Министерство науки и высшего образования Российской федерации

ФЕДЕРАЛЬНОЕ БЮДЖЕТНОЕ ГОСУДАРСТВЕННОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧЕРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт математики и информационных технологий

Кафедра математики и цифровых технологий

**ОТЧЕТ**

по лабораторной работе №1

по дисциплине «Современные средства разработки программного обеспечения»

**Разработка технического задания**

ОГУ 09.03.02.0224.188 О

Руководитель

канд. техн. наук, доцент

\_\_\_\_\_\_\_\_\_\_\_\_ Минина И.В.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

Студент группы 21ИСТ(б)АДМО

\_\_\_\_\_\_\_\_\_\_\_\_ Глазунов Р.В.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

Студент группы 21ИСТ(б)АДМО

\_\_\_\_\_\_\_\_\_\_\_\_ Кучин А.А.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

Студент группы 21ИСТ(б)АДМО

\_\_\_\_\_\_\_\_\_\_\_\_ Полехов Д.А.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

Оренбург 2024

**Содержание**

[1. Общие сведения 3](#_Toc179357065)

[1.1 Назначение документа 3](#_Toc179357066)

[1.2 Наименование системы 3](#_Toc179357067)

[1.3 Сведения о заказчике и исполнителе 3](#_Toc179357068)

[1.4 Основания для выполнения разбор, сроки и финансирование 3](#_Toc179357069)

[1.5 Основания понятия, определения и сокращения 3](#_Toc179357070)

[2 Ход выполнения задания 5](#_Toc179357071)

[2.1 Структура изучаемой системы: определить множество узлов системы и связей между ними. 5](#_Toc179357072)

[2.2 Задачи , выполняемые с помощью программно- технических средств информационной системы: распределить задачи по подсистемам. 6](#_Toc179357073)

[3. Программные и аппаратные средства, обеспечивающие эффективное функционирование системы. 6](#_Toc179357074)

[4. Производительность системы – количество пользователей и частота обращения с их стороны к системе. 7](#_Toc179357075)

[5. Надежность – вероятность того, что система будет выполнять свои функции при заданных условиях в течение требуемого периода времени. 8](#_Toc179357076)

[6. Типы информационных запросов, обслуживаемых системой. 8](#_Toc179357077)

[7. Информационный анализ. 9](#_Toc179357078)

[7.1 Методы и средства передачи данных. 9](#_Toc179357079)

[7.2 Методы и средства передачи данных; 9](#_Toc179357080)

[7.3 Методы и средства хранения данных; 10](#_Toc179357081)

[7.4 Методы и средства представления данных; 10](#_Toc179357082)

[7.5 Методы и средства ввода-вывода данных; 10](#_Toc179357083)

[Заключение 11](#_Toc179357084)

# 1. Общие сведения

**1.1 Назначение документа**

Техническое задание является основным документом, определяющим общие требования и порядок создания веб-приложения по распознаванию рукописных цифр. Требования, изложенные в данном ТЗ, соответствуют передовым образовательным технологиям и не уступают требованиям, предъявляемым к лучшим отечественным и зарубежным программам. Все изменения к данному документу оформляются отдельными согласованными документами.

**1.2 Наименование системы**

Полное наименование приложения – «Распознавание рукописных цифр».

Краткое наименование – РРЦ.

**1.3 Сведения о заказчике и исполнителе**

Заказчик приложения - Оренбургский государственный университет (ОГУ) в лице кафедры геометрии и компьютерных наук.

Исполнители - Студенты группы 21ИСТ(б)АДМО Полехов Дмитрий Андреевич, Кучин Арсений Алексеевич, Глазунов Родион Владимирович.

**1.4 Основания для выполнения разбор, сроки и финансирование**

Разработка ведется на основании рабочей программы дисциплины «Современные средства разработки программного обеспечения».

Система должна быть разработана в течение 2024 года и сдана в опытную эксплуатацию до 09.01.2025.

Работа ведется на безвозмездной основе.

**1.5 Основания понятия, определения и сокращения**

Микросервисная архитектура — это подход, при котором единое приложение строится как набор небольших сервисов, каждый из которых работает в собственном процессе и коммуницирует с остальными используя легковесные механизмы, как правило HTTP.

Микросервис — это функция, отвечающая за один элемент логики.

Spring Framework — это фреймворк с открытым исходным кодом для языка программирования Java.

Spring MVC — это веб-фреймворк Spring. Позволяет создавать веб-сайты или RESTful сервисы (например, JSON/XML) и хорошо интегрируется в экосистему Spring, например, он поддерживает контроллеры и REST контроллеры в ваших Spring Boot приложениях.

Thymeleaf — современный серверный механизм Java-шаблонов для веб- и автономных сред, способный обрабатывать HTML, XML, JavaScript, CSS и даже простой текст.

Объектно-ориентированное программирование (ООП) — это подход, при котором программа рассматривается как набор объектов, взаимодействующих друг с другом.

API (Application programming interface) — это контракт, который предоставляет программа.

Тест-кейс — алгоритм действий для проверки написанной программы. Он подробно описывает короткую последовательность действий, например успешную авторизацию пользователя. В тест-кейсе фиксируют подготовку к проверке, саму диагностику и ожидаемый результат, включая информацию о количестве проверок и нюансах.

**1.6 Актуальность разработки системы**

В современных условиях разработки приложения распознавания рукописных цифр остается актуальной и перспективной задачей в современных технологиях искусственного интеллекта и обработки изображений.

В связи с этим возрастает потребность в системах автоматического распознавания рукописных цифр, которые способны облегчить процесс обработки документов и форм. Системы распознавания, построенные с применением методов искусственного интеллекта, в частности нейронных сетей, позволяют анализировать рукописный ввод и преобразовывать его в цифровой формат с высокой точностью. Такие системы уже активно используются в различных сферах от банковской сферы до архивных учреждений, что подчеркивает их эффективность и востребованность.

Кроме того, область обработки документов, как одна из наиболее требовательных к точности, также требует современных решений для улучшения процесса автоматизации. С увеличением числа электронных форм и документов становится все сложнее ориентироваться в огромных объемах доступной информации. Разработка системы, которая бы автоматически распознавала рукописные цифры с высокой точностью с применением нейронных сетей, имеет большую практическую ценность.

Создание такой системы не только ускорит процесс обработки документов, но и предоставит возможность организациям эффективнее управлять информацией за счет более точного преобразования рукописного ввода в цифровой формат. В условиях высокой нагрузки на административный персонал такие технологии становятся неотъемлемой частью успешных систем управления данными.

Таким образом, разработка системы распознавания рукописных цифр является актуальной задачей, которая отвечает современным требованиям к автоматизации обработки документов и тенденциям развития технологий искусственного интеллекта.

**2 Назначение и цели создания системы**

**2.1 Цели создания системы**

Разработка веб-приложения по распознанию рукописных цифр преследует следующие цели:

анализ существующих моделей обучения, для получения наивысшей точности;

обучение нейронной сети с использованием модели YOLOv8;

изучение микросервисной архитектуры и работы в команде в рамках практической работы.

**2.2 Назначение системы**

Веб-приложение для распознавания рукописного текста (РРЦ) представляет собой сложную систему, предназначенную для автоматизации процесса преобразования рукописных символов в цифровой текст. Это приложение играет важную роль в различных областях, таких как образование, архивирование документов и автоматизация бизнес-процессов.

**2.3 Задачи, решаемые системой**

Облегчение распознавания цифр на рукописных материалах является одной из ключевых задач, решаемых с помощью таких приложений. Современные технологии оптического распознавания символов (OCR) позволяют эффективно извлекать текстовое содержимое из различных источников, включая сканированные документы и фотографии.

**2.4 Область применения системы**

Применение веб-приложений для распознавания текста охватывает широкий спектр задач. Они используются для автоматизации ввода данных, создания цифровых архивов и повышения эффективности работы с документами. В будущем ожидается дальнейшее развитие технологий распознавания, что позволит улучшить качество и скорость обработки рукописных материалов.

Таким образом, веб-приложения для распознавания рукописного текста представляют собой важный инструмент в современном мире, способствующий повышению производительности и упрощению работы с текстовой информацией.

**3 Характеристики объекта автоматизации**

**3.1 Общие сведения**

Распознавание рукописных цифр – это задача компьютерного зрения и машинного обучения, которая заключается в автоматической идентификации и классификации рукописно написанных цифр. Это важная область исследований, имеющая широкое применение в различных сферах, таких как обработка документов, банковские операции и системы безопасности.

Основная аудитория для распознавания рукописных цифр представлена разнообразными группами, каждая из которых имеет свои уникальные интересы и потребности. Исследователи и учёные, академики и студенты в области компьютерного зрения и машинного обучения, наряду с научными сотрудниками в институтах и университетах, заинтересованы в фундаментальных аспектах технологии и проводят исследования для её улучшения. Разработчики и инженеры, работающие над проектами, требующими распознавания рукописных цифр, а также эксперты по безопасности и идентификации, активно используют эту технологию в своей работе. Образовательные организации, такие как учебные заведения и онлайн-школ, также интегрируют распознавание в свой учебный процесс. Не менее важной является аудитория пользователей конечных продуктов, включая тех, кто использует приложения с функцией распознавания, студентов и преподавателей, использующих системы для автоматической оценки домашней работы, и пациентов, которым предлагаются возможности для электронной записи к врачу. Каждая из этих групп имеет свою специфику интересов и потребностей, что делает распознавание рукописных цифр универсальной технологией с широким спектром применения

**3.2 Субъекты объекта автоматизации**

Субъекты объекта автоматизации в контексте распознавания рукописных цифр можно описать следующим образом:

1) пользователи − это конечные пользователи системы, которые взаимодействуют с ней через графический интерфейс. Они предоставляют рукописные цифры для распознавания и получают результаты. Например, студенты могут использовать систему для автоматической проверки домашних работ, а пациенты − для электронной записи к врачу;

2) Администраторы системы − лица, отвечающие за поддержание и обслуживание системы. Они управляют библиотеками моделей, настраивают параметры распознавания и отслеживают производительность системы;

3) Разработчики системы − программисты и инженеры, создающие, поддерживающие и совершенствующие систему распознавания. Они разрабатывают алгоритмы распознавания, обновляют нейронные сети, настраивают интеграции с базами данных и API для получения образов цифр;

4) Поставщики контента − компании или сервисы, предоставляющие образы рукописных цифр для обучения моделей. Они могут предоставлять доступ к своим базам данных через API, обеспечивая систему актуальными данными для обучения;

5) Инфраструктурные субъекты − это серверы, базы данных и другие технические элементы инфраструктуры, на которых работает система. Они обеспечивают хранение и обработку больших объемов данных, обучение моделей и стабильную работу системы в реальном времени.

Эти субъекты взаимодействуют с объектом автоматизации различными способами: пользователи предоставляют данные для распознавания, администраторы управляют системой, разработчики совершенствуют технологию, поставщики контента обеспечивают актуальность данных, а инфраструктура обеспечивает функциональность системы. Их интересы и потребности формируют направления развития технологии и определяют её применение в различных сферах деятельности.

**4 Требования к системе**

**4.1 Требование к системе в целом**

Система для распознавание рукописных цифр (РРЦ) должна быть разработана в виде web-ресурса.

Программа будет обрабатывать входные изображения, на выходе выводить это же изображение с распознанными цифрами и процент их угадывания.

Пользовательский интерфейс должен быть интуитивно понятным и удобным для пользователя с любыми навыками владения компьютера, давая возможность легко выполнить распознавание рукописных цифр на входном изображении, затем показывать все результаты, включая предыдущие.

Система должна точно и эффективно распознавать рукописные числа на изображении используя для этого нейронную модель «YOLOv8» обученную на наборе данных «Mnist». Особенно важно, чтобы система обеспечивала быструю обработку запросов пользователей, минимизируя время отклика и обеспечивая высокую производительность, даже при высокой нагрузке на сервер.

**4.2 Требования к функциям, выполняемым системой**

Диаграмма вариантов использования программы распознавания рукописных цифр представлено в таблице 1.

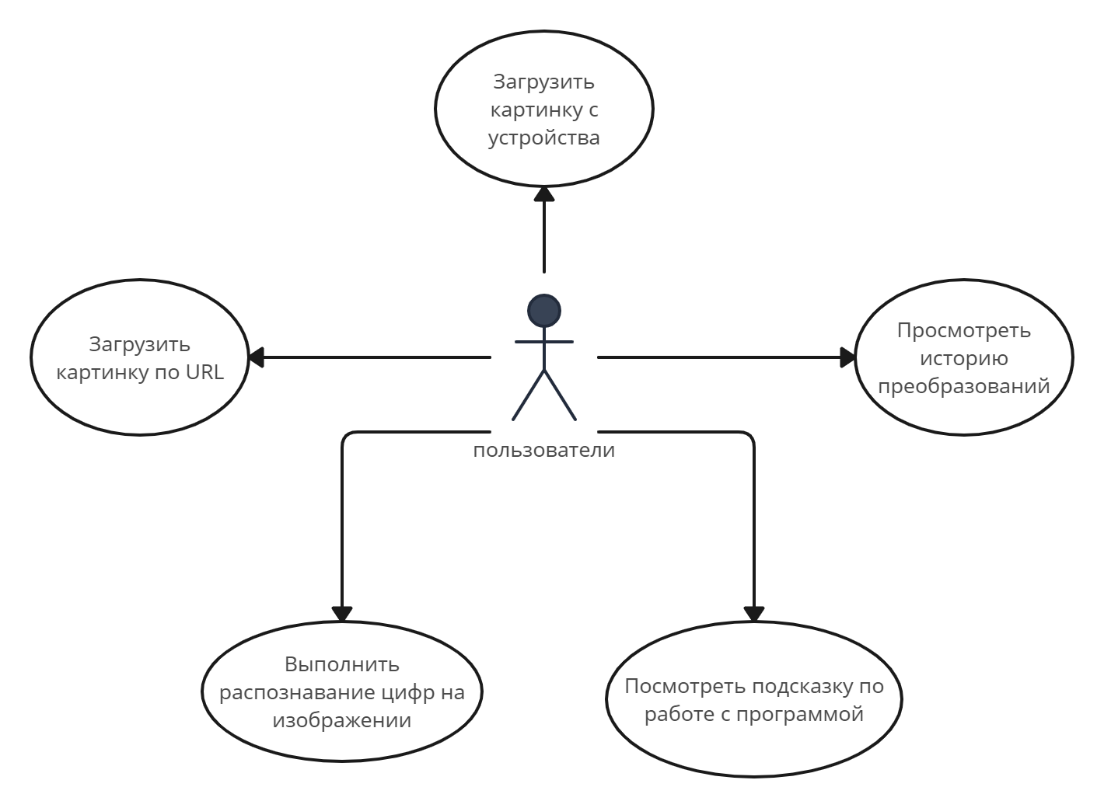
****

Рисунок 1 – Диаграмма вариантов использования

Описание диаграммы вариантов использования программы распознавания рукописных цифр представлено в таблице 1.

Таблица 1 – Описание диаграммы

|  |  |  |
| --- | --- | --- |
| Прецедент | Действие пользователя | Реакция системы |
| Загрузить картинку с устройства | Нажатие кнопки ссылки | Открытие диалогового окна с возможностью выбора файла, также будет «Drag and Drop» |
| Загрузить картинку по URL | Нажатие кнопки файла | Если ссылка действительна, то загрузится картинка из ссылки |
| Посмотреть историю распознаваний | Нажатие кнопки меню «Результаты» | Откроется вкладка сайта с предыдущими результатами |
| Выполнить распознавание цифр на изображении | Нажатие кнопки «Распознать» | После загрузки исходного изображения, начнётся процесс распознавания цифр |
| Посмотреть подсказку по работе с программой | Нажатие кнопки меню «Помощь» | Откроется вкладка с помощью по работе с сайтом |

**5 Состав и содержание работ по созданию (развитию) системы**

**5.1 Перечень работ по созданию АС РМКП**

Разработка системы выполняется на основе каскадной модели жизненного цикла. Каскадная модель − это одна из наиболее ранних и простых моделей разработки программного обеспечения, которая представляет собой последовательный подход к выполнению проектных задач. Она включает в себя несколько четко определенных этапов, каждый из которых должен быть завершен перед переходом к следующему.

К преимуществам каскадной модели относятся: простота, понятность, четкая структура и последовательность.

Основной перечень работ по созданию АС РМКП, их содержание, результаты и сроки представлены в таблице 2. В таблице приведен перечень работ, соответствующий одной итерации жизненного цикла.

Таблица 2 – Перечень работ по созданию АС РМКП

|  |  |  |
| --- | --- | --- |
| Наименование работы | Результат | Сроки |
| Выработка системных требований | Техническое задание, документы спецификаций | 04.09.2024-18.09.2024 |
| Проектирование системы | Готовы архитектура, алгоритмы, и пользовательский интерфейс | 19.09.2024-  05.10.2024 |
| Разработка ПО | Промежуточный результат АС РМКП, реализован бэкенд и фронтенд | 06.10.2024-  06.11.2024 |
| Тестирование системы | Действующий образец АС РМКП, соответствующий требованиям ТЗ, тест кейсы | 07.11.2024-  20.11.2024 |
| Исправление кода | Исправленный код | 21.11.2024-  27.11.2024 |
| Разработка документации | Комплект пользовательской документации | 28.11.2024-  04.12.2024 |
| Установка системы и приемочное тестирование | АС РМКП, соответствующая требованиям ТС, установлена у заказчика и готова к эксплуатации | 05.12.2024-  15.12.2024 |

**6 Порядок контроля и приемки системы**

**6.1 Виды, состав, объем и методы испытаний системы**

Тестирование веб-приложений является важным этапом в процессе разработки, направленным на обеспечение их надежности, производительности и безопасности. Этот процесс включает в себя проверку и оценку различных аспектов веб-приложения, чтобы гарантировать его корректную работу и соответствие требованиям пользователей. Далее будут описаны виды тестирования применимые к РРЦ.

Основные виды тестирования

1. Функциональное тестирование: проверяет, соответствует ли приложение заявленным функциональным требованиям. Это включает в себя тестирование всех функций и возможностей приложения, таких как авторизация, регистрация и другие пользовательские потоки.

2. Тестирование производительности: оценивает, как приложение работает под нагрузкой. Используются инструменты, такие как jMeter, для определения профилей нагрузки и выявления узких мест в производительности.

3. Тестирование безопасности: направлено на выявление уязвимостей, которые могут быть использованы злоумышленниками. Это включает в себя проверку защиты данных и предотвращение несанкционированного доступа.

4. Тестирование удобства использования: оценивает, насколько интуитивно понятен и удобен интерфейс приложения для конечных пользователей.

5. Тестирование совместимости: проверяет, как приложение работает в различных браузерах, операционных системах и устройствах, чтобы обеспечить его доступность для всех пользователей**.**

**6.2 Требования к документированию**

Документы должны быть разработаны следующим образом:

* открытость - все документы должны быть в открытом доступе;
* язык - русский;
* доступный формат отчетности - \*.pdf, \*.docx.

Документы будут загружены на платформе «GitHub» (https://github.com/PoKKu56/numberDetection?tab=readme-ov-file) вместе с кодом проекта.

**7 Источники разработки**

1. Бессмертный, И. А. Системы искусственного интеллекта: учеб. пособие для СПО / И. А. Бессмертный. — 2-е изд., испр. и доп. — М.: Издательство Юрайт, 2018. — 130 с.

2. Гордеев, И. Г. Технология разработки программного обеспечения: учеб. пособие для СПО / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — М.: Издательство Юрайт, 2017. — 235 с.

3. Иванов, В. М. Интеллектуальные системы: учеб. пособие для вузов / В. М. Иванов; под науч. ред. А. Н. Сесекина. — М.: Издательство Юрайт, 2017. — 91 с.

4. Кудрявцев, К. Я. Методы оптимизации: учеб. пособие для вузов / К. Я. Кудрявцев, А. М. Прудников. — 2-е изд. — М.: Издательство Юрайт, 2019. — 140 с.

5. Лаврищева, Е. М. Программная инженерия и технологии программирования сложных систем: учебник для вузов / Е. М. Лаврищева. — 2-е изд., испр. и доп. — М.: Издательство Юрайт, 2019. — 432 с.

6. Сысолетин, Е. Г. Разработка интернет-приложений: учеб. пособие для вузов / Е. Г. Сысолетин, С. Д. Ростунцев; под науч. ред. Л. Г. Доросинского. — М.: Издательство Юрайт, 2019. — 90 с

7. Федоров, Д. Ю. Программирование на языке высокого уровня python : учеб. пособие для прикладного бакалавриата / Д. Ю. Федоров. — 2-е изд., перераб. и доп. — М.: Издательство Юрайт, 2019. — 161 с.

8. Черткова, Е. А. Статистика. Автоматизация обработки информации: учеб. пособие для вузов / Е. А. Черткова; под общ. ред. Е. А. Чертковой. — 2-е изд., испр. и доп. — М.: Издательство Юрайт, 2017. — 195 с.