



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: Создание системы преобразования рукописного математического
текста в LaTeX

Автор ВКРБ:	Ядров Артем Леонидович	()
Руководитель:	Миронов Евгений Сергеевич	()
Консультант:	Кондаратцев Вадим Леонидович	()
Консультант:	—	()
Рецензент:	—	()

К защите допустить

Заведующий кафедрой № 806	Крылов Сергей Сергеевич	()
____ мая 2024 года		

Москва 2024

СОДЕРЖАНИЕ

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ	3
ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ	4
ВВЕДЕНИЕ	5
1 РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА	7
1.1 Высокоуровневая архитектура	7
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	12

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

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

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

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

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. Только в электронной версии научной библиотеки опубликовано 52573694 [1] статей, и все они написаны 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.

Однако, набор даже простых формул в \LaTeX для неподготовленного человека может оказаться достаточно сложным и трудоемким занятием. Для примера возьмем формулу

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

На рисунке 1 показан листинг \LaTeX кода этой формулы:

```
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

Как мы видим, используется много специальных символов (например, символ суммы, произведения, а также дроби, скобки и пр.), которые необходимо знать или тратить время для их поиска на просторах Интернета. В любом случае, требуется каждый раз компилировать pdf-файл для просмотра и проверки результата, что требуется некоторого количества времени.

В настоящее время появляется все больше различных нейросетей

(например, Гигачат, Yandex GPT, ChatGPT, stable diffusion, Midjoney и тд), в том числе преобразующие рукописный текст на изображении в машинный, а также способных генерировать готовый код. Поэтому появляется мотивация для автоматизации процесса преобразования формул из чистового варианта на бумаге в готовый \LaTeX -код.

Однако, мир не стоит на месте, и компания *Mathpix* придумала свое решение [2] этой задачи, которое распространяется по платной подписке, что не удовлетворяет требованию доступности ПО.

Целью работы является разработка программного обеспечения, выполняющего распознавание рукописного научного текста и генерацию готового \LaTeX кода.

Для достижения поставленной цели в работе были решены следующие задачи:

- а) Задача
- б) Задача еще одна:
 - 1) Подзадача для задачи

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

- а) *Python* является основным языком программирования, который использовался для решения задач;
- б) *Tensorflow*
- в) Что-то ещё

Результатом работы является:

- а) Результат

1 РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА

1.1 Высокоуровневая архитектура

Подводка к архитектуре. На рисунке 2 представлена общая архитектура продукта.



Рисунок 2 – Общая архитектура модели

Коррекция перспективы необходима для устранения шума на изображении и получения лучшего результата. Она состоит из нескольких этапов, представленных на рисунке 9. Также на рисунке представлены результаты, получаемые на каждом из этапов обработки.

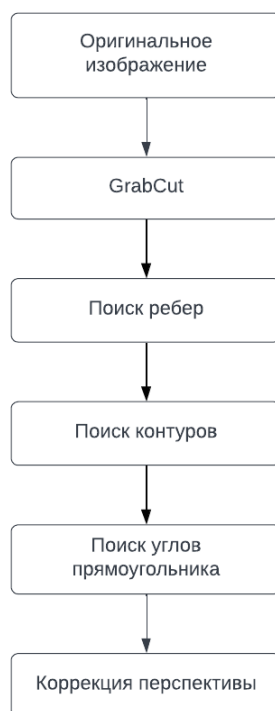


Рисунок 3 – Этапы коррекции перспективы изображения

На начальном этапе мы имеем изображение, показанное на рисунке 4

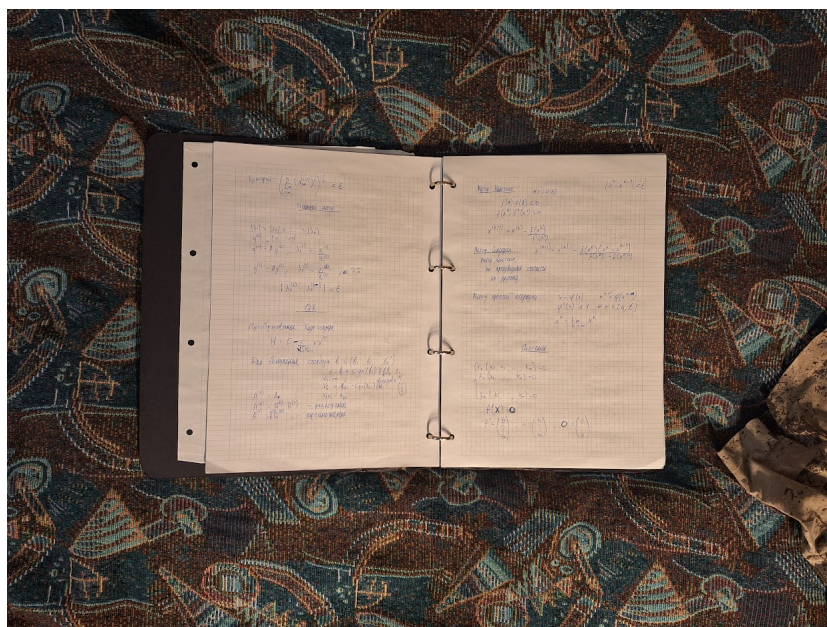


Рисунок 4 – Начальное изображение

Для начала необходимо удалить текст с изображения. Для этого несколько раз производится морфологическая операция закрытия с ядром 3×3 . После этого к изображению применяется алгоритм сегментации *GrabCut* [3]. На выходе данного этапа имеем изображение, представленное на

рисунке 5



Рисунок 5 – Изображение после сегментации алгоритмом *GrabCut*

На следующем этапе необходимо выделить контуры на полученном изображении. Для этого изображение переводится в серый цвет, затем к нему применяется алгоритм Гауссовского размытия [4]. После этого применяется алгоритм поиска ребер Канны [5]. На выходе данного этапа имеем изображение, представленное на рисунке 6

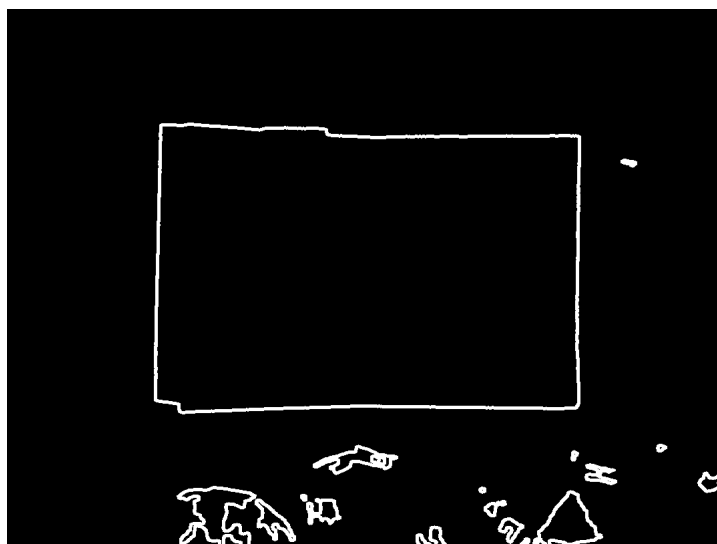


Рисунок 6 – Ребра, найденные на изображении

На этапе нахождения контуров берутся ребра, полученные на предыдущем этапе. Находим контуры с помощью алгоритма, встроенного в *openCV* [6]. Среди найденных контуров оставляем 5 контуров с наибольшей

площадью. На выходе данного этапа имеем изображение, представленное на рисунке 7



Рисунок 7 – Найденные алгоритмом контуры

Теперь необходимо среди оставленных контуров найти координаты вершин прямоугольника (если такой есть). Для этого пытаемся аппроксимировать контур с помощью встроенной в *openCV* [7] функции и найти прямоугольник. После определения прямоугольника сортируем вершины в нужном порядке:

- 1) Левая верхняя вершина
- 2) Правая нижняя вершина
- 3) Правая верхняя вершина
- 4) Левая нижняя вершина

На выходе данного этапа имеем изображение, представленное на рисунке

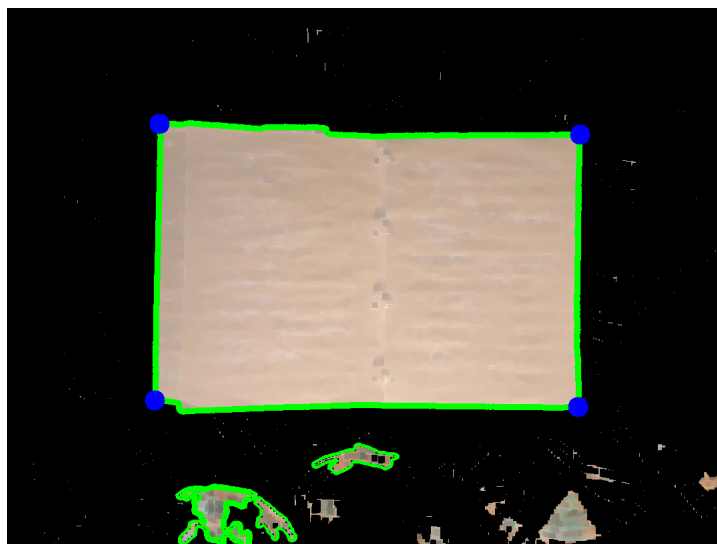


Рисунок 8 – Найденные вершины прямоугольника

На финальном этапе по имеющимся координатам прямоугольника, представляющего лист бумаги, осуществляем коррекцию перспективы. Для этого находим матрицу коррекции [8] и применяем ее к изображению [9]. Получаем результирующее изображение, показанное на рисунке 9

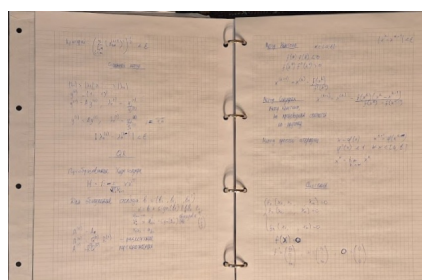


Рисунок 9 – Результирующее изображение

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. eLibrary. Научная электронная библиотека. — 2000. — URL: <https://www.elibrary.ru> (дата обращения 07.03.2024).
2. Mathpix. PDF to LaTeX. — URL: <https://mathpix.com/pdf-to-latex> (дата обращения 09.03.2024).
3. openCV. Interactive Foreground Extraction using GrabCut Algorithm. — URL: https://docs.opencv.org/3.4/d8/d83/tutorial_py_grabcut.html (дата обращения 25.04.2024).
4. openCV. Smoothing Images. — URL: https://docs.opencv.org/4.x/d4/d13/tutorial_py_filtering.html (дата обращения 25.04.2024).
5. openCV. Canny Edge Detection. — URL: https://docs.opencv.org/4.x/da/d22/tutorial_py_canny.html (дата обращения 25.04.2024).
6. openCV. Contours : Getting Started. — URL: https://docs.opencv.org/3.4/d4/d73/tutorial_py_contours_begin.html (дата обращения 25.04.2024).
7. openCV. Structural Analysis and Shape Descriptors. — URL: https://docs.opencv.org/4.x/d3/dc0/group_imgproc_shape.html (дата обращения 25.04.2024).
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. Geometric Image Transformations. — URL: https://docs.opencv.org/4.x/da/d54/group_imgproc_transform.html (дата обращения 25.04.2024).