

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования

**«Южно-Уральский государственный университет
(национальный исследовательский университет)»**

**Высшая школа электроники и компьютерных наук
Кафедра системного программирования**

РАБОТА ПРОВЕРЕНА

Рецензент
Доцент кафедры ВМиИТ
ФГБОУ ВО «ЧелГУ»,
к.ф.-м.н.

_____ А. Ю. Маковецкий

«__»_____ 2024 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой, д.ф.-м.н.,
профессор

_____ Л.Б. Соколинский

«__»_____ 2024 г.

**Разработка веб-приложения для классификации задач ОГЭ
по способам решения**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ЮУрГУ – 02.04.02.2024.308-1389.ВКР

Научный руководитель,
доцент кафедры СП, к.ф.-м.н.
_____ Т.Ю. Маковецкая

Автор работы,
студент группы КЭ-220
_____ Д.О. Еловицова

Ученый секретарь
(нормоконтролер)
_____ И.Д. Володченко
«__»_____ 2024 г.

Челябинск, 2024 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования

**«Южно-Уральский государственный университет
(национальный исследовательский университет)»**

**Высшая школа электроники и компьютерных наук
Кафедра системного программирования**

УТВЕРЖДАЮ

Зав. кафедрой СП

_____ Л.Б. Соколинский

29.01.2024 г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы магистранта

студентке группы КЭ-220

Еловиковой Дарье Олеговне,

обучающейся по направлению

02.04.02 «Фундаментальная информатика и информационные технологии»
(магистерская программа «Машинное обучение и анализ больших данных»)

1. Тема работы (утверждена приказом ректора от 22.04.2024 г. № 764-13/12)

Разработка веб-приложения для классификации задач ОГЭ по способам решения.

2. Срок сдачи студентом законченной работы: 20.05.2024 г.

3. Исходные данные к работе

3.1. Селянкин, В. В. Компьютерное зрение. Анализ и обработка изображений: учебник для вузов. – Санкт-Петербург: Лань, 2021. – 152 с.

3.2. Алексеев, Д. С. Технологии интеллектуального анализа данных: учебное пособие. – Кострома: КГУ им. Н.А. Некрасова, 2020. – 141 с.

3.3. Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O. Scikit-learn: Machine learning in Python. // Journal of machine learning research, 2011. – 64 p.

3.4. Антонио, Д. Библиотека Keras – инструмент глубокого обучения. Реализация нейронных сетей с помощью библиотек Theano и TensorFlow.– Москва: ДМК Пресс, 2018. – 294 с.

4. Перечень подлежащих разработке вопросов

4.1. Обзор научной литературы.

4.2. Формирование функциональных требований.

4.3. Проектирование веб-сервиса для обучения решению задач.

4.4. Реализация приложения для классификации задач ОГЭ по способам решения.

4.5. Тестирование приложения.

5. Дата выдачи задания: 29.01.2024 г.

Научный руководитель,
доцент кафедры СП, к.ф.-м.н.

Т.Ю. Маковецкая

Задание принял к исполнению

Д.О. Еловикова

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	5
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	8
1.1. Описание предметной области	8
1.2. Сравнительный анализ аналогов.....	12
1.3. Нейросетевая модель	17
1.4. Описание набора данных	20
2. ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ	22
2.1. Определение требований	22
2.2. Архитектура системы.....	23
2.3. Проектирование интерфейса веб-приложения	25
2.4. Проектирование REST API веб-приложения.....	28
3. РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ	30
3.1. Программные средства реализации	30
3.2. Реализация веб-приложения	31
3.3. Тестирование веб-приложения.....	34
ЗАКЛЮЧЕНИЕ	37
ЛИТЕРАТУРА.....	38
ПРИЛОЖЕНИЯ.....	40
Приложение А. Страницы тренировки и загрузки файла.....	40
Приложение Б. Код станицы Upload.....	41

ВВЕДЕНИЕ

Актуальность

Современное образование ставит перед школьниками и их преподавателями множество вызовов, и одним из ключевых моментов является успешная подготовка к Основному государственному экзамену (ОГЭ). Выпускники школ должны владеть не только знаниями, но и уметь применять их на практике при решении задач различных типов. Разработка приложения для классификации задач ОГЭ по способам решения является важным шагом в области современного образования.

Актуальность облегчения труда учителя и цифровизации образования в современной России нельзя переоценить, особенно в контексте национальных проектов страны.

Цифровизация образования является одним из приоритетов национального проекта «Образование», так как внедрение современных цифровых технологий в учебный процесс открывает новые возможности для улучшения качества образования и повышения доступности знаний. Это также содействует развитию профессионализма педагогических кадров и оптимизации учебного процесса.

Фокус на облегчении труда учителя здесь играет важную роль. В условиях быстрого темпа жизни и развития информационных технологий учителям становится сложнее эффективно управлять образовательным процессом и оценивать успеваемость учеников. Цифровые инструменты могут помочь автоматизировать рутинные задачи, такие как учет успеваемости, создание интерактивных уроков и индивидуальных программ обучения.

Постановка задачи

Целью выпускной квалификационной работы является разработка веб-приложения для классификации задач ОГЭ по способам решения. Данное приложение станет ценным инструментом для учителей и учеников, предоставляя им возможность быстрого доступа к задачам, сгруппирован-

ным по стратегиям и алгоритмам, что значительно упростит процесс обучения и повысит его результативность. Для достижения поставленной цели необходимо решить представленные ниже задачи.

1. Провести анализ предметной области, а также анализ потребностей и предпочтений потенциальных пользователей.
2. Сформировать полные функциональные требования к разрабатываемому веб-приложению.
3. Выбрать нейросетевую модель на основе подходящей технологии и использовать ее в приложении.
4. Выполнить проектирование и реализацию приложения для формирования и проверки тестовых заданий.
5. Провести тестирование приложения и проверку его переносимости на другие персональные компьютеры.

Структура и содержание работы

Работа состоит из введения, трех глав, заключения и списка литературы. Объем работы составляет 42 страницы, объем списка литературы – 22 источника.

В первой главе предоставляется глубокий анализ предметной области, касающейся подготовки учащихся к государственной итоговой аттестации (ОГЭ). Рассматриваются ключевые проблемы и вызовы, с которыми сталкиваются педагоги и студенты в процессе подготовки к экзамену, а также описываются существующие стратегии их преодоления. Проводится сравнительный обзор существующих веб-приложений, ориентированных на классификацию задач ОГЭ по методам решения, с выделением их сильных и слабых сторон, а также основных функциональных особенностей и параметров. Выполняется сравнительный анализ различных типов нейронных сетей с учетом ряда значимых для данной задачи критериев.

В данном разделе представлена нейросетевая модель, предназначенная для классификации задач ОГЭ по методам решения. Описывается про-

цесс выбора и обоснования используемой нейронной сети, а также раскрываются основные принципы ее функционирования и обучения. Также в данном разделе предоставлено описание набора данных, предназначенного для обучения нейросетевой модели. Описывается процесс сбора и предварительной обработки данных, а также выделяются основные характеристики и специфические черты используемого набора.

Во второй главе описываются требования к веб-приложению для классификации задач ОГЭ по способам решения. Здесь завершен процесс формирования как функциональных, так и нефункциональных требований, учитывающих потребности пользователей и специфику задачи. Выявлены основные аспекты, которые необходимо учитывать при определении требований. Описан использованный для обучения модели набор данных и источники предложений, вошедших в этот набор. Определены ключевые компоненты и их роли в функционировании приложения, что создает прочную основу для последующей реализации.

В третьей главе содержатся описания программных средств, использованных для реализации веб-приложения для классификации задач ОГЭ по способам решения. Выбраны инструменты и технологии, которые обеспечивают эффективную и надежную разработку. Представлены основные функциональные возможности и характеристики приложения, которые обеспечивают его эффективное использование для классификации задач ОГЭ.

Заключительный раздел главы посвящен тестированию веб-приложения для классификации задач ОГЭ. Описан процесс тестирования, включая основные принципы и методы проведения. Представлены результаты тестирования, подтверждающие корректность работы приложения и его готовность к использованию. Проведен анализ результатов тестирования, что позволяет убедиться в высоком качестве и надежности разработанного приложения.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Описание предметной области

Основной государственный экзамен (ОГЭ) представляет собой форму государственной итоговой аттестации, проводимой в целях определения соответствия результатов освоения обучающимися основных образовательных программ основного общего образования требованиям федерального государственного образовательного стандарта.

Для указанных целей используются контрольные измерительные материалы (КИМ), представляющие собой комплексы заданий стандартизированной формы. ОГЭ проводится в соответствии с Федеральным законом от 29.12.2012 г. № 273-ФЗ «Об образовании в Российской Федерации» и Порядком проведения государственной итоговой аттестации по образовательным программам основного общего образования, утвержденным приказом Минпросвещения России и Рособрнадзора от 04.04.2023 г. № 232/551 [4].

Содержание КИМ ОГЭ определяется на основе федерального государственного образовательного стандарта основного общего образования (далее – ФГОС):

1) приказ Министерства просвещения Российской Федерации от 31.05.2021 г. № 287 «Об утверждении федерального государственного образовательного стандарта основного общего образования»;

2) приказ Министерства образования и науки Российской Федерации от 17.12.2010 г. № 1897 (с изменениями 2014–2022 гг.);

3) письмо Рособрнадзора от 19.12.2023 г. №04-389.

Детализированные требования к результатам освоения основной образовательной программы основного общего образования, проверяемые на основе ФГОС 2021 г., являются преемственными по отношению к требованиям ФГОС 2010 г.

При разработке КИМ ОГЭ учитывается содержание федеральной образовательной программы основного общего образования (приказ Министерства просвещения Российской Федерации от 18.05.2023 г. № 370 «Об

утверждении федеральной образовательной программы основного общего образования»).

В рамках выпускной квалификационной работы необходимо провести исследование и выбрать наиболее удобную форму проверки знаний по математике через веб-приложение. Для этого следует рассмотреть различные подходы к тестированию и оценке знаний, а также учесть особенности и потребности пользователей.

Одним из возможных вариантов является использование интерактивных тестов, которые позволяют учащимся проверить свои знания в интерактивном режиме, получая мгновенную обратную связь по каждому заданию. Такой подход стимулирует активную работу учащихся и позволяет им быстро определить свои сильные и слабые стороны.

Другой вариант - использование адаптивной системы тестирования, которая автоматически подстраивает уровень сложности заданий в зависимости от успехов пользователя. Это позволяет более точно оценить знания учащихся и обеспечить индивидуальный подход к каждому из них.

Выбор наиболее удобной формы проверки знаний по математике через веб-приложение должен основываться на анализе потребностей и предпочтений пользователей, а также на оценке эффективности различных подходов к тестированию и оценке знаний. В результате должна быть разработана и реализована система, которая максимально удовлетворяет потребности учащихся и преподавателей и способствует повышению качества образования в области математики.

Описание типа заданий

В данной работе рассматривается задание № 11, относящееся к разделу № 5 «Функции». Правильно решенное задание позволяет определить умение строить графики функций, использовать графики для определения свойств процессов и зависимостей, умение выражать формулами зависимости между величинами. Набор данных для обучения нейросети, состоящий из графиков различных типов функций, состоит из представленных ниже

функций. Образцы задания приведены на рисунках 1–3. В данных заданиях необходимо найти соответствие между предложенными на рисунках графиками и формулами, которые их задают (или знаками коэффициентов, используемых в формулах).

Линейная функция: график линейной функции представляет собой прямую линию. Набор данных для обучения нейросети, состоящий из графиков линейных функций, может содержать различные значения коэффициентов и свободных членов.

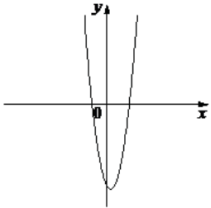
Установите соответствие и впишите ответ.

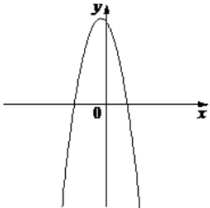
На рисунках изображены графики функций вида $y = ax^2 + bx + c$. Установите соответствие между знаками коэффициентов a и c и графиками функций.

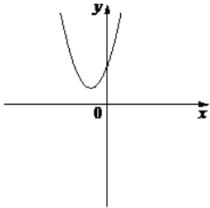
КОЭФФИЦИЕНТЫ

А) $a > 0, c > 0$ Б) $a < 0, c > 0$ В) $a > 0, c < 0$

ГРАФИКИ

1) 

2) 

3) 

В таблице под каждой буквой укажите соответствующий номер.

А	Б	В
▼	▼	▼

Рисунок 1 – Образец задания ОГЭ

Квадратичная функция: график квадратичной функции представляет собой параболу. Набор данных для обучения нейросети, состоящий из графиков квадратичных функций, может содержать различные значения коэффициентов и свободных членов.

Функция обратной пропорциональности: график функции обратной пропорциональности представляет собой гиперболу. Набор данных для обучения нейросети, состоящий из графиков функции обратной пропорциональности, может содержать различные значения коэффициентов и свободных членов.

На рисунке изображены графики функций вида $y = ax^2 + bx + c$. Для каждого графика укажите соответствующие ему значения коэффициента a и дискриминанта D .

Графики

А)

Б)

В)

Г)

Знаки чисел

1) $a > 0, D > 0$

2) $a > 0, D < 0$

3) $a < 0, D > 0$

4) $a < 0, D < 0$

Рисунок 2 – Графики квадратичной функции

Функция корня: график функции корня представляет собой кривую, которая имеет точку перегиба. Набор данных для обучения нейросети, состоящий из графиков функции корня, может содержать различные значения коэффициентов и свободных членов.

Установите соответствие и впишите ответ.

Установите соответствие между функциями и их графиками.

ФУНКЦИИ

А) $y = \frac{1}{3}x + 2$ Б) $y = -4x^2 + 20x - 22$ В) $y = \frac{1}{x}$

ГРАФИКИ

1)

2)

3)

В таблице под каждой буквой укажите соответствующий номер.

А	Б	В
▼	▼	▼

Рисунок 3 – Образец задания 11 ОГЭ

Каждый график функции может быть представлен в виде набора точек, где каждая точка представляет собой пару значений (x, y) , где x – независимая переменная, а y – зависимая переменная. Набор данных может содержать различные значения x и соответствующие значения y для каждой функции.

Набор данных может быть разделен на обучающую и тестовую выборки. Обучающая выборка используется для обучения нейронной сети, а тестовая выборка используется для оценки качества обучения.

Такой набор данных может быть полезен для обучения нейросети распознаванию различных типов функций и их графиков. Он позволяет нейросети научиться различать различные типы функций и предсказывать значения зависимой переменной на основе значений независимой переменной.

1.2. Сравнительный анализ аналогов

Во время изучения предметной области выделились некоторые сервисы, которые являются похожими на приложение для классификации задач ОГЭ по способам решения. Они также предоставляют возможность решать задания для подготовки к экзаменам, созданные в соответствии со спецификациями.

Сайт «Решу ЕГЭ»

Образовательный портал «Решу ЕГЭ» представляет собой дистанционную обучающую систему, предназначенную для подготовки к государственным выпускным или промежуточным мониторингам. Портал был создан творческим объединением «Центр интеллектуальных инициатив», руководителем которого является Гуцин Д. Д. – учитель математики, физики и информатики, почетный работник общего образования РФ, победитель конкурса «Учитель года России – 2007», эксперт Федеральной комиссии по

разработке контрольно-измерительных материалов по математике для проведения единого государственного экзамена по математике и заместитель председателя региональной предметной комиссии ГИА по математике.

Портал предоставляет разнообразные сервисы для подготовки к ГИА.

Одним из них является классификатор экзаменационных заданий, позволяющий последовательно повторять отдельные темы и сразу же проверять свои знания по ним (рисунок 4).

Также есть возможность включения в тренировочные варианты работ произвольного количества заданий каждого экзаменационного типа для организации текущего контроля знаний.



Рисунок 4 – Сайт Решу ЕГЭ

Для любого пользователя доступно прохождение тестирования в формате ЕГЭ текущего года по одному из вариантов или по индивидуальному сгенерированному варианту для проведения итоговых контрольных работ.

Также зарегистрированный пользователь имеет возможность вести статистику выполненных заданий.

Пользователю на сайте представлена подробная информация о правилах проверки экзаменационных работ, включая критерии проверки заданий

с развернутым ответом и возможность проверить задания с открытым ответом в соответствии с этими критериями.

Тестовый балл после выполнения заданий выдается в стобалльной шкале для ЕГЭ и по пятибалльной шкале по ОГЭ.

Задания, используемые на портале, разрабатываются специально для портала «Решу ЕГЭ» и являются интеллектуальной собственностью редакции. Они включают открытые банки заданий ФИПИ, демонстрационные версии экзаменов, задания прошедших экзаменов, диагностические работы, подготовленные Московским институтом открытого образования, а также задания из литературных источников, используемые в соответствии с лицензиями правообладателей. Пользователи портала также могут добавлять в каталоги свои собственные задания, публиковать теоретические материалы, создавать обучающие курсы и переписываться со своими читателями.

Федеральный институт педагогических измерений

«Федеральный институт педагогических измерений» (ФГБНУ «ФИПИ») было создано приказом Министерства образования Российской Федерации от 2 августа 2002 г. № 2990. Его целью является содействие Рособрнадзору в осуществлении полномочий по методическому обеспечению проведения государственной итоговой аттестации (ГИА) по образовательным программам основного общего и среднего общего образования; по организации разработки КИМ для проведения ГИА по образовательным программам основного общего и среднего общего образования и критериев оценивания экзаменационных работ, выполненных на основе этих КИМ ГИА; по определению КИМ для проведения экзамена по русскому языку как иностранному, истории России и основам законодательства Российской Федерации, по разработке, созданию и эксплуатации информационных систем в установленной сфере деятельности; по осуществлению мониторинга в системе образования на федеральном уровне в установленной сфере ведения; по организационно-техническому и научно-методическому определению

КИМ для проведения экзамена по русскому языку как иностранному, истории России и основам законодательства Российской Федерации для иностранных граждан и лиц без гражданства на уровне, соответствующем цели получения разрешения на работу или патента, временное проживание, вида на жительство.

Формирование КИМов и открытый банк экзаменационных заданий являются ключевыми аспектами деятельности ФГБНУ «ФИПИ». Ежегодно Институт разрабатывает контрольные измерительные материалы (КИМ), которые включают в себя задания различного уровня сложности и направлены на оценку знаний учащихся по различным предметам. Эти материалы проходят тщательную проверку и апробацию перед использованием в реальных экзаменах.

Использование открытого банка заданий по математике на сайте ФИПИ имеет ряд преимуществ.

Во-первых, это позволяет ученикам практиковаться в решении задач, которые могут встретиться на экзаменах. Это помогает им лучше подготовиться к экзаменам и повысить свои шансы на успешную сдачу.

Во-вторых, открытый банк заданий по математике на сайте ФИПИ предоставляет учителям возможность разнообразить учебный процесс. Учителя могут использовать задачи из банка для создания собственных уроков или для проверки знаний учеников. Это помогает сделать уроки более интересными и эффективными.

В-третьих, использование открытого банка заданий по математике на сайте ФИПИ способствует развитию навыков самостоятельной работы у учеников. Они могут изучать и решать задачи в удобное для них время, что укрепляет их способность к самообучению и самоорганизации.

В-четвертых, наличие открытого банка заданий на сайте ФИПИ обеспечивает прозрачность и объективность в подготовке к экзаменам. Ученики и учителя имеют доступ к тем же заданиям, которые могут быть на реальных экзаменах, что позволяет им сосредоточиться на наиболее

актуальном и важном материале. Это также гарантирует, что подготовка к экзаменам будет соответствовать официальным стандартам и требованиям.

Открытый банк заданий ГВЭ | Математика

↓
ПОДБОР ЗАДАНИЙ
Кол-во заданий: 126

1
2
3
4
5
6
7
8
...
13

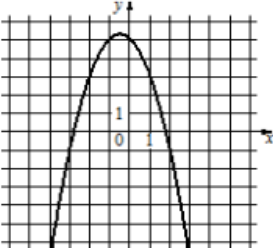
Установите соответствие и впишите ответ.

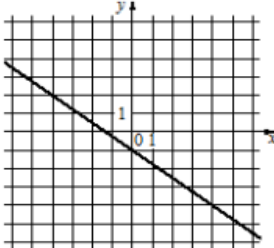
Установите соответствие между функциями и их графиками.

ФУНКЦИИ

А) $y = -x^2 - x + 5$ Б) $y = -\frac{3}{4}x - 1$ В) $y = -\frac{12}{x}$

ГРАФИКИ

1) 

2) 

3) 

В таблице под каждой буквой укажите соответствующий номер.

А	Б	В
▼	▼	▼

i
Номер: 4E8245
★
НЕ РЕШЕНО

ОТВЕТИТЬ

Рисунок 5 – Сайт ФИПИ

Открытый банк экзаменационных заданий ФИПИ представляет собой коллекцию заданий (рисунок 5), которые были использованы в предыдущих экзаменах. Этот банк доступен для всех желающих и позволяет учащимся

ознакомиться с типом и форматом заданий, а также подготовиться к предстоящим экзаменам. Представленные сервисы не учитывают в полной мере особенности подготовки к экзамену в условиях образовательной организации. Согласно требованиям РОСОБРНАДЗОРА в школах в обязательном порядке устанавливается контент-фильтрация, препятствующая использованию ресурсов сети интернет за исключением государственных сервисов. Также как правило пользователи не имеют прав на установку программного обеспечения на персональные компьютеры. Следовательно, приложение должно быть доступно в режиме без установки и через интернет.

1.3. Нейросетевая модель

Обзор методов классификации изображений

Нейронные сети (НС) – вычислительные модели, которые вдохновлены биологическими нейронными сетями человеческого мозга. Они используются для обработки данных и выполнения сложных задач, таких как распознавание образов, классификация, прогнозирование и многое другое.

Обучение нейронных сетей происходит путем настройки весов и параметров модели на основе обучающих данных. Это может быть выполнено с использованием различных методов, таких как обратное распространение ошибки, стохастический градиентный спуск и другие.

Благодаря большой мощности, способности решать сложные задачи и автоматизировать процессы, адаптации к различным типам задач, нейронные сети широко используются в различных областях, включая компьютерное зрение, обработку естественного языка, образование, медицину, финансы, промышленность и многое другое. Но при этом стоит отметить, что нейронные сети часто являются «черными ящиками», что делает их трудно интерпретируемыми и объяснимыми, обучение нейронных сетей требует больших объемов данных, высокой вычислительной мощности и времени. Сложность моделей может привести к переобучению на обучающих данных и плохой обобщающей способности на новых данных.

Нейронные сети имеют различные архитектуры, включая многослойные перцептроны (MLP), сверточные нейронные сети (CNN), рекуррентные нейронные сети (RNN) и глубокие нейронные сети (DNN) [9].

Рекуррентные нейронные сети (RNN) могут быть использованы для классификации изображений, где данные представлены в виде последовательности пикселей. RNN способны обрабатывать последовательности данных и учитывать контекст, что делает их подходящими для обработки изображений.

Для классификации изображений с помощью RNN, мы можем представить изображение в виде последовательности пикселей. Затем мы можем использовать RNN для обработки этой последовательности и извлечения признаков, которые могут быть использованы для классификации.

Классификация изображений с помощью CNN

Сверточные нейронные сети (CNN) являются наиболее популярным методом для классификации изображений. Они специально разработаны для обработки изображений и способны извлекать признаки из изображений более эффективно и точно, чем другие типы нейронных сетей.

Для классификации изображений с помощью CNN, мы можем использовать архитектуру, которая включает несколько слоев свертки, субдискретизации и полносвязных слоев.

Сверточные слои позволяют извлекать признаки из изображений, субдискретизационные слои уменьшают размер изображения, а полносвязные слои позволяют классифицировать изображения на основе извлеченных признаков.

CNN могут быть обучены на больших наборах данных, что позволяет им достигать высокой точности классификации. Они также могут быть обучены на GPU, что значительно ускоряет процесс обучения [11].

Однако, CNN могут быть сложными в настройке и требовать большого объема данных для обучения. Они также могут быть менее эффективными для обработки изображений с большим разрешением или сложной структурой.

Классификация изображений с помощью CNN

Трансформеры – это тип нейронных сетей, которые были изначально разработаны для обработки естественного языка, но в последнее время они также используются для обработки изображений. Они способны обрабатывать последовательности данных и учитывать контекст, что делает их подходящими для классификации изображений.

Для классификации изображений с помощью трансформеров, мы можем представить изображение в виде последовательности пикселей. Затем мы можем использовать трансформер для обработки этой последовательности и извлечения признаков, которые могут быть использованы для классификации.

Трансформеры могут быть более эффективными, чем рекуррентные нейронные сети (RNN), особенно при обработке длинных последовательностей данных. Они также могут быть более точными при классификации изображений, особенно когда данные представлены в виде последовательности пикселей.

Однако, трансформеры могут быть сложными в настройке и требовать большого объема данных для обучения. Они также могут быть менее эффективными для обработки изображений с большим разрешением или сложной структурой.

Классификация изображений с помощью GNN

Графовые нейронные сети (Graph Neural Networks, GNN) представляют собой класс моделей машинного обучения, которые способны обрабатывать данные, представленные в форме графа. В контексте обработки изображений, граф может представлять собой структуру пикселей или регионов

в изображении. Каждый узел графа соответствует определенному пикселю или региону, а ребра между узлами представляют связи между ними.

GNN обучаются на наборах данных, где каждому изображению сопоставляется метка класса. Во время обучения GNN учатся ассоциировать определенные паттерны в графе с конкретными классами. После обучения модель может классифицировать новые изображения, создавая граф на основе их структуры и применяя к нему свои знания о паттернах, связанных с различными классами.

Важно отметить, что GNN могут быть особенно полезны при работе с изображениями, где важны отношения между различными частями изображения, такими как лица в группе людей или объекты в сложной сцене.

1.4. Описание набора данных

Задача классификации объектов является одной из ведущих в области применения нейронной сети. Для решения данной задачи можно подобрать различные алгоритмы реализации. Однако, на первый план выходит задача подготовки данных для обучения. Собранные данные должны полностью соответствовать выбранной модели нейронной сети и предусматривать все возможные сценарии. После сбора данных их необходимо структурировать.

В нашем случае, сбор изображений графиков функций для их классификации с использованием нейронной сети производился из открытых источников: сайты, сборники для подготовки к ОГЭ, учебники. Ожидаемо, возникает проблема повторяемости материала во всех источниках. Данные были сохранены в формате .png и распределены по классам:

- 1) *hyperbola_1*: $k > 1$;
- 2) *hyperbola_2*: $k < -1$;
- 3) *hyperbola_3*: $0 < k < 1$;
- 4) *hyperbola_4*: $-1 < k < 0$;
- 5) *liner_1*: $k > 0, b > 0$;
- 6) *liner_2*: $k < 0, b < 0$;

- 7) liner_3: $k < 0, b > 0$;
- 8) liner_4: $k > 0, b < 0$;
- 9) liner_4: $k = 0$;
- 10) other;
- 11) sqr_1: $a > 0, c > 0$;
- 12) sqr_2: $a < 0, c < 0$;
- 13) sqr_3: $a > 0, c < 0$;
- 14) sqr_4: $a < 0, c > 0$.

В связи с недостаточным количеством собранных таким способом данных для обучения нейронной сети возникла необходимость генерации изображений. Генерация графиков была выполнена на языке программирования Python. При выполнении данной части работы учитывалась комбинация следующих факторов: наличие или отсутствие координатной сетки, толщина и цветность как самого графика, так и координатных осей, размерность рисунка в координатах. В веб приложении генерируются графики аналогичных классов, но одинаковой размерности.

2. ПРОЕКТИРОВАНИЕ ПРИЛОЖЕНИЯ

2.1. Определение требований

Веб-приложение для обучения математике представляет собой инновационный сервис, предназначенный для классификации графиков функций. Этот инструмент предоставляет уникальную возможность загружать изображения графиков функций и получать подробное объяснение их свойств. Основным пользователем приложения должны выступить преподаватели математики, приложение предполагается использовать на уроках математики для отработки знаний по теме «График функции».

Функциональные требования:

- 1) возможность загрузки изображения графика функции;
- 2) возможность получения объяснения графика функции;
- 3) возможность отработки практических навыков на тренировке;
- 4) возможность проверки знаний на экзамене.

Нефункциональные требования:

- 1) веб-приложение должно быть доступно через браузер;
- 2) веб-приложение должно быть оптимизировано для работы с различными графиками функций;
- 3) веб-приложение должно быть удобным и интуитивно понятным в использовании;
- 4) веб-приложение должно быть оптимизировано для работы с настольными ПК и браузерами.

На рисунке 6 изображена диаграмма вариантов использования веб-приложения для классификации задач ОГЭ по способам решения. Главный актер – пользователь веб-приложения, выбирающий различные варианты использования.



Рисунок 6 – Диаграмма вариантов использования

Актерам доступны представленные ниже варианты использования системы.

1. Вариант использования «Загрузка изображения». Пользователь может загрузить изображение графика функции, для которого будет производиться распознавание и выдача рекомендаций по решению задачи. После загрузки, приложение начинает процесс распознавания.

2. Вариант использования «Тренировка». Пользователь может запросить тренировочные задания. Приложение представит случайный график и предложит выбрать правильный ответ.

3. Вариант использования «Экзамен». Пользователь может запросить выдачу тренировочных заданий в течение определенного времени с последующей информацией о проблемных местах.

2.2. Архитектура системы

Архитектуру систему можно декомпозировать на несколько компонентов, которые взаимодействуют между собой для реализации требуемого функционала. На рисунке 7 предоставлена диаграмма компонентов системы.

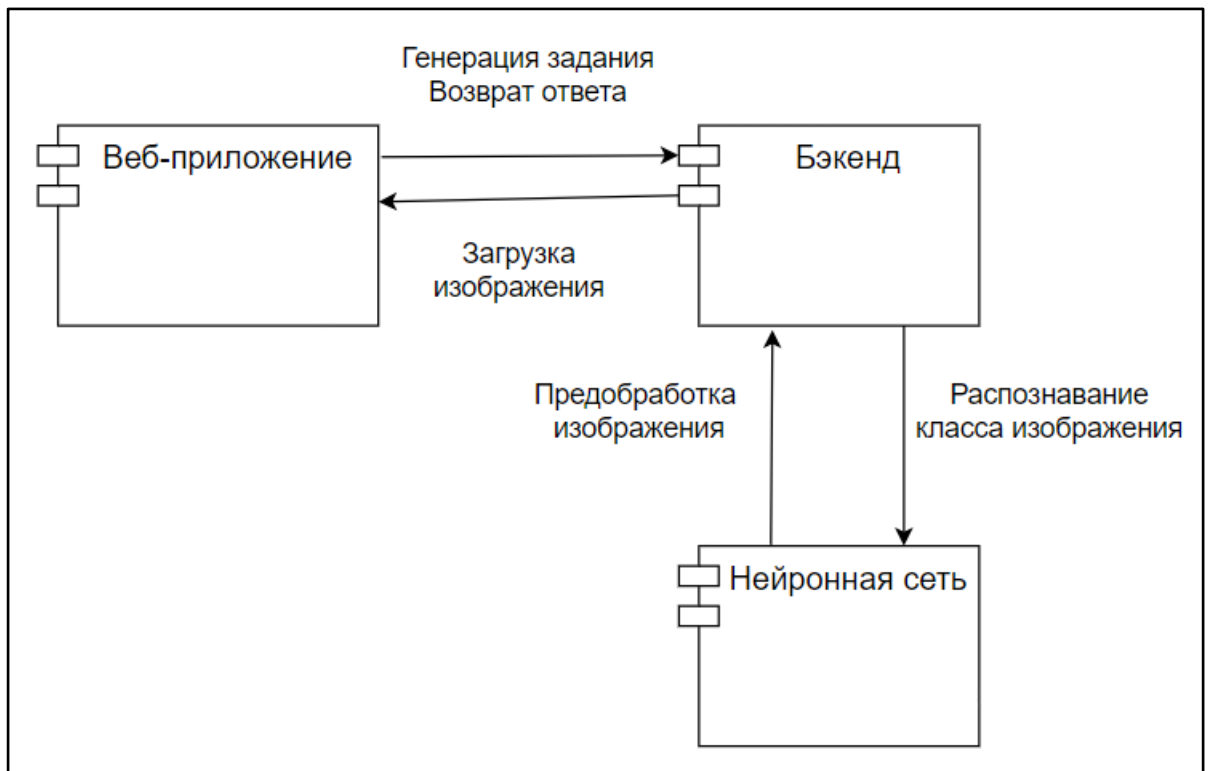


Рисунок 7 – Архитектура системы

Рассмотрим каждый компонент.

1. Обработчик изображений – блок, в котором данные изображений обрабатываются, и передаются в компонент моделей нейросети. Также происходит генерация изображений для передачи веб приложению.
2. Нейронная сеть – компонент обученной модели нейронной сети.
3. Веб-приложение – компонент, отвечающий за организацию передачи данных конечному пользователю, и выполнение операции для удовлетворения клиентских запросов. Основная деятельность системы представляется диаграммой деятельности (рисунок 8), где показывается последовательность действий, происходящих при загрузке изображения.

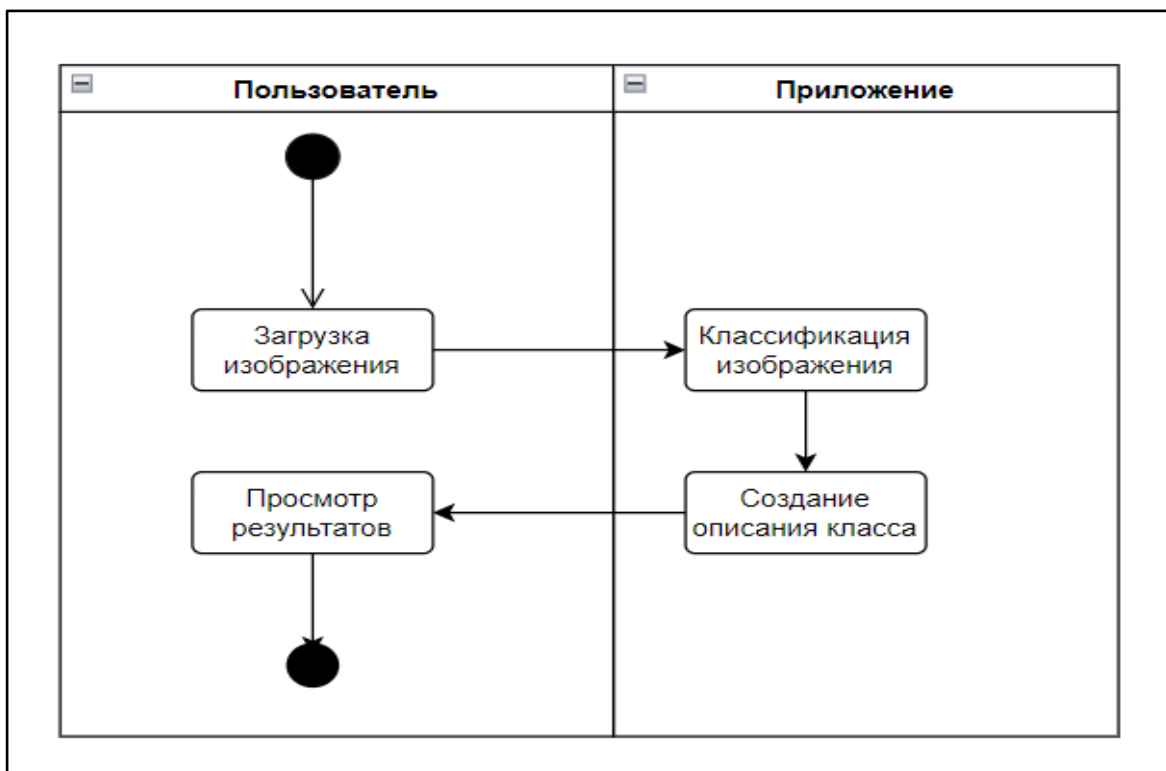


Рисунок 8 – Диаграмма деятельности

2.3. Проектирование интерфейса веб-приложения

Важной частью веб-приложения является графический интерфейс, которым пользователям предоставляется возможность удобного и доступного способа взаимодействия с системой, независимо от операционных систем, и параметров устройств.

Графический интерфейс состоит из нескольких веб страниц, которые играют роль представлений (views) в архитектуре приложения. Рассмотрим их подробно далее.

Главная страница – страница, где отображается форма с кнопками выбора трех основных действий – загрузки изображения для распознавания, режима тренировки и режима экзамена. При выборе первого действия пользователь может загрузить изображение, которое будет обработано системой для распознавания и анализа, содержащегося в нем математического задания. Во втором режиме, режиме тренировки, пользователь получает доступ

к набору задач, которые можно решать в интерактивном режиме с получением мгновенной обратной связи. В режиме экзамена пользователь может проверить свои знания, выполняя серию задач, которые будут оцениваться автоматически, что позволяет получить объективную оценку уровня подготовки. Первоначальный макет страницы можно увидеть на рисунке 9.

Сайт для подготовки к экзамену по математике

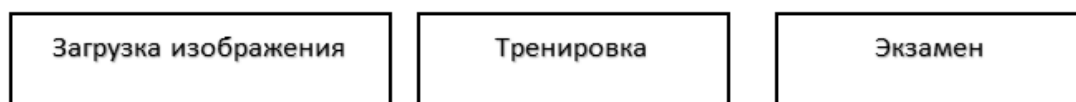


Рисунок 9 – Главная страница

Страница загрузки изображения – на данной странице отображается форма с полями загрузки изображения, после отправки формы, данные проходят валидацию в серверной части системы, при отсутствии ошибок, распознается успешно изображение и переходит на страницу показа объяснений. Макет страницы с полем для загрузки изображения представлен на рисунке 10.

На странице загрузки изображения пользователь может выбрать файл с изображением графика функции и загрузить его, используя удобные инструменты управления. После загрузки изображения система автоматически проверяет его на соответствие требованиям и допустимым форматам, что обеспечивает корректную работу приложения.

Если загруженное изображение проходит валидацию, серверная часть системы обрабатывает его и выполняет распознавание графиков функций. В случае успешного распознавания пользователь перенаправляется на страницу с подробными объяснениями свойств загруженного графика.

Также доступна пакетная загрузка изображений, для классификации некоторого количества изображений. Система классифицирует изображения и возвращает строки с названием файла и классом нарисованного графика, предполагается для использования в проверке работ.

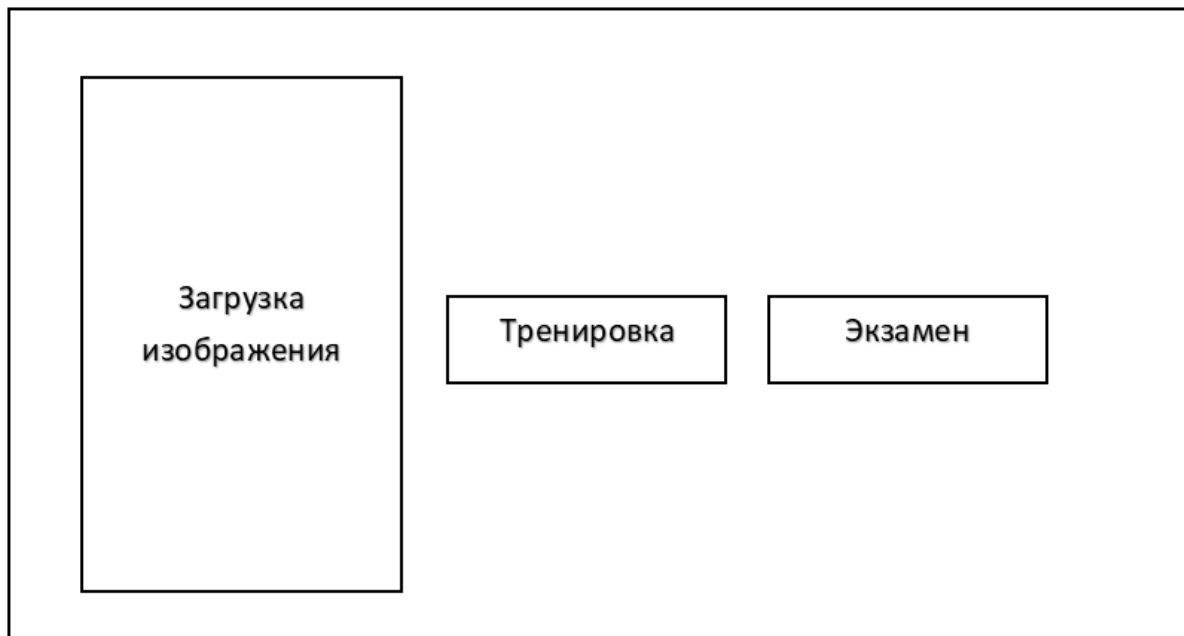


Рисунок 10 – Страница загрузки

Страница загрузки ответа – на данной странице отображается форма с полями загрузки ответа, полученного из серверной части системы.

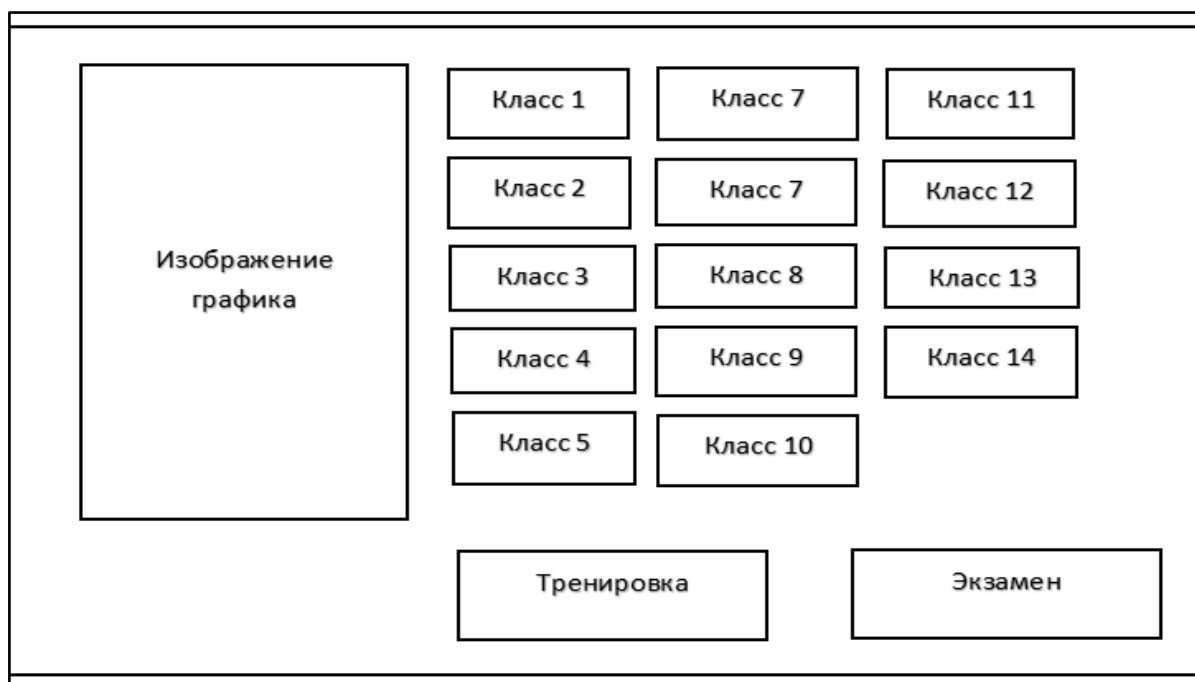


Рисунок 11 – Страница тренировки

Страница тренировки – на данной странице отображается форма с полями загрузки изображения, полученного из серверной части системы и кнопками выбора вариантов ответа. На рисунке 11 представлен макет страницы. Пользователь отмечает ответ кнопкой, система подкрепляет правильный ответ и отнимает баллы за неправильный. При достижении некоторого количества правильных ответов выдает поощрение.

Страница экзамена – на данной странице отображается форма с полями загрузки изображения, полученного из серверной части системы и кнопками выбора вариантов ответа, также результат экзамена и таймер. Время на экзамен ограничено 30 минутами, это обусловлено длительностью урока и необходимостью провести рефлекссию по итогам задания.

Страница ответа – на данной странице отображается образовательная информация и аналитика по решению тренировочных и экзаменационных заданий. Она включает объяснение правил решения задач разных классов и процент успешного решения задач по классам.

2.4. Проектирование REST API веб-приложения

Кроме графического интерфейса, веб-приложение имеет программный интерфейс, который реализуется на базе асинхронного REST API. При проектировании REST API веб-приложения необходимо учитывать ряд важных аспектов, чтобы обеспечить его эффективность, масштабируемость и безопасность. Важно создать простой, понятный и гибкий интерфейс, который будет удовлетворять потребности пользователей и обеспечивать эффективное взаимодействие с веб-приложением. Ниже перечислены доступные клиентским программам запросы.

1. `POST api/upload`. Этот запрос позволяет загружать изображения для классификации нейронной сетью. В теле запроса должны быть поля, содержащие изображение и любые дополнительные данные, необходимые для

классификации. В случае успешного выполнения запроса, сервер отправляет ответ в формате JSON с результатами классификации. Код ответа HTTP – «202 (Accepted)».

2. GET api/graph. Этот запрос позволяет пользователям запрашивать график функции, сгенерированный индивидуально в каждом запросе. Все графики уникальны и генерируются случайным выбором функции и случайным выбором коэффициентов. В ответ на запрос, сервер отправляет данные графика в формате JSON. Код ответа HTTP – «200 (OK)».

3. GET api/answer. Этот запрос позволяет пользователям запрашивать объяснение определенного вопроса. Объяснение представляет собой краткий анализ функции, отмечающий отличительные признаки ее графика. В ответ на запрос, сервер отправляет объяснение в формате JSON. Код ответа HTTP – «200 (OK)».

4. DELETE api/remove. Запрос DELETE позволяет пользователям удалять определенные ресурсы, такие как изображения или записи, связанные с классификацией или другими функциями веб-приложения. После успешного выполнения запроса, сервер отправляет ответ с кодом HTTP «204 (No Content)», подтверждая удаление ресурса.

Таким образом, REST API веб-приложения позволяет пользователям взаимодействовать с нейронной сетью для классификации изображений, запрашивать графики функций и получать объяснения различных вопросов или процессов. После завершения сеанса использования все данные пользователя стираются.

3. РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ

Приложение будет состоять из двух основных частей: серверной и клиентской. Серверная часть будет отвечать за обработку запросов и управление данными, а клиентская часть - за взаимодействие с пользователем и визуализацию информации.

3.1. Программные средства реализации

Для реализации этого проекта были выбраны Python и PyCharm. Python – это мощный и гибкий язык программирования, который широко используется в веб-разработке. PyCharm – это мощная среда разработки, которая предоставляет множество инструментов для упрощения разработки и отладки веб-приложений. Веб-интерфейс приложения разработан с помощью HTML, CSS. Для написания кода и его отладки был использован Postman.

В процессе разработки веб-приложения были использованы следующие библиотеки и фреймворки.

1. Flask (3.0.3) [13]. Веб-фреймворк на базе Python с открытым исходным кодом, предназначенный для быстрого создания небольших веб-приложений с возможностью масштабирования. Часто используется для создания RESTful приложений.

2. ReactJS (3.14.0) [17]. JavaScript-библиотека с открытым исходным кодом для разработки пользовательских интерфейсов.

3. MathJax (3.0) [18]. Библиотека, написанная на JavaScript, для отображения математических символов и формул в любом браузере. Не требует какой-либо настройки и построена на основе MathML.

4. Celery (5.2.7) [22]. Библиотека, реализующая асинхронную очередь задач, основанная на распределенной передаче сообщений.

5. RabbitMQ (3.11.16) [22]. Система обмена сообщениями, которая обеспечивает связь и взаимодействие между различными компонентами программного обеспечения.

6. MobX[21]. Библиотека для управления состоянием в приложениях React, которая позволяет создавать простые и масштабируемые приложения. Она использует концепцию реактивных объектов, что позволяет автоматически обновлять компоненты, когда изменяется состояние, что упрощает управление состоянием и устраняет необходимость ручного отслеживания и обновления зависимостей.

7. KaTeX[22]. Кроссбраузерная библиотека JavaScript, отображающая математические обозначения в веб-браузерах. Особый акцент в ней делается на том, что она быстрая и простая в использовании.

3.2. Реализация веб-приложения

Веб-приложение реализовано на базе фреймворков Flask и React. Также для организации очереди задач в системе, используется Celery и RabbitMQ.

Создается экземпляр приложения Flask, которое будет обслуживать веб-запросы. Используется расширение Flask-CORS для обработки запросов с разных источников.

В коде сервера веб-приложения реализованы представленные ниже функции.

Обработка запросов CORS. Функция `add_cors_headers` представлена в листинге 1, она добавляет заголовки CORS в ответ сервера, что позволяет клиентским приложениям из различных источников обращаться к серверу. Устанавливается путь для загрузки файлов и создается папка для хранения загруженных файлов, если она еще не существует.

Листинг 1 – Функция `add_cors_headers`

```
def add_cors_headers(response):
    response.headers.add('Access-Control-Allow-Origin', '*')
    if request.method == 'OPTIONS':
        response.headers['Access-Control-Allow-Methods'] = 'DELETE, GET,
POST, PUT'
        headers = request.headers.get('Access-Control-Request-Headers')
        if headers:
            response.headers['Access-Control-Allow-Headers'] = headers
    return response
```

Обработка запросов на загрузку файлов представлена в листинге 2. Функция обрабатывает запрос, сохраняет файл на сервере и возвращает сообщение об успешной загрузке.

Листинг 2 – Функция `generate_random_graph`

```
def generate_random_graph():
    # Выбираем случайную функцию из списка
    random_function = random.choice(graph_functions)
    graph_buffer = random_function()
    # Получаем название функции из словаря
    function_name = function_names[random_function]
    # Преобразуем буфер в base64 строку
    #graph_url = base64.b64encode(graph_buffer.getvalue()).decode('utf-8')
    return function_name, graph_buffer
```

Функция `get_graph` вызывает функцию `generate_random_graph` для генерации случайного графика, затем отправляет результаты в очередь RabbitMQ и возвращает их в формате JSON.

Проверка соединения с RabbitMQ: Функция `check_rabbitmq_connection` проверяет, работает ли соединение с RabbitMQ, и выводит соответствующее сообщение в консоль.

Реализация клиентской части веб-приложения написана на React, используется библиотека MobX. Ниже представлен код основной страницы сайта (листинг 3).

Листинг 3 – Основная страница Training

```
const handleClick = async (formulaId, formula) => {
  try {
    if (isCorrectAnswer(formulaId, functionName, formula)) {
      incrementProgress();
    } else {
      decrementProgress();
    }
    const function_n = TrainingStore.fetchGraph();
  } catch (error) {
    console.error('Ошибка при получении данных с сервера:', error);
  }
};

<MathJaxContext>
<div className="flex items-center flex-col h-screen space-y-4 ot">
  <label className="text-3xl text-center">Удачной тренировки!</label>
  <LinearProgress style={{ color: '#00ff00' }} size={24} variant="determinate" value={TrainingStore.progress} />
</div>
<div className="flex space-x-4">
  <div className="flex flex-col w-1/2 space-y-4">
    <Card >
      <CardHeader title="Тренировка" />
```



```

    {TrainingStore.graphUrl && (
      <div className="flex items-start">
        <img
          src={TrainingStore.graphUrl}
          alt="Generated Graph"
          style={{
            maxWidth: '100%',
            maxHeight: '90vh',
            objectFit: 'contain',
          }}
        />
      </div>
    )}
  </Card>
</div>
<div className="flex flex-grow ">
  <Grid container spacing={2}>
    {items.map((item) => (
      <Grid item xs={6} key={item.id}>
        <Button
          className="h-8 w-full"
          onClick={() => handleClick(item.id, item.formula)}
        >
          <MathJax>{\( \${item.formula} \)}</MathJax>
        </Button>
      </Grid>
    )}
  </MathJaxContext>
</div>
export default Training;

```

Данный код представляет собой компонент React, который использует библиотеку MobX для управления состоянием и эффектами. Компонент называется Training и является функциональным компонентом, который использует хук useState для управления состоянием внутри компонента.

Компонент Training использует следующие библиотеки и компоненты: React, MobX, Material-UI, React Router DOM, Axios, MathJax.

Основные функции и логика компонента

Импорт необходимых библиотек и компонентов. Инициализация состояния с помощью хуков useState и useEffect. Обработка нажатия на кнопку с помощью функции handleClick, которая вызывает методы из TrainingStore для проверки ответа и обновления прогресса. Хук useState используется для создания локального состояния. Использование useEffect для инициализации состояния компонента, загрузки данных с сервера и перемешивания формул. Отображение прогресс-бара с помощью

компонента `LinearProgress`. Отображение карточек с формулами и кнопок для выбора ответа. Обработка нажатия на кнопки для навигации между страницами.

Также в компоненте используется декоратор `observer` из `MobX`, который позволяет компоненту реагировать на изменения в хранилище `TrainingStore` и обновлять свое состояние в соответствии с этими изменениями. В отличие от других подходов, таких как `Redux`, `MobX` предлагает более простой и гибкий способ работы с состоянием, используя концепцию наблюдаемых данных и реакций.

Основные принципы использования `MobX` включают наблюдаемые данные, действия, расчеты, реакции, хранилища и компоненты. Наблюдаемые данные отмечаются декоратором и отслеживаются `MobX` для автоматического обновления. Действия, отмеченные декоратором `action`, представляют собой функции, вызываемые для обновления состояния и контролирующие процесс изменения данных.

Хранилища в `MobX` представляют собой объекты, которые содержат наблюдаемые данные, действия и расчеты, связанные с определенной функциональностью приложения, и используются для организации и разделения кода, связанного с состоянием.

3.3. Тестирование веб-приложения

Необходимо проверить работу веб-сервера на предмет его функциональности. Для этого было проведено функциональное тестирование для веб-интерфейса и для REST API. В таблицах 1 и 2 представлены результаты этих тестов.

В результате проведенного функционального тестирования веб-интерфейса и REST API было установлено, что веб-сервер функционирует корректно и обеспечивает необходимый функционал для работы с изображениями графиков функций.

Тестирование показало, что система успешно обрабатывает загрузку изображений, проверяет их формат, а также предоставляет информацию о классе изображения и его признаках через веб-интерфейс и REST API.

Таблица 1 – Результаты функционального тестирования веб-интерфейса

№	Название теста	Входные данные	Ожидаемый результат	Тест пройден?
1	Загрузка изображения	Загружается изображение с графиком одной из функций	Система возвращает класс изображения с разъяснением его признаков	Да
2	Загрузка файла некорректного формата	Загружает файл некорректного расширения, и нажимает на кнопку «Отправить»	Система возвращает ошибку, и отображается сообщение о правильном формате файла	Да
3	Выполнение тренировки	Пользователь на странице тренировки отвечает на вопрос.	Система возвращает количество правильных и неправильных ответов	Да
4	Выполнение экзамена	Пользователь на странице экзамена отвечает на вопросы в течение 20 мин	Система возвращает оценку в зависимости от процента правильных ответов	Да

Для тестирования API была использована программа Postman, в таблице 2 представлен набор тестов REST API сервиса. Тестирование направлено на проверку того, что каждый отдельный компонент или функция системы работает в соответствии с ее описанием и требованиями. В контексте тестирования REST API, это означает проверку того, что каждый запрос к API возвращает ожидаемые результаты и корректно обрабатывает различные типы данных и сценарии использования. В результате функционального тестирования веб-сервера были проверены его возможности для работы с веб-интерфейсом и REST API. Таким образом, бэкенд сайта работает верно.

Все тесты, представленные в таблицах 1 и 2, были успешно пройдены. Это подтверждает работоспособность сервера и его возможность обрабатывать запросы от пользователей, в том числе загрузку изображений, проведение тренировок и экзаменов, а также отправку запросов к API для получения данных и объяснений.

Таблица 2 – Результаты функционального тестирования REST API

№	Название теста	Входные данные	Ожидаемый результат	Тест пройден?
1	Пользователь отправляет изображение, которое необходимо классифицировать	Клиент отправляет запрос GET запрос по адресу /api/upload с изображением	Ответ с данными функции, чей график построен в изображении	Да
2	Пользователь на странице тренировка запрашивает график для самостоятельной классификации по одному из 14 классов	Клиент отправляет запрос GET запрос по адресу /api/graph	График получен, в формате JSON	Да
3	Получение отличительных признаков выбранного класса	Клиент отправляет запрос GET запрос по адресу. GET api/answer.	В ответ на запрос, сервер отправляет объяснение в формате JSON	Да

В целом, результаты тестирования позволяют сделать вывод о том, что веб-сервер работает корректно и эффективно. Ожидаемые функции выполняются в ожидаемом режиме, неожиданных проблем и ошибок при взаимодействии с клиентской частью не возникло.

ЗАКЛЮЧЕНИЕ

В данной работе была разработана веб-приложение для обучения математике с использованием нейронных сетей и методов машинного обучения. В ходе работы были проведены исследования в области обучения математике, а также были изучены существующие веб-приложения для обучения математике.

На основе проведенных исследований была разработана архитектура веб-приложения, которая включает в себя фронтенд и бэкенд части. Фронтенд часть веб-приложения была реализована с использованием HTML, CSS и JavaScript, а бэкенд часть была реализована с использованием Python и Flask.

В ходе работы были разработаны различные функции веб-приложения, такие как создание уроков, добавление задач, оценка результатов и т.д. Также была реализована система авторизации пользователей и система хранения данных в базе данных.

В результате работы было создано веб-приложение, которое позволяет пользователям изучать математику в удобной и интерактивной форме. Веб-приложение предоставляет возможность создавать уроки, добавлять задачи, оценивать результаты и многое другое.

В целом, данная работа представляет собой успешный пример разработки веб-приложения для обучения математике с использованием нейронных сетей и методов машинного обучения. Разработанное веб-приложение может быть использовано как студентами, так и преподавателями для изучения и преподавания математики.

ЛИТЕРАТУРА

1. Полупанов, Д.В. Программирование в Python 3: учебное пособие. // Уфа: БашГУ, 2020. – 164 с.
2. Шакирьянов, Э.Д. Компьютерное зрение на Python. Первые шаги: учебное пособие. – Москва: Лаборатория знаний, 2021. – 163 с.
3. Селянкин, В. В. Компьютерное зрение. Анализ и обработка изображений: учебник для вузов. – Санкт-Петербург: Лань, 2021. – 152 с.
4. Алексеев, Д. С. Технологии интеллектуального анализа данных: учебное пособие. – Кострома: КГУ им. Н.А. Некрасова, 2020. – 141 с.
5. Ян, Э. С. Программирование компьютерного зрения на языке Python. – Москва: ДМК Пресс, 2016. – 312 с.
6. Маккинни У. Python и анализ данных. – М: ДМК Пресс, 2020. – 540 с.
7. Борзунов С. В., Кургалин С. Д. Алгебра и геометрия с примерами на Python. – Санкт-Петербург: Лань, 2021. – 444 с.
8. Гудфеллоу Я., Бенджио И., Курвиль А. Глубокое обучение. – М: ДМК Пресс, 2018. – 652 с.
9. Демидов А. К., Кувшинов Б. М. Искусственный интеллект. – Челябинск: Издательство ЮУрГУ, 2008. – 65 с.
10. Ясницкий Л. Н. Введение в искусственный интеллект. – М.: Академия, 2008. – 174 с.
11. Воронина В. В. Теория и практика машинного обучения. – Ульяновск: УлГТУ, 2017. – 290 с.
12. Федеральная служба по надзору в сфере образования и науки. ФГБНУ «Федеральный институт педагогических измерений». [Электронный ресурс] URL: <https://fipi.ru/> (дата обращения: 25.01.2024 г.).
13. Flask Documetation. [Электронный ресурс] URL: <https://flask.palletsprojects.com/en/3.0.x/> (дата обращения: 25.02.2024 г.).
14. NumPy documentation. [Электронный ресурс] URL: <https://numpy.org/doc/stable/> (дата обращения: 01.02.2024 г.).

15. Matplotlib documentation. [Электронный ресурс] URL: <https://matplotlib.org/stable/index.html> (дата обращения: 20.02.2024 г.).
16. Malhar L. Building Web Apps with Python and Flask: Learn to Develop and Deploy Responsive RESTful Web Applications Using Flask Framework (English Edition). – BPB Publications, 2021. – 262 с.
17. React Documetation. [Электронный ресурс] URL: <https://react.dev/> (дата обращения: 25.02.2024 г.).
18. MathJax documentation. [Электронный ресурс] URL: <https://www.mathjax.org/> (дата обращения: 20.03.2024 г.).
19. Python Flask and Django | Full Stack Python for Web Development: Build Web Applications in Python Using Flask and Django Frameworks. – Emenwa Global, 2022. – 280 с.
20. Букунов С. В. Разработка приложений с графическим пользовательским интерфейсом на языке Python. – Лань, 2023. – 88 с.
21. MobX Documetation. [Электронный ресурс] URL: <https://mobx.js.org/> (дата обращения: 22.04.2024 г.).
22. KaTeX Documetation. [Электронный ресурс] URL: <https://katex.org/docs/node> (дата обращения: 25.04.2024 г.).

ПРИЛОЖЕНИЯ

Приложение А. Страницы тренировки и загрузки файла

Скриншоты разработанного приложения приведены на рисунках 1–2.

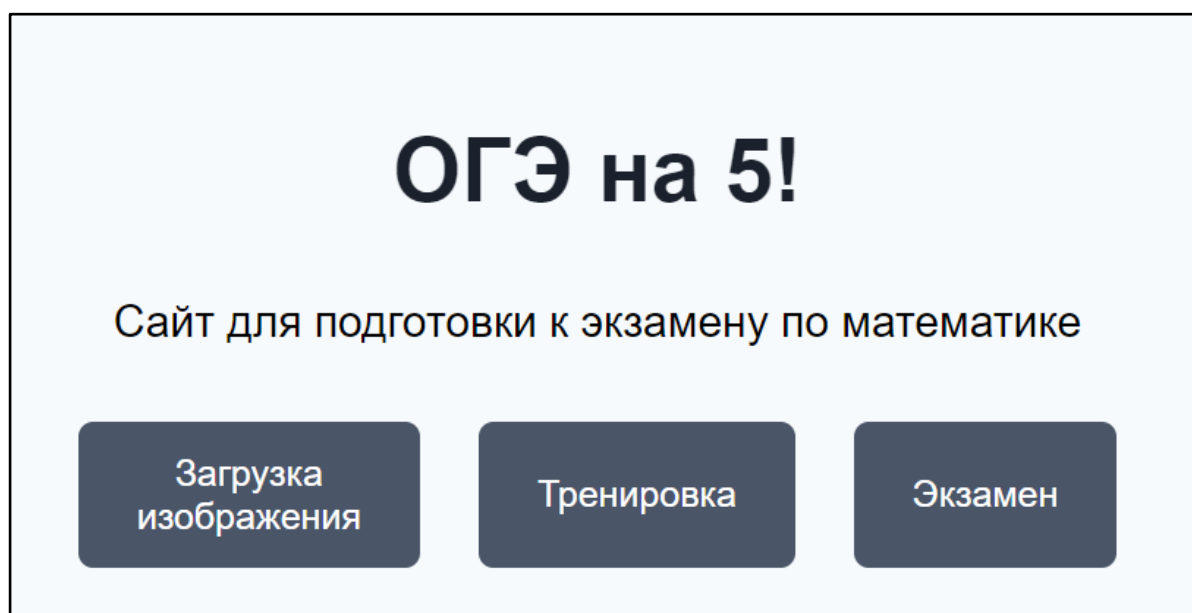


Рисунок 1 – Главная страница

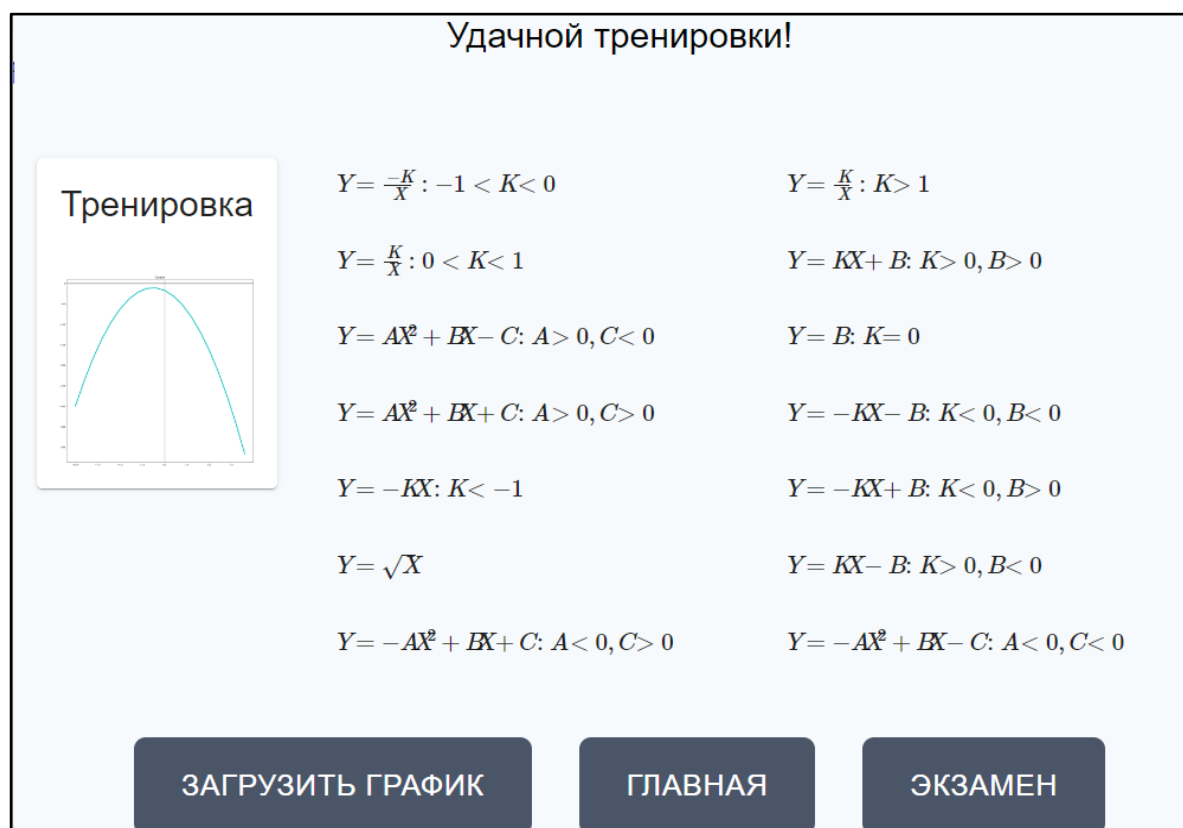


Рисунок 2 – Страница тренировки

Приложение Б. Код страницы Upload

Листинг 1 – Основная страница UPLOAD

```
import './style.css'; // Импортируем ваш CSS-файл
import { useNavigate } from 'react-router-dom';
import axios from 'axios';
import React, { useState } from 'react';
const Description = () => {
  const navigate = useNavigate();
  const handleTrainingClick = () => {
    // Выполните любые необходимые действия перед переходом
    // Например, вы можете сохранить данные формы или активировать API-вызов
    // Затем перейдите на страницу тренировки
    navigate('/training');});
  const handleExamClick = () => {
    // Выполните любые необходимые действия перед переходом
    // Например, вы можете сохранить данные формы или активировать API-вызов
    // Затем перейдите на страницу тренировки
    navigate('/exam');});
  const handleIndexClick = () => {
    // Выполните любые необходимые действия перед переходом
    // Например, вы можете сохранить данные формы или активировать API-вызов
    // Затем перейдите на страницу тренировки
    navigate('/')});
  return (
const UploadForm = () => {
  const [file, setFile] = useState(null);
  const submitFile = async (e) => {
    e.preventDefault();
    const formData = new FormData();
    formData.append('file', file);
    try {
      const res = await axios.post('http://localhost:5000/upload', formData,
        headers: {
          'Content-Type': 'multipart/form-data',
        },
      );
      console.log(res.data);
    } catch (ex) {
      console.log(ex);
    }
  };

  const handleFileUpload = (e) => {
    setFile(e.target.files[0]);
  };

  return (
    <form className="min-h-screen flex flex-col justify-center items-center"
      id="upload-container" onSubmit={submitFile}>
      
      <div>
        <input id="file-input" type="file" name="file" onChange={handleFileUpload} />
        <label htmlFor="file-input">Выберите файл</label>
        <span> или перетащите его сюда</span>
      </div>
      <button type="submit">Отправить</button>
    </form>
  );
};
```

Окончание листинга 1 приложения Б

```
};  
//export default UploadForm;  
const App = () => {  
  return (  
    <div className="min-h-screen">  
      <div className="flex items-start space-x-10">  
        <UploadForm />  
        <Description />  
      </div>  
    </div>  
  );  
};  
  
export default App;
```