



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное  
Образовательное учреждение высшего образования  
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа Инженерная школа ядерных технологий  
Направление подготовки 01.03.02 Прикладная математика и информатика  
Отделение школы (НОЦ) Экспериментальной физики

### ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

Тема работы
Обнаружение и распознавание многочленов не выше четвертой степени с помощью нейросетевого алгоритма

УДК 004.932.75.032.26:512.622

Обучающийся

Группа	ФИО	Подпись	Дата
0В91	Чуваков Виктор Николаевич		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОЭФ ТПУ	Крицкий Олег Леонидович	Кандидат физикоматематических наук		

### КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОСГН ШБИП	Кашук Ирина Вадимовна	к.т.н доцент		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Антоневич Ольга Алексеевна	к.б.н.		

### ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП/ОПОП, должность	ФИО	Ученая степень, звание	Подпись	Дата
Руководитель ООП 01.03.02 «Прикладная математика и информатика»	Крицкий Олег Леонидович	Кандидат физикоматематических наук		



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное  
Образовательное учреждение высшего образования  
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа Инженерная школа ядерных технологий  
Направление подготовки 01.03.02 Прикладная математика и информатика  
Отделение школы (НОЦ) Экспериментальной физики

УТВЕРЖДАЮ:  
Руководитель ООП/ОПОП

\_\_\_\_\_  
(Подпись) (Дата) (ФИО)

### ЗАДАНИЕ на выполнение выпускной квалификационной работы

Обучающийся:

Группа	ФИО
0В91	Чуваков Виктор Николаевич

Тема работы:

Обнаружение и распознавание многочленов не выше четвертой степени с помощью нейросетевого алгоритма	
Утверждена приказом директора (дата, номер)	

Срок сдачи обучающимся выполненной работы:	
--	--

#### ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

<b>Исходные данные к работе</b> <i>(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к функционированию (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.)</i>	Созданная нейронная сеть для обнаружения и распознавание многочленов не выше четвертой степени
<b>Перечень разделов пояснительной записки подлежащих исследованию, проектированию и разработке</b> <i>(аналитический обзор литературных источников с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе)</i>	1. Применение сверточной нейросетевой архитектуры YOLO для распознавания многочленов в pdf файле 2. Тестирование нейросети 3. Проверка эффективности работы нейросетевого алгоритма
<b>Перечень графического материала</b> <i>(с точным указанием обязательных чертежей)</i>	1. Рисунки работы программы 2. График средней точности
<b>Консультанты по разделам выпускной квалификационной работы</b> <i>(с указанием разделов)</i>	
<b>Раздел</b>	<b>Консультант</b>
Социальная	Антоневич Ольга Алексеевна

ответственность	
Финансовый менеджмент	Кашук Ирина Вадимовна
<b>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</b>	

**Задание выдал руководитель:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОЭФ ТПУ	Крицкий Олег Леонидович	Кандидат физикоматематических наук		

**Задание принял к исполнению обучающийся:**

Группа	ФИО	Подпись	Дата
0B91	Чуваков Виктор Николаевич		

## ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа		ФИО	
0В91		Чуваков Виктор Николаевич	
Школа	Инженерная школа ядерных технологий	Отделение (НОЦ)	Экспериментальной физики
Уровень образования	Бакалавриат	Направление/специальность	01.03.02 Прикладная математика и информатика

Тема ВКР:

**Обнаружение и распознавание многочленов не выше четвертой степени с помощью нейросетевого алгоритма**

**Исходные данные к разделу «Социальная ответственность»:**

<b>Введение</b>	<p>– Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика) и области его применения.</p> <p>– Описание рабочей зоны (рабочего места) при разработке проектного решения/при эксплуатации</p>	<p><i>Объект исследования:</i> нейросетевые алгоритмы для распознавания многочленов</p> <p><i>Область применения:</i> обнаружение многочленов в файлах pdf для дальнейшей работы с ними.</p> <p><i>Рабочая зона:</i> офис.</p> <p><i>Размеры помещения:</i> 24 м<sup>2</sup>.</p> <p><i>Количество и наименование оборудования рабочей зоны:</i> 1 персональный компьютер.</p> <p><i>Рабочие процессы, связанные с объектом исследования, осуществляющиеся в рабочей зоне:</i> программная разработка с использованием персонального компьютера</p>
-----------------	--	---

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

<p><b>1. Правовые и организационные вопросы обеспечения безопасности при разработке проектного решения:</b></p> <p>– специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства;</p> <p>– организационные мероприятия при компоновке рабочей зоны.</p>	<p>– СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организации воспитания и обучения, отдыха и оздоровления детей и молодежи»</p> <p>– ФЗ «Об обязательном социальном страховании от несчастных случаев на производстве»</p> <p>– Трудовой кодекс Российской Федерации: федер. Закон от 30 дек. 2001 г. №197-ФЗ Раздел 10</p>
<p><b>2. Производственная безопасность при разработке проектного решения:</b></p> <p>– Анализ выявленных вредных и опасных производственных факторов</p>	<p><b>Вредные факторы:</b></p> <p>– Отклонение показателей микроклимата;</p> <p>– Недостаточная освещённость рабочей зоны;</p> <p>– Повышенный уровень шума на рабочем месте;</p> <p>– Повышенный уровень статического электричества;</p> <p><b>Опасные факторы:</b></p> <p>– Опасность поражения электрическим током;</p>
<p><b>3. Экологическая безопасность при разработке проектного решения</b></p>	<p><b>Воздействие на литосферу:</b> утилизация компьютеров, оргтехники, бумаги.</p> <p><b>Воздействие на гидросферу:</b> производство компьютерной техники.</p>
<p><b>4. Безопасность в чрезвычайных ситуациях при разработке проектного решения</b></p>	<p><b>Возможные ЧС:</b></p> <p>Техногенные аварии (пожар).</p> <p>Наиболее типичная ЧС: пожар</p>

Дата выдачи задания для раздела по линейному графику

**Задание выдал консультант:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент	Антоневич Ольга Алексеевна	к.б.н.		

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
0В91	Чуваков Виктор Николаевич		

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

Студенту:

<b>Группа</b>	<b>ФИО</b>
0В91	Чуваков Виктор Николаевич

<b>Школа</b>	Инженерная школа ядерных технологий	<b>Отделение (НОЦ)</b>	Экспериментальной физики
<b>Уровень образования</b>	Бакалавриат	<b>Направление/специальность</b>	01.03.02 Прикладная математика и информатика

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

1. <i>Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих</i>	Стоимость материальных ресурсов и специального оборудования определены в соответствии с рыночными ценами г. Томска Тарифные ставки исполнителей определены штатным расписанием НИ ТПУ
2. <i>Нормы и нормативы расходования ресурсов</i>	Норма амортизационных отчислений на специальное оборудование
3. <i>Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования</i>	Отчисления во внебюджетные фонды 30 %

**Перечень вопросов, подлежащих исследованию, проектированию и разработке:**

1. <i>Анализ конкурентных технических решений (НИ)</i>	Расчет конкурентоспособности SWOT-анализ
2. <i>Формирование плана и графика разработки и внедрения (НИ)</i>	Структура работ. Определение трудоемкости. Разработка графика проведения исследования
3. <i>Составление бюджета инженерного проекта (НИ)</i>	Расчет бюджетной стоимости НИ
4. <i>Оценка ресурсной, финансовой, бюджетной эффективности (НИ)</i>	Интегральный финансовый показатель. Интегральный показатель ресурсоэффективности. Интегральный показатель эффективности.

**Перечень графического материала**

1. <i>Оценка конкурентоспособности ИР</i>
2. <i>Матрица SWOT</i>
3. <i>Диаграмма Ганта</i>
4. <i>Бюджет НИ</i>
5. <i>Основные показатели эффективности НИ</i>

<b>Дата выдачи задания для раздела по линейному графику</b>	
---	--

**Задание выдал консультант:**

<b>Должность</b>	<b>ФИО</b>	<b>Ученая степень, звание</b>	<b>Подпись</b>	<b>Дата</b>
Доцент ОСГН ШБИП	Кашук Ирина Вадимовна	к.т.н доцент		

**Задание принял к исполнению студент:**

<b>Группа</b>	<b>ФИО</b>	<b>Подпись</b>	<b>Дата</b>
0В91	Чуваков Виктор Николаевич		

## **Реферат**

Учебно-исследовательская работа студента выполнена на 71 страницах, содержит 8 рисунков, 25 источников.

*Ключевые слова:* нейронная сеть; python; сверточная нейросеть; компьютерное зрение; yolo.

Объект исследования: нейросетевой алгоритм распознавания многочленов.

Цель работы: создание нейросетевого алгоритма для обнаружения и распознавания многочленов не выше четвертой степени в файлах pdf.

Методы проведения работы: теоретические (изучение литературы, обзор нейросетевых алгоритмов) и практические (применение нейросетей для обнаружения и распознавания многочленов).

Работа написана в Microsoft Word 2018. Для написания программы на языке программирования Python использовалась среда разработки PyCharm и интерактивная облачная среда для работы с кодом на языке Python «Google Colab».

## **Определения, обозначения, сокращения и нормативные ссылки.**

В данной работе приведены следующие термины с соответствующими определениями:

Нейронная сеть (НС): представляет собой серию нейронов, которые связаны синапсами.

Искусственный нейрон (ИН): самый простой аналоговый элемент преобразования, который моделирует основные идеи о работе живого нейрона.

Сверточная нейронная сеть (СНС): специальная архитектура нейронных сетей, предложенная Яном Лекуном.

Слой: модуль обработки данных, принимающий на входе и возвращающий на выходе один или несколько тензоров.

Функция активации (ФА): функция, которая применяется к выходу каждого нейрона в нейронной сети.

## Оглавление

Введение.....	10
Обзор Литературы .....	11
<b>1. Теоретическая часть.....</b>	<b>13</b>
1.1 Нейронная сеть.....	13
1.2 Функция активации .....	16
1.3 Сверточная нейронная сеть.....	18
1.4 Оптическое распознавание символов.....	18
1.5 Архитектура нейронных сетей YOLO .....	19
<b>2. Практическая часть .....</b>	<b>22</b>
2.1 Установка необходимых библиотек.....	22
2.2 Нейросеть для распознавания многочленов .....	22
2.3 Подготовка pdf файла .....	25
2.4 Работа с данными .....	25
2.5 Проверка работы и результаты .....	27
<b>3. Социальная ответственность .....</b>	<b>29</b>
Введение.....	29
<b>3.1. Правовые и организационные вопросы обеспечения безопасности .....</b>	<b>29</b>
<b>3.2. Производственная безопасность .....</b>	<b>30</b>
<b>3.2.1. Отклонение показателей микроклимата .....</b>	<b>31</b>
3.2.2 Недостаточная освещенность рабочей зоны .....	32
3.2.3. Повышенный уровень шума на рабочем месте.....	33
3.2.4 Повышенный уровень статического электричества .....	35
3.2.5 Опасность поражения электрическим током .....	35
<b>3.3 Экологическая безопасность .....</b>	<b>36</b>
<b>3.4 Безопасность в чрезвычайных ситуациях .....</b>	<b>37</b>
3.4.1 Пожар .....	37
<b>Выводы по разделу .....</b>	<b>38</b>
<b>4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение .....</b>	<b>40</b>
Введение.....	40
<b>4.1 Оценка коммерческого потенциала и перспективности проведения исследований с позиции ресурсоэффективности и ресурсосбережения .....</b>	<b>40</b>
4.1.1 Анализ конкурентных технических решений .....	40
4.1.2 SWOT-анализ.....	42
<b>4.2. Планирование научно-исследовательских работ.....</b>	<b>44</b>
4.2.2 Определение трудоемкости выполнения работ и разработка графика проведения.....	45
<b>4.3 Бюджет научно-технического исследования.....</b>	<b>48</b>
4.3.1 Расчет материальных затрат научно-технического исследования.....	49



4.3.2 Расчет амортизации специального оборудования .....	49
4.3.3 Основная заработная плата исполнителей темы.....	50
4.3.4 Отчисления во внебюджетные фонды (страховые отчисления) .....	53
4.3.5 Накладные расходы .....	53
4.3.6 Бюджет НИР .....	54
<b>4.4 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования.....</b>	<b>55</b>
<b>Выводы по разделу .....</b>	<b>58</b>
<b>Заключение.....</b>	<b>59</b>
<b>Список литературы .....</b>	<b>60</b>
<b>Приложение А.....</b>	<b>64</b>
<b>Приложение Б.....</b>	<b>66</b>
<b>Приложение В.....</b>	<b>68</b>

## **Введение**

В наше время одной из самых быстро развивающихся отраслей информационных технологий является искусственный интеллект, а именно – нейронные сети и нейросетевые алгоритмы. Нейросети могут решить практически любую задачу, которую можно придумать, они способны самообучаться и совершенствоваться практически самостоятельно.

Нейронная сеть является одним из типов машинного обучения, а машинное обучение, в свою очередь, является одним из видов искусственного интеллекта. Работа нейросети основывается на работе нейронов в человеческом мозге. Роль нейронов и нейросети играют различные простейшие процессы. Как нейроны в мозге взаимодействуют между собой, так и процессы в нейросети используют взаимодействие для решения различных задач.

Нейронные сети хорошо работают с изображениями, а так как практически любую информацию можно представить в виде изображения, то использование нейронных сетей актуально практически в любой сфере.

Актуальность данной работы обусловлена активным развитием применения нейронных сетей для решения различных задач в настоящее время и в обозримом будущем.

Целью данной работы является создание нейросетевого алгоритма для обнаружения и распознавания многочленов не выше четвертой степени в файлах pdf.

Задачи, необходимые для достижения цели:

1. Выбор наиболее эффективного алгоритма распознавания многочленов;
2. Применение алгоритма к pdf файлам с многочленами для проверки его эффективности;
3. Оценка результатов работы.

## Обзор Литературы

Из источника [1] возьмём информацию о нейронных сетях, о том, что они из себя представляют, суть их работы, отличия от других алгоритмов.

Из источника [2] возьмём схему работы многослойной нейронной сети, на которой описано взаимодействие входных сигналов, нейронов, слоёв нейронной сети и выходного значения, также в этом источнике рассказывается о выборе нейронной сети в зависимости от типа задачи.

В источнике [3] находится информация о принципе работы искусственного нейрона и схема, иллюстрирующая этот принцип, а также о данных, подающихся на вход нейрона.

Из источника [7] берётся описание функции активации, рассматриваются такие функции активации, как: сигмоида, линейная, ступенчатая, ReLu, tanh. А также говорится о назначении функции активации.

Иллюстрация процесса обучения нейросети будет взята из источника [4], также в этом источнике описан этот процесс обучения.

Из источника [5] возьмём информацию о том, как перевести pdf файл в изображение в среде программирования python (это необходимо для того, чтобы нейросеть могла работать с данными).

В источнике [6] рассказывается о разнице обучения нейронной сети с учителем и без учителя.

О свёрточной нейронной сети, о том, для чего она нужна и как работает информация представлена в [10].

Информацию о OCR (оптическом распознавании символов) возьмём из источника [8].

О том, почему при работе с нейросетями удобнее использовать язык программирования Python говорится в источнике [9].

Из источника [11] будет взята информация о архитектуре нейронной сети YOLO, о том, как она работает и что из себя представляет.

Об различных типах детекторов, и их составе говорится в источнике [12], также там представлен рисунок анатомии различных типов детекторов.

Основная информация об обнаружении объектов при помощи YOLO будет взята из источника [13]. Там приведены фрагменты кода и их описание.

Простое описание алгоритма работы YOLO по шагам будет взято из источника [14].

## **1. Теоретическая часть**

### **1.1 Нейронная сеть**

Структурно любая нейросеть представляет собой совокупность простых процессоров, разделенных на слои, где производятся параллельные вычисления. Между отдельными слоями происходит двусторонний обмен, поэтому последовательность действий, обусловленная наличием нескольких слоев, достаточно условна. [1]

Отличительной особенностью нейронной сети от других алгоритмов является её способность обучаться. Обучение заключается в изменении весовых коэффициентов для нейронов.

Нейросетевой алгоритм представляет из себя вычислительную процедуру, реализованную в виде нейронной сети.

Основой разработки нейросетевого алгоритма является системный подход, при котором процесс решения задачи представляется как функционирование во времени некоторой динамической системы.

Для построения этой системы необходимо определить:

- объект, выступающий в роли входного сигнала нейронной сети;
- объект, выступающий в роли выходного сигнала нейронной сети (например, непосредственно решение или некоторая его характеристика);
- желаемый (требуемый) выходной сигнал нейронной сети;
- структуру нейронной сети (число слоёв, связи между слоями, объекты, служащие весовыми коэффициентами);
- функцию ошибки системы (характеризующую отклонение желаемого выходного сигнала нейронной сети от реального выходного сигнала);
- критерий качества системы и функционал её оптимизации, зависящий от ошибки;
- значение весовых коэффициентов (например, определяемых аналитически непосредственно из постановки задачи, с помощью некоторых численных

методов или процедуры настройки весовых коэффициентов нейронной сети).

Структура нейронной сети:

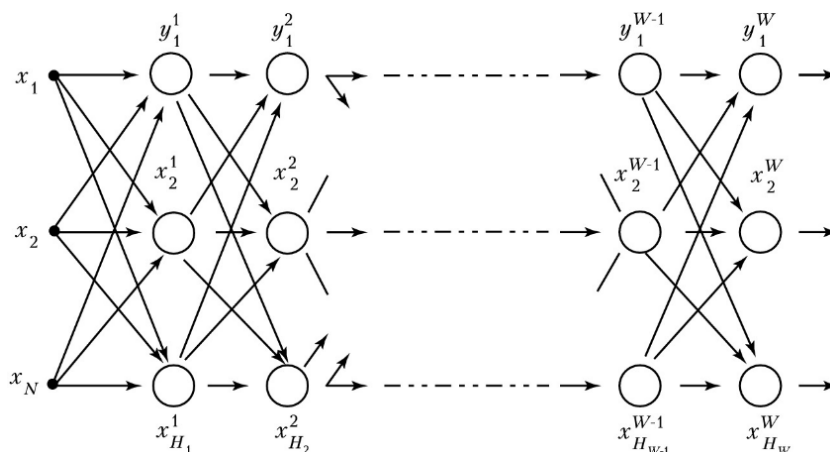


Рис. 1.1. Граф многослойной нейронной сети с последовательными связями [2]

На схеме  $x_i$  – это входные сигналы,  $H_i$  – число нейронов на определённом слое,  $W$  – число слоёв,  $x_{H_W}^W$  – выходной сигнал  $H_W$  нейрона  $W$  слоя,  $y_i^W$  выходное значение  $W$ -го слоя.

Выбор структуры нейронной сети осуществляется в соответствии с особенностями и сложностью задачи. Теоретически число слоёв и число нейронов в каждом слое нейронной сети может быть произвольным, однако фактически оно ограничено ресурсами компьютера или специализированной микросхемы, на которых обычно реализуется нейронная сеть. [2]

Представим графически принцип работы нейронной сети:

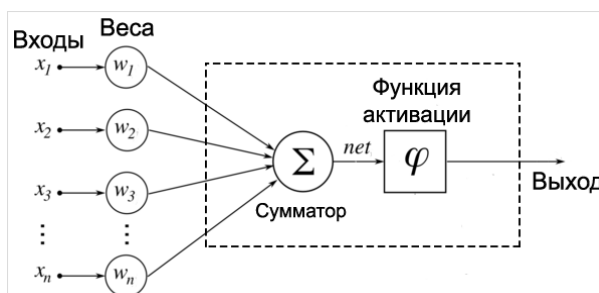


Рис. 1.2. Принцип работы искусственного нейрона [3]

На вход нейрона подаются данные, которые необходимо обработать. Каждому входу присваивается вес (значение), после чего входы и их веса перемножаются и суммируются. К сумматору применяется функция активации, которая выдаёт выходное значение в зависимости от значения сумматора.

Процесс обучения нейронной сети представлен на следующем рисунке:

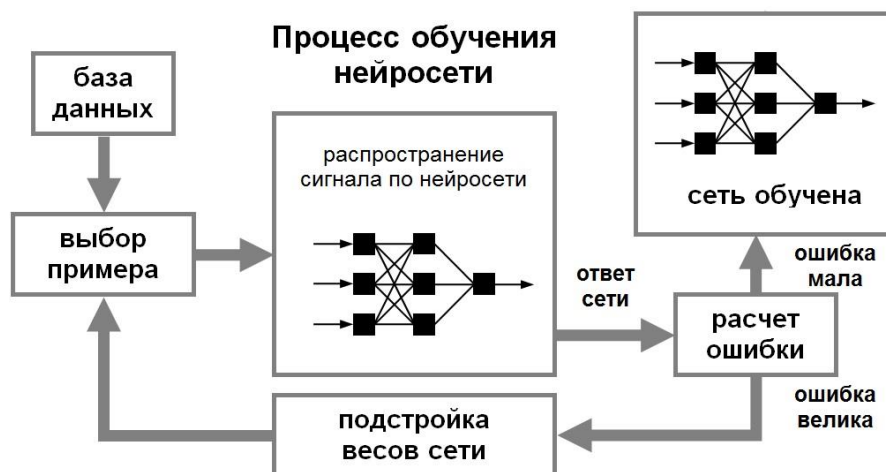


Рис. 1.3. Иллюстрация процесса обучения нейросети [4]

Нейронные сети обучаются при помощи прямого и обратного распространения ошибки. При прямом распространении ошибки выполняется предсказание ответа, а при обратном распространении ошибки происходит сравнение с теоретическим значениями и подстройка параметров. Во время прямого распространения ошибки делается предсказание ответа. При обратном распространении ошибка между фактическим ответом и предсказанным минимизируется. [4]

Можно сказать, что за обучаемость нейронной сети отвечает алгоритм обратного распространения ошибки. Цели обратного распространения ошибки: отрегулировать каждый вес пропорционально тому, насколько он способствует общей ошибке. Если итеративно уменьшать ошибку каждого веса, в конце концов будет ряд весов, которые дают хорошие прогнозы.

В зависимости от данных алгоритмы машинного обучения могут быть поделены на те, что обучаются с учителем и без учителя (*supervised & unsupervised learning*). В задачах обучения без учителя имеется выборка, состоящая из объектов, описываемых набором признаков. В задачах обучения с учителем вдобавок к этому для каждого объекта некоторой выборки, называемой обучающей, известен целевой признак – по сути это то, что хотелось бы прогнозировать для прочих объектов, не из обучающей выборки. Т.е в задачах МО с учителем на обучающей выборке у нас есть “правильные” ответы, а когда задача без учителя - то нет. [6]

Лучше всего с нейронными сетями работать на языке программирования Python. Многочисленные фреймворки и библиотеки Python помогают существенно уменьшить количество времени, необходимого для разработки приложений. [9]

Программная библиотека — это предварительно написанный код, который разработчики используют для решения распространенных задач. Python, с его богатым стеком технологий, имеет обширный набор библиотек для искусственного интеллекта и машинного обучения. Вот некоторые из них:

- Keras, TensorFlow, и Scikit-learn — для машинного обучения
- NumPy — для высокопроизводительных научных вычислений и анализа данных
- SciPy — для продвинутых вычислений
- Pandas — для общего анализа данных
- Seaborn — для визуализации данных

## 1.2 Функция активации

Функция активации определяет выходное значение нейрона в зависимости от результата взвешенной суммы входов и порогового значения. [7]

Функция активации работает со следующим значением нейрона:

$$Y = \sum (weight * input) + bias. \quad (1.1)$$



Значение  $Y$  может быть любым в диапазоне от  $-\infty$  до  $+\infty$ . В действительности нейрон не знает границу, после которой следует активация, поэтому была добавлена функция активации. Она проверяет произведенное нейроном значение  $Y$  на предмет того, должны ли внешние связи рассматривать этот нейрон как активированный, или его можно игнорировать.

Рассмотрим различные функции активации:

#### *Ступенчатая функция активации*

Функция принимает значение 1 (активирована), когда  $Y > 0$  (граница), и значение 0 (не активирована) в противном случае. Если значение  $Y$  больше некоторого порогового значения, считаем нейрон активированным. В противном случае говорим, что нейрон неактивен.

#### *Линейная функция активации*

Линейная функция представляет собой прямую линию и пропорциональна входу ( $A = c \cdot x$ ). Такой выбор активационной функции позволяет получать спектр значений, а не только бинарный ответ. Можно соединить несколько нейронов вместе и, если более одного нейрона активировано, решение принимается на основе применения операции  $\max$  (или  $\text{softmax}$ ).

#### *Сигмоида*

Сигмоида выглядит гладкой и подобна ступенчатой функции ( $A = \frac{1}{1+e^{-x}}$ ). Сигмоида является подходящей функцией для задач классификации. Она стремится привести значения к одной из сторон кривой (например, к верхнему при  $x=2$  и нижнему при  $x=-2$ ). Такое поведение позволяет находить четкие границы при предсказании.

#### *Гиперболический тангенс*

Гиперболический тангенс очень похож на сигмоиду ( $A = \frac{2}{1+e^{-2x}} - 1$ ). Поэтому такая функция имеет те же характеристики, что и у сигмоиды, рассмотренной ранее. Её природа нелинейна, она хорошо подходит для комбинации слоёв, а диапазон значений функции –  $(-1, 1)$ .

#### *ReLU*

ReLU возвращает значение  $x$ , если  $x$  положительно, и  $0$  в противном случае ( $A(x) = \max(0, x)$ ). ReLU менее требовательно к вычислительным ресурсам, чем гиперболический тангенс или сигмоида, так как производит более простые математические операции. Поэтому имеет смысл использовать ReLU при создании глубоких нейронных сетей.

Функция активации является «контролером» для получения вывода о необходимости активации нейрона. Благодаря функциям активации нейронные сети способны порождать более информативные признаковые описания, преобразуя данные нелинейным образом.

### **1.3 Сверточная нейронная сеть**

Сверточная нейросеть используется для распознавания образов на изображении, они состоят из нейронов, которые, в свою очередь, содержат изменяемые веса и смещения. Каждый нейрон получает какие-то входные данные, вычисляет скалярное произведение и, опционально, использует нелинейную функцию активации. Вся сеть по-прежнему представляет собой единственную дифференцируемую функцию оценки: из исходного набора пикселей на одном конце до распределения вероятностей принадлежности к определённому классу на другом конце.

Как и полносвязная нейронная сеть, свёрточная сеть обучается с помощью алгоритма обратного распространения ошибки. Сначала выполняется прямое распространение от первого слоя к последнему, после чего вычисляется ошибка на выходном слое и распространяется обратно. При этом на каждом слое вычисляются градиенты обучаемых параметров, которые в конце обратного распространения используются для обновления весов с помощью градиентного спуска. [10]

### **1.4 Оптическое распознавание символов**

OCR (оптическом распознавании символов) работает следующим образом: системы распознавания текста преобразуют двумерное изображение текста, которое

может содержать машинный печатный или рукописный текст, из его графического представления в машиночитаемый текст. Распознавание текста как процесс обычно состоит из нескольких подпроцессов для выполнения с максимально возможной точностью. [8]

Подпроцессы:

- Предварительная обработка изображения
- Локализация текста
- Сегментация символов
- Распознавание символов
- Постобработка

## **1.5 Архитектура нейронных сетей YOLO**

Для распознавания многочленов будет использоваться архитектура нейронных сетей YOLO (You Only Look Once) – алгоритм обнаружения объектов в реальном времени.

YOLO объединяет то, что когда-то было многоступенчатым процессом, используя единую нейронную сеть для выполнения как классификации, так и прогнозирования ограничивающих рамок для обнаруженных объектов. Таким образом, он оптимизирован для производительности обнаружения и может работать намного быстрее, чем запуск двух отдельных нейронных сетей для обнаружения и классификации объектов по отдельности. [11]

YOLO основан на идее деления изображения на более мелкие изображения. Изображение разбивается на квадратную сетку размером  $S \times S$ .

Ячейка, в которой находится центр объекта является ячейкой, ответственной за обнаружение этого объекта. Каждая ячейка будет предсказывать В ограничивающих рамок и показатель достоверности для каждой рамки. По умолчанию для этой архитектуры модель предсказывает две ограничивающие рамки. Оценка классификации будет от "0.0" до "1.0", причем "0.0" - это самый низкий уровень достоверности, а "1.0" - самый высокий; если в этой ячейке не

существует объекта, оценки достоверности должны быть "0.0", и если модель полностью уверена в своем предсказании, оценка должна быть "1.0". Эти уровни достоверности отражают уверенность модели в том, что в этой ячейке существует объект и что ограничивающая рамка является точной. Каждая из этих ограничивающих рамок состоит из 5 чисел: позиции x, позиции y, ширины, высоты и достоверности. Координаты "(x, y)" представляют местоположение центра прогнозируемой ограничивающей рамки, а ширина и высота являются долями по отношению ко всему размеру изображения. Достоверность представляет собой долговую расписку между предсказанным ограничивающим прямоугольником и фактическим ограничивающим прямоугольником, называемым базовым блоком истинности. Долговая расписка обозначает пересечение через объединение и представляет собой площадь пересечения предсказанных и обоснованных блоков истины, разделенную на площадь объединения тех же самых предсказанных и обоснованных блоков истины.

По сути, нейронная сеть обнаружения объектов обычно состоит из трех частей. Авторы назвали их позвоночником (backbone), шеей (neck) и головой (head). [12]

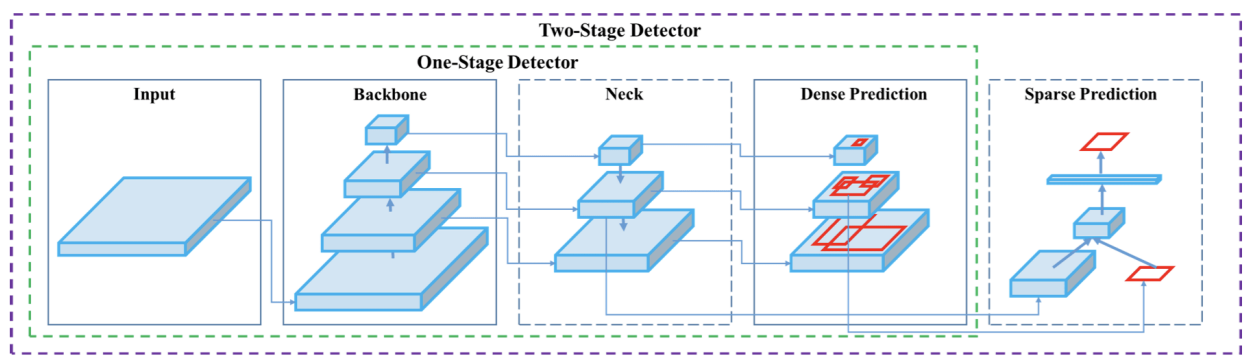


Рис. 1.4. Анатомия различных типов детекторов [12]

Позвоночник обычно представляет собой глубокую архитектуру, которая была предварительно обучена на наборе данных ImageNet без верхних слоев. Обычно это одна из распространенных архитектур CNN, таких как Resnet, VGGNet, Mobilenet и

т.д. YOLO, в частности, использует специализированную версию backbone, называемую 'Darknet'.

Шея обычно состоит из нескольких слоев, целью которых является сбор карт объектов с разных этапов. Распространенными примерами “шейки” являются: Сеть пирамиды объектов (FPN), PAN, SPP, ASPP и т.д.

Голова — это часть модели обнаружения объектов, которая используется для прогнозирования пользовательских классов и рисования ограничивающих рамок вокруг объектов. В зависимости от типа головки можно выделить два типа моделей обнаружения объектов, описанных выше. Эти слои также могут быть описаны в терминах плотных предикторов (одноступенчатые детекторы) или разреженных предикторов (двухступенчатые детекторы).

## 2. Практическая часть

### 2.1 Установка необходимых библиотек

Для работы нейросетевого алгоритма понадобятся следующие библиотеки:

- *opencv-python*: библиотека для обработки изображений и задач компьютерного зрения;

- *pdf2image*: библиотека для перевода pdf файла в изображение;
- *pix2tex*: библиотека для перевода изображения в формат LaTeX;
- *keras==1.2.2*: библиотека для взаимодействия с нейросетями;
- *poppler-utils*: инструмент для работы с документами pdf

Также были импортированы следующие модули:

- *cv2*: библиотека компьютерного зрения, которая предназначена для анализа изображений, а также их классификации и обработки;

- *numpy*: библиотека для работы с многомерными массивами;

- *PIL.Image*: библиотека для открытия, работы и сохранения различных форматов изображений;

- *re*: язык для набора возможных строк, которые вы хотите сопоставить;

### 2.2 Нейросеть для распознавания многочленов

Для распознавания многочленов будет использоваться YOLOv4. Весь код для распознавания многочленов представлен в *Приложение Б*.

Слои модели, с которыми работает нейросеть представлены в *Приложение В*.

На вход нейронной сети подаются слои модели, веса, уровень достоверности (*confidence\_idx*), пороговый индекс (*threshold\_idx*) и разрешение (размерность).

Переработка входных данных нейронной сети и отсеивание всех объектов, уровень достоверности идентификации которых меньше, чем параметр *confidence\_idx* (в данной работе он равен 0,5 – 50%) происходит следующим образом:

Перебирается каждый из выходов слоя и каждое обнаружение объекта:

```
for output in layer_outputs:
    for detection in output:
```

После этого происходит извлечение идентификатора класса (метки) и достоверности (как вероятности) и происходит обнаружение текущего объекта:

```
scores = detection[5:]
class_id = np.argmax(scores)
confidence = scores[class_id]
```

После этого отбрасываются слабые прогнозы:

```
if confidence > self.confidence_idx:
```

Работая с прогнозами больше заданной достоверности, производится масштабирование координаты ограничивающего прямоугольника относительно размер изображения при учете того, что YOLO возвращает центральные координаты (x, y) ограничивающего поля, за которым следуют ширина и высота поля. Используем центральные координаты (x, y) для получения вершины и левый угол ограничительной рамки:

```
box = np.around(detection[0:4], 2)
initial_boxes.append(box)
(centerX, centerY, width, height) = box
x = centerX - (width / 2)
y = centerY - (height / 2)
```

Обновляем наш список координат ограничивающего прямоугольника, достоверности, и идентификаторы класса:

```
boxes.append([x, y, width, height])
confidences.append(confidence)
class_ids.append(class_id)
```

Далее проверяется, был ли обнаружен хотя бы один объект:

```
if len(idxs) > 0:
```

После этого запоминаются перечисленные выше параметры и ограничивающие рамки ставятся вокруг следующего объекта.

После того, как все объекты будут найдены в ограничивающих рамках и их параметры будут запомнены, то можно будет сказать, что нейросеть обучена.

График средней точности в процессе обучения выглядит следующим образом:

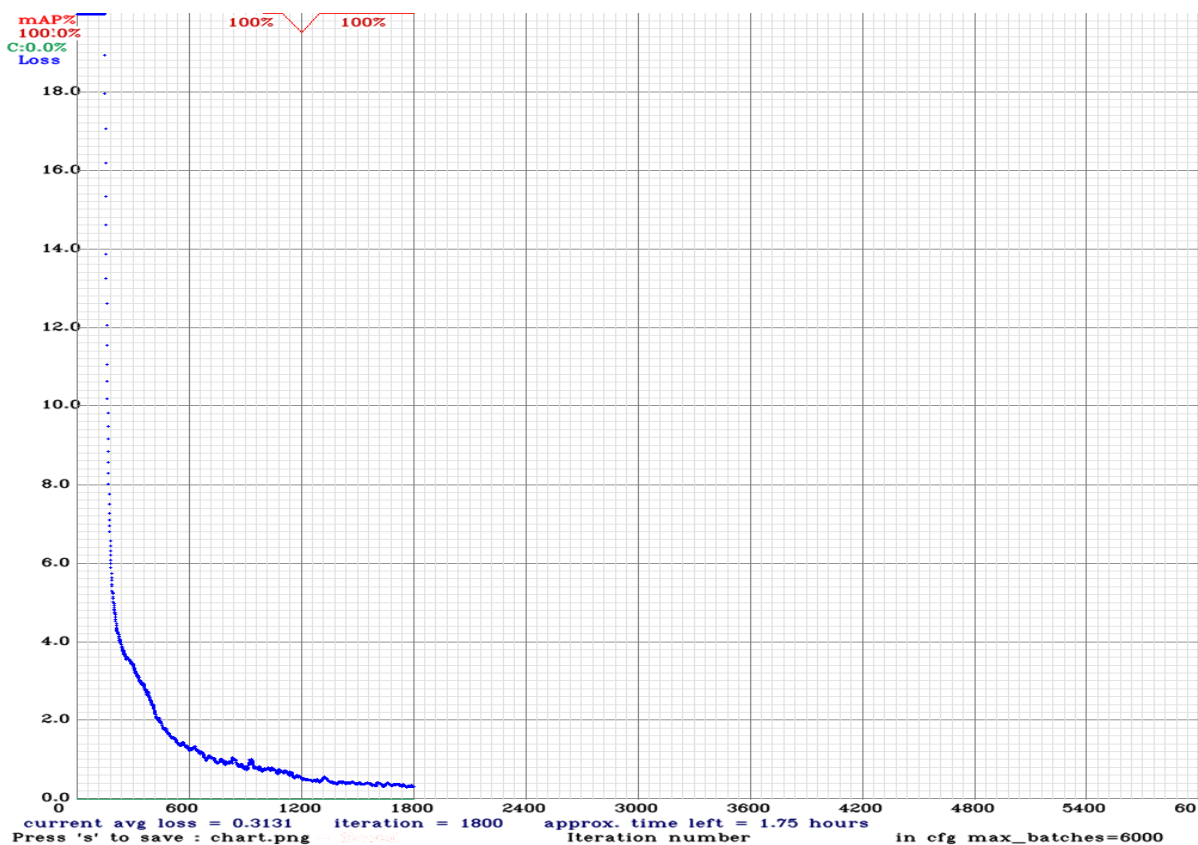


Рис. 2.1. График средней точности.



По графику видно, что после 1800 итераций значение средней ошибки стало равно 0,3131.

### 2.3 Подготовка pdf файла

Так как нейросеть работает с изображениями, то для начала pdf файл переводится в изображение. Для этого используется функция *convert\_from\_path* из библиотеки *pdf2image*.

После этого для лучшего распознавания уменьшаем картинки многочленов на 50%, вводя функцию *resize\_im*, которую можно увидеть в *Приложение А*.

Теперь мы имеем изображение, с которым будет работать нейросеть.

### 2.4 Работа с данными

Многочлены из pdf файла будем преобразовывать в формат LaTeX (алгебраический вид).

При использовании нейросети для распознавания многочленов используются те же параметры, что и при её обучении.

Нейросеть применяется к изображению:

```
for box in boxes:
    box = [i if i>0 else 0 for i in box]
    x,y,w,h = box
    crop = cv_im[y:y+h, x:x+w]
    cropped_elements.append(crop)
    resized = resize_im(crop)
    im_pil = Image.fromarray(resized)
    res = model(im_pil)
```

После этого проверяется, не превышает ли многочлен 4-ю степень:

```
powers = re.findall(r'\^\{(\d+)\}', poly_string)
for power in powers:
```

```
    if int(power) >= 5:
        return True
return False
```

Результат переводится в LaTeX:

```
res = res.replace('\mathbf{x}', 'x')
res = res.replace('{\bf x}', 'x')
res = res.replace('{\bf {x}}', 'x')
res = res.replace('{\bf{x}}', 'x')
print(res + '\\\\')
latex_code.append(res + '\\\\')
```

После этого выводятся многочлены до 4-й степени в формате LaTeX.

Весь код представлен в *Приложение А*.

## 2.5 Проверка работы и результаты

Для проверки нейросетевого алгоритма возьмём следующий pdf файл с многочленами различных степеней:

$$\begin{aligned}3x^3 - 2x^2 + x &= 4 \\4x^4 - 3x^3 + 2x^2 - x &= -10 \\x - 6 &= 0 \\5x^6 - 2x^2 + x &= -11 \\3x^4 - 5x + 9 &= 0 \\x^5 - 1 &= 0 \\7x^2 - 2x^9 + x &= 2 \\3x^4 - 4x^3 + x - 3 &= 0 \\8x^2 - 3x^4 + 2x^3 - x + 3 &= 0 \\10x - 4 &= 0 \\9x^6 - 2x^5 + x - 5 &= 0 \\6x^2 - 8x + 19 &= 0 \\x &= 0 \\11x^2 - 2x^4 + x - 1 &= 0 \\13x^2 - 3x^3 + 2x^4 - x^5 + 12 &= 0 \\7x - 78 &= 0 \\6x^3 - 8x + 1239 &= 0 \\x^2 &= 152 \\11x^6 - 2x^5 + x - 1 &= 0\end{aligned}$$

Рис. 2.2. Данные входного pdf файла

В данном файле 19 многочленов, 13 из которых меньше 4-го порядка.

После применения нейросетевого алгоритма получился следующий результат:

$$\begin{aligned}
 &3x^3 - 2x^2 + x = 4 \\
 &4x^4 - 3x^3 + 2x^2 - x = -10 \\
 &x - 6 = 0 \\
 &3x^4 - 5x + 9 = 0 \\
 &3x^4 - 4x^3 + x - 3 = 0 \\
 &8x^2 - 3x^4 + 2x^3 - x + 3 = 0 \\
 &10x - 4 = 0 \\
 &\mathbf{X} \sim 6 \quad \hat{\mathbf{e}} \times \sim \quad \hat{\mathbf{e}} \\
 &\quad \hat{\mathbf{e}} \\
 &9x - -xx + x - 3 = 0 \\
 &6x^2 - 8x + 19 = 0 \\
 &x = 0 \\
 &11x^2 - 2x^4 + x - 1 = 0 \\
 &\bar{\eta}x - \bar{\eta}8 = 0 \\
 &6x^3 - 8x + 1239 = 0 \\
 &x^2 = 152
 \end{aligned}$$

Рис. 2.3. Результат работы нейросетевого алгоритма

Были выведены все многочлены до 4-го порядка, но в одном из них ( $7x-78=0$ ) нейросеть восприняла 7 как  $\bar{\eta}$ . Помимо этого, нейросеть не смогла корректно распознать многочлен  $9x^6 - 2x^5 + x - 5 = 0$  и выдала следующий результат:

$$\begin{aligned}
 &\mathbf{X} \sim 6 \quad \hat{\mathbf{e}} \times \sim \quad \hat{\mathbf{e}} \\
 &\quad \hat{\mathbf{e}} \\
 &9x - -xx + x - 3 = 0
 \end{aligned}$$

Рис. 2.4. Некорректный вывод многочлена

Нейросеть была испытана и на других файлах с большим количеством многочленов.

В результате нейросеть в основном работает корректно, но иногда возникают ошибки.

Нейросеть может работать с файлами, содержащими многочлены и выявлять из них многочлены до 4-го порядка.

### **3. Социальная ответственность**

#### **Введение**

Объектом разработки данной ВКР является нейросетевой алгоритм для распознавания многочленов. Созданный нейросетевой алгоритм сможет распознавать многочлены в файлах pdf.

Проект выполняется на персональном компьютере (ПК), поэтому в данном разделе проводится анализ опасных и вредных факторов при работе с ПК, влияния этих факторов на окружающую среду и мероприятий по её защите.

Предметом исследования является рабочая зона разработчика, включая компьютерный стол, ПК, клавиатуру, компьютерную мышь и стул. Работы выполнялись в компьютерном классе 427А 10 корпуса ТПУ.

#### **3.1. Правовые и организационные вопросы обеспечения безопасности**

Правовые вопросы организации труда работника, обеспечение его безопасности регулируется трудовым кодексом РФ, а также санитарными нормами и инструкциями. В соответствии со ст.100 Трудового кодекса РФ режим рабочего времени устанавливается правилами внутреннего трудового распорядка, которые, в свою очередь, утверждаются работодателем с учетом мнения представительного органа работников. В данном документе также регламентируется продолжительность рабочего времени, которая не должна быть меньше указанного времени в трудовом договоре, но, в свою очередь, не должна превышать 40 часов в неделю. Также законодательством предусмотрено установление в течение рабочего дня перерывов для питания и отдыха [17]. К актам, устанавливающим количество и продолжительность технологических перерывов, обязательным для исполнения, относятся санитарные нормы и правила [15], которыми установлены гигиенические требования к персональным электронно-вычислительным машинам (ПЭВМ) и организации работы. В них указывается, что в случаях, когда характер работы требует постоянного взаимодействия с видеодисплейными терминалами,

рекомендуется организация перерывов на 10-15 мин. через каждые 45-60 мин. работы.

Согласно ФЗ «Об обязательном социальном страховании от несчастных случаев на производстве» установлены правовые основы обязательного социального страхования и определен порядок возмещения вреда, причиненного жизни и здоровью работника при исполнении им обязанностей по трудовому договору [16]. Обязательное социальное страхование предусматривает обеспечение социальной защиты застрахованных, в качестве которых выступают работники, возмещение вреда, причиненного жизни и здоровью застрахованного при исполнении им обязанностей, обеспечение предупредительных мер по сокращению производственного травматизма и профессиональных заболеваний. Средства на осуществление обязательного социального страхования формируются в основном за счет обязательных страховых взносов страхователей (работодателя).

### 3.2. Производственная безопасность

При разработке какой-либо программы разработчики подвергаются воздействию различных вредных и опасных факторов, которые представлены в таблице 1. В таблице также представлены соответствующие нормативные документы и этапы работ, во время которых разработчики могут столкнуться с их влиянием.

Таблица 3.1 – Возможные опасные и вредные факторы

Факторы (ГОСТ 12.0.003-2015)	Нормативные документы
Отклонение показателей микроклимата	СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания» [18]

Недостаточная освещённость рабочей зоны	СП 52.13330.2016 «Естественное и искусственное освещение» [19]
Повышенный уровень шума на рабочем месте	СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания» [20]
Повышенный уровень статического электричества	ГОСТ Р 53734.1-2014 «Электростатические явления» [21]
Опасность поражения электрическим током	ГОСТ Р 58698-2019 «Защита от поражения электрическим током» [22]

### 3.2.1. Отклонение показателей микроклимата

Отклонение показателей микроклимата на рабочем месте от комфортных непосредственно влияет на здоровье работников. Повышение скорости движения воздуха и понижение температуры может привести к переохлаждению организма путем усиления теплообмена и процесса теплоотдачи при испарении пота. Недостаточная влажность в свою очередь ведет к интенсивному испарению влаги со слизистых оболочек. Это может привести к пересыханию, растрескиванию и затем к заражению болезнетворными бактериями. При разработке программного обеспечения используются персональные компьютеры, которые могут непосредственно влиять на микроклимат путем снижения относительной влажности и повышению температуры в рабочем помещении.

Общие требования к микроклимату производственных помещений регламентируются СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания». Санитарные нормы регулируют оптимальные и допустимые значения показателей в рабочей зоне, соответствующие физиологическим потребностям организма человека, для создания комфортных и безопасных условий труда.

Работа, выполняемая командой разработки программного обеспечения, по энергозатратам относится к категории Ia (производится сидя, сопровождается незначительными физическими усилиями). В таблицах 2 и 3 представлены оптимальные и допустимые значения показателей микроклимата на рабочих местах для данной категории.

Таблица 3.2 – Оптимальные величины показателей микроклимата на рабочих местах

Период года	Температура воздуха, °С	Температура поверхностей, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с
Холодный	22-24	21-25	60-40	0,1
Тёплый	23-25	22-26	60-40	0,1

Таблица 3.3 – Допустимые величины показателей микроклимата на рабочих местах

Период года	Температура воздуха, °С		Температура поверхностей, °С	Относительная влажность воздуха, %	Скорость движения воздуха, м/с	
	диапазон ниже оптимальных величин	диапазон выше оптимальных величин			для диапазона температур воздуха ниже оптимальных величин, не более	для диапазона температур воздуха выше оптимальных величин, не более
Холодный	20,0 – 21,9	24,1 – 25,0	19,0 – 26,0	15 – 75	0,1	0,1
Тёплый	21,0 – 22,9	25,1 – 28,0	20,0 – 29,0	15 – 75	0,1	0,1

В производственных помещениях, где поддерживать допускаемые нормативные величины локального микроклимата не представляется возможным, необходимо проводить мероприятия по защите работников от возможного перегревания и охлаждения. Это достигается разными способами: использование систем местного кондиционирования воздуха; регламентацией периодов работы в неблагоприятном локальном микроклимате и отдыха в помещении с микроклиматом, нормализующим тепловое состояние; уменьшение длительности рабочей смены и др.

### 3.2.2 Недостаточная освещенность рабочей зоны

Недостаточная освещенность рабочей зоны является вредным производственным фактором, приводящим к повышенной утомляемости и



снижению работоспособности человека на предприятии. Продолжительная работа в условиях низкой освещенности приводит к ухудшению зрения.

Нормы естественного, искусственного и совместного освещения регламентируются СП 52.13330.2016 «Естественное и искусственное освещение». Разработка программного обеспечения относится к категории работ высокой точности – Б (наименьший или эквивалентный объект различения 0,30 – 0,50 мм), подразряд 1 (относительная продолжительность зрительной работы при направлении зрения на рабочую поверхность не менее 70%).

В таблице 3.4 представлены требования к освещению рабочего помещения для разряда Б1.

Таблица 3.4 – Требования к освещению рабочего помещения

Искусственное освещение				Естественное освещение	
Освещенность на рабочей поверхности от системы общего освещения, лк	Цилиндрическая освещенность	Объединенный показатель дискомфорта, не более	Коэффициент пульсации освещенности, Кп, %, не более	Коэффициент естественной освещенности, %, при	
				верхнем или комбинированном	боковом
300	100	21	15	3	1

Яркий свет в зоне периферийного зрения заметно увеличивает глазное напряжение. Для снижения влияния вредного фактора недостаточной освещенности необходимо, чтобы уровень естественного освещения рабочего пространства приблизительно совпадал с яркостью дисплея. Проблему недостаточной освещенности помещения можно решить при помощи установки дополнительных осветительных приборов, расширения световых проемов.

### 3.2.3. Повышенный уровень шума на рабочем месте

Превышение уровня шума на рабочем месте создает психологический и физический стресс, снижающий производительность, концентрацию, внимание, повышает утомляемость. Повышение уровня шума на рабочем месте возможно из-за фона, создаваемого работой персональных компьютеров, наличия центральной системы вентиляции и кондиционирования воздуха.

Предельно допустимые показатели уровня звука, звукового давления регламентируются СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания».

Для команды разработчиков программного обеспечения, эти показатели представлены в таблице 3.5.

Таблица 3.5 – Предельно допустимые уровни звукового давления, уровни звука и эквивалентные уровни звука для инженера-программиста

Вид трудовой деятельности, рабочее место	Уровни звукового давления, дБ в октавных полосах со среднегеометрическими частотами, Гц						
	31,5	63	125	250	500	1000	2000
Творческая деятельность, руководящая работа с повышенными требованиями, научная деятельность, конструирование и проектирование, программирование, преподавание и обучение, врачебная деятельность. Рабочие места в помещениях дирекции, проектноконструкторских бюро, расчетчиков, программистов вычислительных машин, в лабораториях для теоретических работ и обработки данных, приема больных в здравпунктах	86	71	61	54	49	45	42

Для снижения уровня шума в производственном помещении можно использовать защитные звукопоглощающие экраны. Для любого оборудования необходимо регулярно проводить техническое обслуживание, так как загрязнение может увеличить производимый шум.

### 3.2.4 Повышенный уровень статического электричества

Статическое электричество является опасным производственным фактором, проявление которого может нанести вред здоровью человека (ожоги) или привести к пожару и другим чрезвычайным ситуациям.

При работе за компьютером статический заряд может накапливаться, если нет хорошего контакта с землей или влажность/ионизация воздуха превышает допустимые нормы. Статический разряд в производственных помещениях рассматриваемого типа при условии соответствии нормам микроклимата и организации работ при воздействии на человека вызывает дискомфорт.

Допустимые показатели уровня статического электричества на производстве регламентируются ГОСТ Р 53734.1-2014 «Электростатические явления». В таблице 3.6 представлены уровни восприятия электростатического заряда человеком.

Таблица 3.6 – Уровни восприятия людьми электростатического заряда и ответной реакции при емкости тела в 200 пФ

Энергия разряда, мДж	Реакция	Потенциал тела, В
0,1	Разряд ощутим	1000
0,9	Четко ощутим	3000
6,4	Неприятный шок	8000

Для уменьшения накапливаемого статического заряда при работе за компьютером необходимо организовать антистатические рабочие места, соблюдать установленную норму влажности воздуха и поддерживать чистоту помещения, поскольку пыль обладает свойствами диэлектрика.

### 3.2.5 Опасность поражения электрическим током

Под электробезопасностью подразумевается система технических и организационных мероприятий, направленных на защиту людей от опасного воздействия электрического тока, статического электричества и электромагнитного поля. Значения вышеперечисленных факторов регулируются ГОСТ Р 58698-2019.

Таблица 3.7 – Пороги напряжения прикосновения для реагирования

Характер реагирования	Пороги напряжения, В
Реакция испуга	2 (переменный ток)
	8 (постоянный ток)
Мышечная реакция	20 (переменный ток)
	40 (постоянный ток)

Меры предосторожности для основной защиты от поражения электрическим током:

- использование защитных ограждений или оболочек;
- размещение опасных для жизни и здоровья человека участков электропроводов и приборов вне зоны досягаемости рукой;
- ограничение напряжения или питание должно осуществляться от безопасного источника питания;
- автоматическое отключение питания (защитное устройство, которое будет отключать систему, питающую электрическое оборудование в случае замыкания)

Защита от поражения электрическим током может осуществляться посредством системы безопасного сверхнизкого напряжения (БСНН) и защитного сверхнизкого напряжения (ЗСНН).

### 3.3 Экологическая безопасность

Программное обеспечение не оказывает влияния на окружающую среду, так как его разработка и использование происходит при помощи персональных компьютеров, однако использование самого компьютера может оказывать влияние на окружающую среду. Так в случаях нагрева материнской платы и корпуса монитора происходит выброс в воздух вредных веществ.

В производстве компьютеров и их комплектующих используются материалы, которые при неправильной утилизации компьютерной техники могут стать причиной загрязнения литосферы. Утилизировать компьютер необходимо после извлечения его компонент, их сортировки и отправки на повторное использование,

это необходимо делать на специально отведённых полигонах с присутствием квалифицированного персонала. Стоит также учитывать, что в технологических процессах производства компьютеров и их комплектующих образуются производственные сточные воды, которые могут являться фактором загрязнения гидросферы.

Соблюдение всех норм при использовании и утилизации компьютерной техники позволяет уменьшить вредное воздействие на окружающую среду.

Основной проблемой охраны окружающей среды в компьютерных лабораториях является утилизация люминесцентных ламп. Все люминесцентные лампы содержат ртуть (в дозах от 1 до 70 мг), ядовитое вещество 1-го класса опасности. Такая доза может причинить вред здоровью при повреждении лампы. Хранение и удаление отходов (в данном случае - люминесцентных ламп) осуществляются в соответствии с требованиями Постановления Правительства Российской Федерации от 28.12.2020 № 2314 «Об утверждении Правил обращения с отходами производства и потребления в части осветительных устройств, электрических ламп, ненадлежащие сбор, накопление, использование, обезвреживание, транспортирование и размещение которых может повлечь причинение вреда жизни, здоровью граждан, вреда животным, растениям и окружающей среде» [23]. Наполненную тару с отходами закрывают герметически стальной крышкой, при необходимости заваривают и передают по договору специализированным предприятиям, имеющим лицензию на их утилизацию.

### **3.4 Безопасность в чрезвычайных ситуациях**

#### **3.4.1 Пожар**

Причинами возникновения пожара при работе с ЭВМ может служить короткое замыкание проводки, в том числе в следствии неисправности прибора, сильный перегрев ЭВМ в результате его использования в режиме повышенной нагрузки.

Для предотвращения возникновения пожара, необходимо проводить периодическую своевременную диагностику оборудования и электрической

проводки, соблюдать нормы при работе с ЭВМ, обеспечить наличие средств пожаротушения в рабочем помещении, готовых к эксплуатации. Здание, в котором находится помещение с ЭВМ, тоже должно отвечать требованиям пожарной безопасности, для этого необходимо наличие охранно-пожарной сигнализации, плана эвакуации, углекислотных огнетушителей с проверенным клеймом, табличек с указанием направления к эвакуационному выходу. При появлении возгорания необходимо сообщить в службу пожарной охраны адрес и место возникновения пожара.

По СП 12.13130.2009 [24] рабочее место разработчика относится к категории «В», (пожароопасные), потому что в данном помещении присутствует пыль, вещества и материалы, способные при взаимодействии с воздухом гореть.

В здании, в котором проводится расположено рабочее место имеются такие средства пожаротушения, как огнетушители, существует противопожарная сигнализация, датчики дыма, имеются ответственные за противопожарную безопасность люди, имеется план эвакуации.

### **Выводы по разделу**

В данном разделе были рассмотрены правовые и организационные вопросы безопасности при работе и компоновке рабочего места.

Рабочее место, удовлетворяет всем требованиям безопасности. Требуемое освещение обеспечивается за счет нескольких энергосберегающих ламп. Уровень шума находится в допустимом диапазоне. Микроклиматические условия соблюдаются за счет системы отопления в холодное время и проветривания в теплое время. Для предотвращения влияния монотонности производиться перерывы и легкая разминка. Защита от воздействия электрического тока обеспечивается путем проверки состояния ПК и соблюдения правил безопасности при работе с ним. Помещение имеет все необходимые компоненты для обеспечения пожарной безопасности.

Согласно ПУЭ, рабочее место относится к 1 категории по электробезопасности: помещения без повышенной опасности.

Для работы в офисе персоналу присваивается 1 группа по электробезопасности.

Согласно СанПиН 1.2.3685-21 работа разработчика относится к категории 1б, так как разработчик проводит большую часть времени за компьютером, передвигается из кабинета в кабинет или по коридору.

Согласно СП 12.13130.2009 рабочее место разработчика относится к категории «В», (пожароопасные), так как в данном помещении присутствует вещества и материалы, способные при взаимодействии с воздухом гореть.

Согласно Постановлению от 31 декабря 2020 года №2398 [25] категория объекта, в данном случае ПЭВМ, оказывающего негативное воздействие на окружающую среду, является 4.

Также были затронуты темы экологической безопасности, уменьшение вредного влияния на окружающую среду и человека, и безопасности в чрезвычайных ситуациях, правил поведения людей в чрезвычайных ситуациях и превентивных мер по их предупреждению.

## **4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение**

### **Введение**

Основная цель данного раздела – оценить перспективность развития и планировать финансовую и коммерческую ценность конечного продукта, представленного в рамках исследовательской программы. Коммерческая ценность определяется не только наличием более высоких технических характеристик над конкурентными разработками, но и тем, насколько быстро разработчик сможет ответить на следующие вопросы – будет ли продукт востребован на рынке, какова будет его цена, каков бюджет научного исследования, какое время будет необходимо для продвижения разработанного продукта на рынок.

Данный раздел, предусматривает рассмотрение следующих задач:

- Оценка коммерческого потенциала разработки.
- Планирование научно-исследовательской работы;
- Расчет бюджета научно-исследовательской работы;
- Определение ресурсной, финансовой, бюджетной эффективности

исследования.

Цель данной НИР – создание нейросетевого алгоритма для распознавания многочленов. Созданный нейросетевой алгоритм сможет распознавать многочлены в файлах pdf.

### **4.1 Оценка коммерческого потенциала и перспективности проведения исследований с позиции ресурсоэффективности и ресурсосбережения**

#### **4.1.1 Анализ конкурентных технических решений**

В ходе исследования были рассмотрены два конкурирующих приложения, способных распознавать математические выражения:

- 1) Mathway — приложение для решения математических задач;



2) Photomath — мобильное приложение, использующее камеру телефона для распознавания математических уравнений и отображения пошагового решения на экране.

Детальный анализ необходим, т.к. каждое программное обеспечение имеет свои достоинства и недостатки. В таблице 4.1 показано сравнение разработок-конкурентов и разработки данной ВКР с точки зрения технических и экономических критериев оценки эффективности.

Таблица 4.1 – Сравнение конкурентных приложений

Критерии оценки	Вес критерия	Баллы			Конкурентоспособность		
		Б <sub>ф</sub>	Б <sub>к1</sub>	Б <sub>к2</sub>	К <sub>ф</sub>	К <sub>к1</sub>	К <sub>к2</sub>
1	2	3	4	5	6	7	8
<b>Технические критерии оценки ресурсоэффективности</b>							
1. Простота эксплуатации	0.12	3	4	4	0.36	0.48	0.48
2. Потребность в ресурсах памяти	0.06	3	4	3	0.18	0.24	0.18
3. Точность результатов	0.15	4	2	3	0.6	0.3	0.45
4. Предоставляемые возможности	0.14	3	4	4	0.42	0.56	0.49
5. Скорость работы	0.05	4	4	3	0.2	0.2	0.15
<b>Экономические критерии оценки эффективности</b>							
1. Конкурентоспособность продукта	0,12	4	4	3	0.42	0.48	0.36
2. Уровень проникновения на рынок	0,11	4	5	4	0.44	0.495	0.385
3. Цена	0,04	5	5	5	0.18	0.18	0.18
4. Предполагаемый срок эксплуатации	0,09	4	3	3	0.36	0.225	0.27
5. Послепродажное обслуживание	0,08	4	4	4	0.28	0.32	0.28
6. Срок выхода на рынок	0,03	4	3	3	0.12	0.075	0.075
7. Наличие сертификации разработки	0,01	2	5	4	0.015	0.045	0.04
<b>Итого</b>	<b>1</b>	42	45	41	3.575	3.6	3.34

Расчет конкурентоспособности, на примере стабильности срабатывания, определяется по формуле:

$$K = \sum V_i * B_i = 3.575, \quad (4.1)$$

где K – конкурентоспособность проекта; V<sub>i</sub> – вес показателя (в долях единицы); B<sub>i</sub> – балл показателя.

Согласно проведенному анализу, можно сделать вывод, что разрабатываемый алгоритм примерна наравне со своими конкурентами.

Сильными сторонами нейросетевого алгоритма являются его простота, предполагаемый срок эксплуатации и точность результатов.

Слабыми сторонами является наличие сертификата и предоставляемые возможности приложения, так как нейросетевой алгоритм сможет лишь распознавать многочлены.

#### 4.1.2 SWOT-анализ

Для исследования внешней и внутренней среды проекта, в этой работе проведен SWOT-анализ с детальной оценкой сильных и слабых сторон исследовательского проекта, а также его возможностей и угроз.

Первый этап, составляется матрица SWOT, в которую описаны слабые и сильные стороны проекта и выявленные возможности и угрозы для реализации проекта, которые проявились или могут появиться в его внешней среде, приведены в таблице 4.2.

Таблица 4.2 – Матрица SWOT-анализа

<b>Сильные стороны</b>	<b>Слабые стороны</b>
С1. Возможность определять степень многочлена.	Сл1. Отсутствие ссылок и материалов для соответствующих научных исследований.
С2. Более свежая информация, которая была использована для разработки проекта.	Сл2. Вероятность получения неверных результатов работы.
С3. Низкая цена разработки.	Сл3. Маленькая область применения нейросети.
<b>Возможности</b>	<b>Угрозы</b>
В1. Появление потенциального спроса на новые разработки.	У1. Снижение стоимости разработок конкурентов.
В2. Дальнейшее улучшение нейросети, увеличение возможностей.	У2. Появление зарубежных аналогов и более ранний их выход на рынок.

На втором этапе на основании матрицы SWOT строятся интерактивные матрицы возможностей и угроз, позволяющие оценить эффективность проекта, а также надежность его реализации. Соотношения параметров представлены в таблицах 4.3–4.6.

Таблица 4.3 – Интерактивная матрица проекта «Возможности проекта и сильные стороны»

<b>Сильные стороны проекта</b>				
<b>Возможности проекта</b>		C1	C2	C3
	B1	-	+	-
	B2	+	+	-

Таблица 4.4 – Интерактивная матрица проекта «Возможности проекта и слабые стороны»

<b>Слабые стороны проекта</b>				
<b>Возможности проекта</b>		Сл1	Сл2	Сл3
	B1	-	-	-
	B2	-	+	+

Таблица 4.5 – Интерактивная матрица проекта «Угрозы проекта и сильные стороны»

<b>Сильные стороны проекта</b>				
<b>Угрозы проекта</b>		C1	C2	C3
	У1	-	-	+
	У2	-	-	-

Таблица 4.6 – Интерактивная матрица проекта «Угрозы проекта и слабые стороны»

<b>Слабые стороны проекта</b>				
<b>Угрозы проекта</b>		Сл1	Сл2	Сл3
	У1	-	-	-
	У2	-	+	+

Результаты анализа представлены в итоговую таблицу 4.7.

Таблица 4.7 – Итоговая таблица SWOT-анализа

	<b>Сильные стороны научно-исследовательского проекта</b> С1. Возможность определять степень многочлена. С2. Более свежая информация, которая была использована для разработки проекта. С3. Низкая цена разработки.	<b>Слабые стороны научно-исследовательского проекта</b> Сл1. Отсутствие ссылок и материалов для соответствующих научных исследований. Сл2. Вероятность получения неверных результатов работы. Сл3. Маленькая область применения нейросети.
<b>Возможности</b> В1. Появление потенциального спроса на новые разработки. В2. Дальнейшее улучшение нейросети, увеличение возможностей.	<b>Направления развития</b> В1С2. Благодаря используемой свежей информации появится спрос на новые разработки в этой области. В2С1С2. Увеличение возможностей нейросети за счёт её улучшения и дальнейшего обучения, основанного на свежей информации.	<b>Сдерживающие факторы</b> В2Сл2Сл3. Возможные осложнения с улучшением нейросети из-за неверных результатов и небольшой области применения этой нейросети.
<b>Угрозы</b> У1. Снижение стоимости разработок конкурентов. У2. Появление зарубежных аналогов и более ранний их выход на рынок.	<b>Угрозы развития</b> У1С3. Зарубежные аналоги будут пользоваться большим спросом не зависимо от цены разработки.	<b>Уязвимости:</b> У2Сл2Сл3. Зарубежные аналоги могут давать более точные результаты и иметь более обширную область применения

В результате SWOT-анализа показано, что на преимущества и недостатки присутствуют в равной мере. Данные недостатки, которые на данный момент на практике не устранены, но в теории уже есть возможности для их устранения. Результаты анализа учтены в дальнейшей научно-исследовательской разработке.

## 4.2. Планирование научно-исследовательских работ

Планирование комплекса научно-исследовательских работ осуществляется в порядке:

- определение структуры работ в рамках научного исследования;
- определение количества исполнителей для каждой из работ;
- установление продолжительности работ;
- построение графика проведения научных исследований.

Для оптимизации работ удобно использовать классический метод линейного планирования и управления.

Результатом такого планирования является составление линейного графика выполнения всех работ. Порядок этапов работ и распределение исполнителей для данной научно-исследовательской работы, приведен в таблице 4.8.

Таблица 4.8 – Перечень этапов, работ и распределение исполнителей

Основные этапы	№ раб	Содержание работ	Должность исполнителя
Разработка технического задания	1	Составление и утверждение технического задания, утверждение плана-графика	Научный руководитель
	2	Календарное планирование выполнения работ	Инженер, научный руководитель
Выбор способа решения поставленной задачи	3	Обзор научной литературы	Инженер
	4	Выбор методов исследования	Инженер
Теоретические и экспериментальные исследования	5	Планирование эксперимента	Инженер, научный руководитель
	6	Подготовка образцов для эксперимента	Инженер
	7	Проведение эксперимента	Инженер
Обобщение и оценка результатов	8	Обработка полученных данных	Инженер
	9	Оценка правильности полученных результатов	Инженер, Научный руководитель
Оформление отчета по НИР	10	Составление пояснительной записки	Инженер

#### 4.2.2 Определение трудоемкости выполнения работ и разработка графика проведения

При проведении научных исследований основную часть стоимости разработки составляют трудовые затраты, поэтому определение трудоемкости проводимых работ является важным этапом составления сметы.

Для определения ожидаемого (среднего) значения трудоемкости использована следующая формула:

$$t_{ож\ i} = \frac{3t_{min\ i} + 2t_{max\ i}}{5}, \quad (4.1)$$

где  $t_{ож\ i}$  – ожидаемая трудоемкость выполнения  $i$ -ой работы, человеко-дни;

$t_{min\ i}$  – минимально возможная трудоемкость выполнения заданной  $i$ -ой работы, человеко-дни;

$t_{max i}$  – максимально возможная трудоемкость выполнения заданной  $i$ -ой работы, человеко-дни.

Зная величину ожидаемой трудоемкости, можно определить продолжительность каждой  $i$ -ой работы в рабочих днях  $T_{pi}$ , при этом учитывается параллельность выполнения работ разными исполнителями. Данный расчёт позволяет определить величину заработной платы.

$$T_{pi} = \frac{t_{ожi}}{Ч_i}, \quad (4.2)$$

где  $T_{pi}$  – продолжительность одной работы, рабочие дни;

$t_{ожi}$  – ожидаемая трудоемкость выполнения одной работы, человеко-дни;

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

Для перевода длительности каждого этапа из рабочих в календарные дни, необходимо воспользоваться формулой (4.3):

$$T_{ki \text{ мнж}} = T_{pi} * k_{\text{кал}}, \quad (4.3)$$

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

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

$k_{\text{кал}}$  – календарный коэффициент.

Календарный коэффициент определяется по формуле:

$$k_{\text{кал мнж}} = \frac{T_{\text{кал}}}{T_{\text{кал}} - T_{\text{вых}} - T_{\text{пр}}} = \frac{365}{365 - 104 - 14} = 1.48, \quad (4.4)$$

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

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

$T_{np}$  – общее количество праздничных дней в году (2023 год).

Расчеты временных показателей проведения научного исследования обобщены в таблице 4.9.

Таблица 4.9 – Временные показатели проведения научного исследования

Название работы	Трудоёмкость работ						Длительность работ в рабочих днях $T_{pi}$	Длительность работ в календарных днях $T_{ki}$
	$t_{min}$ , чел-дни		$t_{max}$ , чел-дни		$t_{ож}$ , чел-дни			
	Исп.1	Исп.2	Исп.1	Исп.2	Исп.1	Исп.2		
1	2	3	4	5	6	7	8	9
1. Составление и утверждение технического задания, утверждение плана-графика	2	-	4	-	2,8	-	2,8	4
2. Календарное планирование выполнения работ	1	3	3	4	1,8	3,4	2,6	4
3. Обзор научной литературы	-	6	-	10	-	7,6	7,6	11
4. Выбор методов исследования	-	3	-	5	-	3,8	3,8	6
5. Планирование эксперимента	2	6	4	8	2,8	6,8	4,8	7
6. Подготовка образцов для эксперимента	-	5	-	7	-	5,8	5,8	9
7. Проведение эксперимента	-	15	-	20	-	17	17	25
8. Обработка полученных данных	-	10	-	15	-	12	12	18
9. Оценка правильности полученных результатов	2	3	4	5	2,8	3,8	3,3	5
10. Составление пояснительной записки		8		10	-	8,8	8,8	13
<b>Итого:</b>	7	59	15	84	10,2	69	68,5	102

Примечание: Исп. 1 – научный руководитель, Исп. 2 – инженер.

На основе таблицы составлен календарный план-график выполнения проекта с использованием диаграммы Ганта (таблица 4.10).

Таблица 4.10 – Диаграмма Ганта

№	Вид работ	Исп	$T_{ki}$ , кал. дн.	Продолжительность работ												
				февр			март			апр			май			
				1	2	3	1	2	3	1	2	3	1	2	3	
1	Составление и утверждение технического задания, утверждение плана-графика	Исп1	4	■												
2	Календарное планирование выполнения ВКР	Исп1 Исп2	4	■												
3	Обзор научной литературы	Исп2	11		■											
4	Выбор методов исследования	Исп2	6			■										
5	Планирование эксперимента	Исп1 Исп2	7			■										
6	Подготовка образцов для эксперимента	Исп2	9				■									
7	Проведение эксперимента	Исп2	25					■								
8	Обработка полученных данных	Исп2	18							■						
9	Оценка правильности полученных результатов	Исп1 Исп2	5									■				
10	Составление пояснительной записки	Исп2	13											■		

Примечание:



– Исп. 1 (научный руководитель), ■ – Исп. 2 (инженер)

### 4.3 Бюджет научно-технического исследования

При планировании бюджета научно-технического исследования учитывались все виды расходов, связанных с его выполнением. В этой работе использовать следующую группировку затрат по следующим статьям:

- материальные затраты выпускной квалификационной работы (ВКР);
- основная заработная плата исполнителей темы;
- дополнительная заработная плата исполнителей темы;
- отчисления во внебюджетные фонды (страховые отчисления);
- накладные расходы ВКР.



#### 4.3.1 Расчет материальных затрат научно-технического исследования

Материальные затраты — это затраты организации на приобретение сырья и материалов для создания готовой продукции.

Так как данное исследование связано с теоретическими разработками, то в качестве материальных затрат будут взяты следующие материалы:

Таблица 4.11 – материальные затраты

Наименование материалов	Цена за ед., руб.	Кол-во, ед.	Сумма, руб.
Комплекс канцелярских принадлежностей	340	4	1 200
Картридж для лазерного принтера	3 490	1	3 490
Итого:			8 290

Цены приняты на основании прайс-листа поставщика материалов:  
<https://papirus-tomsk.ru>, <https://mvideo.ru>.

#### 4.3.2 Расчет амортизации специального оборудования

Расчет сводится к определению амортизационных отчислений, так как оборудование было приобретено до начала выполнения данной работы и эксплуатировалось ранее, поэтому при расчете затрат на оборудовании учитываем только рабочие дни по данной теме.

Расчет амортизации проводится следующим образом:

Норма амортизации: рассчитывается по формуле:

$$H_A = \frac{1}{n}, \quad (4.5)$$

где  $n$  - срок полезного использования в количестве лет.

Амортизация оборудования рассчитывается по формуле:

$$A = \frac{N_A * I}{12} * m, \quad (4.6)$$

где  $B$  – итоговая сумма, тыс. руб.;  $m$ – время использования, мес.

При выполнении научно-исследовательского проекта использовался ПЭВМ - Asus. Срок полезного использования данного ноутбука по паспорту составляет 3 года.

Таблица 4.12 – Затраты на оборудование

№	Наименование оборудования	Кол-во, шт.	Срок полезного использования, лет	Цены единицы оборудования, тыс. руб.	Общая стоимость оборудования, тыс. руб.
1	ПЭВМ	1	3	30	30
<b>Итого</b>		30 тыс. руб.			

Рассчитаем норму амортизации для ноутбука, с учётом того, что срок полезного использования составляет 3 года:

$$N_A = \frac{1}{3} = 0,33.$$

Общую сумму амортизационных отчислений находим следующим образом:

$$A = \frac{0,33 * 30000}{12} * 3 = 2475 \text{ руб.}$$

#### 4.3.3 Основная заработная плата исполнителей темы

В данном разделе рассчитывается заработная плата инженера и руководителя, помимо этого необходимо рассчитать расходы по заработной плате, определяемые трудоемкостью проекта и действующей системой оклада.

Основная заработная плата  $Z_{\text{осн}}$  одного работника рассчитывается по следующей формуле:

$$Z_{\text{осн}} = Z_{\text{дн}} * T_p, \quad (4.7)$$

где  $Z_{\text{дн}}$  – среднедневная заработная плата, руб.;  $T_p$  – продолжительность работ, выполняемых работником, раб.дн. (таблица 4.9).

Среднедневная заработная плата рассчитывается по формуле:

Для шестидневной рабочей недели (рабочая неделя руководителя):

$$Z_{\text{дн}} = \frac{Z_m * M}{F_d} = \frac{51285 * 10.3}{246} = 2147.3 \text{ руб}, \quad (4.8)$$

где  $Z_m$  – месячный должностной оклад работника, руб.;

$F_d$  – действительный годовой фонд рабочего времени научно-технического персонала, раб. дней;

$M$  – количество месяцев работы без отпуска в течение года:

– при отпуске в 28 раб. дня –  $M = 11,2$  месяца, 5-дневная рабочая неделя;

– при отпуске в 56 раб. дней –  $M = 10,3$  месяца, 6-дневная рабочая неделя.

Для пятидневной рабочей недели (рабочая неделя инженера):

$$Z_{\text{дн}} = \frac{Z_m * M}{F_d} = \frac{33150 * 11.2}{213} = 1743.1 \text{ руб}. \quad (4.9)$$

Должностной оклад работника за месяц:

– для руководителя:

$$Z_m = Z_{\text{мс}} * (1 + k_{np} + k_d) * k_p = 26300 * (1 + 0.3 + 0.2) * 1,3 = 51285 \text{ руб}, \quad (4.10)$$

где  $Z_{мс}$  – заработная плата, согласно тарифной ставке, руб.;

$k_{np}$  – премиальный коэффициент, равен 0,3;

$k_d$  – коэффициент доплат и надбавок, равен 0,2;

$k_p$  – районный коэффициент, равен 1,3 (для г. Томска).

– для инженера:

$$Z_m = Z_{мс} * (1 + k_{np} + k_d) * k_p = 17000 * (1 + 0.3 + 0.2) * 1,3 = 33150 \text{ руб.} \quad (4.11)$$

Таблица 4.13 – Баланс рабочего времени исполнителей

Показатели рабочего времени	Руководитель	Инженер
Календарное число дней	365	365
Количество нерабочих дней	52/14	104/14
- выходные дни		
- праздничные дни		
Потери рабочего времени	48/5	24/10
- отпуск		
- невыходы по болезни		
Действительный годовой фонд рабочего времени	246	213

Таблица 4.14 – Расчет основной заработной платы исполнителей

Исполнители НИ	$Z_{мс}$ , руб	$k_{np}$	$k_d$	$k_p$	$Z_m$ , руб	$Z_{дн}$ , руб	$T_p$ , раб.дн.	$Z_{осн}$ , руб
Руководитель	26300	0,3	0,2	1,3	51285	2147,3	13,5	28988,6
Инженер	17000	0,3	0,2	1,3	33150	1743,1	68,5	119402,4
Итого:								148391

Дополнительная заработная плата определяется по формуле:

– для руководителя:

$$Z_{доп} = k_{доп} * Z_{осн} = 0.15 * 28988.6 = 4348.3 \text{ руб,} \quad (4.12)$$

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

– для инженера:

$$Z_{\text{доп}} = k_{\text{доп}} * Z_{\text{осн}} = 0.15 * 119402.4 = 17910.4 \text{ руб.} \quad (4.13)$$

#### 4.3.4 Отчисления во внебюджетные фонды (страховые отчисления)

Отчисления во внебюджетные фонды определяется по формуле:

– для руководителя:

$$Z_{\text{внеб}} = k_{\text{внеб}} * (Z_{\text{осн}} + Z_{\text{доп}}) = 0.3 * (28988.6 + 4348.3) = 10001.1 \text{ руб.} \quad (4.14)$$

где  $k_{\text{внеб}}$  – коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд ОМС и социальное страхование). Общая ставка взносов составляет в 2023 году – 30% (ст. 425, 426 НК РФ).

– для инженера:

$$Z_{\text{внеб}} = k_{\text{внеб}} * (Z_{\text{осн}} + Z_{\text{доп}}) = 0.3 * (119402.4 + 17910.4) = 41193.8 \text{ руб.} \quad (4.15)$$

#### 4.3.5 Накладные расходы

Накладные расходы включают в себя следующие расходы: печать ксерокопирование материалов исследования, оплата услуг связи и т.д. Сумма 5 статьи затрат, рассчитанных выше, приведена в таблице ниже и используются для расчета накладных расходов.

Таблица 4.15 – Группировка затрат по статьям

Статьи					
1	2	3	4	5	6
Амортизация	Сырье, материалы	Основная заработная плата	Дополнительная заработная плата	Отчисления на социальные нужды	Итого без накладных расходов
2475	8 290	148391	22258,7	51194,9	232609,6

Величина накладных расходов определяется по формуле:

$$Z_{\text{накл}} = (\text{сумма статей}) * k_{np} = (232609.6) * 0,2 = 46521.9 \text{ руб, (4.16)}$$

где  $k_{np}$  – коэффициент, учитывающий накладные расходы. Величина коэффициента принимается равной 0,2.

#### 4.3.6 Бюджет НИР

На основании полученных данных по отдельным статьям затрат составляется калькуляция плановой себестоимости ВКР «Обнаружение и распознавание многочленов не выше четвертой степени с помощью нейросетевого алгоритма» по форме, приведенной в таблице 4.16. В таблице также представлено определение бюджета затрат двух конкурирующих проектов. Различия будут отражены в ЗП сотрудников и количестве оборудования (будет взято большее количество ноутбуков). Над Photomath работало 4 инженера, над Mathway – 3. Средняя заработная плата программиста была взята из источника: <https://www.avito.ru/tomsk/vakansii?cd=1&q=программист>.

Таблица 4.16 – Группировка затрат по статьям

№	Наименование статьи	Сумма, руб.			Примечание
		Текущий Проект	Mathway	Photomath	
1	Материальные затраты НИР	8290	8290	8290	Пункт 4.2.3.1
2	Затраты на специальное оборудование	30000	90000	120000	Пункт 4.2.3.2
3	Затраты по основной заработной плате исполнителей темы	148391	409779	546133	Пункт 4.2.3.3
4	Затраты по дополнительной заработной плате исполнителей темы	22258,7	61466,9	81920,1	Пункт 4.2.3.3

5	Отчисления во внебюджетные фонды	51194,9	141373,8	188415,9	Пункт 4.2.3.4
6	Накладные расходы	46521,9	128469,4	171217,6	Пункт 4.2.3.5
Бюджет затрат НИР		306656,5	839379,1	1115976,6	Сумма ст. 1- 6

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

Для определения эффективности исследования рассчитан интегральный показатель эффективности научного исследования путем определения интегральных показателей финансовой эффективности и ресурсоэффективности.

**Интегральный показатель финансовой эффективности** научного исследования получен в процессе оценки бюджета затрат трех программных обеспечений. Для этого наибольший интегральный показатель реализации технической задачи принят за базу расчета (как знаменатель), с которым соотносятся финансовые значения по всем программными обеспечениями.

В качестве аналогов данной ВКР рассмотрены:

- 1) Mathway — приложение для решения математических задач;
- 2) Photomath — мобильное приложение, использующее камеру телефона для распознавания математических уравнений и отображения пошагового решения на экране.

Интегральный финансовый показатель разработки рассчитывается как:

$$I_{\text{финр}}^{\text{исп.}i} = \frac{\Phi_{pi}}{\Phi_{\text{max}}}, \quad (4.17)$$

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

$\Phi_{pi}$  — стоимость  $i$ -го варианта исполнения;

$\Phi_{\text{max}}$  — максимальная стоимость исполнения.

$$\Phi_{\text{тек.пр.}} = 306656,5; \Phi_{\text{Mathway}} = 839379,1; \Phi_{\text{Photomath}} = 1115976,6.$$

$$I_{\text{финр}}^{\text{тек.пр.}} = \frac{306656,5}{1115976,6} = 0,27;$$

$$I_{\text{финр}}^{\text{Mathway}} = \frac{839379,1}{1115976,6} = 0,75;$$

$$I_{\text{финр}}^{\text{Photomath}} = \frac{1115976,6}{1115976,6} = 1.$$

В результате расчета консолидированных финансовых показателей по трем вариантам разработки вариант 1 (текущий проект) с меньшим перевесом признан считается более приемлемым с точки зрения финансовой эффективности.

**Интегральный показатель ресурсоэффективности** вариантов выполнения ВКР ( $I_{pi}$ ) определен путем сравнительной оценки их характеристик, распределенных с учетом весового коэффициента каждого параметра (таблица 4.17).

Таблица 4.17 – Сравнительная оценка характеристик конкурентов ВКР

Объект исследования Критерии	Весовой коэффициент параметра	Текущий проект	Mathway	Photomath
1. Простота эксплуатации	0,2	7	9	8
2. Потребность в ресурсах памяти	0,12	5	6	5
3. Точность результатов	0,3	8	5	6
4. Предоставляемые возможности	0,28	6	8	7
5. Скорость работы	0,1	8	8	6
ИТОГО	1	6,88	7,06	6,56

Расчет интегрального показателя для разрабатываемого проекта:

$$I_{\text{тек.пр.}} = 0,2 * 7 + 0,12 * 5 + 0,3 * 8 + 0,28 * 6 + 0,1 * 8 = 6,88;$$

$$I_{\text{Mathway}} = 0,2 * 9 + 0,12 * 6 + 0,3 * 5 + 0,28 * 8 + 0,1 * 8 = 7,06;$$



$$I_{Photomath} = 0,2 * 8 + 0,12 * 5 + 0,3 * 6 + 0,28 * 7 + 0,1 * 6 = 6,56.$$

Интегральный показатель эффективности вариантов исполнения разработки вычисляется на основании показателя ресурсоэффективности и интегрального финансового показателя по формуле:

$$I_{исп.i} = \frac{I_{р-исп.i}}{I_{финр.i}}. \quad (4.18)$$

$$I_{тек.пр.} = \frac{6,88}{0,27} = 25,5;$$

$$I_{Mathway.} = \frac{7,06}{0,75} = 9,41;$$

$$I_{Photomath} = \frac{6,56}{1} = 6,56.$$

Далее интегральные показатели эффективности каждого конкурента ВКР сравнивались с интегральными показателями эффективности других конкурентов с целью определения сравнительной эффективности проекта (таблица 4.18).

Сравнительная эффективность проекта ( $\mathcal{E}_{ср}$ ):

$$\mathcal{E}_{ср} = \frac{I_{исп.i}}{I_{исп.тек.пр.}}. \quad (4.19)$$

Таблица 4.18 – Сравнительная эффективность разработки

№ п/п	Показатели	Текущий проект	Mathway	Photomath
1	Интегральный финансовый показатель разработки	0,27	0,75	1
2	Интегральный показатель ресурсоэффективности разработки	6,88	7,06	6,56
3	Интегральный показатель эффективности	25,5	9,41	6,56
4	Сравнительная эффективность вариантов исполнения	1	0,37	0,26

Сравнение среднего интегрального показателя сопоставляемых конкурентов позволило сделать вывод о том, что наиболее финансово- и ресурсоэффективным является текущий проект. Наш проект является более эффективным по сравнению с конкурентами.

### **Выводы по разделу**

В результате выполнения целей раздела можно сделать следующие выводы:

1. Результатом анализа конкурентных технических решений является выявление сильных и слабых сторон программного обеспечения относительно конкурентов.

2. В ходе планирования для руководителя и инженера был разработан график реализации этапа работ, который позволяет оценивать и планировать рабочее время исполнителей. Определено следующее: общее количество календарных дней для выполнения работ составляет 102 дней; общее количество дней, в течение которых работал инженер, составляет 98 дней; общее количество дней, в течение которых работал руководитель, составляет 20 дней;

3. Для оценки затрат на реализацию проекта разработан проектный бюджет, который составляет 306656,5 руб;

4. Результат оценки эффективности ИР показывает следующие выводы:

1) значение интегрального финансового показателя ИР составляет 0,27, что является показателем того, что ИР является финансово выгодной по сравнению с аналогами;

2) значение интегрального показателя ресурсоэффективности ИР составляет 6,88, по сравнению с 7,06 и 6,56;

3) значение интегрального показателя эффективности ИР составляет 25,5, по сравнению с 9,41 и 6,56, и является наиболее высоким, что означает, что техническое решение, рассматриваемое в ИР, является наиболее эффективным вариантом исполнения.

## **Заключение**

В результате проделанной работы были сделаны следующие основные выводы:

1. Построен нейросетевой алгоритм для распознавания многочленов. Алгоритм использует архитектуру сверточной нейронной сети YOLO.
2. Полученное после 1800 итераций значение средней ошибки равно 0,3131.
3. В результате модель каждый раз отбирает необходимые многочлены до 4-го порядка, но при этом имеются ошибки при распознавании некоторых символов или многочленов выше 4-го порядка, которые некорректно отображаются в результате работы нейросети.

## Список литературы

1. Нейронные сети: как работают и где используются [Электронный ресурс] // GeekBrains – образовательный портал. 2022. URL: <https://gb.ru/blog/nejronnye-seti/> (дата обращения: 25.12.2022).
2. Нейронные сети [Электронный ресурс] // Большая российская энциклопедия. 2005-2019. URL: [https://bigenc.ru/technology\\_and\\_technique/text/4114009](https://bigenc.ru/technology_and_technique/text/4114009) (дата обращения: 25.12.2022).
3. Нейронные сети – простыми словами [Электронный ресурс] // future2day.ru. 2022. URL: <https://future2day.ru/nejronnye-seti/> (дата обращения: 25.12.2022).
4. Обучение нейронной сети [Электронный ресурс] // Neuronus.com. 2013-2018. URL: <https://neuronus.com/theory/nn/238-obucheniya-nejronnoi-seti.html> (дата обращения: 26.12.2022).
5. Python. Чтение содержимого PDF с помощью OCP (оптическое распознавание символов) [Электронный ресурс] // GeeksforGeeks. 2022. URL: <https://www.geeksforgeeks.org/python-reading-contents-of-pdf-using-ocr-optical-character-recognition/> (дата обращения: 26.12.2022).
6. Машинное обучение [Электронный ресурс] // Продвинутое программирование на Python МФТИ. 2022. URL: [http://cs.mipt.ru/advanced\\_python/lessons/lab25.html](http://cs.mipt.ru/advanced_python/lessons/lab25.html) (дата обращения: 27.12.2022).
7. Функция активации нейросети [Электронный ресурс] // Neurohive. 2022. URL: <https://neurohive.io/ru/osnovy-data-science/activation-functions/> (дата обращения: 27.12.2022).
8. Tesseract OCR in Python with Pytesseract & OpenCV [Электронный ресурс] // Nanonets. 2022. URL: <https://nanonets.com/blog/ocr-with-tesseract/> (дата обращения: 27.12.2022).
9. Почему Python – лучший язык для машинного обучения и ИИ? [Электронный ресурс] // Pythonist. 2020. URL: <https://pythonist.ru/pochemu-python-luchshij-yazyk-dlya-mashinnogo-obucheniya-i-ii/> (дата обращения: 27.12.2022).

10. Свёрточные нейронные сети для распознавания образов [Электронный ресурс] // itnan.ru. 2022. URL: <https://itnan.ru/post.php?c=1&p=456186> (дата обращения: 27.12.2022).
11. YOLO Explained [Электронный ресурс] // Medium. 2023. URL: <https://medium.com/analytics-vidhya/yolo-explained-5b6f4564f31> (дата обращения: 15.03.2023).
12. YOLO-v4 Object Detector [Электронный ресурс] // reckoning.dev. 2023. URL: <https://reckoning.dev/blog/yolo-v4/> (дата обращения: 15.03.2023).
13. Как обнаруживать объекты, используя YOLO, OpenCV и PyTorch [Электронный ресурс] // Записки преподавателя. 2012-2023. URL: <https://waksoft.susu.ru/2021/05/19/kak-vypolnit-obnaruzhenie-obektov-yolo-s-pomoshhyu-opencv-i-pytorch-v-python/> (дата обращения: 15.03.2023).
14. Как работает Object Tracking на YOLO и DeepSort [Электронный ресурс] // Хабр. 2006-2023. URL: <https://habr.com/ru/articles/514450/> (дата обращения: 15.03.2023).
15. СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организации воспитания и обучения, отдыха и оздоровления детей и молодежи» // Гарант – информационный портал [Электронный ресурс]. – Режим доступа: <https://www.garant.ru/products/ipo/prime/doc/74993644/> (дата обращения: 20.04.2023).
16. ФЗ «Об обязательном социальном страховании от несчастных случаев на производстве» // Электронный фонд правовой и нормативно-технической документации [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/901713539> (дата обращения: 20.04.2023).
17. Трудовой кодекс (ТК РФ) «Рабочее время» // Электронный фонд правовой и нормативно-технической документации [Электронный ресурс]. – Режим доступа: <http://base.garant.ru/12125268/> (дата обращения: 20.04.2023).
18. СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания» // Электронный фонд правовой и нормативно-технической документации

[Электронный ресурс]. 2021. – Режим доступа: <https://docs.cntd.ru/document/573500115> (дата обращения: 20.04.2023).

19. СП 52.13330.2016 «Естественное и искусственное освещение» // Электронный фонд правовой и нормативно-технической документации [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/456054197> (дата обращения: 20.04.2023).

20. СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания» // Электронный фонд правовой и нормативно-технической документации [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/573500115> (дата обращения: 20.04.2023).

21. ГОСТ Р 53734.1-2014 «Электростатические явления» // Электронный фонд правовой и нормативно-технической документации [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/1200111323> (дата обращения: 20.04.2023).

22. ГОСТ Р 58698-2019 «Защита от поражения электрическим током» // Электронный фонд правовой и нормативнотехнической документации [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/1200170001> (дата обращения: 20.04.2023).

23. Постановление Правительства Российской Федерации от 28.12.2020 № 2314 «Об утверждении Правил обращения с отходами производства и потребления в части осветительных устройств, электрических ламп, ненадлежащие сбор, накопление, использование, обезвреживание, транспортирование и размещение которых может повлечь причинение вреда жизни, здоровью граждан, вреда животным, растениям и окружающей среде» [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/573275660> (дата обращения: 20.04.2023).

24. СП 12.13130.2009 «Системы противопожарной защиты. Эвакуационные пути и выходы» [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/1200071156> (дата обращения: 20.04.2023).

25. Постановлению от 31 декабря 2020 года №2398 «Об утверждении критериев отнесения объектов, оказывающих негативное воздействие на окружающую среду, к объектам I, II, III и IV категорий» [Электронный ресурс]. – Режим доступа: <https://docs.cntd.ru/document/573292854> (дата обращения: 20.04.2023).

## Приложение А

```
!pip install q keras==1.2.2
pip install opencv-python
pip install pdf2image
pip install pix2tex
!apt-get install poppler-utils
import PIL.Image
if not hasattr(PIL.Image, 'Resampling'):
    PIL.Image.Resampling = PIL.Image
import re
from cnn_class import Cnn
from PIL import Image
from pdf2image import convert_from_path
from pix2tex.cli import LatexOCR
import numpy as np
import cv2

cnn = Cnn('yolov4-tiny-obj_best.weights', 'yolov4-tiny-obj.cfg', 0.5, 0.5, 640)
filePath = 'Polynoms.pdf'
doc = convert_from_path(filePath)

def resize_im(img):
    scale_percent = 50
    width = int(img.shape[1] * scale_percent / 100)
    height = int(img.shape[0] * scale_percent / 100)
    dim = (width, height)
    resized = cv2.resize(img, dim, interpolation = cv2.INTER_AREA)
    return resized

def has_high_power(poly_string):
    powers = re.findall(r'\^\{(\d+)\}', poly_string)
    for power in powers:
        if int(power) >= 5:
            return True

    return False

def get_latex(cv_im):
    latex_code = []
    boxes, conf, classes = cnn.get_bboxes(cv_im, rel=False)
    if boxes is None:
        return latex_code
    boxes.sort(key=lambda x: x[1])
    model = LatexOCR()
    cropped_elements = []
    for box in boxes:

        box = [i if i>0 else 0 for i in box]
        x, y, w, h = box
```



```

crop = cv_im[y:y+h, x:x+w]
cropped_elements.append(crop)
resized = resize_im(crop)
im_pil = Image.fromarray(resized)
res = model(im_pil)
if has_high_power(res):
    continue
res = res.replace('\mathbf{x}', 'x')
res = res.replace('\bf x', 'x')
res = res.replace('\bf {x}', 'x')
res = res.replace('\bf{x}', 'x')
print(res + '\\\\')
latex_code.append(res + '\\\\')
return latex_code

latex = []

for img in doc:
    cv_im = np.array(img)
    cv_im_fst_half = cv_im[:int(cv_im.shape[0]/2), :]
    cv_im_scn_half = cv_im[int(cv_im.shape[0]/2):, :]
    latex += get_latex(cv_im_fst_half)
    latex += get_latex(cv_im_scn_half)

print(latex)

```

## Приложение Б

```
import cv2
import numpy as np
class Cnn:
    net: cv2.dnn_Net
    confidence_idx: float
    threshold_idx: float

    def __init__(self, weights_path: str, config_path: str, confidence_idx,
threshold_idx, size):

        self.net = cv2.dnn.readNetFromDarknet(config_path, weights_path)
        self.confidence_idx = confidence_idx
        self.threshold_idx = threshold_idx
        ln = self.net.getLayerNames()
        self.ln = [ln[i - 1] for i in self.net.getUnconnectedOutLayers()]
        self.size = size

    def get_bboxes(self, image, rel=True) -> (list[list[float]], list[float]):

        blob = cv2.dnn.blobFromImage(image, 1 / 255.0, (self.size, self.size),
swapRB=True, crop=False)

        self.net.setInput(blob)
        layer_outputs = self.net.forward(self.ln)

        boxes = []
        initial_boxes = []
        final_boxes = []

        confidences = []
        final_confidences = []

        class_ids = []
        final_class_ids = []

        for output in layer_outputs:
            for detection in output:

                scores = detection[5:]
                class_id = np.argmax(scores)
                confidence = scores[class_id]
                if confidence > self.confidence_idx:
                    box = np.around(detection[0:4], 2)
                    initial_boxes.append(box)
                    (centerX, centerY, width, height) = box
                    x = centerX - (width / 2)
                    y = centerY - (height / 2)
                    boxes.append([x, y, width, height])
                    confidences.append(confidence)
                    class_ids.append(class_id)

            idxs = cv2.dnn.NMSBoxes(boxes, confidences, self.confidence_idx,
self.threshold_idx)
            if len(idxs) > 0:
                H, W = image.shape[:2]
                for i in idxs:
                    if not rel:
                        (centerX, centerY, width, height) = initial_boxes[i] *
np.array([W, H, W, H])
                        x1 = int(centerX - (width / 2))
```

```

        y1 = int(centerY - (height / 2))
        # x2 = int(centerX + (width / 2))
        # y2 = int(centerY + (height / 2))
        box = [x1, y1, int(width), int(height)]
        final_boxes.append(box)
    else:
        final_boxes.append(initial_boxes[i])
        final_confidences.append(confidences[i])
        final_class_ids.append(class_ids[i])
    return final_boxes, final_confidences, final_class_ids
else:
    return None, None, None

def draw_boxes(self, boxes, image, confidences=None, classes=None, labels=None,
color=(0, 255, 0), thickness=6,):
    (H, W) = image.shape[:2]
    copy = image.copy()

    if boxes is None:
        return copy

    if confidences is not None:

        confidences = np.around(confidences, 2)
        for bb, conf, class_id in zip(boxes, confidences, classes):
            (centerX, centerY, width, height) = bb * np.array([W, H, W, H])
            x1 = int(centerX - (width / 2))
            y1 = int(centerY - (height / 2))
            x2 = int(centerX + (width / 2))
            y2 = int(centerY + (height / 2))
            cv2.rectangle(copy, (x1, y1), (x2, y2), color, thickness)

            cv2.putText(copy, '{}: {:.2f}'.format(labels[class_id], conf), (x1,
y1 - 10), cv2.FONT_HERSHEY_SIMPLEX,
                    10, color,
                    thickness)
            cv2.circle(copy, (int(centerX), int(centerY)), 1, color, -1)
    else:
        for bb in boxes:
            (centerX, centerY, width, height) = bb * np.array([W, H, W, H])
            x1 = int(centerX - (width / 2))
            y1 = int(centerY - (height / 2))
            x2 = int(centerX + (width / 2))
            y2 = int(centerY + (height / 2))
            cv2.rectangle(copy, (x1, y1), (x2, y2), color, thickness)
    return copy

```

## Приложение В

```
[net]
# Testing
#batch=1
#subdivisions=1
# Training
batch=64
subdivisions=1
width=640
height=640
channels=3
momentum=0.9
decay=0.0005
angle=0
saturation = 1.5
exposure = 1.5
hue=.1
learning_rate=0.00261
burn_in=1000
max_batches = 6000
policy=steps
steps=4800,5400
scales=.1,.1
[convolutional]
batch_normalize=1
filters=32
size=3
stride=2
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=64
size=3
stride=2
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=64
size=3
stride=1
pad=1
activation=leaky
[route]
layers=-1
groups=2
group_id=1
[convolutional]
batch_normalize=1
filters=32
size=3
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=32
size=3
stride=1
```

```
pad=1
activation=leaky
[route]
layers = -1,-2
[convolutional]
batch_normalize=1
filters=64
size=1
stride=1
pad=1
activation=leaky
[route]
layers = -6,-1
[maxpool]
size=2
stride=2
[convolutional]
batch_normalize=1
filters=128
size=3
stride=1
pad=1
activation=leaky
[route]
layers=-1
groups=2
group_id=1
[convolutional]
batch_normalize=1
filters=64
size=3
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=64
size=3
stride=1
pad=1
activation=leaky
[route]
layers = -1,-2
[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky
[route]
layers = -6,-1
[maxpool]
size=2
stride=2
[convolutional]
batch_normalize=1
filters=256
size=3
stride=1
pad=1
activation=leaky
```

```
[route]
layers=-1
groups=2
group_id=1
[convolutional]
batch_normalize=1
filters=128
size=3
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=128
size=3
stride=1
pad=1
activation=leaky
[route]
layers = -1,-2
[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky
[route]
layers = -6,-1
[maxpool]
size=2
stride=2
[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=leaky
#####
[convolutional]
batch_normalize=1
filters=256
size=1
stride=1
pad=1
activation=leaky
[convolutional]
batch_normalize=1
filters=512
size=3
stride=1
pad=1
activation=leaky
[convolutional]
size=1
stride=1
pad=1
filters=18
activation=linear
[yolo]
```

```
mask = 3,4,5
anchors = 10,14, 23,27, 37,58, 81,82, 135,169, 344,319
classes=1
num=6
jitter=.3
scale_x_y = 1.05
cls_normalizer=1.0
iou_normalizer=0.07
iou_loss=ciou
ignore_thresh = .7
truth_thresh = 1
random=0
resize=1.5
nms_kind=greedynms
beta_nms=0.6
[route]
layers = -4
[convolutional]
batch_normalize=1
filters=128
size=1
stride=1
pad=1
activation=leaky
[upsample]
stride=2
[route]
layers = -1, 23
[convolutional]
batch_normalize=1
filters=256
size=3
stride=1
pad=1
activation=leaky
[convolutional]
size=1
stride=1
pad=1
filters=18
activation=linear
[yolo]
mask = 0,1,2
anchors = 10,14, 23,27, 37,58, 81,82, 135,169, 344,319
classes=1
num=6
jitter=.3
scale_x_y = 1.05
cls_normalizer=1.0
iou_normalizer=0.07
iou_loss=ciou
ignore_thresh = .7
truth_thresh = 1
random=0
resize=1.5
nms_kind=greedynms
beta_nms=0.6
```