



**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(национальный исследовательский университет)»**

**Институт (Филиал) № 8 «Компьютерные науки и прикладная математика» Кафедра 806
Группа М8О-408Б-20 Направление подготовки 01.03.02 «Прикладная математика и
информатика»**

Профиль Информатика

Квалификация: бакалавр

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

БАКАЛАВРА

на тему: Программный инструмент для распознавания научного текста и преобразования его в L^AT_EX код

Автор ВКРБ: Ядров Артем Леонидович (_____)
Руководитель: Миронов Евгений Сергеевич (_____)
Консультант: Зайцев Валентин Евгеньевич (_____)
Консультант: — (_____)
Рецензент: — (_____)

К защите допустить

Заведующий кафедрой № 806 Крылов Сергей Сергеевич (_____
мая 2024 года

РЕФЕРАТ

Выпускная квалификационная работа бакалавра состоит из 53 страниц, 41 рисунка, 23 использованных источников, 1 приложения.

ОБРАБОТКА ИЗОБРАЖЕНИЙ, ПРЕОБРАЗОВАНИЕ ТЕКСТА, МАШИННОЕ ОБУЧЕНИЕ, НЕЙРОННЫЕ СЕТИ

Объектом исследования в данной работе является научный текст и способы его обработки.

Цель работы — разработка платформы, выполняющей распознавание научного текста и генерацию готового L^AT_EX-кода.

Основное содержание работы состояло в разработке алгоритма распознавания формул на изображении научного текста и последующей генерации L^AT_EX-кода данного текста.

Основными результатами работы, полученными в процессе разработки, является обученная для поиска формул на изображении нейросетевая модель, модуль для генерации L^AT_EX-кода.

Результаты разработки предназначены для автоматического преобразования научного текста в L^AT_EX-код.

Использование результатов данной работы позволяет сократить время верстки научного документа, а также оцифровывать отсканированные источники.

СОДЕРЖАНИЕ

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ	4
ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ	5
ВВЕДЕНИЕ	6
1 ПОСТАНОВКА ЗАДАЧИ И ТЕОРЕТИЧЕСКИЕ ПРЕДПОСЫЛКИ	9
1.1 Пользовательские сценарии	9
1.1.1 Авторизация пользователя	10
1.1.2 Преобразование фотографии	11
1.1.3 Преобразование файла формата .pdf	11
1.1.4 Кроссплатформенная обработка данных	12
1.1.5 Контроль обучения классификатора	12
1.1.6 Замена нейросетевых моделей	13
1.2 Метрики	14
1.2.1 Метрики нейросетевых моделей	14
2 РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА	16
2.1 Высокоуровневая архитектура нейронного модуля	16
2.2 Коррекция перспективы	18
2.2.1 Неправильное распознавание	22
2.3 Сегментация на абзацы	24
2.3.1 Сегментация изображения	25
2.3.2 Сегментация pdf файла	28
2.4 Выделение формул	32
2.4.1 Модель	32
2.4.2 Данные для обучения	33
2.4.3 Обучение	35
2.4.4 Корректировка результата	39
2.5 Генерация L ^A T _E X кода	41
3 Тестирование программного продукта	46
3.1 Демонстрация программного продукта	46
ЗАКЛЮЧЕНИЕ	50
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	51
ПРИЛОЖЕНИЕ А Исходный код	53

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

В настоящей выпускной квалификационной работе бакалавра применяют следующие термины с соответствующими определениями:

LATEX — система компьютерной вёрстки, широко используемая для создания научных и технических документов. Она основана на языке разметки TeX, созданном Дональдом Кнутом, и включает в себя множество макросов и стилей, которые облегчают работу с текстом и форматирование документов

True Positives (TP) — количество правильных предсказаний положительного класса

False Positives (FP) — количество неправильных предсказаний положительного класса (когда модель предсказывает положительный класс, а на самом деле объект принадлежит к отрицательному классу)

Precision — одна из метрик машинного обучения, используется для оценки точности модели в отношении положительных предсказаний. Вычисляется по формуле

$$Precision = \frac{TP}{TP + FP}$$

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

В настоящей выпускной квалификационной работе бакалавра применяют следующие сокращения и обозначения:

mAP — Mean Average Precision

IoU — Intersection over Union

PDF — Portable Document Format

DVI — Device Independent file format

GRPC — Google Remote Procedure Call

API — Application Programming Interface

ВВЕДЕНИЕ

В России на постоянной основе проводятся научные исследования во многих областях. Результаты этих исследований публикуются в виде научных статей, которые являются важным инструментом для распространения новых знаний и научных открытий. Например, по данным "Научной электронной библиотеки" опубликовано 52573694 [1] статей, и большинство из них написано с помощью \LaTeX — мощного инструмента для верстки и оформления математических формул и научных текстов, который позволяет создавать качественные и профессионально оформленные статьи. Также с помощью \LaTeX можно готовить конспекты к предметам, причем это может делать как преподаватель, так и студент.

Так как \LaTeX — система верстки со своим особенным синтаксисом, то набор даже простых формул в нем может оказаться сложным процессом. Покажем весь процесс написания \LaTeX -кода на следующем простом примере

$$f(x, y, \alpha, \beta) = \frac{\sum_{n=1}^{\infty} A_n \cos\left(\frac{2n\pi x}{\nu}\right)}{\prod \mathcal{F}_g(x, y)} \quad (1)$$

На рисунке 1 показан листинг \LaTeX -кода формулы 1:

```
1 f(x, y, \alpha, \beta) = \frac{%
2   \sum \limits_{n=1}^{\infty} A_n \cos%
3     \left( \frac{2 n \pi x}{\nu} \right)%
4   }{%
5     \prod \mathcal{F}_g(x, y)%
6   }%
7 }
```

Рисунок 1 – Листинг формулы 1

На данном листинге мы наблюдаем большое разнообразие команд \LaTeX . Все команды в \LaTeX начинаются со знака '\', аргументы команд должны заключаться в фигурные скобки '{}', а опции команды — в квадратные '[]'. Также на листинге приведен пример вложенности команд друг в друга. Данные особенности усложняют не только чтение, но и в первую очередь написание кода документа.

После написания исходного кода документа, необходимо произвести

компиляцию в один из поддерживаемых форматов: *.pdf* и *.dvi*. На данном этапе отсеиваются все синтаксические ошибки, компилятор укажет (не всегда однозначно и понятно) на них в логе. Однако, такие ошибки как пропущенный или неправильно написанный символ, выравнивание и пр. не являются синтаксическими, и обнаружить их можно лишь при просмотре готового документа. После исправления ошибки необходимо заново компилировать документ, что увеличивает время написания документа.

Автоматическая генерация \LaTeX -кода позволит сократить количество компиляций и уменьшить допускаемые человеком ошибки в написании формул, что ускорит процесс верстки документов.

Также существует множество источников, оцифрованных в форматы *pdf*, *djvu* и др. Хранение в отсканированном виде источников обладает недостатками, такими как невозможность поиска по тексту и быстрого перемещения по нему. Возможность преобразовывать данные файлы в \LaTeX -код позволит получить данные источники в оцифрованном, а не отсканированном виде, что устранит приведенные недостатки.

Стоит отметить, что уже существуют решения, позволяющие выполнять генерацию \LaTeX -кода по изображению, например, *Mathpix* [2]. Однако, данное решение обладает рядом недостатков:

- Распространение по платной подписке;
- Отсутствие открытого исходного кода;
- Ограниченнная поддержка языков: основным поддерживаемым языком является английский.

В настоящее время появляется все больше различных нейросетей (например, Гигачат [3], ChatGPT [4], Midjourney [5] и тд). Некоторые из них обучены под задачи распознавания текста, в том числе математического. Также существуют модели, способные генерировать готовый код.

На основании вышенаписанного возникает идея автоматизации процесса преобразования научного текста на изображении в готовый \LaTeX -код.

Целью работы является разработка платформы, выполняющей распознавание научного текста и генерацию готового \LaTeX -кода.

Для достижения поставленной цели в работе были решены следующие задачи:

- Определение пользовательских сценариев;
- Определение требований к платформе;

- Проектирование архитектуры платформы;
- Разработка сервиса для распознавания научного текста;
- Разработка сервиса для синхронизации локальных и облачных пользовательских данных;
- Разработка *Desktop* приложения для взаимодействия пользователя с сервером;
- Тестирование и анализ прототипа платформы.

Для разработки программного обеспечения необходимо изучить технологии и методы, решающие поставленные задачи. Работа основывается на следующих библиотеках, технологиях, алгоритмах:

- *Python* является основным языком программирования, который использовался для решения задач;
- *Tensorflow* — библиотека для запуска и обучения моделей;
- *YOLO* — модель, созданная для классификации объектов на изображении;
- *openCV* — библиотека для обработки изображения;
- *tkinter* — *Python* библиотека для работы с *GUI*;
- *grpc* — фреймворк для удаленного вызова процедур;
- *Weights & Biases* — сервис для отслеживания метрик обучения.

Результатом работы является:

- Сервис для коррекции перспективы изображения;
- Сервис для нахождения формул на изображении;
- Дообученная модель *YOLO*, позволяющая находить формулы на изображении;
- Сервис для преобразования найденных формул в *LAT_EX*-код;
- *Desktop* приложение с авторизацией в *Google Drive* и взаимодействием с удаленным сервером с помощью *grpc*.

1 ПОСТАНОВКА ЗАДАЧИ И ТЕОРЕТИЧЕСКИЕ ПРЕДПОСЫЛКИ

Концепция пользовательских сценариев (*usecases*) является ключевым элементом в разработке программного обеспечения и системного анализа. Они помогают детально описать взаимодействие между пользователем и системой, выявить требования и ожидания к функциональности системы. Пользовательские сценарии не только улучшают понимание разрабатываемой системы, но и способствуют выявлению возможных проблем на ранних этапах проектирования, что позволяет сократить затраты на доработку и повысить качество конечного продукта.

Далее будут представлены конкретные сценарии, охватывающие основные функции системы.

1.1 Пользовательские сценарии

Прежде чем определять конкретные пользовательские сценарии, покажем высокоуровневую архитектуру системы. Она представлена на рисунке 2.

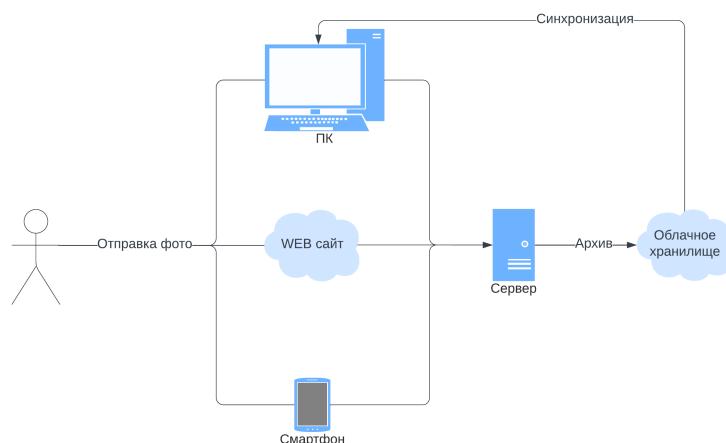


Рисунок 2 – Высокоуровневая архитектура системы

Под пользователем приложения будем подразумевать пользователя одного из типов приложения:

- *WEB* приложение;
- *Desktop* приложение;
- мобильное приложение.

Для начала определим основные роли целевых пользователей:

- Пользователь приложения;
- Разработчик — программист, занимающийся разработкой конкретных частей приложения;
- Архитектор — работник, занимающийся планированием и разработкой высокоуровневой архитектуры приложения;
- Администратор системы — специалист, отвечающий за настройку, управление и контроль работы системы;
- Инженер по машинному обучению — специалист, занимающийся разработкой моделей машинного обучения;
- Аналитик данных — специалист, занимающийся оценкой качества обучения модели, выбором подходящих метрик, а также оптимизацией процесса обучения.

1.1.1 Авторизация пользователя

Участники: пользователь любого из типов приложения

Предусловие: пользователь не авторизован

Постусловие: пользователь авторизован в один из сервисов и предоставлен доступ к облачному хранилищу

Сценарий: Для авторизации с помощью *Google* необходимо реализовать следующий сценарий, изображенный на рисунке 3:

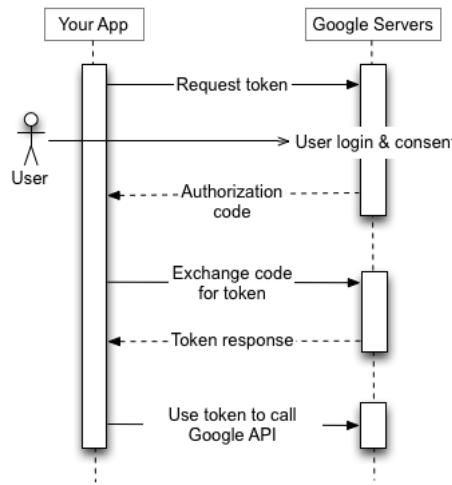


Рисунок 3 – Сценарий сетевого взаимодействия при авторизации пользователя [6]

По сути *Google* делает всё сам на этапе создания сервиса внутри кода. Если приложение не видит токена для авторизации, то происходит

автоматическое перенаправление на сайт авторизации. Дальнейший токен сохраняется локально.

Для замены аккаунта достаточно удалить локальный токен пользователя и перезапустить сервис *Google*. Стоит отметить, что лучше сохранять токен для дальнейшего быстрого входа в сервис. Но так как мы разрабатываем прототип, то и этого решения будет вполне достаточно.

1.1.2 Преобразование фотографии

Участники: пользователь приложения

Предусловие: пользователь сделал фото научного текста и открыл его в приложении

Постусловие: пользователь получает готовый L^AT_EX код текста, полученного на изображении в виде архива из исходного кода и *pdf* файла

Сценарий:

- Производится автоматическая коррекция перспективы фотографии;
- Производится автоматическое сегментация текста на абзацы;
- С приложения на сервер отправляется изображение, а также таблица с координатами начала и конца абзацев;
- С сервера на приложение отправляется таблица с найденными формулами;
- Пользователь проверяет правильность распознавания формул и вносит корректиды;
- Приложение отправляет на сервер таблицу с финальными формулами;
- Сервер загружает архив с L^AT_EX кодом и *pdf* файлом в *Google Drive*, а также отсылает его пользователю;
- *Desktop* приложение обновляет папку *Google Drive* и загружает в локальное хранилище последний архив.

При необходимости пользователь может корректировать точки перспективы, а также точки абзацев на изображении.

1.1.3 Преобразование файла формата .pdf

Участники: пользователь приложения

Предусловие: пользователь загрузил файл научного текста в формате .*pdf*

в приложение

Постусловие: пользователь получает готовый L^AT_EX код текста, полученного на изображении в виде архива из исходного кода и pdf файла

Сценарий: В случае с .pdf файлом мы предполагаем, что нет необходимости корректировать перспективу.

- Производится автоматическое сегментация текста на абзацы;
- С приложения на сервер отправляется изображение, а также таблица с координатами начала и конца абзацев;
- С сервера на приложение отправляется таблица с найденными формулами;
- Пользователь проверяет правильность распознавания формул и вносит корректиды;
- Приложение отправляет на сервер таблицу с финальными формулами;
- Сервер загружает архив с L^AT_EX кодом и pdf файлом в *Google Drive*, а также отсылает его пользователю;
- *Desktop* приложение обновляет папку *Google Drive* и загружает в локальное хранилище последний архив.

1.1.4 Кроссплатформенная обработка данных

Участники: пользователь приложения

Предусловие: пользователь загружает фото/pdf файл с мобильного устройства, пользователь авторизован в *Desktop* версии приложения

Постусловие: пользователь получает готовый L^AT_EX код текста на *Desktop* устройство

Сценарий:

- С мобильного приложения на сервер отправляется изображение;
- Сервер обрабатывает изображение;
- Сервер отправляет данные на *Desktop* клиент.

1.1.5 Контроль обучения классификатора

Так как самая важная часть системы (распознавание текста) реализуется на основе нейросетевых алгоритмов (о чем будет подробно рассказано позже), важно контролировать процесс обучения. Для этого необходимо иметь доступ

к метрикам модели, а также обновлять обучающие наборы данных, проводить тестирование модели на новых данных, проводить валидацию в процессе обучения.

Важно быть уверенным в том, что модель не переобучена и способна обобщать данные.

Для этих целей очень хорошо подходит платформа *Weights & Biases* [7]

Участники: аналитик данных

Предусловие: существует рабочая система, обученная модель

Постусловие: выводится информация об обучении модели

Сценарий:

- Пользователь заходит на веб-сайт *wandb* [7];
- Пользователь выбирает интересующую его версию модели;
- Платформа выводит всю информацию о модели.

1.1.6 Замена нейросетевых моделей

Необходимо предусмотреть возможность замены одной модели на другую. Это является важным требованием к системе, поскольку позволяет улучшать эффективность и точность работы всей системы:

- Это дает возможность адаптировать систему под изменяющиеся условия. Если модель устаревает или не удовлетворяет требованиям, необходимо иметь возможность быстро ее заменить;
- Это позволяет снизить затраты на разработку и поддержку системы. Вместо разработки или поддержки своей модели, можно заменить текущую модель на готовое открытое решение;
- Это повышает гибкость системы. При изменении требований к системе, текущая модель может не удовлетворять новым требованиям.

Участники: инженер по машинному обучению, разработчик системы, аналитик данных

Предусловие: существует система

Постусловие: существует рабочая система, удовлетворяющая текущим требованиям

Сценарий:

- Инженер по машинному обучению разрабатывает новую модель или находит открытое решение;

- Разработчик системы встраивает новую модель в систему;
- Аналитик проверяет работоспособность модели, а также удовлетворение модели текущим требованиям;
- По результатам проверки:
 - в случае удовлетворения требованиям старая модель заменяется на новую;
 - в случае неудовлетворения требованиям цикл повторяется.

1.2 Метрики

Мы определили ожидаемое поведение системы. Теперь мы установим критерии для оценки систем, за которыми ведется наблюдение. Оптимальный метод оценки отдельных компонентов системы заключается в использовании метрик, так как они позволяют сравнивать разные состояния системы и выбирать наилучшее из них.

Метрики могут быть различными и зависят от специфики конкретной системы. Например, для оценки производительности может использоваться метрика времени ответа системы на запрос пользователя, а для оценки надежности - частота отказов системы.

Важно отметить, что выбор метрик должен основываться на целях и задачах проекта. Каждая система имеет свои особенности и требования к работе, поэтому необходимо определить наиболее значимые параметры для ее функционирования.

Кроме того, использование метрик позволяет проводить анализ данных о работе системы и делать выводы об эффективности ее работы. Это помогает оптимизировать процессы и повышать качество продукта.

В данной работе ключевую роль играют метрики нейросетевых моделей.

1.2.1 Метрики нейросетевых моделей

Метрики, применяемые к нейросетевым моделям, можно глобально разделить на:

- Метрики качества модели — показатели, позволяющие оценивать эффективность работы модели;
- Метрики потерь — показатели, позволяющие оценивать степень ошибки или несоответствия между прогнозами модели и

реальными данными.

В качестве метрик качества модели мы выбрали следующие:

- *Precision* — точность обнаруженных объектов, указывающая, сколько обнаружений было выполнено правильно;
- *Recall* — способность модели идентифицировать все экземпляры объектов на изображениях;
- *mAP50* — средняя точность, рассчитанная при пересечении, превышает порог объединения (*IoU*), равный 0,50. Это показатель точности модели, учитывающий только "простые" обнаружения;
- *mAP50 – 95* — среднее значение средней точности, рассчитанное при различных пороговых значениях *IoU*, варьирующихся от 0,50 до 0,95. Оно дает полное представление о производительности модели на разных уровнях сложности обнаружения.

В качестве метрик потерь мы выступает метрика *cls loss* (*Classification loss*) — потеря классификации, определяющая погрешность в прогнозируемых вероятностях классов для каждого объекта на изображении по сравнению с исходным значением. Меньшее значение этой метрики означает, что модель более точно предсказывает класс объектов

2 РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА

Конечный продукт является целостной системой, и необходимо рассказать про каждую ее часть. Ключевым элементом системы является нейронный модуль, обеспечивающий обработку данных, предоставленных пользователем, с использованием методов нейронного обучения. Поэтому необходимо начать именно с него.

2.1 Высокоуровневая архитектура нейронного модуля

Поскольку задача преобразования изображения в текст, выполняемая модулем, не является тривиальной, было решено разбить ее на несколько этапов. Схема разбиения представлена на рисунке 4

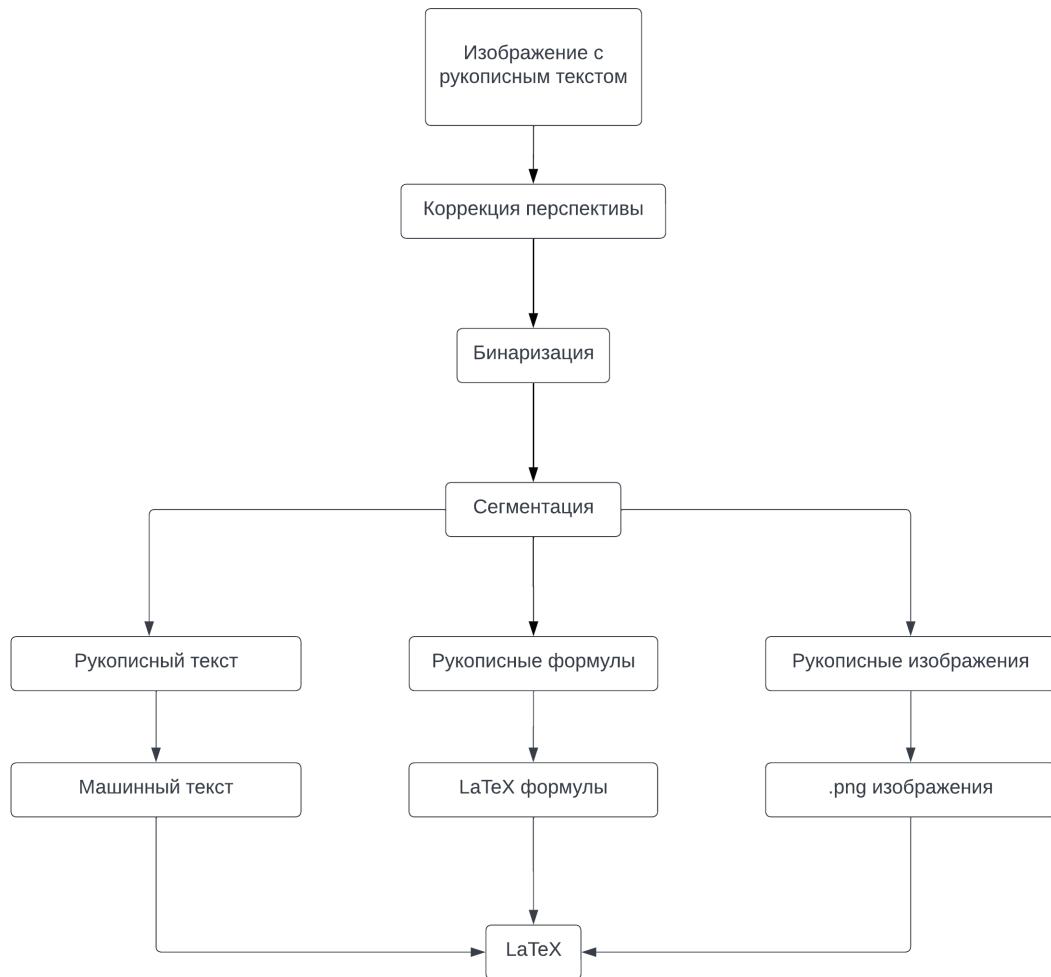


Рисунок 4 – Общая архитектура модели

Каждый из этапов данной схемы можно охарактеризовать входными

данными и результатом работы модуля (выходными данными). Такой подход обладает рядом преимуществ:

- Гибкость и масштабируемость: Модульная структура позволяет легко добавлять новые компоненты или модифицировать существующие без необходимости переписывать всю модель;
- Ускорение процесса обучения: Благодаря возможности параллельной обработки данных, модульные нейронные сети обучаются быстрее, чем монолитные модели;
- Улучшение качества модели: Разделение модели на модули позволяет специалистам сосредоточиться на оптимизации каждого компонента, что в итоге приводит к улучшению общей производительности модели;
- Простота внедрения новых технологий: Модульная архитектура облегчает внедрение новых технологий и подходов, таких как трансферное обучение или диффузионные модели;
- Улучшение производительности: некоторые модули могут исполняться в препроцессинге на клиентской машине, что ослабляет нагрузку на сервер.

Однако, имеются и недостатки:

- Проблемы с совместимостью: Разные модули могут использовать различные архитектуры, форматы данных и методы обучения, что может привести к проблемам совместимости;
- Риск ухудшения производительности: Неправильно спроектированные или плохо интегрированные модули могут снизить общую производительность модели;
- Необходимость в дополнительных ресурсах: Для обучения и развертывания модульных моделей часто требуются дополнительные ресурсы, такие как GPU или TPU.

Несмотря на недостатки, в современных реалиях важно уметь быстро и без проблем масштабироваться и заменять при необходимости один компонент другим, поэтому было принято решение использовать модульную архитектуру.

Подробнее разберем каждый этап данной схемы.

2.2 Коррекция перспективы

Коррекция перспективы необходима для устранения шума на изображении и получения лучшего результата. В данной работе коррекция перспективы осуществляется при помощи библиотеки для обработки изображений *openCV*. Использование данной библиотеки обладает следующими преимуществами:

- Простота внедрения: *openCV* предоставляет готовые функции для коррекции перспективы, например, *cv2.warpPerspective* [8];
- Отсутствие необходимости в больших данных для обучения;
- Детерминированность: результат всегда является детерминированным, так как основан на точных математических алгоритмах;
- Производительность: библиотека написана на производительном языке *C++* с использованием различных оптимизаций;
- Небольшие вычислительные затраты: по сравнению с нейросетевыми моделями, использование *openCV* требует значительно меньших ресурсов;
- Поддержка различных платформ: библиотека является кросс-платформенной и поддерживает основные платформы, такие как *Windows*, *Linux*, *macOS*, *Android*, *iOS*;
- Поддержка и документация: *openCV* обладает широким сообществом, а также обширной документацией с примерами использования тех или иных алгоритмов.

Сам алгоритм коррекции перспективы состоит из нескольких этапов, представленных на рисунке 5.

Необходимо подчеркнуть, что процесс коррекции перспективы базируется исключительно на алгоритмах анализа изображений без привлечения сложных алгоритмов нейросетевого обучения. Поэтому в целях экономии ресурсов сервера, а в следствии улучшения производительности, было принято решение выполнять данный этап на ЭВМ клиента.

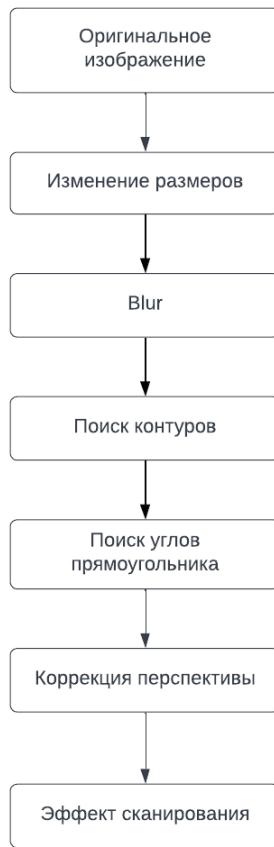


Рисунок 5 – Этапы коррекции перспективы изображения

На начальном этапе мы имеем изображение, показанное на рисунке 6

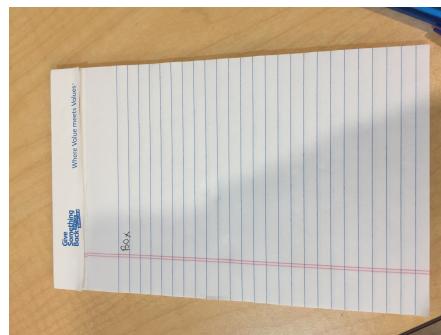


Рисунок 6 – Начальное изображение

Для начала необходимо удалить текст с изображения. Для этого преобразуем изображение в серый цвет и применим к нему размытие Гаусса [9]. На выходе данного этапа имеем изображение, представленное на рисунке



Рисунок 7 – Изображение после размытия Гаусса

Для поиска контуров необходимо выделить ребра. Для этого используется алгоритм Канни [10]. На выходе имеем изображение, представленное на рисунке 8

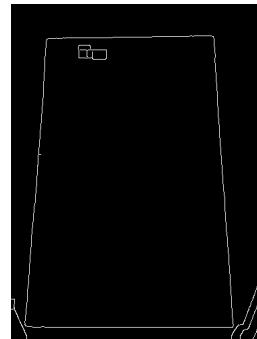


Рисунок 8 – Ребра, найденные на изображении

После нахождения ребер поиск контуров осуществляется двумя способами:

- a) С помощью алгоритма *Line Segment Detector* [11]
- б) С помощью встроенного в *openCV* алгоритма поиска контуров [12]

Опишем подробнее поиск контуров на основе найденных линий: после прохода алгоритма имеем изображение, представленное на рисунке 9

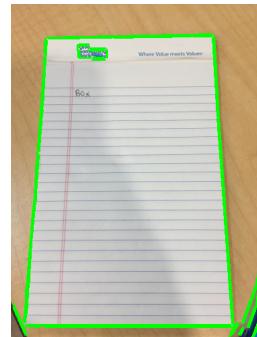


Рисунок 9 – Найденные на изображении линии с помощью алгоритма *LSD*

Контур определяется как пересечение горизонтальных и вертикальных линий. На выходе имеем найденные углы контура, показанные на рисунке 10

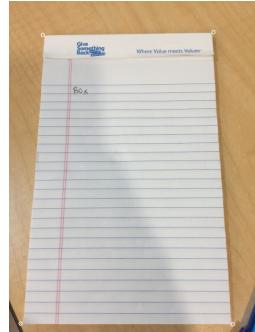


Рисунок 10 – Найденные на изображении контуры на основе линий

С помощью библиотеки *openCV* контуры находятся следующим образом:

- а) Находятся 5 наибольших по площади контуров
- б) Найденные контуры проверяются на количество углов, минимальную площадь контура

Среди всех подходящих контуров выбирается наибольший по площади, как показано на рисунке 11

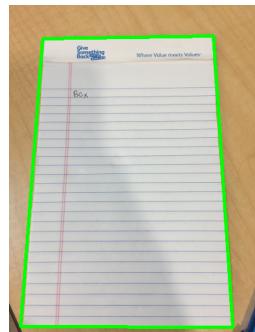


Рисунок 11 – Найденные алгоритмом контуры

На основе найденного контура, представляющего лист бумаги, осуществляем коррекцию перспективы. Для этого находим матрицу коррекции [8] и применим ее к изображению [13]. Получаем результирующее изображение, показанное на рисунке 12

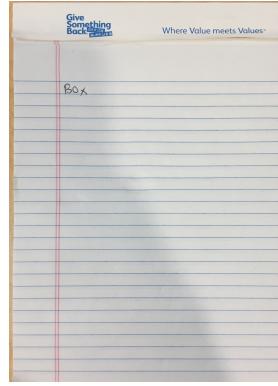


Рисунок 12 – Изображение с коррекцией перспективы

Далее необходимо добавить эффект сканирования. Эффект достигается путем применения к композиции небольшого размытия и серого изображения алгоритма сегментации *Adaptive Threshold* [14]. В конечном итоге имеем результирующее изображение показанное на рисунке 13

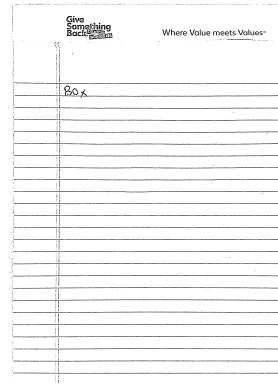


Рисунок 13 – Результирующее изображение

2.2.1 Неправильное распознавание

Стоит отметить, что данный алгоритм не является универсальным. Например, если на изображении находится посторонний шум (например, палец на бумаге, экран монитора, большое здание в виде прямоугольника), то вероятность получения неверного результата кратно возрастает. Именно с целью защиты от таких случаев в конечном продукте пользователь должен удостовериться в правильности найденного контура и отредактировать границы контура при необходимости. Пример входного изображения, дающего неверный результат, и найденные на нем контуры приведены на рисунках 14 и 15 соответственно.

***o*-обозначения**

Асимптотическая верхняя граница, предоставляемая *O*-обозначениями, может описывать асимптотическое поведение функции с разной точностью. Граница $2n^2 = O(n^2)$ дает правильное представление об асимптотическом поведении функции, а граница $2n = O(n^2)$ — нет. Для указания того, что верхняя граница не является асимптотически точной оценкой функции, применяются *o*-обозначения. Приведем формальное определение множества $o(g(n))$ (произносится как “о малое от g от n ”):

$$\begin{aligned} o(g(n)) = \{f(n) : & \text{ для любой положительной константы } c > 0 \\ & \text{ существует константа } n_0 > 0, \text{ такая, что} \\ & 0 \leq f(n) < cg(n) \text{ для всех } n \geq n_0\} \end{aligned}$$

Например, $2n = o(n^2)$, но $2n^2 \neq o(n^2)$.

Определения *O*-обозначений и *o*-обозначений похожи между собой. Основное отличие в том, что определение $f(n) = O(g(n))$ ограничивает функцию $f(n)$ неравенством $0 \leq f(n) \leq cg(n)$ лишь для некоторой константы $c > 0$, а определение $f(n) = o(g(n))$ ограничивает ее неравенством $0 \leq f(n) < cg(n)$ для всех констант $c > 0$. Интуитивно понятно, что в *o*-обозначениях функция $f(n)$ пренебрежимо мала по сравнению с функцией $g(n)$ при n , стремящемся к бесконечности, т.е.

$$\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} = 0. \quad (3.1)$$

Некоторые авторы используют этот предел в качестве определения *o*-обозначений. Добавим, что определение, данное в этой книге, накладывает на безымянную функцию ограничение, согласно которому она должна быть асимптотически неотрицательной.

***ω*-обозначения**

По аналогии *ω*-обозначения соотносятся с *Ω*-обозначениями точно так, как *o*-обозначения с *O*-обозначениями. С помощью *ω*-обозначений указывается нижний предел, не являющийся асимптотически точной оценкой. Один из возможных способов определения *ω*-обозначения следующий:

$$f(n) \in \omega(g(n)) \text{ тогда и только тогда, когда } g(n) \in o(f(n)).$$

Формально же $\omega(g(n))$ (произносится как “омега малое от g от n ”) определяется как множество

$$\begin{aligned} \omega(g(n)) = \{f(n) : & \text{ для любой положительной константы } c > 0 \\ & \text{ существует константа } n_0 > 0, \text{ такая, что} \\ & 0 \leq cg(n) < f(n) \text{ для всех } n \geq n_0\} \end{aligned}$$

Рисунок 14 – Пример неправильно распознанного изображения: входное изображение

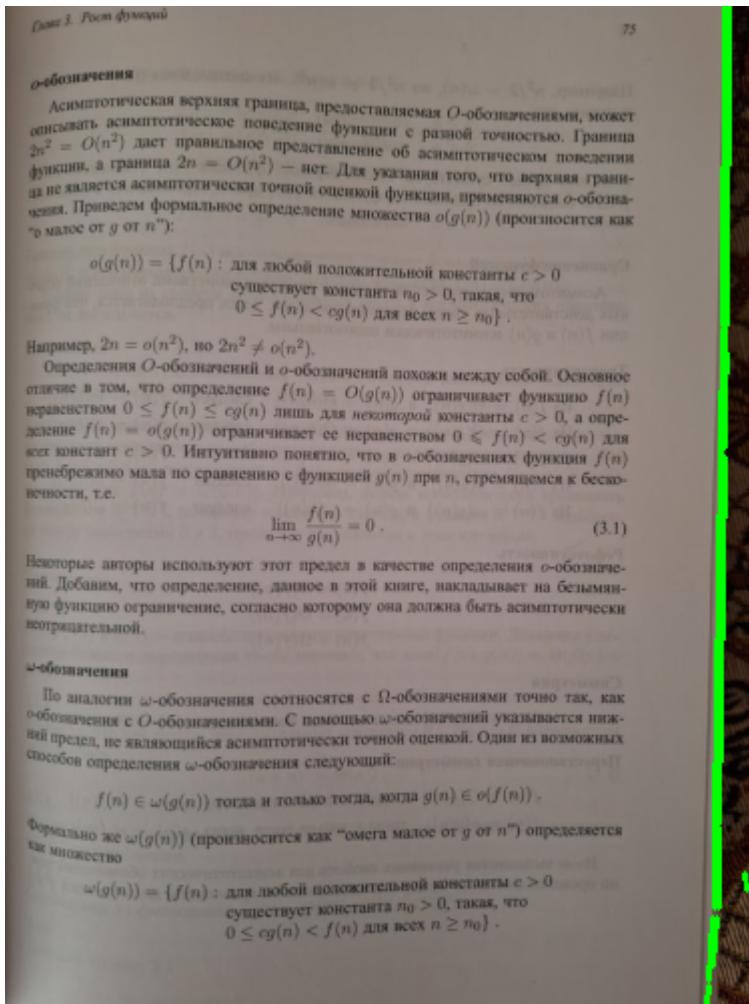


Рисунок 15 – Пример неправильно распознанного изображения: найденные контуры

2.3 Сегментация на абзацы

Сегментация — процесс разделения входного изображения на отдельные области, называемые сегментами, представляющие собой уникальную часть изображения. Цель сегментации заключается в улучшении понимании и интерпретации изображения, а также в анализе и классификации объектов на изображении. Примером сегментации может служить распознавание машин с видеокамеры дорожного наблюдения, обнаружение дорожных знаков с видеорегистратора и тд.

В данной работе сегментация служит для:

- Выделения абзацев;
- Отделения текста от формул, рисунков и пр.

Цели являются независимыми друг от друга, поэтому их можно выделить на отдельные этапы сегментации, а также выполнять параллельно.

Выделение абзацев требуется для дальнейшего сохранения структуры текста в L^AT_EX коде. Данный вид сегментации, как и коррекция перспективы, осуществляется при помощи алгоритмов обработки изображения. Поэтому данный этап можно также выполнять на ЭВМ пользователя.

Требуется понять, что такое абзац в контексте изображения. Будем считать абзацем последовательность строк текста, находящихся друг от друга на небольшом расстоянии.

2.3.1 Сегментация изображения

Рассмотрим алгоритм сегментации изображения на абзацы на примере изображения, представленного на рисунке 16

12.2 (a) Permittee shall not move, use, destroy, cut or remove or permit to be moved, used, destroyed, cut or removed any timber, cactus, native plants, standing trees or products of the land except that which is necessary for the use of the Subject Land, and then only with the prior written approval of Permittee. For undeveloped land, the Permittee must submit a plant survey prior to the removal of any native plant. If the removal or destruction of plants protected under the Arizona Native Plant Law (A.R.S. § 3-901 *et seq.*, or any successor statutes) is necessary to the use of the Subject Land, Permittee shall also obtain the prior written approval of the Arizona Department of Agriculture. In the event the Permittee removes the native plants, the Permittee must pay a vegetation fee to the Permittee and this fee is not a reimbursable improvement.

(b) Permittee is responsible for treatment of all regulated and restricted noxious weeds listed by the Arizona Department of Agriculture.

ARTICLE 13 PERMITTEE SHALL PROTECT AND RESTORE SUBJECT LAND

13.1 In the event of known trespass on the Subject Land resulting in damage thereto, Permittee shall notify Permittee and appropriate law enforcement authorities.

13.2 Upon abandonment, cancellation, revocation or termination of this Permit, Subject Land shall be restored to its original condition, to the satisfaction of the Permittee. Such restoration shall include, but shall not be limited to, removal of any and all material, equipment, facilities, temporary structures, or debris, deposited by Permittee on Subject Land. If Permittee fails to remove all such material, equipment, facilities, temporary structures, or debris within a reasonable period, as determined by the Permittee, they shall be forfeited and become the property of the State, but Permittee shall remain liable for the cost of removal of all materials and for restoration of the site.

ARTICLE 14 MISCELLANEOUS

14.1 It is understood by Permittee that the establishment of any water right, or rights, shall be by and for the State of Arizona, and no claim thereto shall be made by said Permittee; such rights shall attach to and become appurtenant to the Subject Land.

14.2 This Permit is granted subject to all the provisions and requirements thereto, and to the present laws relating to State Lands, and all amendments, revisions or repeals of all existing laws, the same as though they were fully set forth herein. No provisions of this Permit shall create any vested right in Permittee.

14.3 In the event of a dispute between the parties to this Permit, it is agreed to use arbitration to resolve the dispute but only to the extent required by A.R.S. § 12-1518; and in no event shall arbitration be employed to resolve a dispute which is otherwise subject to administrative review by the Department.

23 – Communication Permit 05/05 (Rev. 8/2010)

-12-

Рисунок 16 – Входное изображение алгоритма

Сначала необходимо бинаризовать изображение. Для этого используется размытие Гаусса [9], представленное на рисунке 17, а затем преобразование

Трешхолда [14], представленное на рисунке 18

12.2 (a) Permittee shall not move, use, destroy, cut or remove or permit to be moved, used, destroyed, cut or removed any timber, cactus, native plants, standing trees or products of the land except that which is necessary for the use of the Subject Land, and then only with the prior written approval of Permittee. For undeveloped land, the Permittee must submit a plant survey prior to the removal of any native plants. If the removal or destruction of plants protected under the Arizona Native Plant Law (A.R.S. § 3-395 et seq., or any successor statute) is necessary to the use of the Subject Land, Permittee shall also obtain the prior written approval of the Arizona Department of Agriculture. In the event the Permittee removes the native plants, the Permittee must pay a vegetation fee to the Permittee and this fee is not a reimbursable improvement.

(b) Permittee is responsible for treatment of all regulated and restricted noxious weeds listed by the Arizona Department of Agriculture.

**ARTICLE 13
PERMITTEE SHALL PROTECT AND RESTORE SUBJECT LAND**

13.1 In the event of known trespass on the Subject Land resulting in damage thereto, Permittee shall notify Permittee and appropriate law enforcement authorities.

13.2 Upon abandonment, cancellation, revocation or termination of this Permit, Subject Land shall be restored to its original condition, to the satisfaction of the Permittee. Such restoration shall include, but shall not be limited to, removal of any and all material, equipment, facilities, temporary structures, or debris, deposited by Permittee on Subject Land. If Permittee fails to remove all such material, equipment, facilities, temporary structures, or debris within a reasonable period, as determined by the Permittee, they shall be forfeited and become the property of the State, but Permittee shall remain liable for the cost of removal of all materials and for restoration of the site.

**ARTICLE 14
MISCELLANEOUS**

14.1 It is understood by Permittee that the establishment of any water right, or rights, shall be by and for the State of Arizona, and no claim thereto shall be made by said Permittee; such rights shall attach to and become apportioned to the Subject Land.

14.2 This Permit is granted subject to all the provisions and requirements thereto, and to the present laws relating to State Lands, and all amendments, revisions or repeals of all existing laws, the same as though they were fully set forth herein. No provisions of this Permit shall create any vested right in Permittee.

14.3 In the event of a dispute between the parties to this Permit, it is agreed to use arbitration to resolve the dispute but only to the extent required by A.R.S. § 12-1518; and in no event shall arbitration be employed to resolve a dispute which is otherwise subject to administrative review by the Department.

Рисунок 17 – Изображение после размытия Гаусса

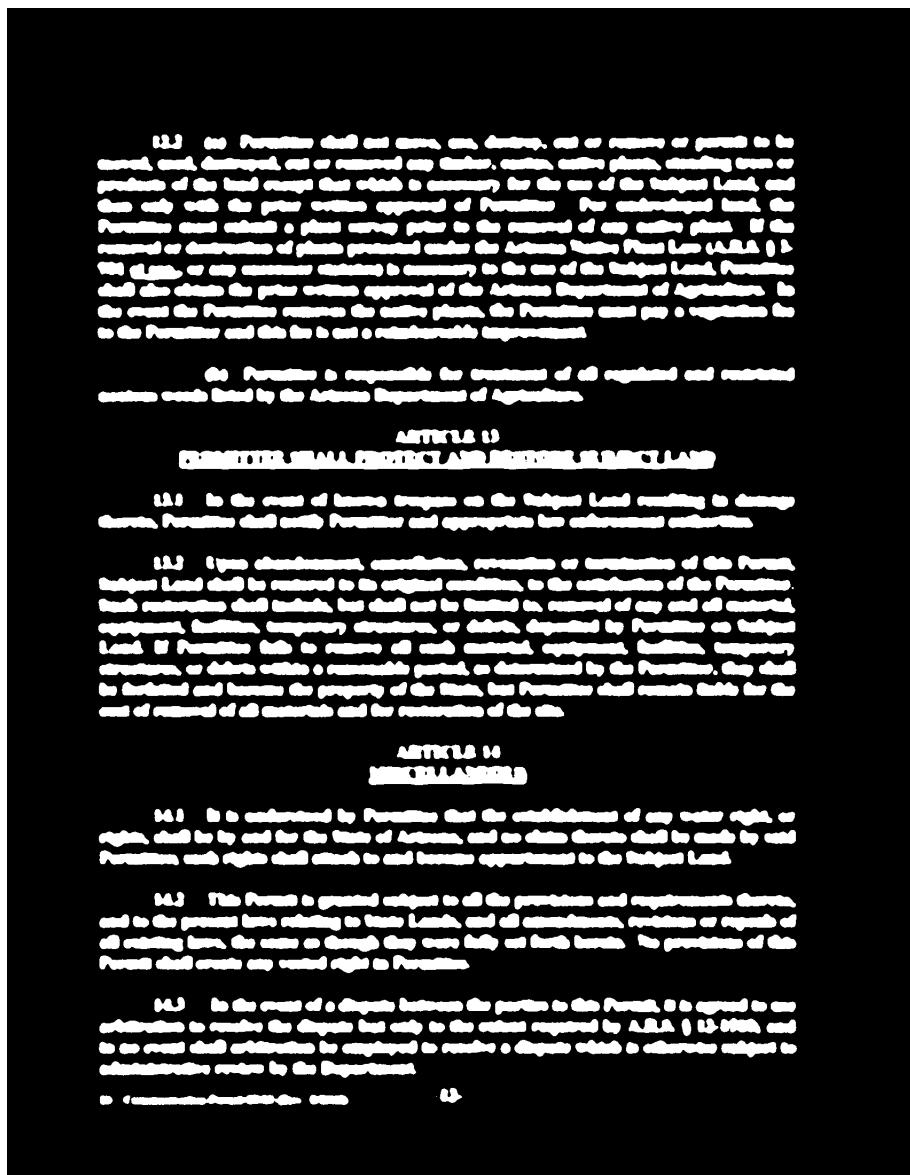


Рисунок 18 – Бинаризованное изображение

Теперь необходимо найти и скорректировать все абзацы. Для этого найдем все контуры изображения с помощью метода из библиотеки *openCV* [12] и находящиеся рядом контуры объединим. В результате получим абзацы, границы которых изображены на рисунке 19

<p>12.2 (a) Permittee shall not move, use, destroy, cut or remove or permit to be moved, used, destroyed, cut or removed any timber, cactus, native plants, standing trees or products of the land except that which is necessary for the use of the Subject Land, and then only with the prior written approval of Permittor. For undeveloped land, the Permittee must submit a plan survey prior to the removal of any native plant. If the removal or destruction of plants protected under the Arizona Native Plant Law (A.R.S. § 3-901 <i>et seq.</i>, or any successor statutes) is necessary to the use of the Subject Land, Permittee shall also obtain the prior written approval of the Arizona Department of Agriculture. In the event the Permittee removes the native plants, the Permittee must pay a vegetation fee to the Permittor and this fee is not a reimbursable improvement.</p> <p>(b) Permittee is responsible for treatment of all regulated and restricted noxious weeds listed by the Arizona Department of Agriculture.</p>
<p style="text-align: center;">ARTICLE 13 PERMITTEE SHALL PROTECT AND RESTORE SUBJECT LAND</p>
<p>13.1 In the event of known trespass on the Subject Land resulting in damage thereto, Permittee shall notify Permittor and appropriate law enforcement authorities.</p>
<p>13.2 Upon abandonment, cancellation, revocation or termination of this Permit, Subject Land shall be restored to its original condition, to the satisfaction of the Permittor. Such restoration shall include, but shall not be limited to, removal of any and all material, equipment, facilities, temporary structures, or debris, deposited by Permittee on Subject Land. If Permittee fails to remove all such material, equipment, facilities, temporary structures, or debris within a reasonable period, as determined by the Permittor, they shall be forfeited and become the property of the State, but Permittee shall remain liable for the cost of removal of all materials and for restoration of the site.</p>
<p style="text-align: center;">ARTICLE 14 MISCELLANEOUS</p>
<p>14.1 It is understood by Permittee that the establishment of any water right, or rights, shall be by and for the State of Arizona, and no claim thereto shall be made by said Permittee; such rights shall attach to and become appurtenant to the Subject Land.</p>
<p>14.2 This Permit is granted subject to all the provisions and requirements thereto, and to the present laws relating to State Lands, and all amendments, revisions or repeals of all existing laws, the same as though they were fully set forth herein. No provisions of this Permit shall create any vested right in Permittee.</p>
<p>14.3 In the event of a dispute between the parties to this Permit, it is agreed to use arbitration to resolve the dispute but only to the extent required by A.R.S. § 12-1518; and in no event shall arbitration be employed to resolve a dispute which is otherwise subject to administrative review by the Department.</p>

23 – Communication Permit 05/05 (Rev. 8/2010) -12-

Рисунок 19 – Найденные абзацы

2.3.2 Сегментация pdf файла

В настоящее время файл формата *pdf* широко распространен, и важно уметь обрабатывать данный формат, так как многие публикации и статьи хранятся именно в этом формате.

Существует несколько типов *pdf* файлов, такие как:

- **Программно генерируемые pdf файлы:** эти *pdf* создаются программно, например, предмет данной работы, *LATEX* можно преобразовать в *pdf* файл. Данный тип файла может содержать различные компоненты, например, изображения, текст, ссылки;
- **Традиционные отсканированные документы:** такие *pdf* файлы создаются из неэлектронных носителей, например, при

сканировании. В данном случае *pdf* выступает в роли контейнера для изображений. Поэтому с элементами на изображениях нельзя никак взаимодействовать.

В случае с отсканированными документами необходимо поступать как с изображением по описанному выше алгоритму. Однако, в случае с программно генерируемым файлом мы можем извлекать из него необходимую информацию, такую как: абзацы, изображения, таблицы и тд. На данном этапе нас интересует извлечение абзацев. Для задачи парсинга *pdf* файла использовалась библиотека *PyMuPDF* [15]. С помощью этой библиотеки можно легко извлекать абзацы текста. На рисунке 20 представлен пример одной из страниц *pdf* файла. На рисунке 21 представлены результаты сегментации на абзацы средствами библиотеки.

Лабораторная работа №1 по курсу дискретного анализа: сортировка за линейное время

Выполнил студент группы 08-208 МАИ Ядрев Артем.

Условие

1. Требуется разработать программу, осуществляющую ввод пар «ключ-значение», их упорядочивание по возрастанию ключа указанным алгоритмом сортировки за линейное время и вывод отсортированной последовательности.
2. Вариант задания определяется типом ключа (и соответствующим ему методом сортировки) и типом значения:
 - (a) Поразрядная сортировка.
 - (b) Тип ключа: телефонные номера, с кодами стран и городов в формате +<код страны> <код города> телефон.
 - (c) Тип значения: строки фиксированной длины 64 символа, во входных данных могут встречаться строки меньшей длины, при этом строка дополняется до 64-х нулевыми символами, которые не выводятся на экран.

Метод решения

Алгоритм поразрядной сортировки достаточно прост. Достаточно пройтись по всем разрядам справа налево (конечно, интуиция подсказывает, что начать лучше со старшего разряда, но в третьем издании книги Кормена "Алгоритмы. Построение и анализ" в главе 8.3 "Поразрядная сортировка" отлично показано, что начинать лучше с крайнего правого разряда) и применить устойчивую сортировку. Лично я использовал сортировку подсчетом, так как мы имеем ограниченное количество элементов (10 цифр).

Описание программы

Весь код содержится в файле **main.cpp**: класс **TObject**, в котором хранится ключ, разделенный на разряды (код страны, код города и непосредственно сам номер телефона), и значение. С целью экономии памяти я использовал **char[]** вместо **std::string**.

Дневник отладки

С самого начала я использовал **Make** для отправки своего решения. Однако при отладке непосредственно у себя на устройстве я использовал ключ **-fsanitize=address**, который потом на чекере использовал много памяти, из-за чего я превышал заданную память. Также была попытка использовать **std::string**, что достаточно сильно влияло на расход

Рисунок 20 – Пример страницы pdf

Лабораторная работа №1 по курсу дискретного анализа: сортировка за линейное время

Выполнил студент группы 08-208 МАИ Ядрев Артем.

Условие

1. Требуется разработать программу, осуществляющую ввод пар «ключ-значение», их упорядочивание по возрастанию ключа, указанным алгоритмом сортировки за линейное время и вывод отсортированной последовательности.
2. Вариант задания определяется типом ключа (и соответствующим ему методом сортировки) и типом значения:
 - (a) Поразрядная сортировка.
 - (b) Тип ключа: телефонные номера, с кодами стран и городов в формате +<код страны> <код города> телефон.
 - (c) Тип значения: строки фиксированной длины 64 символа, во входных данных могут встречаться строки меньшей длины, при этом строка дополняется до 64-х нулевыми символами, которые не выводятся на экран.

Метод решения

Алгоритм поразрядной сортировки достаточно прост. Достаточно пройтись по всем разрядам справа налево (конечно, интуиция подсказывает, что начинать лучше со старшего разряда, но в третьем издании книги Кормена "Алгоритмы. Построение и анализ" в главе 8.3 "Поразрядная сортировка" отлично показано, что начинать лучше с крайнего правого разряда) и применить устойчивую сортировку. Лично я использовал сортировку подсчетом, так как мы имеем ограниченное количество элементов (10 цифр).

Описание программы

Весь код содержится в файле `main.cpp`: класс `TObject`, в котором хранится ключ, разделенный на разряды (код страны, код города и непосредственно сам номер телефона), и значение. С целью экономии памяти я использовал `char[]` вместо `std::string`.

Дневник отладки

С самого начала я использовал Make для отправки своего решения. Однако при отладке непосредственно у себя на устройстве я использовал ключ `-fsanitize=address`, который потом на чекере использовал много памяти, из-за чего я превышал заданную память. Также была попытка использовать `std::string`, что достаточно сильно влияло на расход



Рисунок 21 – Пример страницы pdf с размеченными абзацами

Мы можем наблюдать четко выделенные абзацы. Для сравнения к этому же изображению применим алгоритм сегментации изображения, описанный выше. На рисунке 22 показан результат применения этого алгоритма к изображению на рисунке 20

Лабораторная работа №1 по курсу дискретного анализа: сортировка за линейное время

Выполнил студент группы 08-208 МАИ Ядроб Артем.

Условие

1. Требуется разработать программу, осуществляющую ввод пар «ключ-значение», их упорядочивание по возрастанию ключа указанным алгоритмом сортировки за линейное время и вывод отсортированной последовательности.

2. Вариант задания определяется типом ключа (и соответствующим ему методом сортировки) и типом значения:

(a) Поразрядная сортировка.

(b) Тип ключа: телефонные номера, с кодами стран и городов в формате +<код страны> <код города> телефон.

(c) Тип значения: строки фиксированной длины 64 символа, во входных данных могут встретиться строки меньшей длины, при этом строка дополняется до 64-х нулевыми символами, которые не выводятся на экран.

Метод решения

Алгоритм поразрядной сортировки достаточно прост. Достаточно пройтись по всем разрядам справа налево (конечно, интуиция подсказывает, что начать лучше со старшего разряда, но в третьем издании книги Кормена "Алгоритмы. Построение и анализ" в главе 8.3 "Поразрядная сортировка" отлично показано, что начинать лучше с крайнего правого разряда) и применить устойчивую сортировку. Лично я использовал сортировку подсчетом, так как мы имеем ограниченное количество элементов (10 цифр).

Описание программы

Весь код содержится в файле `main.cpp`: класс `TObject`, в котором хранится ключ, разделенный на разряды (код страны, код города и непосредственно сам номер телефона), и значение. С целью экономии памяти я использовал `char[]` вместо `std::string`.

Дневник отладки

С самого начала я использовал Mako для отправки своего решения. Однако при отладке непосредственно у себя на устройстве я использовал ключ `-fsanitize=address`, который потом на чекере использовал много памяти, из-за чего я превышал заданную память. Также была попытка использовать `std::string`, что достаточно сильно влияло на расход

1

Рисунок 22 – Пример страницы pdf с размеченными абзацами алгоритмом сегментации изображения

2.4 Выделение формул

2.4.1 Модель

Для выделения формул недостаточно одних алгоритмов обработки изображений. Существует множество моделей, выполняющих распознавание объектов на изображении. Одной из таких моделей является *YOLO* [16].

Данная модель обладает следующими преимуществами:

- Быстродействие. Нейросеть работает в реальном времени, поэтому её используют для распознавания объектов на фото и видео ”здесь и сейчас”;
- Точность. Нейросеть *YOLO* умеет распознавать объекты разных размеров в пределах одного кадра;
- Универсальность. Нейросеть *YOLO* способна определять как хорошо знакомые ей объекты, так и те, с которыми она ещё не сталкивалась;
- Простота. Модель *YOLO* можно запросто запускать и дообучать с помощью *Tensorflow* [17].

Однако, данная модель не специализирована на какой-то одной задаче, поэтому будет проигрывать специализированным моделям.

На основании плюсов данной модели, а также на основании самой архитектуры системы, позволяющую при необходимости легко заменить выбранную модель на другую, было принято решение использовать для распознавания формул модель *YOLO* последней версии *v8*.

2.4.2 Данные для обучения

В качестве датасета для обучения был взят датасет *ICDAR – 2021* (*International Conference on Document Analysis and Recognition*). Он содержит набор изображений текстовых документов, содержащих формулы, а также для каждого изображения имеется набор координат ограничивающих рамок (”*bounding boxes*”) для каждой формулы. Размер датасета составляет 5171 изображений. На рисунках 23 и 24 показаны примеры входного изображения и изображения с метками формул соответственно.

- Three-Family Condition: To have three families we must require

$$6 = \lambda\eta(\eta - nc_1). \quad (2.9)$$

The second such rule is associated with the anomaly cancellation requirement that

$$[W_Z] = c_2(TZ) - c_2(V_{Z1}) - c_2(V_{Z2}), \quad (2.10)$$

where $[W_Z]$ is the class associated with non-perturbative five-branes in the bulk space of the Hořava-Witten theory. Vector bundles V_{Z1} and V_{Z2} are located on the “observable” and “hidden” orbifold planes respectively. In this talk, for simplicity, we will always take V_{Z2} to be the trivial bundle. Hence, gauge group E_8 remains unbroken on the “hidden” sector, $c_2(V_{Z2})$ vanishes and condition (2.10) simplifies accordingly. Using the definition

$$[W_Z] = \frac{1}{2}q_*[W], \quad (2.11)$$

condition (2.10) can be pulled-back onto X to give

$$[W] = c_2(TX) - c_2(V). \quad (2.12)$$

It follows that

$$[W] = \sigma_*W_B + c(F - N) + dN \quad (2.13)$$

where

$$W_B = 12c_1 - \eta \quad (2.14)$$

and

$$c = c_2 + \left(\frac{1}{24}(n^3 - n) + 11 \right) c_1^2 - \frac{1}{2} \left(\lambda^2 - \frac{1}{4} \right) n\eta(\eta - nc_1) - \sum_i \kappa_i^2, \quad (2.15)$$

$$d = c_2 + \left(\frac{1}{24}(n^3 - n) - 1 \right) c_1^2 - \frac{1}{2} \left(\lambda^2 - \frac{1}{4} \right) n\eta(\eta - nc_1) - \sum_i \kappa_i^2 + \sum_i \kappa_i. \quad (2.16)$$

The class $[W_Z]$ must represent an actual physical holomorphic curve in the Calabi-Yau threefold Z since physical five-branes are required to wrap around it. Hence, $[W_Z]$ must be an effective class and, hence, its pull-back $[W]$ is an effective class in the covering threefold X . This leads to the following rule.

- Effectiveness Condition: For $[W]$ to be an effective class, we require

$$W_B \text{ is effective in } B, \quad c \geq 0, \quad d \geq 0. \quad (2.17)$$

Рисунок 23 – Пример входного изображения

- Three-Family Condition: To have three families we must require

$$\delta = \lambda\eta(\eta - nc_1). \quad (2.9)$$

The second such rule is associated with the anomaly cancellation requirement that

$$[W_Z] = c_2(TZ) - c_2(V_{Z1}) - c_2(V_{Z2}), \quad (2.10)$$

where $[W_Z]$ is the class associated with non-perturbative five-branes in the bulk space of the Hořava-Witten theory. Vector bundles V_{Z1} and V_{Z2} are located on the “observable” and “hidden” orbifold planes respectively. In this talk, for simplicity, we will always take V_{Z2} to be the trivial bundle. Hence, gauge group \mathbb{Z}_2 remains unbroken on the “hidden” sector. $c_2(V_{Z2})$ vanishes and condition (2.10) simplifies accordingly. Using the definition

$$[W_Z] = \frac{1}{2}q_*[W], \quad (2.11)$$

condition (2.10) can be pulled-back onto X to give

$$[W] = c_2(TX) - c_2(V). \quad (2.12)$$

It follows that

$$[W] = \sigma_*W_B + c(F - N) + dN \quad (2.13)$$

where

$$W_B = 12c_1 - \eta \quad (2.14)$$

and

$$c = c_2 + \left(\frac{1}{24}(n^3 - n) + 11 \right) c_1^2 - \frac{1}{2} \left(\lambda^2 - \frac{1}{4} \right) n\eta(\eta - nc_1) - \sum_i \kappa_i^2, \quad (2.15)$$

$$d = c_2 + \left(\frac{1}{24}(n^3 - n) - 1 \right) c_1^2 - \frac{1}{2} \left(\lambda^2 - \frac{1}{4} \right) n\eta(\eta - nc_1) - \sum_i \kappa_i^2 + \sum_i \kappa_i. \quad (2.16)$$

The class $[W_Z]$ must represent an actual physical holomorphic curve in the Calabi-Yau threefold since physical five-branes are required to wrap around it. Hence, $[W_Z]$ must be an effective class and, hence, its pull-back $[W]$ is an effective class in the covering threefold X . This leads to the following rule.

- Effectiveness Condition: For $[W]$ to be an effective class, we require

$$W_B \text{ is effective in } B, \quad c \geq 0, \quad d \geq 0. \quad (2.17)$$

Рисунок 24 – Пример изображения с метками формул

2.4.3 Обучение

Обучение проводилось на платформе *Kaggle* [18] с логированием в системе *wandb* [7]. В качестве графического процессора использовались две видеокарты *Tesla T4*. После обучения модели мы переходим к оценки ее эффективности. Посмотрим на полученные с помощью *wandb* графики описанных выше метрик.



Рисунок 25 – Графики потерь

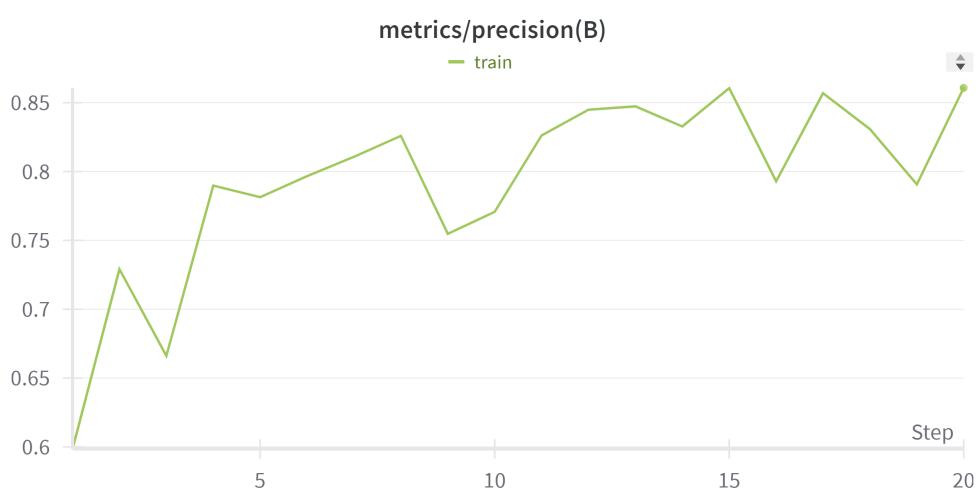


Рисунок 26 – График точности



Рисунок 27 – Графики mAP

На каждом из графиков 25 ... 27 по оси oX отложен номер шага, на котором снимались показатели. Весь этап обучения, состоящий из N эпох равномерно делится на 20 этапов. На каждом этапе снимаются метрики, попадающие в результирующий график.

После обучения модель была протестирована на тестовых изображениях. На рисунках 28 и 29 показаны результаты тестирования.

[Formula: 0.226](#) [Formula: 0.444](#)
 usual approach [1,3] when the factor i enters the boundary conditions for real fields and thus makes the boundary value problem ill-defined. The fact that parity odd fields can get an imaginary factor in the Euclidean space has been observed long ago [10] in the context of chiral theories. It is easy to see that the field [Formula: 0.840](#) A_μ is of odd parity from the [Formula: 0.843](#) point of view. Indeed, the last term in (1) can be [Formula: 0.864](#) $\int_M \partial_a(\varepsilon^{ab} A_\mu \partial_b X^\mu)$. Here A_μ couples to the parity-odd quantity ε^{ab} . We are not able to compare our results to those of previous related papers [1,3] where this mathematical subtlety was just ignored, they are compatible after the replacement $A \rightarrow iA$. [Formula: 0.822](#) [Formula: 0.771355](#)

Let us expand the action (1) around an arbitrary background X , $X = X + \xi$. To calculate the one-loop (in the field theory sense) effective action we need only the part which is quadratic in ξ . [Formula: 0.375](#)

$$S_2 = \frac{1}{2} \left[\frac{1}{2} \int d^2 z \sqrt{h} h^{ab} \partial_a \xi_\mu \partial_b \xi^\mu + \right. \\ \left. + \frac{1}{2} \int_{\partial M} \left(F_{\nu\mu} \xi^\nu \dot{\xi}^\mu + \dot{X}^\mu \partial_\nu F_{\rho\mu} \xi^\nu \xi^\rho \right) \right] \quad (2)$$

Clearly the one-loop effective action is [Formula: 0.890](#)

$$W = \frac{1}{2} \log \det(-\Delta \delta_{\mu\nu}), \quad (3)$$

with the scalar Laplacian [Formula: 0.884](#) with the boundary condition [Formula: 0.884](#)

$$(-\partial_N \delta_{\mu\nu} + F_{\mu\nu} \partial_\tau + \dot{X}^\rho (\partial_\nu F_{\mu\rho})) \xi^\nu|_{\partial M} = 0, \quad (4)$$

where $\partial_N = \frac{1}{2}(\partial_\tau \Gamma + \Gamma \partial_\tau)$ is the derivative with respect to the inward pointing unit normal vector N . It is useful to rewrite (4) by adding and subtracting a term with $\mu \leftrightarrow \nu$ as [Formula: 0.808](#)

$$\mathcal{B}\xi|_{\partial M} = \left(\partial_N + \frac{1}{2}(\partial_\tau \Gamma + \Gamma \partial_\tau) + S \right) \xi|_{\partial M} = 0, \quad (5)$$

where [Formula: 0.911](#)

$$\Gamma_{\mu\nu} = -F_{\mu\nu}, \quad S_{\mu\nu} = -\frac{1}{2} \dot{X}^\rho (\partial_\nu F_{\mu\rho} + \partial_\mu F_{\nu\rho}). \quad (6)$$

Thus the boundary condition (5) also ensures the hermiticity of the Laplace operator. Indeed, [Formula: 0.887](#)

$$\int_M (\xi'_\mu \Delta \xi^\mu - \xi_\mu \Delta \xi'^\mu) = \int_{\partial M} (-\xi'_\mu \partial_N \xi^\mu + \xi_\mu \partial_N \xi'^\mu) = 0, \quad (7)$$

if ξ and ξ' satisfy (5), when the hermiticity of $\frac{1}{2}(\partial_\tau \Gamma + \Gamma \partial_\tau) + S$ is used. The boundary condition (5) contains tangential derivatives and belongs to the Gilkey-Smith class¹. [Formula: 0.306](#)

¹Sometimes the boundary conditions with tangential derivatives are called “mixed” [11]. In the theory of the heat equation asymptotics the name “mixed” is reserved for a completely different type of the boundary conditions [12].

Рисунок 28 – Предсказание модели на тестовом изображении

this dual situation, the physics is described by the dual of the condensed moose in figure 2. The condensates in this case are

$$\langle \psi_{i,i+1} \chi_{i+1,i+1} \rangle \sim 4\pi f^3 V_{i,i+1} \quad i = 1, \dots, N. \quad (5.1)$$

Again the physics is five dimensional for $a \ll r \ll R$, but it is a different fifth dimension, dynamically generated by a different set of interactions and with a different set of gauge bosons.

The transition from (2.3) to (5.1) is theoretically fascinating, but somewhat puzzling, and we will not discuss it in detail here. But it is important to understand the approach to the transition because it bears on the possibly phenomenologically relevant question of how large the five dimensional coupling can be. For example the heaviest of the KK modes has a mass of order $g(\Lambda_s) f_s$ parametrically lighter than the scale where G_s gets strong, Λ_s . How similar can we make these scales? Can we increase g to the region of strong coupling as well?

For simplicity, let us take Λ to infinity so that the physics appears five dimensional at arbitrarily long distances. What happens as we change the ratio of Λ to Λ_s ? For $\Lambda \ll \Lambda_s$ where the analysis of section 2 applies, the residual gauge interactions at distances large compared to a are very weak. In the four dimensional language, this is obvious because the gauge coupling $g_5^2 = g^2 a$, and its effects at distances of order a are suppressed by powers of g_5^2/a . In the four dimensional language, one might worry that there is something wrong with this argument at distances large compared to a . This worry is groundless. The weak gauge group breaks the condensate (2.3) down to a residual gauge group with coupling of order g^2/N and the infrared weak $\Lambda \rightarrow \infty$.

What happens as we increase Λ/Λ_s ? The gauge coupling $g_5^2 = g^2 a$ but its effects remain tiny at large distances. We know that at some point as $\Lambda \rightarrow \Lambda_s$ an ecological disaster will occur, dramatically changing the nature of the long distance physics. It is reasonable to suppose that the cataclysm will happen abruptly at some point $\Lambda \approx \Lambda_s$ where both g_5^2 and $g^2 a$ get large. The only signal at large distances of impending doom is that as $\Lambda \rightarrow \Lambda_s$, $g_5^2 = g^2 a$ gets large compared to a . This signals the imminent breakdown of the effective theory because dimensional couplings in an effective theory must not be large compared to the appropriate power of the cut-off. Even though the tree level interactions are still weak at long distances, the theory is losing control of its quantum corrections, a warning that anarchy is about to be loosed upon the world.

6 Other completions

The fifth dimension has appeared in the condensed moose because the non-linear sigma model fields allow the gauge field to “hop” from one site to the next. Since we could have obtained this directly as a latticization of the five-dimensional gauge theory, we might ask why we need the original moose model at all. The reason is that latticization in the fifth dimension does not cut-off divergences from large four-momenta: the four-dimensional non-linear sigma model of (2.5) is non-renormalizable, becoming strongly coupled at a scale $\sim 4\pi f_s$. That is, this theory requires a UV completion. But this is familiar problem, with familiar solution. The moose model we have constructed provides a UV completion in the same way that QCD

Рисунок 29 – Предсказание модели на тестовом изображении

2.4.4 Корректировка результата

Как видно на рисунке 28, некоторые найденные формулы пересекаются

друг с другом или находятся одна внутри другой. Для устранения этого дефекта используется алгоритм *NMS (Non-Maximum Suppression)* [19], который реализован в *PyTorch*. Так, например, на рисунке 30 показан результат применения алгоритма к изображению, представленному на рисунке 28

usual approach [1, 3] when the factor i enters the boundary conditions for real fields and thus makes the boundary value problem ill-defined. The fact that parity odd fields can get an imaginary factor in the Euclidean space has been observed long ago [10] in the context of chiral theories. It is easy to see that the field A_μ is of odd parity from the point of view [10]. Indeed, the last term in (1) can be written as $\int_M \partial_a(\varepsilon^{ab} A_\mu \partial_b X^\mu)$. Here A_μ couples to the parity-odd quantity ε^{ab} . We are able to compare our results to those of previous related papers [1, 3] where this mathematical subtlety was just ignored, they are compatible after the replacement $A \rightarrow iA$.
Formula: 0.771355

Let us expand the action (1) around an arbitrary background X , $X = X + \xi$. To calculate the one-loop (in the field theory sense) effective action we need only the part which is quadratic in ξ :
Formula: 0.699

$$\begin{aligned} S_2 &= \frac{1}{2\pi\alpha'} \left[\frac{1}{2} \int_M d^2z \sqrt{h} h^{ab} \partial_a \xi_\mu \partial_b \xi^\mu + \right. \\ &\quad \left. + \frac{1}{2} \int_{\partial M} \left(F_{\nu\mu} \xi^\nu \dot{\xi}^\mu + \dot{X}^\mu \partial_\nu F_{\mu\rho} \xi^\nu \xi^\rho \right) \right] \end{aligned} \quad (2)$$

Clearly the one-loop effective action is:
Formula: 0.890

$$W = \frac{1}{2} \log \det(-\Delta \delta_{\mu\nu}), \quad (3)$$

with the scalar Laplacian Δ and with the boundary condition
Formula: 0.884

$$(-\partial_N \delta_{\mu\nu} + F_{\mu\nu} \partial_\tau + \dot{X}^\rho (\partial_\nu F_{\mu\rho})) \xi^\nu|_{\partial M} = 0, \quad (4)$$

where $\partial_N = N^\mu \partial_\mu$ is the derivative with respect to the inward pointing unit normal vector N^μ . It is useful to rewrite (4) by adding and subtracting a term with $\mu \leftrightarrow \nu$ as
Formula: 0.885

$$\mathcal{B}\xi|_{\partial M} = \left(\partial_N + \frac{1}{2} (\partial_\tau \Gamma + \Gamma \partial_\tau) + S \right) \xi|_{\partial M} = 0, \quad (5)$$

where
Formula: 0.911

$$\Gamma_{\mu\nu} = -F_{\mu\nu}, \quad S_{\mu\nu} = -\frac{1}{2} \dot{X}^\rho (\partial_\nu F_{\mu\rho} + \partial_\mu F_{\nu\rho}). \quad (6)$$

Thus the boundary condition (5) also ensures the hermiticity of the Laplace operator. Indeed
Formula: 0.887

$$\int_M (\xi'_\mu \Delta \xi^\mu - \xi_\mu \Delta \xi'^\mu) = \int_{\partial M} (-\xi'_\mu \partial_N \xi^\mu + \xi_\mu \partial_N \xi'^\mu) = 0, \quad (7)$$

if ξ and ξ' satisfy (5), when the hermiticity of $\frac{1}{2}(\partial_\tau \Gamma + \Gamma \partial_\tau) + S$ is used. The boundary condition (5) contains tangential derivatives and belongs to the Gilkey-Smith class¹.

¹Sometimes the boundary conditions with tangential derivatives are called “mixed” [11]. In the theory of the heat equation asymptotics the name “mixed” is reserved for a completely different type of the boundary conditions [12].

Рисунок 30 – Скорректированное с помощью алгоритма
NMS предсказание модели

2.5 Генерация LATEX кода

После сегментации мы имеем представление о том, как расположены

абзацы в тексте, где расположены формулы и тд. Теперь мы можем приступить к генерации \LaTeX кода. Будем отделять текст и с помощью алгоритмов *OCR* (*optical character recognition*). В частности, в данной работе для распознавания текста используется *Python* библиотека *pytesseract* [20], а для преобразования изображения формул в \LaTeX код используется модель *LaTeX – OCR* [21].

Для правильной работы моделей необходимо предоставлять им корректные данные: для распознавания текста на изображении не должно быть формул и наоборот. Для этого необходимо на основании имеющихся координат абзацев и координат формул разбить абзацы на подабзацы так, чтобы каждый подабзац содержал в себе либо текст, либо формулу. На рисунке 31 приведен пример простого текста, содержащего математические формулы:

Вот небольшой пример текста с математическими формулами. Пусть x и y будут переменными, представляющими числа. Мы можем использовать эти переменные для описания различных математических выражений. Например, мы можем записать уравнение прямой в виде $y = mx + c$, где m - это наклон прямой, а c - это её точка пересечения с осью y .

Давайте рассмотрим ещё один пример. Пусть у нас есть квадратное уравнение $ax^2 + bx + c = 0$. Его корни могут быть найдены с помощью формулы дискриминанта:

$$D = b^2 - 4ac$$

Если $D > 0$, то у уравнения два действительных корня x_1 и x_2 , которые можно найти по формулам:

$$x_1 = \frac{-b + \sqrt{D}}{2a}, \quad x_2 = \frac{-b - \sqrt{D}}{2a}$$

Если $D = 0$, то у уравнения есть один действительный корень x , который находится по формуле:

$$x = -\frac{b}{2a}$$

Наконец, если $D < 0$, то у уравнения нет действительных корней, а только комплексные.

Рисунок 31 – Пример входного текста

После применения к данному тексту всех алгоритмов, он будет разбит на части, как показано на рисунке 32. Зеленым цветом выделен текст, красным -

формулы.

Вот небольшой пример текста с математическими формулами. Пусть x и y будут переменными, представляющими числа. Мы можем использовать эти переменные для описания различных математических выражений. Например, мы можем записать уравнение прямой в виде $y = mx + c$, где m - это наклон прямой, а c - это её точка пересечения с осью y .

Давайте рассмотрим ещё один пример. Пусть у нас есть квадратное уравнение $ax^2 + bx + c = 0$. Его корни могут быть найдены с помощью формулы дискриминанта:

$$D = b^2 - 4ac$$

Если $D > 0$, то уравнения два действительных корня x_1 и x_2 , которые можно найти по формулам:

$$x_1 = \frac{-b + \sqrt{D}}{2a}, \quad x_2 = \frac{-b - \sqrt{D}}{2a}$$

Если $D = 0$, то уравнения есть один действительный корень x , который находится по формуле:

$$x = -\frac{b}{2a}$$

Наконец, если $D < 0$, то уравнения нет действительных корней, а только комплексные.

Рисунок 32 – Разбиение текста, представленного на рисунке
31

Разбиение по строкам текста показано для наглядности. На деле каждое разбиение является прямоугольником, который может содержать в себе несколько строк. Для этого ненужная часть верхних строк закрашивается белым цветом и не участвует в дальнейшем распознавании текста.

Теперь мы можем преобразовать исходный текст в L^AT_EX код. Формулы, находящиеся в тексте, будем выделять символом '\$', а находящиеся в отдельных абзацах формулы будем выделять через [...] .

Для входного текста, представленного на рисунке 31, будем иметь следующий L^AT_EX код, показанный на рисунке 33

```

1 Вот небольшой пример текста с математическими формулами. Пусть
2 $x$ и $y$ будут переменными, представляющими числа. Мы можем
3 использовать эти переменные для описания различных
4 математических выражений. Например, мы можем записать уравнение
5 прямой в виде  $y = mx + c$ , где  $m$  – это наклон прямой, а  $c$  –
6 это её точка пересечения с осью  $y$ .
7
8
9 Давайте рассмотрим ещё один пример. Пусть у нас есть
10 квадратное уравнение  $ax^2 + bx + c = 0$ . Его корни могут быть
11 найдены с помощью формулы дискриминанта:
12
13
14
15 Если  $D > 0$ , то у уравнения два действительных корня  $x_1$  и  $x_2$ , которые можно найти по формулам:
16
17
18
19
20
21
22

```

Рисунок 33 – Листинг текста 31

В данном примере не хватает преамбулы. Для этого сформируем ее из популярных пакетов L^AT_EX. На рисунке 34 показан листинг преамбулы

```
1  \documentclass{article}
2  \usepackage[T2A]{fontenc} % Кодировка шрифта
3  \usepackage[utf8]{inputenc} % Кодировка ввода
4  \usepackage[english, russian]{babel} % Пакет для поддержки
      русского языка
5  \usepackage{amsmath} % Для использования математических формул
6
7  \begin{document}
8  ...
9  \end{document}
10
```

Рисунок 34 – Листинг преамбулы документа

3 Тестирование программного продукта

Разработав отдельные компоненты продукта, необходимо протестировать ее работоспособность на тестовом изображении.

3.1 Демонстрация программного продукта

Для создания *Graphical User Interface (GUI)* использовался язык *Python* и библиотека *tkinter* [22], а также *matplotlib* [23].

При первом входе в приложение будет произведена авторизация путем протокола *OAuth 2.0*. На рисунке 35 показано окно авторизации через облачное хранилище (в данном случае *Google*)

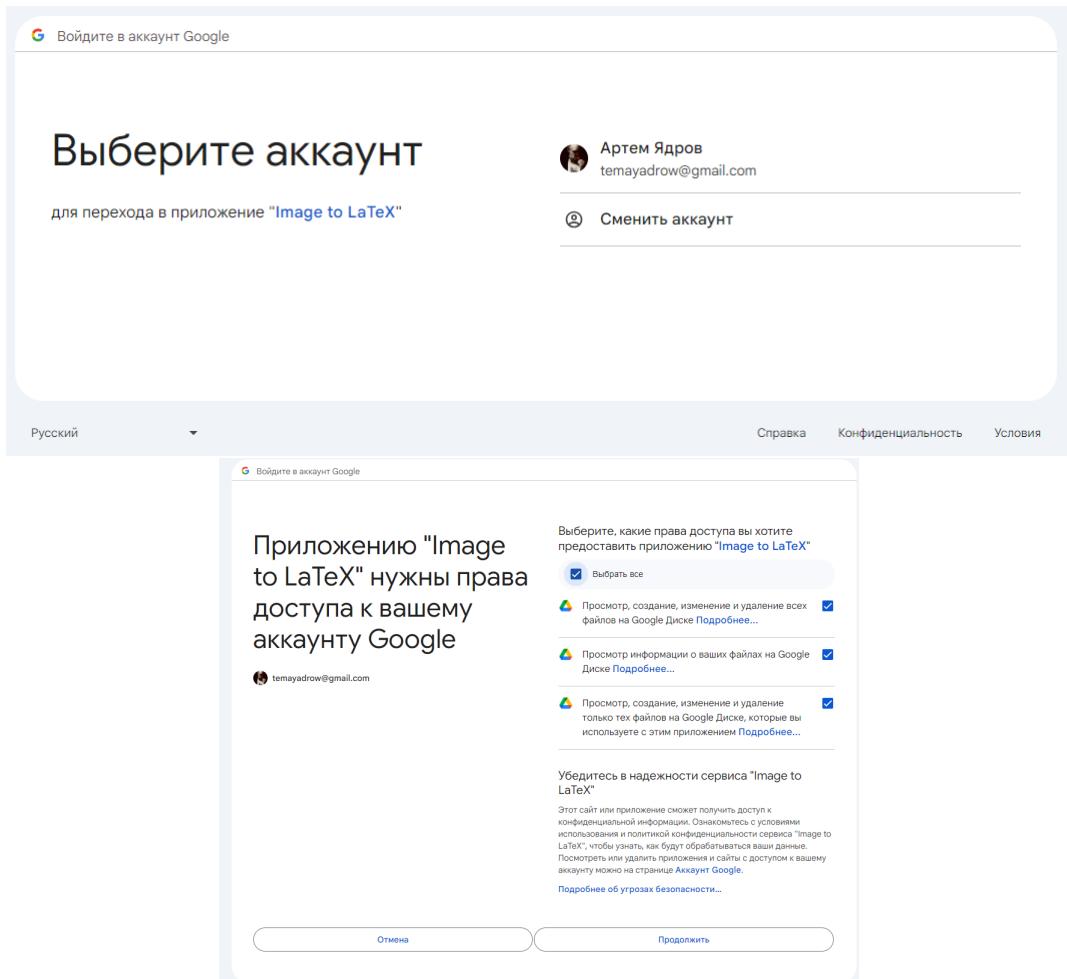


Рисунок 35 – Окно авторизации через облачное хранилище

После выбора изображения, согласно схеме, представленной на рисунке 4, начинается процесс коррекции перспективы. В случае неудачного изображения может понадобиться коррекция пользователя. В таком случае появляется окно, представленное на рисунке 36

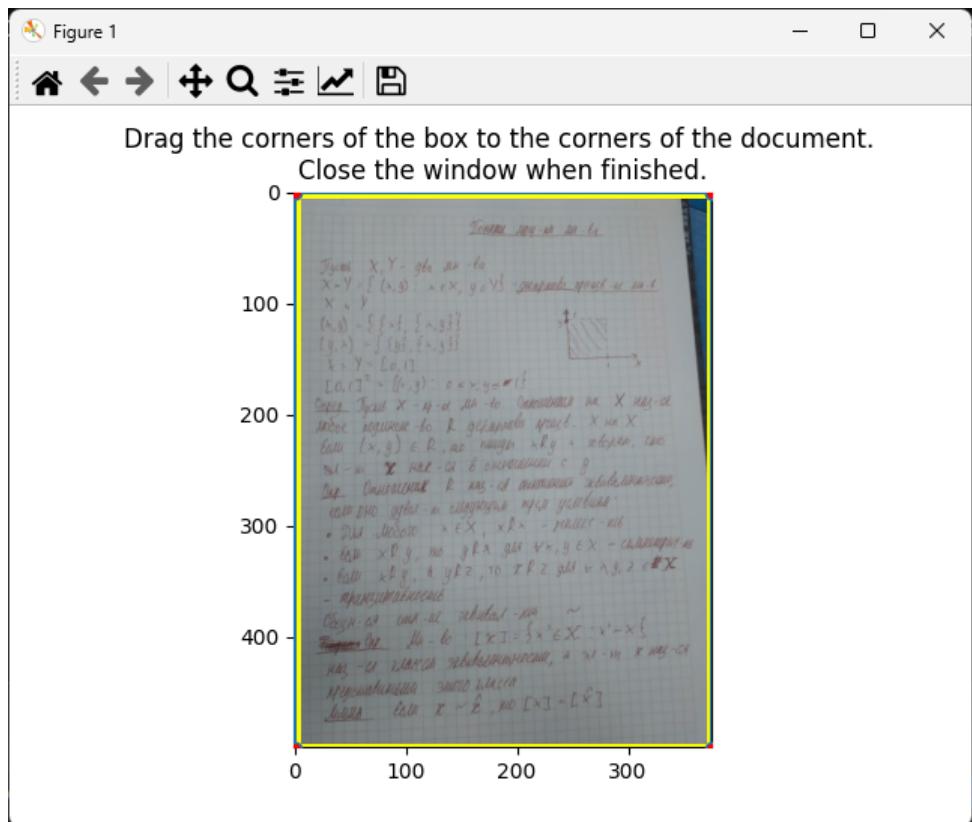


Рисунок 36 – Окно пользовательского ввода для коррекции перспективы

Дальнейшие этапы работы программы продемонстрируем на отсканированном изображении, представленном на рисунке 37

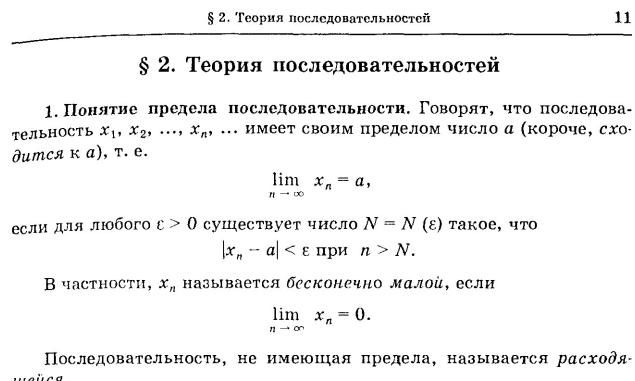


Рисунок 37 – Тестовое изображение

Изображение с коррекцией перспективы выводится на экран пользователя, а также отправляется на сервер, где происходит выделение формул. На клиентской машине в это время производится сегментация на абзацы и дальнейшая ее корректировка, как показано на рисунке 38

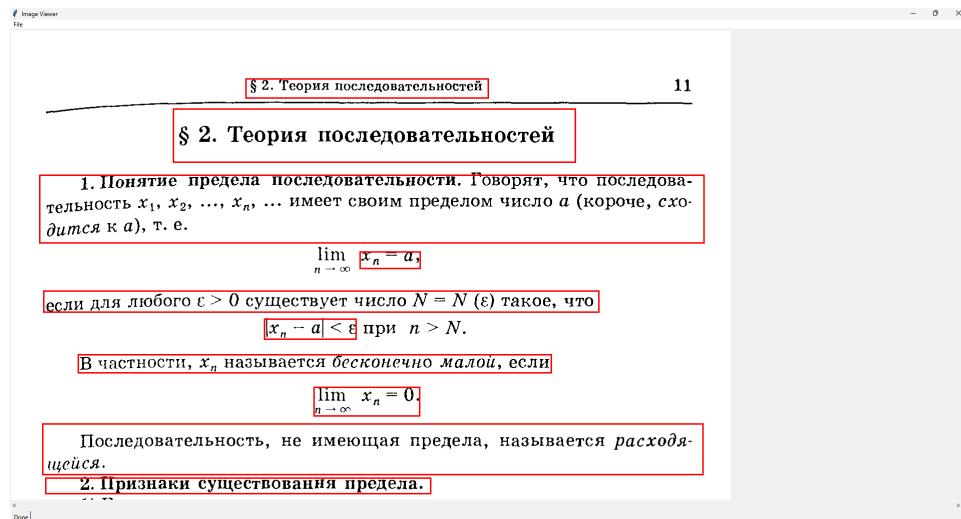


Рисунок 38 – Окно для пользовательского редактирования разбиения текста на абзацы

После получения ответа с сервера с координатами найденных формул клиент производит коррекцию формул, как показано на рисунке 39

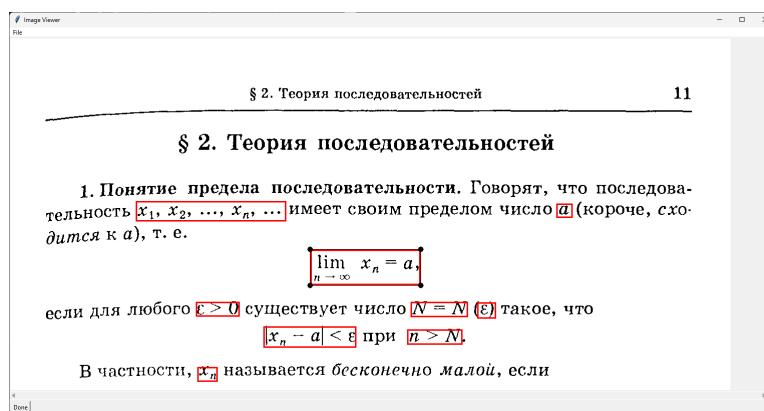


Рисунок 39 – Окно для пользовательского редактирования найденных формул

После обработки на сервере скорректированных данных происходит генерация L^AT_EX кода. В частности, для изображения, представленного на рисунках 38 и 39 сгенерированный код представлен на рисунке 40

```

1 \documentclass{article}
2 \usepackage[T2A]{fontenc} % Кодировка шрифта
3 \usepackage[utf8]{inputenc} % Кодировка ввода
4 \usepackage[english, russian]{babel} % Пакет для поддержки
      русского языка
5 \usepackage{amsmath} % Для использования математических формул
6
7 \begin{document}Теория
8 последовательностейТеория
9
10 последовательностей
11
12 1. Понятие предела последовательности. Говорят, что
    последовательность  $x_1, x_2, \dots, x_n, \dots$  имеет своим
    пределом число  $a$  короче(, сходится к  $a$ ), т.e..
13
14 [
15 \lim_{n \rightarrow \infty} x_n = a
16 ] если
17
18 для любого  $\epsilon > 0$  существует число  $N = N(\epsilon)$ 
    такое, что
19
20  $|x_n - a| < \epsilon$  при  $n > N$ 
21
22 частности,  $x_n$  называется бесконечно малой, если
23 [
24 \lim_{n \rightarrow \infty} x_n = 0
25 ] Последовательность
26
27 , не имеющая предела, называется расходящейся.
28
29 \end{document}

```

Рисунок 40 – Листинг изображения, представленного на
рисунке 37

ЗАКЛЮЧЕНИЕ

В результате выполнения выпускной квалификационной работы бакалавра была разработана система, выполняющая распознавание научного текста и генерацию его \LaTeX кода. Были решены следующие задачи:

- определены пользовательские сценарии;
- определены требования к платформе;
- спроектирована архитектура платформы;
- спроектирован и разработан рабочий прототип платформы;
- произведено тестирование прототипа на реальном изображении.

Развитие данной платформы, спроектированной и реализованной в данной работе, может позволить перевести существующие отсканированные документы в оцифрованный вид. На данном этапе платформа не способна конкурировать с аналогичными решениями, например, *Mathpix* [2].

Для успешного конкурирования необходимо улучшить качество распознавания формул, поддержать распознавание изображений в тексте, добавить распознавание рукописного текста, научиться распознавать структуру текста (заголовки, списки, выделение курсивом, полужирным и пр.).

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. eLibrary. Научная электронная библиотека. — URL: <https://www.elibrary.ru> (дата обращения 07.03.2024).
2. Mathpix. PDF to LaTeX. — URL: <https://mathpix.com/pdf-to-latex> (дата обращения 07.03.2024).
3. SberDevices. Гигачат. — URL: <https://developers.sber.ru/gigachat> (дата обращения 07.03.2024).
4. OpenAI. Chat GPT. — URL: <https://chatgpt.com/> (дата обращения 07.03.2024).
5. Midjourney. Midjourney. — URL: <https://www.midjourney.com/home> (дата обращения 07.03.2024).
6. Google. Using OAuth 2.0 to Access Google APIs. — URL: <https://developers.google.com/static/identity/protocols/oauth2/images/flows/authorization-code.png> (дата обращения 08.03.2024).
7. Weights & Biases. — URL: <https://wandb.ai> (дата обращения 27.04.2023).
8. OpenCV. Geometric Image Transformations. — URL: https://docs.opencv.org/4.x/da/d54/group__imgproc__transform.html (дата обращения 25.04.2024).
9. OpenCV. Smoothing Images. — URL: https://docs.opencv.org/4.x/d4/d13/tutorial_py_filtering.html (дата обращения 25.04.2024).
10. OpenCV. Canny Edge Detection. — URL: https://docs.opencv.org/4.x/da/d22/tutorial_py_canny.html (дата обращения 25.04.2024).
11. pylsd. Line Segment Detector. — URL: <https://github.com/primetang/pylsd> (дата обращения 25.04.2024).
12. OpenCV. Contours : Getting Started. — URL: https://docs.opencv.org/3.4/d4/d73/tutorial_py_contours_begin.html (дата обращения 25.04.2024).
13. OpenCV. Geometric Image Transformations. — URL: https://docs.opencv.org/4.x/da/d54/group__imgproc__transform.html (дата обращения 25.04.2024).
14. OpenCV. Image Thresholding. — URL: https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html (дата обращения 25.04.2024).
15. PyMuPDF. — URL: <https://github.com/pymupdf/PyMuPDF> (дата обращения 28.04.2023).

16. YOLO v.8. — URL: <https://github.com/ultralytics/ultralytics> (дата обращения 27.04.2024).
17. Tensorflow. — URL: <https://www.tensorflow.org> (дата обращения 27.04.2024).
18. Kaggle. — URL: <https://www.kaggle.com/> (дата обращения 27.04.2023).
19. non-maximum suppression. — URL: <https://pandysong.github.io/blog/post/nms-algorithm-non-maximum-suppression/> (дата обращения 27.04.2023).
20. Python Tesseract. — URL: <https://github.com/madmaze/pytesseract> (дата обращения 30.04.2023).
21. LaTeX-OCR. — URL: <https://github.com/lukas-blecher/LaTeX-OCR> (дата обращения 30.04.2023).
22. tkinter. — URL: <https://tkdocs.com/> (дата обращения 01.05.2023).
23. matplotlib. — URL: <https://matplotlib.org/stable/index.html> (дата обращения 25.04.2023).

ПРИЛОЖЕНИЕ А

Исходный код

На рисунке А.1 изображен QR-код со ссылкой на *GitHub* репозиторий с исходным кодом разработанного программного продукта.



Рисунок А.1 – QR-код на репозиторий