML5 позволяет быстрее и проще различать футер, заголовки, панель навигации. Дело в том, что для достижения таких целей теперь используются новые теги для упрощенного назначения основных элементов в разметке;
- Использование элементов HTML5 позволяет сделать более видимым семантическое ядро каждой страницы;
- Высокое удобство пользования. Теперь возможно создание сайтов с HTML5, используя новые элементы, включая <canvas>, <audio>, <video>. Выполнены улучшения для интеграции с содержимым SVG. Таким образом, при использовании новых элементов отсутствует необходимость в установке таких плагинов, как Flash [8].

CSS3 – это новый стандарт оформления HTML документов значительно расширяющий возможности предыдущего стандарта CSS2.1.

Многие возможности, которые были труднодоступны в CSS2.1, то есть требовали использования дополнительных внешних программ (таких как Adobe

Photoshop), скриптов (таких как JavaScript) или специальных «хитростей» могут легко достигаться в CSS3 за счет использования новых свойств оформления.

CSS3 предоставляет следующие возможности:

- Сформировать сферические и линейные градиенты;
- Создавать элементы со сглаженными углами;
- Добавлять тени как к самим элементам, так и их тексту;
- Применять небезопасные шрифты (без опасений, что браузер пользователя не будет их поддерживать);
- Использовать более гибкий подход к оформлению фоновой картинки элементов;
- Использовать новые способы задания цвета;
- Использовать и создавать различные эффекты переходов и анимации, а также многое другое [9].

2.2 Выбор среды разработки

Для создания и проектирования приложения в качестве среды разработки было выбрано решение от компании JetBrains – WebStorm, которое обладает мощным функционалом для разработки.

Выбранная среда разработки подходит одновременно для front-end-разработки и для разработки серверной части на Node.js. Главное достоинство WebStorm – это удобный редактор, поддерживающий все основные языки web-разработки – JavaScript, HTML, CSS.

Также можно отметить некоторые ключевые возможности, помогающие разработчику:

- Отладка Javascript кода;
- Интеграция с системами управления версиями (Git, GitHub, Subversion и т.д.);
- Встроенный менеджер пакетов npm.

2.3 Выбор системы управления базами данных

В настоящее время любая информационная система хранит какие-либо данные, а с ростом количества пользователей и объемов данных, хранить данные в оперативной памяти неэффективно. На данный момент практически любая информационная система имеет в своей структуре базу данных (БД).

Существует большое количество различных систем управления базами данных (СУБД), каждая из которых имеет свои преимущества и недостатки. В последнее время набирают популярность NoSQL хранилища. Основная цель NoSQL баз данных – предоставление дополнительных возможностей в отличие от реляционных систем. Например, это хранение пар ключ-значение, за счет такого хранения получается значительное увеличение производительности, или хранение неструктурированных наборов данных с которыми проще будет работать.

В качестве СУБД для более простого взаимодействия с NoSQL хранилищем была выбрана открытая СУБД – MongoDB.

MongoDB – это система хранения данных, основанная на принципе хранения документов в BSON (Binary JSON) формате. К достоинствам данной СУБД можно отнести:

- распределенный доступ к данным, расположенных на нескольких серверах;
- возможность параллельного извлечения данных;
- быстрое извлечение простых структур данных;
- возможность хранить неструктурную информацию.

Для графического представления хранилища использовался кроссплатформенный графический клиент для MongoDB – Robomongo, представляющий собой достаточно простую и удобную систему для тестирования и поддержки баз данных.

2.4 Используемые средства и фреймворки

Фреймворки являются неотъемлемой частью программного обеспечения, позволяющей упростить и ускорить процесс разработки.

Фреймворк это набор всевозможных библиотек или инструментов для быстрой разработки рутинных задач. Чаще всего использует одну из распространенных архитектур приложения для разделения проекта на логические сегменты (модули). Главная цель фреймворка, предоставить программисту удобную среду для проекта с большим и хорошо расширяемым функционалом.

В ходе анализа технологий были рассмотрены следующие фреймворки для Node.js: Express, Koa, Nari. Сравнение web-фреймворков приведено в таблице 2.

Таблица 2 – Сравнение web-фреймворков

Фреймворк	Достоинства	Недостатки
Express	Популярность, 5-летнее развитие, постоянная поддержка со стороны разработчиков, подходит для любых приложений	Множество рутинных задач, сложное промежуточное обеспечение, отсутствие встроенной обработки ошибок
Koa	Небольшой, промежуточное связывание проще, использует генераторы ES6	Нестабилен, необходимость написания промежуточного обеспечения самостоятельно
Nari	Отлично подходит для крупных проектов, ориентирован на повторное использование	Отсутствие примеров и приложений с открытым исходным кодом

В результате проведенного анализа веб-фреймворков, для серверной части был выбран фреймворк Express. Express – это очень гибкий web-фреймворк для приложений Node.js, предоставляющий обширный набор функций для мобильных и web-приложений. Имея в своем распоряжении множество служебных методов HTTP и промежуточных обработчиков, фреймворк позволяет создавать надежный API быстро и легко. Однако альтернативные варианты Koa и Nari не отстают и показывают реальные перспективы, охватывая стандарт ES6. Koa и Nari в основном ориентированы

на опытных разработчиков, а также на более крупные и сложные приложения. Express является более универсальным фреймворком, способным решать задачи различной сложности.

На клиентской стороне приложения, для создания нестандартных элементов интерфейса и адаптивной верстки использовались такие популярные фреймворки, как:

- jQuery;
- Bootstrap.

jQuery – это Javascript-фреймворк, созданный для упрощения взаимодействия с HTML-элементами документа, в том числе: получение доступа к элементу, изменение его свойств и атрибутов, манипуляции элементами, создание событий и удобный интерфейс для работы с AJAX [10].

Bootstrap представляет собой CSS-фреймворк для создания адаптивных web-приложений.

К достоинствам Bootstrap можно отнести:

- Сетка. Простота позиционирования элементов на странице;
- Адаптивность. В Bootstrap адаптивность включена по умолчанию для каждого проекта, и позволяет оптимизировать его не только под большие экраны, но и под смартфоны и планшеты. Надстройка под конкретные задачи;
- Формы и другие элементы. Легкое управление современным дизайном шаблонов для событий, кнопок, вкладок и других элементов;
- Динамика. В библиотеке фреймворка содержится большое количество готовых динамичных элементов, которые часто применяются в современных сайтах;
- Универсальность. Совершенно не обязательно в проекте использовать все возможности Bootstrap, можно ограничиться определенным набором.

2.5 Выбор системы контроля версий

Система контроля версий – это система, которая регистрирует все изменения в файлах, а в дальнейшем позволяет вернуться к их более ранним версиям и определить, кем и когда были сделаны конкретные изменения.

Системы контроля версий подразделяются на распределенные и централизованные.

Распределенные системы контроля версий (Git, Bazaar, Mercurial) используют распределенную модель вместо традиционной клиент-серверной модели, то есть все изменения хранятся в локальном хранилище на компьютере и при необходимости синхронизируются с другими. В этом случае устраняется зависимость от центрального сервера, можно вести работу даже без сетевого соединения с ним.

В централизованных системах контроля версий (Subversion, CVS, Perforce) есть центральный сервер, на котором хранятся все данные, и ряд клиентов, получающих из него копии файлов. В таких системах всегда понятно, кто чем занимается в проекте, и их проще администрировать.

Для работы было решено использовать распределенную систему в силу удобства, она обеспечивает возможность сохранять код без соединения с сервером. К тому же, в работе не было необходимости поддерживать постоянную связь с сервером, вполне подходило локальное хранилище. Сравнение распределенных систем приведено в таблице 3.

Таблица 3 – Сравнение распределенных систем контроля версий

Система контроля версий	Достоинства	Недостатки
Git	Высокая производительность, популярность, поддержка локальных веток	Сложность использования, необходимость оперировать большими строками хэшей вместо маленьких номеров (SHA1-хэш)

Продолжение таблицы 3

Mercurial	Интуитивность, простота, невозможность изменения истории	Откат изменений лишь на один коммит Невозможность извлечения части рабочей копии проекта
Bazaar	Интуитивно понятный, хороший встроенный графический интерфейс	Медленный, непопулярный

Была выбрана система Git по ряду причин:

- Git сохраняет на вашем компьютере полную версию всей истории изменений;
- В Git не нужно иметь доступа к серверу, чтобы сохранить в системе контроля версий измененный файл. Система контроля версий позволяет сравнивать и восстанавливать только те изменения, которые были сохранены в ней. Git может сохранить новую версию файла в любой момент;
- Благодаря хранению файлов локально Git быстрее. Это действительно так – в большинстве операций, которые вы будете выполнять, Git окажется быстрее, поскольку все происходит локально на компьютере. Конечно, когда будет происходить загрузка изменений в репозиторий на удаленный сервер, это потребует времени;
- Git обеспечивает целостность файлов. Для всех файлов, которые сохраняются в системе контроля версий, Git использует хэш SHA1. На практике это исключает возможность изменения какого-либо файла таким образом, чтобы система контроля версий не заметила этого.
- Ранее был получен опыт работы с данной распределенной системой, что сократило время на изучение другой системы контроля версий;
- Наличие ресурса GitHub, на котором размещены тысячи репозиториях на различных языках.

3 ПРОЕКТИРОВАНИЕ WEB-ПРИЛОЖЕНИЯ

3.1 Проектирование архитектуры web-приложения

В основе архитектуры лежит паттерн MVC. Данный паттерн представляет собой схему разделения управляющей логики, пользовательского интерфейса, данных приложения на три отдельных компонента: модель, представление и контроллер – таким образом, что модификация каждого компонента может осуществляться независимо [11].

- Модель (Model) предоставляет данные и реагирует на команды контроллера, изменяя свое состояние;
- Представление (View) отвечает за отображение данных модели пользователю, реагируя на изменения модели;
- Контроллер (Controller) интерпретирует действия пользователя, оповещая модель о необходимости изменений.

Схема Model-View-Controller представлена на рисунке 2.

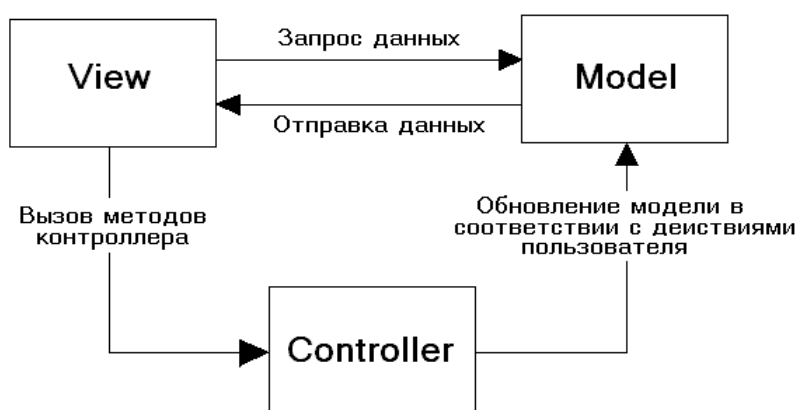


Рисунок 2 – Схема Model-View-Controller

Используя данный паттерн, была определена структура файлов web-приложения, которая выглядит следующим образом.

Описание структуры:

- app.js – содержит главный код приложения;
- core – представляет собой файлы ядра приложения, библиотеки, модули и т.п.;

- `javascripts` – содержит необходимые для работы скрипты (добавление изображения, обработка событий личного кабинета, галереи и пр.);
- `stylesheets` – содержит стили для различных страниц web-приложения;
- `model` – представляет собой папку с моделями (коллекциями). Здесь описывается структура коллекции, которая хранится в базе данных;
- `view` – представляет собой папку с представлениями;
- `routes` – содержит маршруты.

Для представления общей конфигурации и топологии распределенной программной системы была использована диаграммы развертывания UML (рисунок 3).

Диаграмма развертывания предназначена для визуализации элементов и компонентов программы, которые существуют лишь на этапе ее исполнения. При этом представляются только компоненты программы, являющиеся исполняемыми файлами или динамическими библиотеками.

Левая часть диаграммы представляет собой клиента, который с помощью web-браузера по протоколу HTTP взаимодействует с сервером.

Web-сервер представляет собой некую исполняемую среду, реализованную с помощью программной платформы Node.js и web-фреймворка Express. В основу фреймворка Express положены три модуля: Application Logic, Express Middlewares, Route Handler. Application Logic содержит в себе всю бизнес-логику приложения. Express Middlewares представляет собой промежуточный уровень, осуществляющий предобработку запросов. Route Handler организует маршрутизацию клиентских запросов. Благодаря использованию Express осуществляется поддержка паттерна MVC [12].

Правая часть диаграммы представляет собой сервер базы данных, который с помощью специального драйвера взаимодействует с web-сервером.

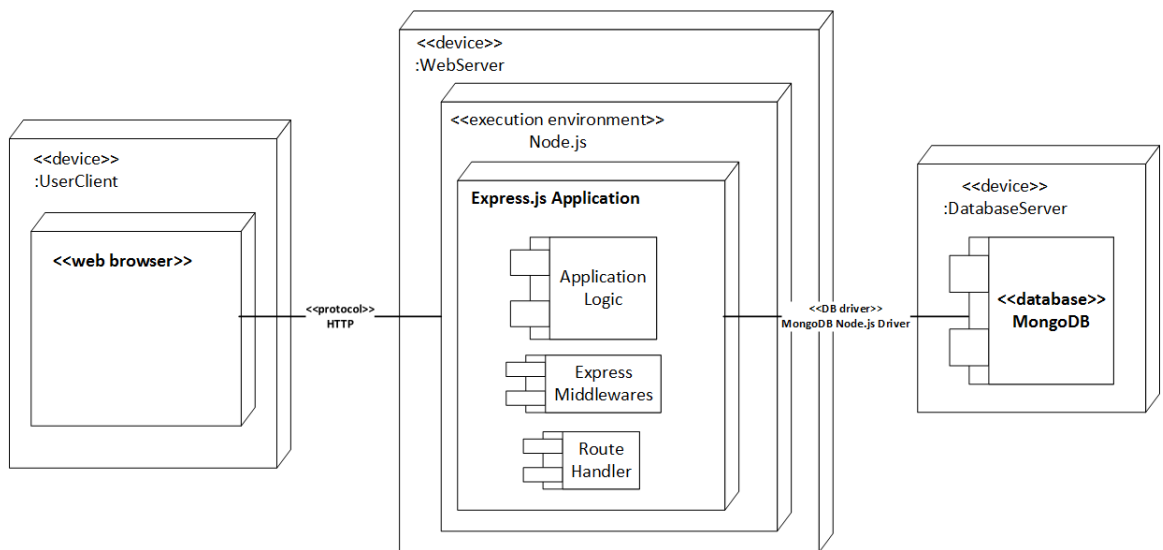


Рисунок 3 – Диаграмма развертывания

Предоставляемые возможности web-приложения представлены на диаграмме вариантов использования (рисунок 4).



Рисунок 4 – Диаграмма вариантов использования для роли «Пользователь»

3.2 Проектирование базы данных

Всю модель устройства базы данных в MongoDB можно представить следующим образом:

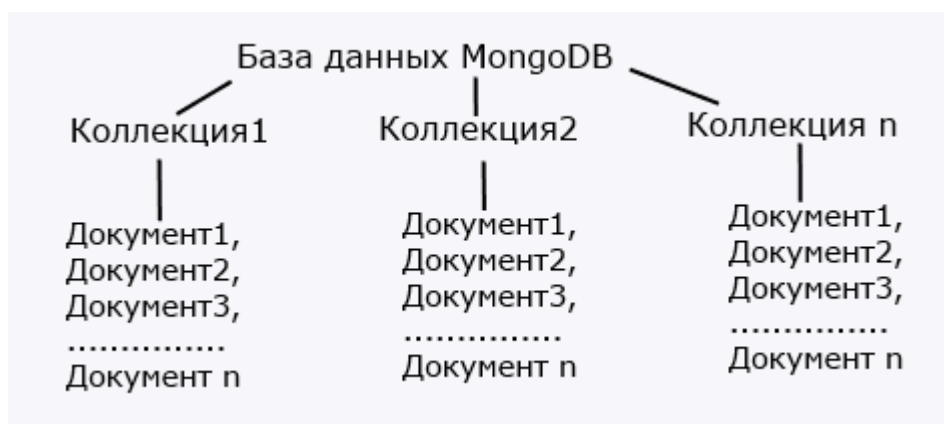


Рисунок 5 – Структура базы данных в MongoDB

В реляционных базах данных содержимое составляют таблицы, в mongodb же база данных состоит из коллекций.

Каждая коллекция имеет свое уникальное имя – произвольный идентификатор, состоящий из не более 128 различных символов.

В отличие от реляционных баз данных MongoDB не использует табличное устройство с четко заданным количеством столбцов и типов данных. MongoDB является документо-ориентированной системой, в которой центральным понятием является документ [13].

Документ можно представить как объект, хранящий некоторую информацию. Документ представляет набор пар ключ-значение.

Для работы web-приложения было необходимо спроектировать структуру базы данных. Было создано 4 сущности:

- User. Коллекция для хранения информации о пользователях.
- Image. Коллекция для хранения информации об изображении.
- UserImage. Сводная коллекция, необходимая для связи пользователя и добавленных им изображений («Избранного»).
- ImageObjects. Коллекция-справочник. Данная сущность позволяет сопоставить к изображению наборы объектов.

Схема разработанной базы данных представлена на рисунке 6.

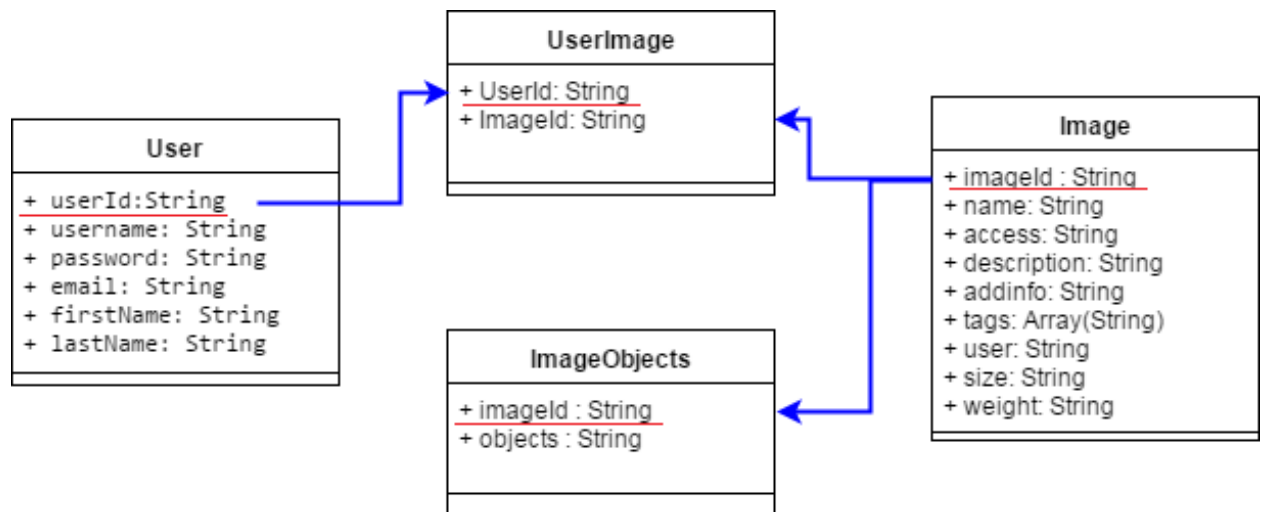


Рисунок 6 – Структура базы данных

4 РЕАЛИЗАЦИЯ WEB-ПРИЛОЖЕНИЯ

4.1 Используемые модули Node.js

Для разработки необходимого функционала web-приложения подключались различные модули, каждый из которых выполнял соответствующую функцию. Процедура поиска и подключения модулей довольно проста, т.к. серверная платформа Node.js обладает отличительной особенностью: наличием большого количества модулей, возможности которых покрывают большинство рутинных задач, которые возникают перед разработчиком.

Наиболее значимыми модулями web-приложения являются:

- Passport – модуль для авторизации в node.js. Используется для регистрации и авторизации пользователей, а также валидации введенных пользователем данных;
- Formidable – модуль node.js для синтаксического анализа данных формы, особенно загрузки файлов. Используется для анализа данных формы, которые приходят на сервер;
- Fs – модуль для работы с файлами в node.js. В web-приложении выполняет функцию чтения и записи файлов;
- Mongoose – это ORM (Object-Relational Mapping) для MongoDB сделанная под node.js. Необходим для создания подключения к базе данных MongoDB и работе с коллекциями;
- Express – быстрый, минималистичный, гибкий web-фреймворк для приложений Node.js. Данный модуль упрощает операции по созданию сервера, работе с запросами и маршрутизацией. Данный фреймворк поддерживает паттерн MVC, разделяя бизнес логику от пользовательского интерфейса и данных приложения.

Все вышеперечисленные модули были подключены к проекту через npm (Node.js Package Manager) – менеджер пакетов, входящий в состав Node.js.

4.2 Регистрация и авторизация

Для авторизации пользователей использовался модуль Passport. Для хэширования паролей использовался модуль bcrypt, который позволяет производить одностороннее хэширование в синхронном и асинхронном режиме.

4.3 Каталог изображений

Для представления каталога изображений в виде галереи использовались каскадные таблицы стилей. Для добавления анимации к элементам страницы использовались псевдоэлементы и псевдоклассы, такие как: after, hover, before, focus.

Галерея является частичным представлением и подгружается на страницу с помощью технологии AJAX. Таким образом, загрузка изображений в каталоге при переходе на другую страницу каталога или при фильтрации изображений по тегам происходит быстрее.

Для загрузки и редактирования изображений использовался компонент modal (модальное окно), предоставляемый bootstrap. Передача данных от клиента к серверу также осуществлялось по технологии AJAX.

4.4 Семантический редактор

Для реализации панели инструментов для семантической разметки изображений был создан класс toolbox. При реализации данного класса использовалась последняя спецификация языка JavaScript – ES6, которая предоставляет удобный синтаксис по работе с классами.

Для отображения графических данных использовался встроенный в HTML5 элемент – canvas.

4.5 Система контроля версий

Для работы использовался web-сервис для хостинга IT-проектов Github. Для каждой задачи создавалась своя ветвь, что позволяет разрабатывать web-приложение параллельно и независимо. Более того, функционал, находящийся

в отдельной ветке, никак не влияет на ту функциональность, что уже реализована. Данный подход позволяет повысить стабильность и безопасность проекта. В ходе работы было создано 10 ветвей, 56 коммитов.

4.6 Разделение обязанностей участников командного проекта

Данная работа является совместным проектом, где у каждого из участников были свои обязанности.

Выполнение задания в составе команды из двух человек позволило решить поставленные задачи в более полном объеме. В команде обязанности были разделены так, что один из разработчиков реализовывал клиентскую часть, а другой – серверную часть. Тестирование web-приложения выполнялось совместно. Стадии и этапы разработки приведены в таблице 4.

Таблица 4 – Стадии и этапы разработки

№	Этап	Срок	Ответственный
1.	Выбор и утверждение задания	10.01.17 – 15.01.17	Иовлев Ю.Д. Кузнецова А.С.
2.	Аналитический обзор, выбор технологий разработки	16.01.17 – 5.02.17	
3.	Проектирование программы		
3.1.	Выбор структуры базы данных	6.02.17 – 15.02.17	Кузнецова А.С.
3.2	Выбор и планирование архитектуры web-приложения	16.02.17 –	Иовлев Ю.Д. Кузнецова А.С.
3.3	Спецификация основных функций	25.02.17	
3.4	Продумывание интерфейса приложения	26.02.17 – 28.02.17	Иовлев Ю.Д. Кузнецова А.С.
4.	Реализация web-приложения		
4.1	Серверная часть		
4.1.1	Создание базы данных	01.03.17 – 04.03.17	Иовлев Ю.Д.

Продолжение таблицы 4

4.1.1	Запуск сервера на локальном компьютере	05.03.17 – 07.03.17	Иовлев Ю.Д.
4.1.2	Обработка добавления изображения на сервер	08.03.17 – 13.03.17	Иовлев Ю.Д.
4.1.3	Обработка изменений при редактировании изображения	14.03.17 – 16.03.17	Иовлев Ю.Д.
4.1.4	Обработка действий по каталогу изображений	17.03.17 – 24.03.17	Иовлев Ю.Д.
4.1.5	Реализация поиска изображений на сервере по тегам	25.03.17 – 02.04.17	Иовлев Ю.Д.
4.1.6	Обработка добавления изображения пользователем в «избранное»	03.04.17 – 09.04.17	Иовлев Ю.Д.
4.1.6	Обработка открытия/сохранения изображений и объектов в семантическом редакторе	10.04.17 – 15.05.17	Иовлев Ю.Д.
4.2	Клиентская часть		
4.2.1	Создание окна добавления/редактирования изображения	01.03.17 – 06.03.17	Кузнецова А.С.
4.2.2	Создание каталога изображений	07.03.17 – 20.03.17	Кузнецова А.С.
4.2.3	Реализация поиска изображений по тегам	21.03.17 – 01.04.17	Кузнецова А.С.
4.2.4	Создание страницы семантического редактора	02.04.17 – 10.04.17	Кузнецова А.С.

Продолжение таблицы 4

4.2.5	Реализация функционала по семантической разметке изображения	11.04.17 – 01.05.17	Кузнецова А.С.
4.2.6	Сохранение изображения в формат PNG	02.05.17 – 07.05.17	Кузнецова А.С.
4.2.7	Сохранение/загрузка размеченных объектов в формат JSON	08.05.17 – 15.05.17	Кузнецова А.С.
5.	Тестирование web-приложения		
5.1.	Тестирование основных сценариев	16.05.17 –	Кузнецова А.С. Иовлев Ю.Д.
5.2.	Тестирование некорректных действий	24.05.17	
6.	Составление документации	25.05.17 – 01.06.17	

5 ТЕСТИРОВАНИЕ РАЗРАБОТАННЫХ ПРОГРАММНЫХ СРЕДСТВ

Тестирование программного обеспечения – проверка соответствия между реальным и ожидаемым поведением программы, осуществляемая на конечном наборе тестов, выбранном определенным образом.

Рассмотрим два основных подхода к выполнению тестов: подход ручного тестирования и подход автоматического исполнения тестов.

Проводимые сравнения показывают тенденцию современного тестирования, ориентированную на автоматизацию процесса тестирования и генерацию тестового кода, что позволяет справляться с большими объемами данных и тестов, необходимых для обеспечения качества при производстве программных продуктов.

Сравнение ручного и автоматизированного тестирования представлено в таблице 5.

Таблица 5 – Сравнение ручного и автоматизированного тестирования

Показатель	Ручное	Автоматизированное
Проверка результата	Гибкая, позволяет оценивать нечетко сформулированные критерии	Строгая. Нечетко сформулированные критерии проверяются путем сравнения с эталоном
Задание входных значений	Гибкость в задании данных	Входные значения строго заданы
Повторяемость	Низкая	Высокая
Надежность	Низкая	Высокая
Чувствительность к изменениям в проекте	Низкая	Высокая
Скорость выполнения тестов	Низкая	Высокая
Возможность генерации тестов	Отсутствует	Поддерживается

В ходе разработки приложения использовалось как ручное, так и автоматизированное тестирование.

5.1 Автоматическое тестирование

На каждом этапе разработки приложения проводилось тестирование уже реализованных частей. Разработка с параллельным тестированием основывается на повторении коротких циклов разработки: написание теста, покрывающего дальнейшие изменения; реализация задуманного; запуск подготовленного теста.

Тест – это процедура, которая позволяет либо подтвердить, либо опровергнуть работоспособность кода. Каждый тест описывает собой предполагаемое поведение системы. Сами тесты были разделены на две категории:

- integration тесты – тесты, имитирующие поведение пользователя и направленные на тестирование системы целиком;
- unit тесты – тесты, предназначенные для тестирования отдельных модулей приложения.

В качестве фреймворков для написания автоматических тестов использовалась библиотека mocha, should.js, и supertest.

Mocha служит для организации описаний тест-кейсов, should.js предоставляет синтаксис для осуществления различных проверок, а supertest представляет собой надстройку над http-клиентом, которая позволяет проверять результаты http-запросов [14].

Приведем пример простого теста. Например, для того чтобы проверить корректность http-запроса можно написать следующий тест.

```
var req = require('supertest')
var application = require(__dirname + '/../app')

describe('GET /', function () {
  it('should contain text "Success"', function (done) {
    req(application)
      .get('/')
      .expect(/Success/, done)
  })
})
```

В данном тесте проверяется, что ответ, приходящий от сервера будет успешным. Если на каком-то этапе разработки ответ, приходящий с сервера будет отличаться от «Success», то ответ на данный тест будет некорректным.

Таким образом, с помощью небольших и простых тестов, происходит покрытие всего кода приложения. При написании некоторого блока кода на очередном этапе разработки происходит проверка корректности всей системы. Это позволяет повысить стабильность разрабатываемого приложения.

В ходе тестирования было написано 20 тестов, которые проверяли работоспособность серверной части приложения. На промежуточных этапах на некоторые тесты система давала неправильный ответ. В конечном итоге, после внесения исправлений и корректировок, система проходит все тесты успешно.

5.2 Ручное тестирование

Ручное тестирование – часть процесса тестирования на этапе контроля качества в процессе разработки программного обеспечения. Оно проводится тестировщиками или обычными пользователями путем моделирования возможных сценариев действия пользователя.

Задача тестировщика заключается в поиске наибольшего количества ошибок. Он должен хорошо знать наиболее часто допускаемые ошибки и уметь находить их за минимально короткий период времени.

После разработки программных средств было проведено ручное тестирование силами двух разработчиков, был составлен протокол тестирования, который предоставляет подробную информацию о проведенном тестировании: описание теста, ожидаемый результат и фактический результат, полученный в ходе выполнения теста (таблица 6).

Таблица 6 – Протокол тестирования

Варианты использования	Действие	Пример	Ожидаемый результат	Фактический результат
Авторизация	Успешная авторизация	Ввод логина и пароля на главной странице приложения	Пользователь будет авторизован, произойдет перенаправление в каталог изображений	Пользователь авторизован, текущей страницей является страница с каталогом изображений.
	Неудачная авторизация	Ввод логина и пароля на главной странице приложения	Пользователю выведется сообщение о некорректности введенных данных	На экране появилось сообщение «Неправильный логин или пароль»
	Выход	Нажатие кнопки «Выйти» на верхней панели приложения	Произойдет выход пользователя из системы, а также перенаправление на главную страницу с полями для авторизации	Пользователь вышел из системы, текущей страницей является основная страница с формой для авторизации
Работа с изображениями	Добавление/редактирование изображения	Добавление изображения с компьютера. Не заполнены обязательные поля	На экране будет отображено сообщение-предупреждение о том, что необходимо заполнить обязательные поля	На экране появилось сообщение-предупреждение о том, что необходимо заполнить обязательные поля

Продолжение таблицы 6

Работа с изображениями		Добавление изображения с закрытым уровнем доступа	Изображение будет добавлено в избранное текущего пользователя, а в общий каталог добавлено не будет	Добавление изображения произошло только в «избранное» текущего пользователя
	Удаление изображения из общего каталога автором изображения	Удаление изображения из каталога	Выбранное изображение должно удалиться из общего каталога, а также из «избранного» других пользователей	Выбранное изображение удалено из общего каталога, а также из «избранного» других пользователей
	Поиск по тегам	Поиск по тегу «колесо», «дверь» в каталоге изображений	Будут показываться только те изображения, которые содержат теги «колесо» и «дверь» одновременно	В результате фильтра были показаны только те изображения, которые содержат теги «колесо» и «дверь» одновременно

Продолжение таблицы 6

<p>Работа с изображениями</p>	<p>Добавление изображения в «избранное»</p>	<p>Нажатие кнопки «Добавить в избранное» для изображения, которое уже было добавлено</p>	<p>Будет показано сообщение, что данное изображение уже содержится в «избранном»</p>	<p>Появилось сообщение, что данное изображение уже содержится в «избранном»</p>
<p>Работа в семантическом редакторе</p>	<p>Открытие изображения в редакторе</p>	<p>Нажатие «Открыть в редакторе» под изображением</p>	<p>Произойдет открытие новой страницы с редактором, в редактор будет подгружено выбранное изображение</p>	<p>Появилось новое окно с семантическим редактором, в редакторе изображение соответствует выбранному изображению</p>
	<p>Добавление нового набора объектов</p>	<p>Добавление нового набора, наполнение его объектами, а также изменение другого набора. Нажатие кнопки «Сохранить на сервер»</p>	<p>Изменения всех наборов должны уйти на сервер</p>	<p>На сервере сохранены все изменения в наборах</p>

Продолжение таблицы 6

Работа в семантическом редакторе	Удаление узла объекта	Удаление последнего добавленного узла. Данный узел является последним для объекта	Будет удален весь объект и удаление продолжится в порядке, обратном добавлению узлов	Объект с единственным узлом удален и удаление продолжилось в порядке, обратном добавлению узлов
	Выгрузка результатов разметки в формате png	Нажатие в меню на «Сохранить как...» и выбор формата png	Произойдет загрузка размеченного изображения на локальный компьютер пользователя в папку указанную по умолчанию в браузере	В папку «Загрузки» добавился файл в формате png, содержащий размеченное изображение
	Выгрузка результатов разметки в формате json	Нажатие в меню на «Сохранить как...» и выбор формата json	Произойдет сохранение геометрии и классов выделенных объектов в формат json, с последующей загрузкой файла на компьютер пользователя	В папку «Загрузки» добавился файл в формате json, содержащий геометрию и классы размеченных объектов

В результате тестирования ошибок и сбоев в системе выявлено не было, приложение является работоспособным и может быть использовано в области компьютерного зрения и машинного обучения.

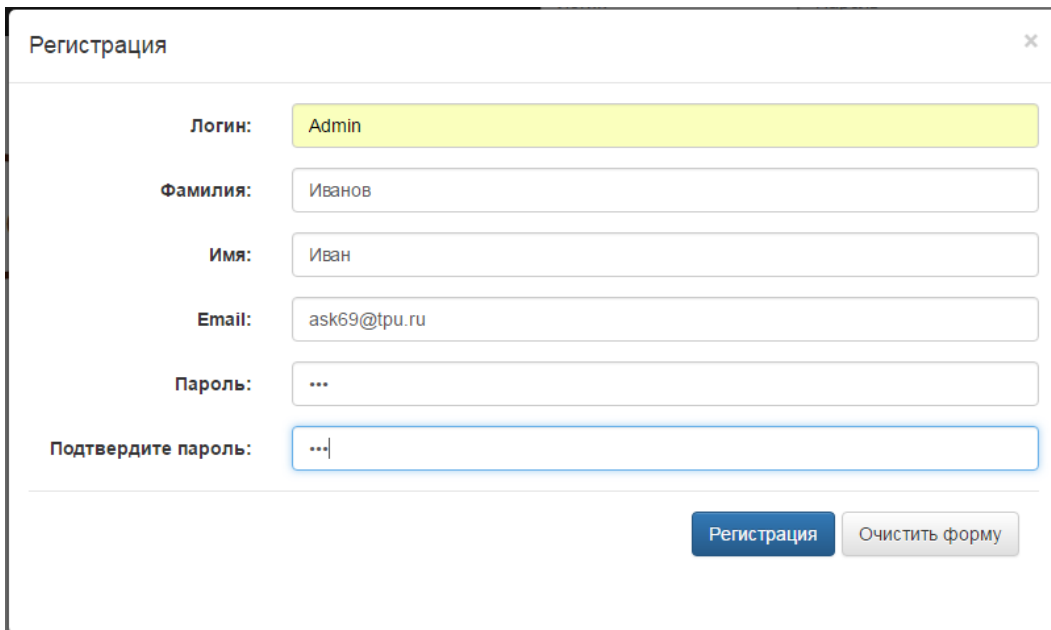
6 РЕЗУЛЬТАТЫ РАБОТЫ

6.1 Функционал приложения

Функционал web-приложения предоставляет пользователям следующие возможности.

1) Регистрация

Пользователю необходимо ввести логин, фамилию, имя, email и пароль. После прохождения процедуры, у зарегистрированного пользователя будет возможность отредактировать свой профиль (рисунок 7).



Регистрация

Логин: Admin

Фамилия: Иванов

Имя: Иван

Email: ask69@tpu.ru

Пароль: ...

Подтвердите пароль: ...

Регистрация Очистить форму

Рисунок 7 – Страница регистрации

2) Загрузка изображения

Существует одиночная и множественная загрузка. При загрузке одного изображения некоторая часть информации подгружается автоматически, такая как: наименование файла, его размер и вес. Остальную информацию заполняет пользователь, к ней относятся: теги, новое наименование файла, описание, дополнительно. Также, пользователю необходимо указать уровень доступа изображения.

Пример добавления одного изображения представлен на рисунке 8.

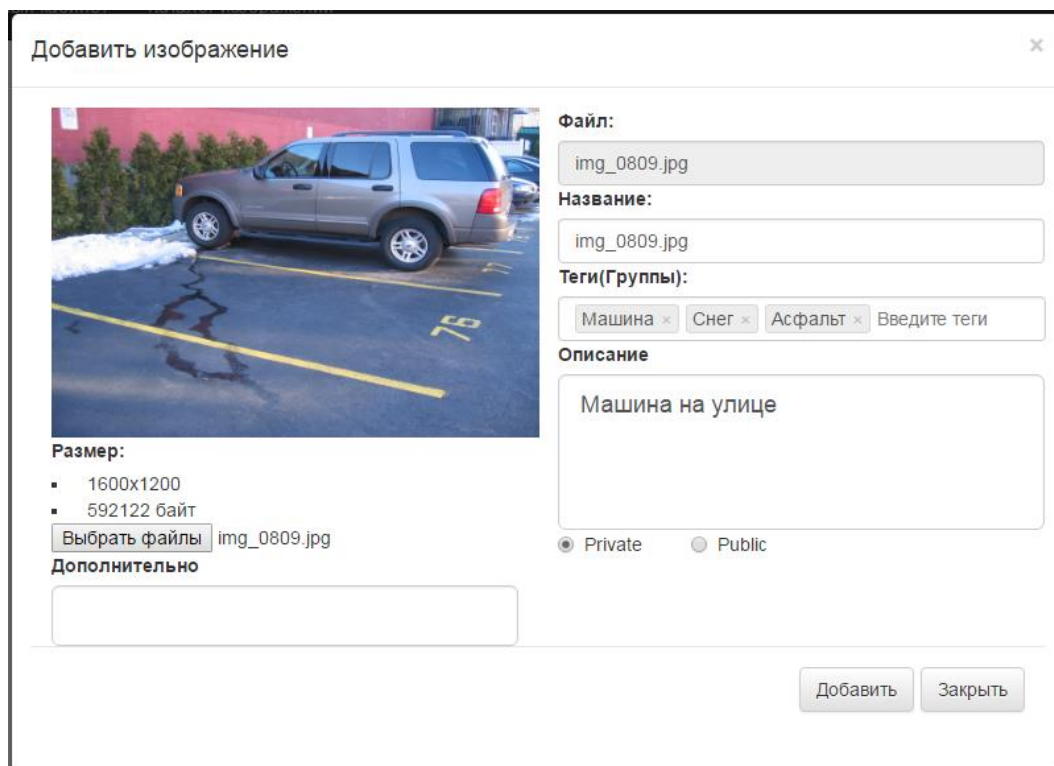


Рисунок 8 – Добавление изображения

При загрузке множества изображений, файлы для загрузки отображаются в виде списка. Пользователь указывает теги, общие для данной группы изображений, а также описание, дополнительную информацию и уровень доступа. Данная информация является общей для всех добавляемых изображений (рисунок 9).

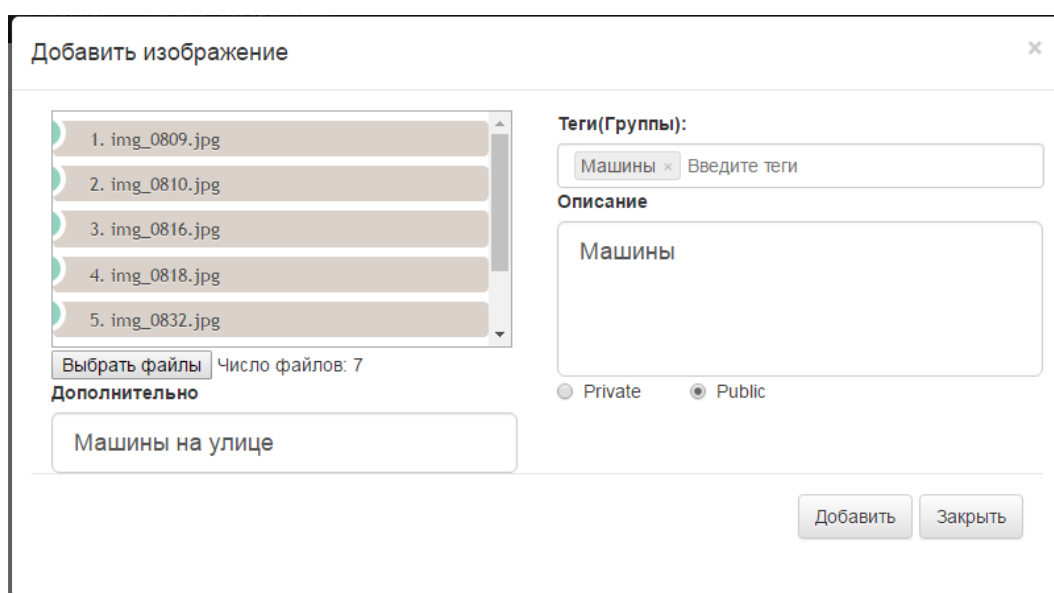


Рисунок 9 – Добавление группы изображений

3) Управление уровнем доступа изображений

Существует два уровня доступа изображений:

- Private – личное (закрытое). Данное изображение будет отображаться только автору, т.е. пользователю, добавившее данное изображение в каталог.

- Public – публичное (открытое). Данное изображение будет отображаться в каталоге всем пользователям, с возможностью его редактирования другими пользователями.

4) Редактирование изображения

При наведении на изображение появляются кнопки для редактирования, удаления, добавления в «избранное», сохранения на компьютер (рисунок 10).

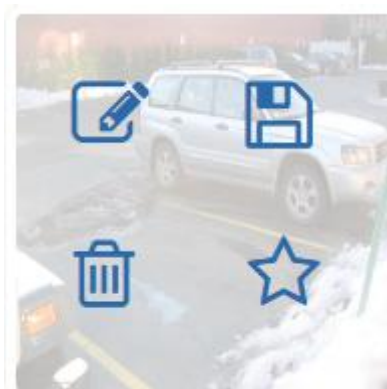


Рисунок 10 – Режим наведения на изображение

Страница редактирования аналогична странице добавления изображения.

5) Просмотр галереи изображений (общего каталога)

Общий каталог содержит все изображения с открытым уровнем доступа, а также личные изображения, автором которых является текущий пользователь (рисунок 11).

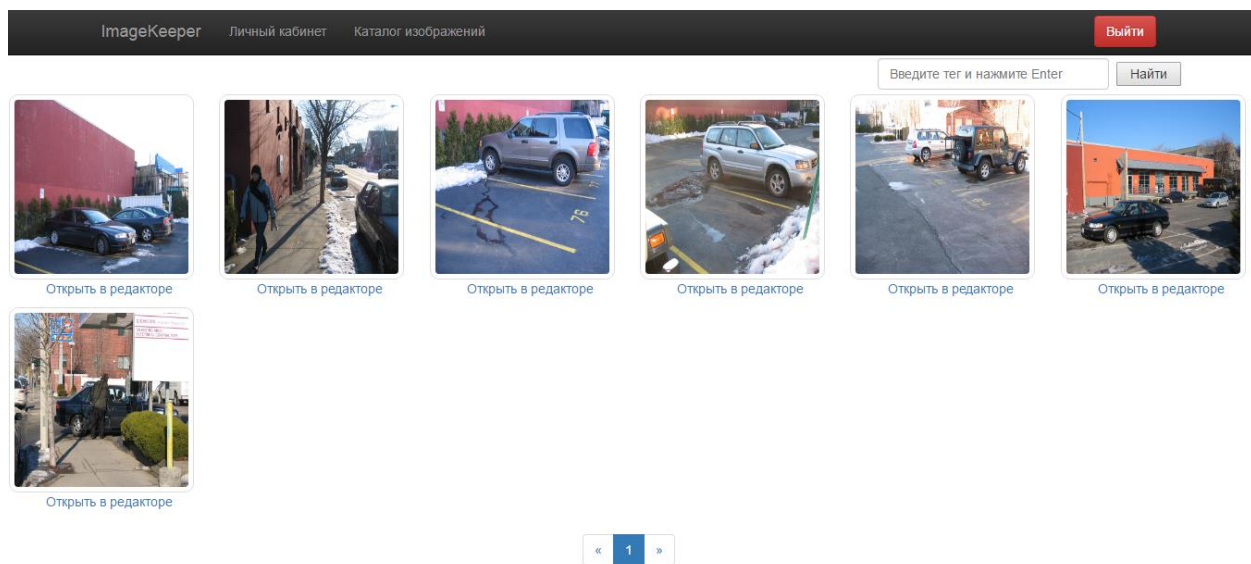


Рисунок 2 – Общий каталог изображений

6) Добавить изображение из общего каталога к себе (избранное)

Осуществляется с помощью нажатия на звездочку в режиме наведения мыши на изображение. После этого изображение будет отображаться в личном кабинете. Соответствие между пользователем и изображением поддерживается с помощью сводной таблицы UserImage в базе данных.

7) Осуществлять поиск изображений в каталоге по тегам

Изображение будет считаться найденным, если в его списке тегов присутствуют все теги, введенные для поиска. Результат поиска изображений с тегом «Снег» в каталоге изображений представлен на рисунке 12.

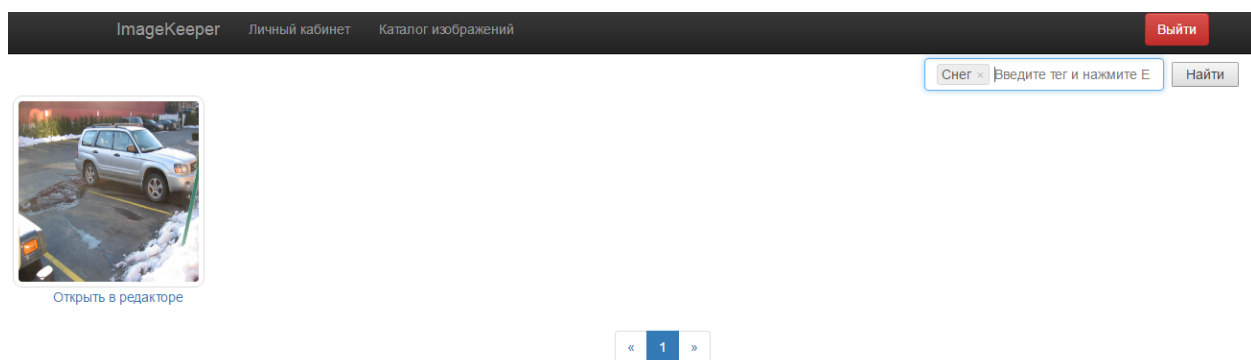


Рисунок 3 – Результат поиска по тегу «Снег»

Изменение изображений, отображаемых после ввода поисковых тегов, происходит с помощью технологии AJAX.

8) Выделение объектов на изображении.

Разработанное web-приложение позволяет выделять объекты на изображении, относить их к определенным классам, а также объединять изображения в наборы объектов. Для одного изображения может быть создано несколько наборов.

Страница редактора изображения, состоит из меню, изображения с размеченными объектами, списка добавленных объектов, панели редактирования. Размер изображения фиксирован и составляет 640x480 (рисунок 13).

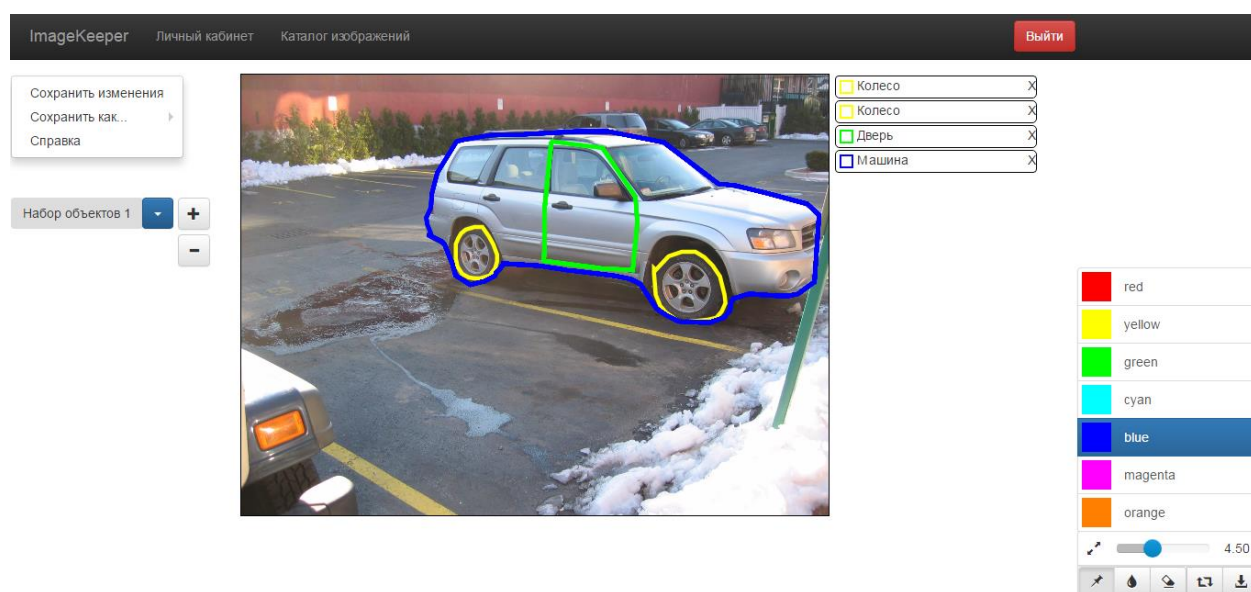


Рисунок 4 – Семантический редактор

9) Выгрузка изображения в формате PNG

На странице редактора расположено меню, в котором содержится возможность сохранения изображения на компьютер. Также, изображение выгружается вместе с размеченными объектами (рисунок 14).



Рисунок 5 – Экспортированное изображение

10) Загрузка и выгрузка объектов в формате JSON

Web-приложение позволяет сохранять объекты в формате JSON. Одновременная выгрузка возможна только одного набора объектов (рисунок 15).

```

▼ 0 {2}
imageId : 58f5ec3c66aa5cc41587fd84
objects : [{"points":[{"x":86,"y":274.890625},
{"x":81,"y":293.890625},
{"x":157,"y":311.890625},
{"x":160,"y":285.890625},
{"x":86,"y":274.890625}], "borderColor": "#f0000", "borderSize": 2, "fillingColor": "transparent", "name": "Homep"}, {"points":
[{"x":323,"y":296.890625},
{"x":315,"y":341.890625},
{"x":315,"y":377.890625},
{"x":317,"y":396.890625},
{"x":326,"y":408.890625},
{"x":340,"y":393.890625},
{"x":349,"y":358.890625},
{"x":350,"y":324.890625},
{"x":349,"y":298.890625},
{"x":338,"y":277.890625},
{"x":325,"y":280.890625},
{"x":323,"y":296.890625}], "borderColor": "#0000ff", "borderSize": 2, "fillingColor": "transparent", "name": "Koleco"}], {"points":
[{"x":463,"y":300.890625},
{"x":455,"y":325.890625},

```

Рисунок 6 – Набор объектов в формате JSON

Реализованное web-приложение применяется для семантической разметки изображений, используемых в области компьютерного зрения. В частности, подготовленные тестовые данные могут использоваться для обучения нейронных сетей. Искусственные нейронные сети позволяют автоматизировать процесс считывания номеров автомобилей, анализа медицинских снимков, дефектоскопии на производстве, распознавание лиц и многое другое.

7 СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ

Обеспечение производственной и экологической безопасности является необходимым условием реализации любых проектов, в том числе конструкторских и исследовательских. Обеспечение безопасности, в общем, предполагает создание безопасных и благоприятных рабочих условий для всех лиц, задействованных в работах, предусмотренных проектом, а также условий, обеспечивающих экологическую безопасность окружающей среды.

Первичным этапом в задаче обеспечения безопасности труда является выявление возможных причин потенциальных несчастных случаев, производственных травм, профессиональных заболеваний, аварий и пожаров. Дальнейшими этапами являются разработка мероприятий по устранению выявленных причин и их реализация. Потенциальные причины и риски, а также конкретный набор мероприятий по их устранению, определяются спецификой выполняемых работ и априорными условиями труда (в частности, видом и состоянием рабочих мест исполнителей) [15].

Данная работа посвящена разработке веб-приложения для семантической разметки изображений. Основными исполнителями работы являются программисты. Объектом исследования является компьютерное зрение и применение web-репозитория для решения задач кластеризации и классификации изображений.

На рабочем месте возможно возникновение вредных факторов, таких как: недостаточная освещенность рабочего места, повышение уровня шума, повышенная либо пониженная температура воздуха. Также на данном рабочем месте могут иметь место проявления опасных факторов среды, например, поражение электрическим током. Возможной чрезвычайной ситуацией на рабочем месте является лишь возникновение пожара.

7.1 Производственная безопасность

Производственные условия на рабочем месте характеризуются наличием различных опасных и вредных производственных факторов, оказывающих

негативное влияние на работников. Под вредными факторами, понимают такие факторы трудового процесса и рабочей среды, которые характеризуются потенциальной опасностью для здоровья, в частности способствуют развитию каких-либо заболеваний, приводят к повышенной утомляемости и снижению работоспособности. При этом вредные факторы проявляются при определенных условиях таких как интенсивность и длительность воздействия. Опасные производственные факторы способны моментально оказать влияние на здоровье работника: привести к травмам, ожогам или к резкому ухудшению здоровья работников в результате отравления или облучения [16].

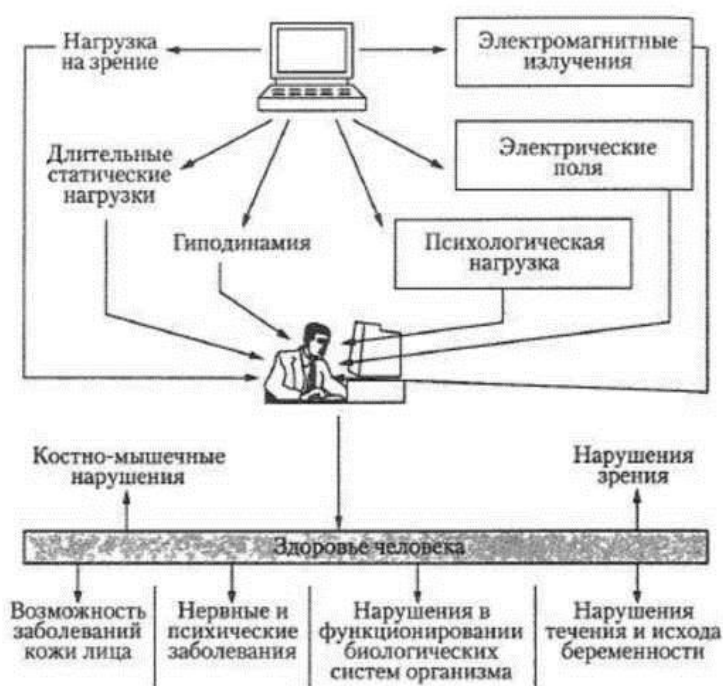


Рисунок 16 – Факторы отрицательного воздействия компьютера на здоровье человека

Сегодня в стране более 50% лиц, работающих на ПЭВМ (персональной электронной вычислительной машине) или совмещающих эту работу с другой, имеют функциональные отклонения в состоянии здоровья. Защита от отрицательного воздействия ПЭВМ является одной из важнейших медико-биологических и социальных задач.

Факторы отрицательного воздействия ПЭВМ на человека показаны на рисунке 16.

7.1.1 Анализ выявленных вредных факторов при разработке и эксплуатации проектируемого решения

7.1.1.1 Отклонение показателей микроклимата в помещении

Одним из необходимых благоприятных условий труда является обеспечение в помещениях нормальных метеорологических условий, оказывающих существенное влияние на самочувствие человека. Метеорологические условия в производственных помещениях (микроклимат), зависят от ряда особенностей технологического процесса, а также внешних условий (климата, сезона, условий вентиляции и отопления) [17].

Для безопасной работы необходимо соблюдать показатели микроклимата. Ниже приведены оптимальные величины показателей микроклимата на рабочих местах производственных помещений по СанПиН 2.2.4.548-96 (табл. 7). Работа программиста относится к категории 1а, потому что уровень энергозатрат не превышает 139 Вт.

Таблица 7 - Допустимые параметры микроклимата на рабочем месте

Период года	Категория работы	Температура воздуха, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	1а	20 - 25	15 - 75	0,1
Теплый	1а	21 - 28	15 - 75	0,1

Оптимальные значения перечисленных параметров для работ с ПК, установленные санитарными нормами, приведены в таблице 8.

Таблица 8 - Оптимальные значения показателей микроклимата

Период года	Температура воздуха, °С	Относительная влажность воздуха, %	Скорость движения
Холодный	22-24	40-60	0,1
Теплый	23-25	40-60	0,1

7.1.1.2 Недостаточная освещенность помещений

Для работы за компьютером важное значение имеет освещение кабинета. Недостаточная освещенность приводит к снижению контрастной чувствительности, понижению остроты зрения.

Освещение должно включать в себя как естественное, так и искусственное. Для источников искусственного освещения применяют люминесцентные лампы типа ЛБ.

Минимальный размер объекта различия входит в диапазон 0,5 до 1,0 мм, следовательно, работа относится к разряду IV. Подразряд Г, т.к. контраст объектов различия с фоном большой, сам фон светлый. В соответствии с СП 52.13330.2011 норма освещенности в кабинете должна быть $E_n = 200$ лк [18].

Пульсация при работе с компьютером не должна превышать 5%.

Увеличение коэффициента пульсации освещенности снижает зрительную работоспособность, повышает утомляемость, воздействует на нервные элементы коры головного мозга и фоторецепторные элементы сетчатки глаз.

Для снижения пульсации необходимо использовать светильники, в которых лампы работают от переменного тока частотой 400 Гц и выше [19].

7.1.2 Анализ выявленных опасных факторов при разработке и эксплуатации проектируемого решения

7.1.2.1 Электробезопасность

Одним из выявленным опасных факторов является поражение электрическим током, так как напряжение считается безопасным при $U < 42$ В, а вычислительная техника питается от сети 220 В с частотой 50 Гц. Ток является опасным, так как ток с частотой 20-100 Гц наиболее опасен. Результатом воздействия на организм человека электрического тока могут быть электрические травмы, электрические удары, и даже смерть.

Виды электротравм: местные электротравмы, к ним относятся: электрический ожог, электрические знаки, металлизация кожи, механические

повреждения). Особую опасность представляют электрические травмы, которые выглядят в виде ожогов.

Электрический ожог возникает на том месте тела человека, в котором контакт происходит с токоведущей частью электроустановки. Электроожоги сопровождаются кровотечениями, омертвением отдельных участков тела. Лечатся они гораздо труднее и медленнее обычных термических.

В результате механического повреждения могут разорваться кровеносные сосуды, нервные ткани, а также случаются вывихи суставов и даже переломы костей. Такие повреждения могут возникнуть в результате сокращений мышц под действием тока, который проходит через тело человека.

Электрические знаки в основном безболезненны, они могут возникнуть у 20% пострадавших от тока. Иногда электрические знаки выглядят в виде царапин, ушибов, бородавок, мозолей, также они представляют собой серые или бледно-желтые пятна круглоовальной формы с углублением в центре [20].

Чтобы защититься от поражения током, необходимо:

- обеспечить недоступность токоведущих частей от случайных прикосновений;
- электрическое разделение цепи;
- устранять опасности поражения при проявлении напряжения на разных частях.

При работе с компьютером при прикосновении к его элементам могут возникнуть токи статического электричества, которые в свою очередь имеют свойство притягивать пыль и мелкие частицы к экрану. Пыль на экране ухудшает видимость, а при подвижности воздуха может попасть на кожу лица и в легкие, что вызывает заболевание кожи и дыхательных путей.

Для предотвращения этого существуют специальные шнуры питания с заземлением и экраны для снятия статического электричества, это поможет защититься от статического электричества, а также необходимо проводить регулярную влажную уборку рабочего помещения.

По электробезопасности рабочее место относится к помещениям без повышенной опасности поражения людей электрическим током характеризуются отсутствием условий, создающих повышенную или особую опасность. К ним относятся жилые помещения, лаборатории, конструкторские бюро, заводоуправление, конторские помещения и другие [21].

Степень воздействия зависит от продолжительности работы и индивидуальных особенностей организма.

Для снижения уровня воздействия, необходимо:

- экранирование экрана монитора;
- соблюдать оптимально расстояние от экрана;
- рационально размещать оборудование (если имеется несколько компьютеров, то расстояние между боковыми и задними стенками компьютеров должно быть 1,22 м);
- организовывать перерывы 10-15 минут через каждые 45-60 минут работы.

7.1.2.2 Пожаровзрывобезопасность

Одними из наиболее вероятных и разрушительных видов ЧС являются пожар или взрыв на рабочем месте. Пожарная безопасность представляет собой единый комплекс организационных, технических, режимных и эксплуатационных мероприятий по предупреждению пожаров и взрывов [22].

Причинами возгораний в рабочей зоне являются:

- резкие перепады напряжения;
- короткое замыкание в проводке, когда рубильник не отключен;
- пожар в соседнем помещении;
- короткое замыкание в розетке.

7.2 Экологическая безопасность

Непосредственно с выполнением данной работы, могут быть связаны негативно влияющие на экологию факторы, сопутствующие эксплуатации ПК. В частности, аспектами негативного влияния являются, отходы и выбросы,

имеющие место на этапе производства ПК, а также отходы, связанные с неполной их утилизацией. Кроме того, компьютерная техника является набором приборов, потребляющих электроэнергию, в связи с чем, нерациональное их использование может быть также расценено, как необоснованная нагрузка на окружающую среду [23].

Эксплуатация компьютерной техники может сопровождаться следующими негативными факторами влияния на окружающую среду:

- локальное повышение электромагнитного и радиоактивного фона;
- повышение интенсивности звукового фона (слышимый шум, инфра- и ультразвук);
- образование твердых отходов (компьютерный лом, бумага и т.п.) и жидких отходов (сточные воды);
- неоправданное потребление электроэнергии (связано с использованием ПК не на полную мощность в течение всего его рабочего времени) и прочее.

Также в СанПиН 2.2.2/2.4.1340-03 даются следующие общие рекомендации по снижению опасности для окружающей среды, исходящей от компьютерной техники:

- применять оборудование, соответствующее санитарным нормам и стандартам экологической безопасности;
- применять расходные материалы с высоким коэффициентом использования и возможностью их полной или частичной регенерации;
- отходы в виде компьютерного лома утилизировать;
- использовать экономные режимы работы оборудования.

На основе выполненного выше анализа стоит отметить, что современные ПК практически не оказывают негативного влияния на окружающую среду, посредством электромагнитных (в разных диапазонах частот спектра) излучениях. Кроме того, для современных ПК характерен низкий уровень производимых шумов. Таким образом, при дальнейшем

рассмотрении проблемы целесообразно остановиться на последних двух факторах влияния.

При использовании данных рекомендаций, возможно, существенно сократить наносимый экологии вред и снизить действующие значения вредных факторов до приемлемого минимума [24].

7.3 Безопасность в чрезвычайных ситуациях

Вследствие того, что ведется работа с компьютером, существует возможность короткого замыкания или неисправности проводов. Питаются электроустановки посредством кабельных линий, являющихся особо пожароопасными. Именно поэтому наиболее типичная чрезвычайная ситуация – это пожар.

Так как часть данных в базе изображений является скрытой, ещё одной угрозой может являться кибертерроризм. Сущность кибертерроризма заключается в оказании противоправного воздействия на информационные системы, совершенного в целях создания опасности причинения вреда жизни, здоровью или имуществу неопределенного круга лиц путем создания условий для аварий и катастроф техногенного характера либо реальной угрозы такой опасности [25].

7.3.1 Инструкция безопасности внутри помещений

В рабочее время каждый работник должен:

- Постоянно содержать в чистоте и порядке свое рабочее место;
- Проходы, выходы не загромождать различными предметами и оборудованием;
- Не допускать нарушение пожарной безопасности со стороны посторонних лиц;
- Протирать полы, стены и оборудование горючими растворами запрещается;
- Не подключать самовольно электроприборы, исправлять эл. сеть и предохранители;

- Не пользоваться открытым огнем в служебных и рабочих помещениях;
- Не курить, не бросать окурки и спички в служебных и рабочих помещениях;
- Не накапливать и не разбрасывать бумагу и другие легковоспламеняющиеся материалы и мусор;
- Не пользоваться электронагревательными приборами в личных целях с открытыми спиралями;
- Не оставлять включенными без присмотра электрические приборы и освещение;
- Не вешать плакаты, одежду и другие предметы на электророзетки, выключатели и другие электроприборы.

7.3.2 Обязанности работающих в помещениях

В обязанности работающих в помещениях входит:

- Знать внутреннюю планировку здания помещении, расположение лестничных клеток, основных и запасных эвакуационных выходов, средств пожаротушения (огнетушителей, внутренних пожарных кранов).
- Уметь пользоваться средствами пожаротушения, знать их тактико-технические данные.
- Неукоснительно подчиняться сигналам оповещения о пожаре, срочно покидать помещении.

7.3.3 Необходимые действия при возникновении пожара в помещении

Действия при возникновении пожара в помещении:

- Немедленно сообщить о случившемся в службу спасения по телефону 01.
- Приступить к тушению пожара имеющимися в помещении средствами пожаротушения.
- Если ликвидировать очаг пожара своими силами не представляется возможным, выйти из помещению и закрыть дверь, не запирая ее на замок.

7.4 Правовые и организационные вопросы обеспечения

безопасности

7.4.1 Правовые нормы трудового законодательства

Продолжительность рабочего дня не должна превышать 40 часов в неделю. Возможно сокращение рабочего времени. Для работников, возраст которых меньше 16 лет – не более 24 часа в неделю, от 16 до 18 лет – не более 35 часов, как и для инвалидов I и II группы. Также рабочее время зависит от условий труда: для работников, работающих на рабочих местах с вредными условиями для жизни – не больше 36 часов в неделю

Светопроемы должны быть ориентированы преимущественно на север и северо-восток. В рабочем кабинете имеется три окна, два ориентированные на север и одно на северо-запад.

Площадь на одно рабочее место с ПЭВМ для взрослых пользователей должна составлять не менее 6,0м² [26].

Требования к организации и оборудованию рабочих мест с ПЭВМ по СанПиН 2.2.2/2.4.1340–03 представлены в таблице 9. Таблица 9 – Требования к организации и оборудованию рабочих мест с ПЭВМ

Требование	Требуемые значения параметров	Значения параметров в комнате
Высота рабочей поверхности стола	680 – 800 мм	740 мм
Расположение монитора от глаз пользователя	600 – 700 мм	640 мм
Расположение клавиатуры на поверхности стола от края	100 – 300 мм	190 мм
Высота стула над полом (для роста 161170 см)	420 мм	420 мм
Угол наклона монитора	0 – 30 градусов	5 градусов

Рабочее место также необходимо оборудовать подставкой для ног, которая отсутствует на рабочем месте. Рабочий стул должен быть подъемно – поворотным, для регулировки высоты и угла наклона [27].

7.4.2 Организационные мероприятия обеспечения безопасности

Рабочее место должно обеспечивать возможность удобного выполнения работ, учитывать размеры рабочей зоны, а также необходимость передвижения в ней работающего.

Невыполнение требований к расположению и компоновке рабочего места может привести к получению работником производственной травмы или развития у него профессионального заболевания. Рабочее место должно соответствовать требованиям ГОСТ 12.2.032-78.

Конструкция оборудования и рабочего места при выполнении работ в положении сидя должна обеспечивать оптимальное положение работающего, которое достигается регулированием высоты рабочей поверхности, высоты сидения, оборудованием пространства для размещения ног и высотой подставки для ног. Оптимальные параметры рабочего места при работе с ЭВМ представлены в таблице 10:

Таблица 10 – Оптимальные параметры рабочего места при работе с ЭВМ

Параметры	Значение параметра	Реальные значения
Высота рабочей поверхности стола	От 600 до 800 мм	700
Высота от стола до клавиатуры	Около 20 мм	20
Высота клавиатуры	600–700, мм	600
Удаленность клавиатуры от края стола	Не менее 80 мм	100
Удаленность экрана монитора от глаз	500–700, мм	500
Высота сидения	400–500, мм	500
Угол наклона монитора	0–30, град.	20
Наклон подставки ног	0–20, град.	0

Конструкция рабочего кресла поддерживает рациональную рабочую позу, а также позволяет изменить позу, чтобы снизить статическое напряжение мышц шейно–плечевой области и спины, это позволяет предотвратить утомление.

Не рекомендуется располагать компьютеры рядом друг с другом в целях уменьшения действия переменного электрического поля [28].

Разработанный продукт позволит наполнять общую базу изображений, что сократит время на поиск необходимых для обучения изображений. Кроме того, существует множественная загрузка изображений на сервер, что обобщает изображения с общей тематикой и сокращает время на добавление однотипных изображений и исключает дублирование общей для добавляемых изображений информации.

Благодаря встроенному сематическому редактору, специалисты в области компьютерного зрения смогут производить семантическую разметку изображений. Интерфейс данного редактора прост и интуитивно понятен, а функциональные возможности удовлетворяют требованиям.

Разработанные программные средства позволят улучшить условия работы специалистов в области компьютерного зрения, снизить трудозатраты на подготовку тестовых данных и создать репозиторий для хранения изображений.

РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ**Введение**

Целью раздела «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение» является проектирование и создание конкурентоспособных разработок, технологий, отвечающих современным требованиям в области ресурсоэффективности и ресурсосбережения.

Данные программные средства позволят сократить время и трудозатраты на подготовку данных для решения задач машинного обучения и компьютерного зрения.

Для достижения поставленной цели было необходимо решить следующие задачи:

- Выполнить обзор существующих решений в исследуемой области.
- Исследовать технологии, библиотеки, фреймворки и прочее программное обеспечение, которое может быть использовано в процессе разработки.
- Реализовать web-репозиторий для хранения изображений.
- Создать и наполнить базу изображений.
- Протестировать приложение, составить протокол тестирования.
- Разработать программные средства для выполнения семантической сегментации изображений.
- Актуальность разработки
- 1. Экономическая актуальность

В результате выполнения проекта произойдет снижение затрат времени на подготовку тестовых данных, а также трудозатрат работников, что повысит скорость анализа и классификации изображений.

- 2. Социальная актуальность

Методы компьютерного зрения могут быть использованы в микроскопии, оптической когерентной томографии, цифровой голографии, робототехнике, космической отрасли. Примером информации, которая может быть получена из таких данных, является обнаружение опухолей, атеросклероза или других злокачественных изменений.

Также данная разработка позволит улучшить условия работы специалистов в области компьютерного зрения и машинного обучения.

- 3. Техническая актуальность

Разработанная система является кроссплатформенной. Именно поэтому существует возможность работать с системой с любой программной платформы, на любом устройстве.

Система разработана с использованием современных средств разработки, как при работе с изображениями, так и с большими объемами данных. Данные современные средства отличаются скоростью работы и надежностью.

Критерии эффективности

Таблица 11 – Критерии эффективности разработки

Тип показателя	Показатель
Экономические	Снижение затрат времени на подготовку тестовых данных
Социальные	Улучшение условий работы специалистов в области компьютерного зрения и машинного обучения
Технические	Увеличение скорости анализа и классификации изображений
	Независимость от программной платформы
	Возможность хранения большого количества изображений

8.1 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения

8.1.1 Потенциальные потребители результатов исследования

Для анализа потребителей результатов исследования необходимо рассмотреть целевой рынок и провести его сегментирование.

Целевой рынок представляет собой сегменты рынка, на котором в будущем возможна продажа разработки.

Сегментирование – это разделение покупателей на однородные группы, для каждой из которых может потребоваться определенный товар.

Произведем анализ рынка потенциальных потребителей. Поскольку разработка представляет собой прикладное программное обеспечение, предназначенное для семантической разметки изображений, то к первичным потребителям можно отнести людей, занимающихся научно-исследовательской деятельностью в области машинного обучения и компьютерного зрения.

В процессе сегментирования рынка, прикладное программное обеспечение было разбито на группы по его назначению (общего назначения, развлекательного назначения, специального назначения и профессионального назначения). Среди потребителей были выделены 3 группы пользователей программного обеспечения (студенты, специалисты, инженеры).

Таблица 12 – Карта сегментирования рынка

		Прикладное программное обеспечение			
		ПО общего назначения	ПО развлекательного назначения	ПО специального назначения	Профессиональное ПО
Потребители	Студенты				
	Специалисты				
	Инженеры				

Так как разработанное программное обеспечение можно отнести частично к специальному и профессиональному, то основным сегментов рынка выбрана область разработки программного обеспечения для специалистов и инженеров.

В будущем наиболее привлекательным сегментом рынка являются студенты и внедрение разработанного программного обеспечение в образовательный процесс для подготовки специалистов в области компьютерного зрения.

8.1.2 Анализ конкурентных технических решений

Наиболее интересными, с точки зрения аналогий, представляются использование системы VGG Image Annotator и сервиса LabelMe.

VGG Image Annotator – это проект с открытым исходным кодом, разработанный в группе Visual Geometry Group в Оксфордском университете. С помощью этого автономного приложения можно выделять регионы изображений и создавать текстовое описание этих регионов. Такие области

изображения и их описания полезны для контролируемой подготовки алгоритмов машинного обучения.

LabelMe – это инструмент аннотации изображений на основе web-технологий, который позволяет пользователям размечать изображения и делиться аннотациями со всем миром.

Оба конкурента являются аналогами разрабатываемого приложения, поэтому сравнение проводилось именно с ними (табл. 13).

Таблица 13 – Оценочная карта сравнения конкурентных технических решений

Критерии оценки	Вес критерия	Баллы			Конкурентоспособность		
		Б _ф	Б _{к1}	Б _{к2}	К _ф	К _{к1}	К _{к2}
1	2	3	4	5	7	8	9
Технические критерии оценки ресурсоэффективности							
1. Безопасность системы	0.25	4	2	2	1	0.5	0.5
2. Потребность в ресурсах памяти	0.05	4	3	3	0.2	0.15	0.15
3. Функциональная мощность (предоставляемые возможности)	0.3	5	3	4	1.5	0.9	1.2
4. Простота эксплуатации	0.1	4	2	3	0.4	0.2	0.3
Экономические критерии оценки эффективности							
5. Конкурентоспособность продукта	0.05	5	5	4	0.25	0.25	0.2
6. Уровень проникновения на рынок	0.05	1	4	4	0.05	0.2	0.2
7. Поддержка продукта	0.15	5	4	3	0.75	0.6	0.45
8. Цена продукта	0.05	5	3	4	0.25	0.15	0.2
Итого	1				4.4	2.95	3.2

В результате проведения анализа можно заключить, что уязвимость конкурентных технологических решений связана, прежде всего, с недостаточно широкими функциональными возможностями систем и сложностью пользовательского интерфейса для работы с системой.

Преимуществом собственной разработки можно считать то, что данный продукт на рынке является уникальным. Полных аналогов разрабатываемой системы не существует, поэтому под аналогами понимались схожие продукты и решения. Также, сильной стороной системы является простота интерфейса и удобство пользования, в отличие от конкурентов.

8.1.3 SWOT-анализ

На основе анализа рынка и конкурентных технических решений необходимо составить матрицу SWOT-анализа, в которой показаны сильные и слабые стороны проекта, возможности и угрозы для разработки. Матрица SWOT представлена в таблице 14.

Таблица 14 – SWOT-анализ

	Сильные стороны научно-исследовательского проекта:	Слабые стороны научно-исследовательского проекта:
	<p>C1. Использование системы на различных операционных системах.</p> <p>C2. Удобство пользования.</p> <p>C3. Постоянная поддержка системы.</p> <p>C4. Новизна и актуальность разрабатываемой системы.</p> <p>C5. Простота интерфейса.</p> <p>C6. Выгрузка изображений в различные распространённые и удобные для работы форматы.</p>	<p>Сл1. Неполная адаптация системы под мобильные устройства.</p> <p>Сл2. Уязвимости в безопасности при хранении паролей пользователей в cookies.</p>

Продолжение таблицы 14

<p>Возможности: В1. Присутствие новых возможностей системы. В2. Возможность работы системы на разных платформах.</p>	<p>Направления развития: В2С1С2 – модификация системы под мобильные устройства. В1С3 – повышение безопасности системы.</p>	<p>Сдерживающие факторы: В2С2 – необходимость поддержки безопасности системы на разных платформах.</p>
<p>Угрозы: У1. Развитие и появление аналогов разрабатываемой системы. У2. Непопулярность продукта на рынке.</p>	<p>Угрозы развития: У1С4 – развитие аналогов могут лишить систему преимуществ в новизне. У2С2С3С5 – непопулярность продукта на рынке лишает систему преимущества в постоянной поддержке, простоте интерфейса и удобстве пользования.</p>	<p>Уязвимости: У1Сл1 – становятся более очевидными недостатки системы перед конкурентами (неполная поддержка мобильных устройств).</p>

На втором этапе проведения SWOT-анализа проводится составление интерактивных матриц проекта, в которых производится анализ соответствия параметров SWOT каждого с каждым. Соотношения параметров представлены в таблицах 15-18.

Таблица 15 – Интерактивная матрица для сильных сторон и возможностей

Сильные стороны проекта							
Возможности проекта		C1	C2	C3	C4	C5	C6
	B1	-	-	+	-	-	0
	B2	+	+	-	-	-	-

Таблица 16 – Интерактивная матрица для слабых сторон и возможностей

Слабые стороны проекта			
Возможности проекта		Сл1	Сл2
	B1	-	-
	B2	-	+

Таблица 17 – Интерактивная матрица для сильных сторон и угроз

Сильные стороны проекта							
Угрозы проекта		C1	C2	C3	C4	C5	C6
	У1	-	-	-	+	-	-
	У2	-	+	+	-	+	-

Таблица 18 – Интерактивная матрица для слабых сторон и угроз

Слабые стороны проекта			
Угрозы проекта		Сл1	Сл2
	У1	+	-
	У2	-	-

8.2 Определение возможных альтернатив проведения научных исследований

Для определения альтернативных путей проведения научных исследований и вариантов реализации технической части используется морфологический подход. Морфологическая матрица для составляющих реализации рассматриваемой разработки представлена в таблице 19.

Таблица 19 – Морфологическая матрица

	1	2	3
А. Фреймворк для разработки системы	Node.js Express	ASP.NET MVC	Yii
Б. Фреймворк для разработки интерфейса	Bootstrap, Kendo UI	Bootstrap, jQuery UI	Yiistrap
В. Операционная система	Windows	Unix	Кроссплатформенность
Г. Среда разработки	JetBrains WebStorm	Visual Studio Code	Sublime Text
Д. Хранение данных системы	Хранение данных в базе данных	Хранение в кэше системы	Хранение данных в облачном хранилище
Е. Исполнитель	Один студент	Два студента	Программист

Из полученной морфологической матрицы выделим 3 наиболее функционально востребованных реализаций проекта:

- Исполнение 1. A1B2B3Г1Д1Е2
- Исполнение 2. A2B2B1Г2Д2Е1
- Исполнение 3. A3B3B3Г3Д3Е3

8.3 Планирование научно-исследовательских работ

8.3.1 Структура работ в рамках научного исследования

При организации работ в рамках научно-исследовательской работы необходимо планировать занятость каждого участника проекта в работе. На данном этапе определяется полный перечень работ, а также распределение времени работ между всеми участниками. В качестве структуры, показывающей необходимые данные, используется линейный график работ, представленный в таблице 20.

Таблица 20 – Перечень этапов, работ и распределение исполнителей

Основные этапы	№ работ	Содержание работ	Должность исполнителя
Разработка и утверждение функциональных возможностей системы	1	Постановка задач	Руководитель
	2	Составление технической документации	Студенты
Анализ исследуемой области	3	Обзор существующих решений в исследуемой области	Студенты
	4	Исследование технологий, библиотек, фреймворков необходимых для разработки	Студенты
	5	Проектирование системы	Студенты
	6	Утверждение выбранных технологий и проекта	Руководитель, студенты
Реализация	7	Реализация серверной части	Студенты
	8	Реализация клиентской части	Студенты
	9	Реализация функционала для семантической разметки изображений	Студенты

Продолжение таблицы 20

Тестирование системы	10	Проведение тестирования системы	Студенты
	11	Составление протокола тестирования	Студенты
Ввод системы в эксплуатацию	12	Наполнение базы данных изображениями	Студенты
Ввод системы в эксплуатацию	13	Развертывание системы на хостинге	Студенты
Анализ и оформление результатов	14	Оценка полученных результатов	Руководитель, студенты
	15	Составление пояснительной записки	Студенты

8.3.2 Определение трудоемкости выполнения работ

Для определения трудоемкости выполнения работ необходимо на основе экспертной ожидаемой трудоемкости выполнения каждой работы рассчитать длительность работ в рабочих и календарных днях для каждого из вариантов исполнения работ последующим формулам:

$$t_{ож\ i} = \frac{3t_{\min\ i} + 2t_{\max\ i}}{5}$$

Где $t_{ож\ i}$ – ожидаемая трудоемкость выполнения i -ой работы чел.-дн.;

$t_{\min\ i}$ – минимально возможная трудоемкость выполнения заданной i -ой работы (оптимистическая оценка), чел.-дн.;

$t_{\max\ i}$ – максимально возможная трудоемкость выполнения заданной i -ой работы (пессимистическая оценка), чел.-дн.

Исходя из ожидаемой трудоемкости работ, определяется продолжительность каждой работы в рабочих днях T_p , учитывающая параллельность выполнения работ несколькими исполнителями. Такое вычисление необходимо для обоснованного расчета заработной платы, так как удельный вес зарплаты в общей сметной стоимости научных исследований составляет около 65 %.

$$T_{pi} = \frac{t_{ожi}}{Ч_i}$$

Где T_{pi} – продолжительность одной работы, раб. дн.;

$t_{ожi}$ – ожидаемая трудоемкость выполнения одной работы, чел.-дн.;

$Ч_i$ – численность исполнителей, выполняющих одновременно одну и ту же работу на данном этапе, чел.

Над выполнением дипломного проекта на протяжении всего времени участвовало два студента.

Расчеты продолжительности работ представлены в таблице 21.

8.3.3 Разработка графика проведения научного исследования

Для сравнительно небольших по объему научных работ, наиболее удобным и наглядным является построение ленточного графика проведения научных работ в форме диаграммы Ганта.

Диаграмма Ганта – горизонтальный ленточный график, на котором работы по теме представляются протяженными во времени отрезками, характеризующимися датами начала и окончания выполнения данных работ.

Для удобства построения графика, длительность каждого из этапов работ из рабочих дней следует перевести в календарные дни. Для этого необходимо воспользоваться следующей формулой:

$$T_{ki} = T_{pi} \cdot k_{кал},$$

где T_{ki} – продолжительность выполнения i -й работы в календарных днях;

T_{pi} – продолжительность выполнения i -й работы в рабочих днях;

$k_{кал}$ – коэффициент календарности.

Коэффициент календарности определяется по следующей формуле:

$$k_{кал} = \frac{T_{кал}}{T_{кал} - T_{вых} - T_{пр}},$$

где $T_{\text{кал}}$ – количество календарных дней в году;

$T_{\text{вых}}$ – количество выходных дней в году;

$T_{\text{пр}}$ – количество праздничных дней в году.

Рассчитанные значения в календарных днях по каждой работе T_{ki} необходимо округлить до целого числа.

Для расчета коэффициент календарности подсчитаем количество рабочих и выходных дней в 2017 году. Всего в году 247 рабочих дней и 118 выходных и праздничных дней. Исходя из полученных данных, рассчитывается коэффициент календарности:

$$k_{\text{кал}} = \frac{T_{\text{кал}}}{T_{\text{кал}} - T_{\text{вых}} - T_{\text{пр}}} = \frac{365}{247} = 1,4778$$

На основе таблицы 21 построен календарный план-график проведения работ (табл. 22).

Таблица 21 – Временные показатели проведения научного исследования

Название работы	Исполнители	Трудоёмкость работ									Длительность работ в рабочих днях T_{pi}	Длительность работ в календарных днях T_{ki}				
		t_{min} , чел-дни			t_{max} , чел-дни			$t_{ожл}$, чел-дни				Исп.1	Исп.2	Исп.3		
		Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3						
Постановка задач	Руководитель	4	4	2	9	9	4	6	6	3	3	6	3	4	8	4
Составление технической документации	Студенты	10	10	3	25	20	8	16	14	5	8	14	5	12	20	8
Обзор существующих решений в исследуемой области	Студенты	2	2	1	3	3	1	2	2	1	1	2	1	2	4	2
Исследование технологий, библиотек, фреймворков необходимых для разработки	Студенты	2	2	1	3	3	3	2	2	2	1	2	2	2	4	3
Проектирование системы	Студенты	5	4	2	8	5	4	6	4	3	3	4	3	4	6	4
Утверждение выбранных технологий и проекта	Руководитель, студенты	2	2	1	3	3	1	2	2	1	1	2	1	2	4	2
Реализация серверной части	Студенты	30	24	10	55	29	17	40	26	13	20	26	13	30	40	20

Продолжение таблицы 21

Реализация клиентской части	Студенты	20	20	8	45	45	11	30	30	9	15	30	9	22	44	14
Реализация функционала для семантической разметки изображений	Студенты	18	16	5	33	24	10	24	22	7	12	22	7	18	32	10
Проведение тестирования системы	Студенты	10	10	3	25	20	8	16	14	5	8	14	5	12	22	8
Составление протокола тестирования	Студенты	8	10	2	13	20	7	10	14	4	5	14	4	8	20	6
Наполнение базы данных изображениями	Студенты	2	4	2	7	9	4	4	6	3	2	6	3	3	8	5
Развертывание системы на хостинге	Студенты	8	4	2	13	9	4	10	6	3	5	6	3	7	10	4
Оценка полученных результатов	Руководитель, студенты	4	2	2	9	7	4	6	4	3	3	4	3	4	6	4
Составление пояснительной записки	Студенты	14	20	6	24	35	11	22	26	8	11	26	8	16	40	12
Итого								196	178	70	98	178	70	146	268	106

Таблица 22 – Календарный план-график проведения работ

№ работ	Вид работ	Исполнители	T _{ки} , кал. дн.	Продолжительность выполнения работ														
				январь			февраль			март			апрель			май		
				1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
1	Постановка задач	Руководитель	4	■														
2	Составление технической документации	Студенты	12	■	■													
3	Обзор существующих решений в исследуемой области	Студенты	2		■													
4	Исследование технологий, библиотек, фреймворков необходимых для разработки	Студенты	2		■													
5	Проектирование системы	Студенты	4			■												
6	Утверждение выбранных технологий и проекта	Руководитель, студенты	2			■	■											
7	Реализация серверной части	Студенты	30			■	■	■	■									
8	Реализация клиентской части	Студенты	22					■	■	■								
9	Реализация функционала для семантической разметки изображений	Студенты	18								■	■	■					

8.4 Бюджет научно-технического исследования

В состав бюджета выполнения работ по научно-технической работе включает вся себя стоимость всех расходов, необходимых для их выполнения. При формировании бюджета используется группировка затрат по следующим статьям:

- основная заработная плата исполнителей темы;
- дополнительная заработная плата исполнителей темы;
- отчисления во внебюджетные фонды (страховые отчисления);
- накладные расходы.

8.4.1 Расчет материальных затрат НТИ

Данная статья включает стоимость всех материалов, используемых при разработке проекта.

Расчет материальных затрат осуществляется по следующей формуле:

$$Z_m = (1 + k_T) \cdot \sum_{i=1}^m \Pi_i \cdot N_{расхи}$$

где m - количество видов материальных ресурсов, потребляемых при выполнении научного исследования;

$N_{расхи}$ – количество материальных ресурсов i -го вида, планируемых к использованию при выполнении научного исследования (шт., кг, м, м² и т.д.);

Π_i – цена приобретения единицы i -го вида потребляемых материальных ресурсов (руб./шт., руб./кг, руб./м, руб./м² и т.д.);

k_T – коэффициент, учитывающий транспортно-заготовительные расходы.

Величина коэффициента (k_T), отражающего соотношение затрат по доставке материальных ресурсов и цен на их приобретение, зависит от условий договоров поставки, видов материальных ресурсов, территориальной удаленности поставщиков и т.д. Транспортные расходы принимаются в пределах 15-25% от стоимости материалов. Материальные затраты, необходимые для данной разработки, заносятся в таблицу 23.

Таблица 23 – Материальные затраты

Наименование	Ед. измерения	Количество			Цена за ед., Тыс.руб.			Затраты на материалы, (З _м), тыс.руб.		
		Исп .1	Ис п.2	Ис п.3	Исп .1	Исп. 2	Исп .3	Ис п.1	Исп .2	Исп .3
Амортизация оборудования	шт	1	2	1	4	4	6	4	8	6
JetBrains WebStorm	шт	0	2	0	0	3,6	0	0	7,2	0
Sublime Text	шт	0	0	1	0	0	3,9	0	0	3,9
Облачное хранилище	мес	0	0	5	0	0	10,3	0	0	51,5
Итого								4	15,2	61,4

В ходе разработки использовалось имеющееся оборудование, поэтому в материальные расходы внесены затраты на его амортизацию за пол года.

Также в материальные расходы занесены затраты на приобретение лицензий для используемого программного обеспечения.

8.4.2 Расчет основной заработной платы исполнителей системы

В данную статью расходов включается заработная плата научного руководителя и студентов, а также премии и доплаты. Расчет выполняется на основе трудоемкости выполнения каждого этапа и величины почасовой оплаты работы исполнителей.

Основной расчет фонда заработной платы выполняется по формуле:

$$Z_{зп} = Z_{осн} + Z_{доп}$$

где $Z_{осн}$ – основная заработная плата;

$Z_{доп}$ – дополнительная заработная плата (12-20 % от $Z_{осн}$).

Основная заработная плата ($Z_{осн}$) руководителя (лаборанта, инженера) от предприятия (при наличии руководителя от предприятия) рассчитывается по следующей формуле:

$$Z_{осн} = Z_{дн} \cdot T_p,$$

где $Z_{осн}$ – основная заработная плата одного работника;

T_p – продолжительность работ, выполняемых научно-техническим работником, раб. часов.

$Z_{дн}$ – среднедневная заработная плата работника, руб.

Таблица 24 – Расчет основной заработной платы

№ п/п	Наименование этапов	Исполнители по категориям	Трудо-емкость, чел.-дн.	Заработная плата, приходящаяся на один чел.-дн., тыс. руб.	Всего заработная плата по тарифу (окладам), тыс. руб.
1	Постановка задач	Руководитель	6	1.3	7,8
2	Составление технической документации	Студенты	16	0.5	8
3	Обзор существующих решений в исследуемой области	Студенты	2	0.5	1
4	Исследование технологий, библиотек, фреймворков необходимых для разработки	Студенты	2	0.5	1
5	Проектирование системы	Студенты	6	0.5	3
6	Утверждение выбранных технологий и проекта	Руководитель, студенты	2	1.3, 0.5	3,6
7	Реализация серверной части	Студенты	40	0.5	20
8	Реализация клиентской части	Студенты	30	0.5	14

Продолжение таблицы 24

9	Реализация функционала для семантической разметки изображений	Студенты	24	0.5	12
10	Проведение тестирования системы	Студенты	16	0.5	8
11	Составление протокола тестирования	Студенты	10	0.5	5
12	Наполнение базы данных изображениями	Студенты	4	0.5	2
13	Развертывание системы на хостинге	Студенты	10	0.5	5
14	Оценка полученных результатов	Руководитель, студенты	6	1.3, 0.5	10,8
15	Составление пояснительной записки	Студенты	22	0.5	11
Итого:					104,4

Среднедневная заработная плата рассчитывается по формуле:

$$Z_{\text{дн}} = \frac{Z_{\text{м}} \cdot M}{F_{\text{д}}},$$

где $Z_{\text{м}}$ – месячный должностной оклад работника, руб.;

M – количество месяцев работы без отпуска в течение года (для научного руководителя – 11.2 месяца; для студента – 10.4 месяцев);

$F_{\text{д}}$ – действительный годовой фонд рабочего времени научно-технического персонала, раб. дн. (для научного руководителя – 190 раб дн., для студента – 178 раб. дн.)

Таблица 25 – Баланс рабочего времени

Показатели рабочего времени	Руководитель	Студент	Студент
Календарное число дней	9	146	146
Количество нерабочих дней	2	45	45
- выходные дни			
- праздничные дни			
Потери рабочего времени	0	0	0
- отпуск			
- невыходы по болезни			
Действительный годовой фонд рабочего времени	190	178	178

Месячный должностной оклад работника:

$$Z_m = Z_{tc} \cdot (1 + k_{пр} + k_d) \cdot k_p,$$

где Z_{tc} – заработная плата (ставка в час), руб.;

$k_{пр}$ – премиальный коэффициент, равный 0,3 (т.е. 30% от Z_{tc});

k_d – коэффициент доплат и надбавок. В данной работе был принят за 0.

k_p – районный коэффициент, равный 1,3 (для Томска).

Ставка в час руководителя проекта составляет 300 р., студента – 100р, а программиста 400р. Время работы руководителя в день по проекту составляет 2 часа, студентов, программистов – также 2 часа

Расчет основной заработной платы показан в таблице 26.

Таблица 26 – Расчёт основной заработной платы

Исполнители	Z_{tc} , руб.	T_p , раб. дн.			$k_{пр}$	k_p	$Z_{осн.}$, руб.		
		Исп1	Исп2	Исп3			Исп1	Исп2	Исп3
Руководитель	300	7	7	7	0,3	1,3	7098	7098	7098
Студент	100	146	268	-			98696	90584	-
Программист	400	-	-	106			-	-	143312
Итого:							105794	97682	150410

Исходя из полученных данных, исполнение 1 не является самым экономически выгодным вариантом, но является более предпочтительным. Так

как два студента могут обеспечить большую функциональность системы, по сравнению с выполнением проекта одним студентом.

8.4.3 Дополнительная заработная плата исполнителей темы

Здесь учитываются величина предусмотренных Трудовым кодексом РФ доплат за отклонение от нормальных условий труда и выплат, связанных с обеспечением гарантий и компенсаций.

Расчет дополнительной заработной платы ведется по следующей формуле:

$$Z_{\text{доп}} = k_{\text{доп}} \cdot Z_{\text{осн}}$$

где $k_{\text{доп}}$ – коэффициент дополнительной заработной платы (на стадии проектирования принимается равным 0,15).

Таблица 27 – Результаты расчета дополнительной заработной платы

Исполнители	Z _{осн} , руб.			k _{доп}	Z _{доп} , руб.		
	Исп. 1	Исп. 2	Исп.3		Исп.1	Исп.2	Исп.3
Руководитель	7098	7098	7098	0,15	1064,7	1064,7	1064,7
Студент	98696	90584	-	0,15	14804,4	13587,6	-
Программист	-	-	143312	0,15	-	-	21496,8
Итого					15869,1	14652,3	22561,5

8.4.4 Отчисления во внебюджетные фонды (страховые отчисления)

В данной статье расходов отражаются обязательные отчисления по установленным законодательством Российской Федерации нормам органам государственного социального страхования (ФСС), пенсионного фонда (ПФ) и медицинского страхования (ФФОМС) от затрат на оплату труда работников. Величина отчислений во внебюджетные фонды определяется исходя из следующей формулы:

$$Z_{\text{внеб}} = k_{\text{внеб}} \cdot (Z_{\text{осн}} + Z_{\text{доп}})$$

где $k_{\text{внеб}}$ – коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд обязательного медицинского страхования и пр.).

На 2017 г. в соответствии с Федеральным законом от 24.07.2009 №212-ФЗ установлен размер страховых взносов равный 30%.

Отчисления во внебюджетные фонды представлены в таблице 28.

Таблица 28 – Отчисления во внебюджетные фонды

Исполнитель	Основная заработная плата, руб.			Дополнительная заработная плата, руб.		
	Исп.1	Исп.2	Исп.3	Исп.1	Исп.2	Исп.3
Руководитель проекта	7098	7098	7098	1064,7	1064,7	1064,7
Студент-дипломник	98696	90584	-	14804,4	13587,6	-
Программист	-	-	143312	-	-	21496,8
Коэффициент отчислений во внебюджетные фонды	0,3					
Итого						
Исполнение 1	36498,93					
Исполнение 2	33700,29					
Исполнение 3	51891,45					

8.4.5 Расчет накладных расходов

Накладные расходы учитывают прочие затраты организации, не попавшие в предыдущие статьи расходов: печать и ксерокопирование материалов исследования, оплата услуг связи, электроэнергии, размножение материалов и т.д. Их величина определяется по следующей формуле:

$$З_{\text{накл}} = (\text{сумма статей } 1 \div 7) \cdot k_{\text{нр}}$$

где $k_{\text{нр}}$ – коэффициент, учитывающий накладные расходы.

Величину коэффициента накладных расходов можно взять в размере 16%. В рассматриваемом случае были использованы лишь 4 статьи, поэтому деление производится на 4.

$$\text{Для исполнения 1: } З_{\text{накл}} = 39541,51 \cdot 0,16 = 6326,64$$

$$\text{Для исполнения 2: } З_{\text{накл}} = 36512,45 \cdot 0,16 = 5841,99$$

$$\text{Для исполнения 3: } З_{\text{накл}} = 56231,09 \cdot 0,16 = 8996,97$$

8.4.6 Формирование бюджета затрат научно-исследовательского

проекта

Сумма затрат по всем статьям расходов рассчитывается заносится на данном этапе в таблицу 29.

Таблица 29 – Бюджет затрат научно-исследовательского проекта

Наименование статьи	Сумма, руб.			Примечание
	Исп.1	Исп.2	Исп.3	
Материальные затраты НТИ	4	15,2	61,4	Пункт 8.4.1
Затраты по основной заработной плате исполнителей темы	105794	97682	150410	Пункт 8.4.2
Затраты по дополнительной заработной плате исполнителей темы	15869,1	14652,3	22561,5	Пункт 8.4.3
Отчисления во внебюджетные фонды	36498,93	33700,29	51891,45	Пункт 8.4.4
Накладные расходы	6326,64	5841,99	8996,97	16 % от суммы ст. 1-4
Бюджет затрат НТИ	164492,67	151891,78	233921,32	Сумма ст. 1- 5

8.4.7 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования

Определение эффективности происходит на основе расчета интегрального показателя эффективности научного исследования. Его нахождение связано с определением двух средневзвешенных величин: финансовой эффективности и ресурсоэффективности.

$$I_{\text{финр}}^{\text{исп.}i} = \frac{\Phi_{ri}}{\Phi_{\text{max}}}$$

где $I_{\text{финр}}^{\text{исп.}i}$ – интегральный финансовый показатель разработки;

Φ_{ri} – стоимость i -го варианта исполнения;

Φ_{max} – максимальная стоимость исполнения научно-исследовательского проекта (в т.ч. аналоги).

Исполнение 1: $I_{\text{финр}} = 164492,67/233921,32 = 0,703$

Исполнение 2: $I_{\text{финр}} = 151891,78/233921,32 = 0,649$

Исполнение 3: $I_{\text{финр}} = 233921,32/233921,32 = 1$

Полученная величина интегрального финансового показателя разработки отражает соответствующее численное удешевление стоимости разработки в разгах.

Интегральный показатель ресурсоэффективности вариантов исполнения объекта исследования можно определить следующим образом:

$$I_{pi} = \sum a_i \cdot b_i,$$

где I_{pi} – интегральный показатель ресурсоэффективности для i -го варианта исполнения разработки;

a_i – весовой коэффициент i -го варианта исполнения разработки;

b_i^a, b_i^p – бальная оценка i -го варианта исполнения разработки, устанавливается экспертным путем по выбранной шкале оценивания;

n – число параметров сравнения.

Расчет интегрального показателя ресурсоэффективности рекомендуется проводить в форме таблицы (табл. 30).

Таблица 30 – Сравнительная оценка характеристик вариантов исполнения проекта

Объект исследования Критерии	Весовой коэффициент параметра	Исп.1	Исп.2	Исп.3
1. Функциональная мощность (предоставляемые возможности)	0,21	5	4	5
2. Скорость работы	0,13	5	4	4
3. Потребность в ресурсах памяти	0,21	4	3	3
4. Зависимость от платформы	0,18	4	3	4
5. Удобство эксплуатации	0,12	4	4	4
6. Качество предоставляемого интерфейса	0,15	4	3	4
7. Открытость исходного кода	0,02	4	4	3
ИТОГО	1			

$$I_{p-исп1} = 0,21 * 5 + 0,13 * 5 + 0,21 * 4 + 0,18 * 4 + 0,12 * 4 + 0,15 * 4 + 0,02 * 4 = 4,42$$

$$I_{p-исп2} = 0,21 * 5 + 0,13 * 5 + 0,21 * 4 + 0,18 * 4 + 0,12 * 4 + 0,15 * 4 + 0,02 * 4 = 3,54$$

$$I_{p-исп3} = 0,21 * 5 + 0,13 * 5 + 0,21 * 4 + 0,18 * 4 + 0,12 * 4 + 0,15 * 4 + 0,02 * 4 = 4,06$$

Интегральный показатель эффективности вариантов исполнения разработки ($I_{исп.i}$) определяется на основании интегрального показателя ресурсоэффективности и интегрального финансового показателя по формуле:

$$I_{исп.1} = \frac{I_{p-исп1}}{I_{финр}}$$

$$I_{исп1} = 4,42 / 0,703 = 6,29$$

$$I_{исп2} = 3,54 / 0,649 = 5,45$$

$$I_{исп3} = 4,06 / 1 = 4,06$$

Полученное значение интегрального показателя эффективности исполнения разработки превысило максимальный балл оценки. Таким образом, результат работы можно считать положительным, так как оценка интегрального показателя ресурсоэффективности близка к максимальной, при этом стоимость разработки ниже, чем у ряда аналогов, рассмотренных при анализе конкурентных решений.

Сравнение интегрального показателя эффективности вариантов исполнения разработки позволит определить сравнительную эффективность проекта (см.табл.31) и выбрать наиболее целесообразный вариант из предложенных. **Сравнительная эффективность проекта (\mathcal{E}_{cp}):**

$$\mathcal{E}_{cp} = \frac{I_{исп.1}}{I_{исп.2}}$$

Сравнительная эффективность разработки представлена в таблице 31.

Таблица 31 – Сравнительная эффективность разработки

№ п/п	Показатели	Исп.1	Исп.2	Исп.3
1	Интегральный финансовый показатель разработки	0,703	0,649	1
2	Интегральный показатель ресурсоэффективности разработки	4,42	3,54	4,06
3	Интегральный показатель эффективности	6,29	5,45	4,06
4	Сравнительная эффективность вариантов исполнения	1,55	1,34	1

Заключение

В ходе оценки перспективности и альтернатив проведения научного исследования с позиции ресурсоэффективности и ресурсосбережения коммерческого потенциала для работы «Разработка веб-приложения для семантической разметки изображений» были определены потенциальные потребители разработки – специалисты, занимающиеся научно-исследовательской деятельностью в области машинного обучения и компьютерного зрения.

Также был произведен анализ конкурентных технических решений, результаты которого показали, что разрабатываемое приложение имеет некоторые функциональные возможности, отличающие разработку от аналогов на рынке.

Сильные и слабые стороны решения, его возможности и угрозы, а также корреляция этих показателей были определенных в ходе SWOT-анализа. Полученные показатели позволили определить направление развития разработки для достижения наибольшей востребованности среди целевой аудитории. Результаты SWOT-анализа показали, что необходимо направить развитие приложения в сторону кроссплатформенности и адаптивности под различные мобильные устройства.

Определение перечня этапов и работ в рамках проведения научного исследования позволило структурировать и упорядочить запланированные этапы, а также распределить ответственных исполнителей-участников проекта.

Созданный перечень этапов и работ в рамках проведения научного исследования лег в основу структуры календарного плана-графика, необходимого для детального планирования времени выполнения определенного этапа научного исследования.

Исходя из полученных данных и проведенного анализа эффективности, можно сделать вывод, что первый вариант исполнения является наиболее эффективным с позиции ресурсоэффективности, поскольку его интегральные показатели ресурсоэффективности разработки и эффективности выше, чем у других рассмотренных вариантов. Стоимость первого исполнения является не самой низкой, однако функциональный потенциал системы намного выше. Именно поэтому был выбран первый вариант исполнения.

В выполненной дипломной работе были достигнуты экономические и технические критерии эффективности за счет функциональных возможностей разработки, а также социальные за счет востребованности таких систем на рынке.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы была достигнута основная цель работы – разработано web-приложение для семантической разметки изображений. Основой для создания является программная платформа Node.js, а также библиотека jQuery и фреймворк Bootstrap.

Данная разработка позволит подготавливать и структуризировать данные для настройки и обучения классификаторов при решении задач компьютерного зрения и машинного обучения. Создание открытой базы данных изображений позволяет сократить время на подготовку тестовых данных, а также время на поиск необходимых изображений, благодаря системе тегов.

В процессе разработки все поставленные задачи были выполнены:

- Выполнен обзор области компьютерного зрения и машинного обучения;
- Реализован web-репозиторий для хранения изображений;
- Создана и наполнена начальная база изображений;
- Разработаны программные средства для выполнения семантической сегментации изображений.

Главным достоинством разработанного web-приложения является его функциональность, быстродействие, а также отсутствие у данной разработки полных аналогов в открытом доступе. Благодаря открытой базе изображений, существует возможность многократного использования сегментированного изображения для решения различных задач классификации и кластеризации.

В качестве направления развития можно отметить создание встроенного автоматического инструмента для распознавания наиболее распространенных образов. Данная перспектива позволит в большей степени автоматизировать процесс подготовки тестовых данных.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1) Научный интернет-журнал. Control Engineering Россия // Компьютерное зрение [Электронный ресурс] – URL: http://controleng.ru/wp-content/uploads/CE_0149_2014.pdf.
- 2) Image segmentation. Википедия [Электронный ресурс] – URL: https://en.wikipedia.org/wiki/Image_segmentation.
- 3) Компьютерное зрение. Современный подход / Форсайт Д.А., Понс Ж. – Вильямс, 2004. – 56 с.
- 4) VGG Image Annotator. Web-сервис [Электронный ресурс] – URL: http://www.robots.ox.ac.uk/~vgg/software/via/via_demo.html.
- 5) LabelMe. Web-сервис [Электронный ресурс] – URL: <http://labelme.csail.mit.edu/Release3.0/>.
- 6) Szoter. Web-сервис [Электронный ресурс] – URL: <http://www.szoter.com/#intro>.
- 7) Официальный сайт Node.js. // Платформа Node.js. [Электронный ресурс] – URL: <https://nodejs.org/en/>.
- 8) Самоучитель HTML. Руководство HTML. [Электронный ресурс] – URL: <http://htmlbook.ru/html5>.
- 9) CSS // Каскадные таблицы стилей. URL: <http://htmlbook.ru/samcss>.
- 10) Введение в jQuery. jQuery [Электронный ресурс] – URL: http://jquery.page2page.ru/index.php5/Введение_в_jQuery.
- 11) Паттерны для новичков: MVC vs MVP vs MVVM. // Хабрахабр [Электронный ресурс] – URL: <https://habrahabr.ru/post/215605/>.
- 12) Express. Проект Фонда Node.js [Электронный ресурс] – URL: <http://expressjs.com/ru/>.
- 13) Сайт о программировании. Введение в MongoDB [Электронный ресурс] – URL: <https://metanit.com/nosql/mongodb/1.1.php>.
- 14) Web разработка на node.js и express. Глава 2 – тестирование приложения.// Хабрахабр [Электронный ресурс] – URL: <https://habrahabr.ru/post/158185/>.

- 15) Безопасность жизнедеятельности: Учебник для вузов /Под ред. К.З.Ушакова. – М.: Изд-во Московского гос. горного университета, 2000.– 430 с.
- 16) ГОСТ 12.0.003-74 (с измен. №1, октябрь 1978г., переиздание 1999г.) «Классификация вредных и опасных производственных факторов».
- 17) СанПиН 2.2.4.548 – 96. «Гигиенические требования к микроклимату производственных помещений».
- 18) СНиП 23-05-95. «Естественное и искусственное освещение».
- 19) СНиП 23-05-2010 «Нормы Коэффициента пульсации помещений административных зданий».
- 20) ГОСТ 12.1.009-76 «Электробезопасность. Термины и определения».
- 21) НПБ 105-95. «Определение категорий помещений и зданий по взрывопожарной и пожарной опасности».
- 22) СНиП 21-01-97. «Пожарная безопасность зданий и сооружений».
- 23) ГОСТ 30772-2001 «Ресурсосбережение. Обращение с отходами. Термины и определения».
- 24) СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы».
- 25) Гаврилов Ю.В. Современный терроризм: сущность, типология, проблемы противодействия / Ю.В. Гаврилов, Л.В. Смирнов. – М.: ЮИ МВД РФ, 2003. – 66 с.
- 26) Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 3.07.2016) // Электронный фонд правовой и нормативно-технической документации. URL: <http://docs.cntd.ru/document/901807664> (дата обращения: 22.05.2017).
- 27) ГОСТ 12.2.061-81 «ССБТ. Оборудование производственное. Общие требования безопасности к рабочим местам».
- 28) ГОСТ 12.2.032-78 «ССБТ. Рабочее место при выполнении работ сидя. Общие эргономические требования».