МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра информационных технологий**

**КУРСОВАЯ РАБОТА**

**РАСПОЗНАВАНИЕ И РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ УРАВНЕНИЙ**

Работу выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_В.С. Прокопенко

(подпись, дата)

Направление подготовки 02.03.03 курс 3

Направленность (профиль) Математическое обеспечение и администрирование информационных систем

Научный руководитель

канд. пед. наук, доц. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н.Ю. Добровольская

(подпись, дата)

Нормоконтролер

ст. преп. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.В. Харченко

(подпись, дата)

Краснодар

2021

**РЕФЕРАТ**

Курсовая работа 30 страниц, 18 рисунков, 7 таблиц, 7 источников.

НЕЙРОННЫЕ СЕТИ, РАСПОЗНАВАНИЕ, СИСТЕМЫ ЛИНЕЙНЫХ УРАВНЕНИЙ

Цель курсовой работы – реализовать приложение, распознающее систему линейных уравнений и решающую ее методом Гаусса.

В курсовой работе рассмотрены принципы построения нейронных сетей, изучены их архитектуры области применения, проведен реферативный обзор приложений, позволяющих распознавать печатные и рукописные математические примеры и решать их.

В практической части курсовой работы предложено приложение, которое с помощью различных библиотек и фреймворком на языке Python решает задачу распознавания системы линейных уравнений, заданных в печатной форме. На вход программы подается изображение любого графического формата, затем система решается методом Гаусса.

**СОДЕРЖАНИЕ**

[Введение 4](#_Toc89975174)

[1 Реферативный обзор приложений решения математических задач 5](#_Toc89975175)

[1.1 Приложение Photomath 5](#_Toc89975176)

[1.2 Приложение Malmath 6](#_Toc89975177)

[1.3 Приложение Mathway 7](#_Toc89975178)

[1.4 Приложение Mathpix 8](#_Toc89975179)

[1.5 Приложение MyScript Calculator 10](#_Toc89975180)

[2 Механизмы распознавания символов 12](#_Toc89975181)

[3 Используемый инструментарий и функционал приложения 19](#_Toc89975182)

[3.1 Структура работы приложения 20](#_Toc89975183)

[3.2 Эксперимент 23](#_Toc89975184)

[Заключение 29](#_Toc89975185)

[Список использованных источников 30](#_Toc89975186)

# **ВВЕДЕНИЕ**

Нейронная сеть – это последовательность нейронов, соединенных между собой синапсами. Структура нейронной сети пришла в мир программирования прямиком из биологии. Благодаря такой структуре, машина обретает способность анализировать и даже запоминать различную информацию. Нейронные сети также способны не только анализировать входящую информацию, но и воспроизводить ее из своей памяти.

Нейронные сети используются для решения сложных задач, одной из которых является распознавание. Распознавание – в настоящее время, самое широкое применение нейронных сетей. Используется в Google, когда вы ищете фото или в камерах телефонов, когда оно определяет положение вашего лица и выделяет его и многое другое. Этот процесс можно применить для помощи учащимся. Например, обучаемый решает самостоятельно пример и ему необходимо убедиться в правильности ответа. Тогда ученик можно обратиться к приложению, система может распознать условие из учебника и предложить ответ.

1. **Реферативный обзор приложений решения математических задач**

Рассмотрим предметную область, представленную некоторыми разработками, определим полезные свойства в каждом приложении и применим их в своем алгоритме.

* 1. **Приложение Photomath**

PhotoMath полезный и наглядный инструмент для решения математических уравнений, в основе которого лежит технология распознавания данных (OCR-технология), поступающих с камеры мобильного устройства [1].

Для того чтобы рассчитать уравнение необходимо навести камеру смартфона на задание, выбрать границы уравнения (можно откорректировать вручную), сфотографировать пример и подождать, пока на экране появится решение (Рисунок 1). Приложение умеет сканировать задачи как с бумаги, так и с экрана, но во втором случае ему может потребоваться больше времени на обработку. Однако рукописный текст не всегда точно распознает выражения.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1 – Пример работы приложения Photomath

Образовательная часть приложения заключается в том, что приложение выдает не только правильный ответ, но и позволяет просмотреть все этапы решения математического уравнения. Ко всем решенным с помощью приложения примерам можно вернуться в любой момент, поскольку они автоматически складируются в специальном разделе.

Основные задачи данного приложения заключаются в получении быстрого ответа к математической задаче и предоставить пользователю решение.

* 1. **Приложение Malmath**

Malmath решает математические задачи с подробным описанием каждого действия, а также предоставляет график (Рисунок 2). Приложение умеет решать интегралы, производные, пределы, логарифмы, тригонометрические уравнения и неравенства, примеры с корнями и модулями. По крайней мере большую часть. Однако, вам придется вручную вводить условие задания, так как функция распознавания с помощью камеры здесь не предусмотрена. Что касается его способностей, то они ограничиваются лишь задачами средней сложности с более скромным, чем у остальных приложений [2].

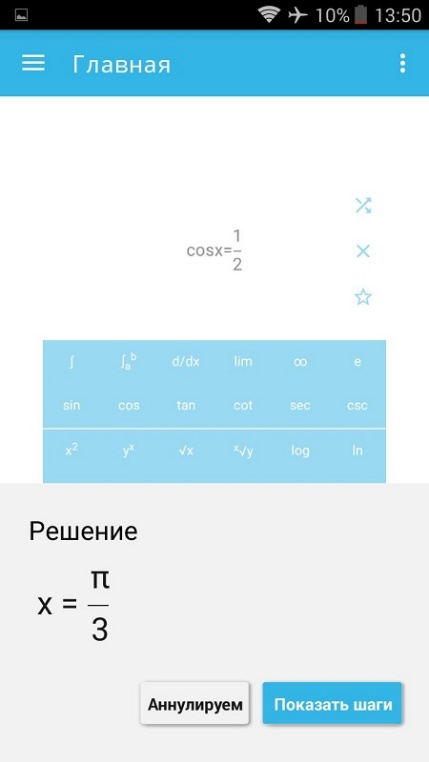


Рисунок 2 – Интерфейс приложения Malmath

* 1. **Приложение Mathway**

Приложение является своеобразным инструментом для решения задач, который, помимо школьного курса математики, охватывает математический анализ, статистику, тригонометрию, линейную алгебру и даже химию, так же позволяет достаточное сложные и интересные задачи. Практически тот же интерфейс в виде мессенджера, в котором все действия происходят будто в диалоге с виртуальным помощником [3].

Mathway является одним из самых интересных в плане качества решения, среди онлайн калькуляторов, так как данное приложение дает наиболее развернутые пошаговые решения задач. Калькулятор поможет вам не только с решением уравнений, неравенств и прочих сложных выражений, но также сумеет построить графики, может найти число молекул в определенной массе тела (Рисунок 3).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 3 – Получение ответа в приложении Mathway

Mathway для Android тоже решает задачи с помощью камеры устройства, однако эта функция плохо реализована. Интерфейс камеры в программе крайне минималистичный, так как отсутствует области распознавания. Камера захватывает только часть приложения, отсюда и ошибки. К тому же камера автоматически настроена на макро-режим, поэтому алгоритмы распознавания часто плохо срабатывают и выдают неправильный ответ.

* 1. **Приложение Mathpix**

Mathpix – первое приложение, которое позволяет вам решать и визуализировать решения, распознавая рукописный текста, включая сложные формулы (Рисунок 4). Mathpix стремится заменить дорогие и устаревшие графические калькуляторы, чтобы обеспечить бесплатное и интересное учебное пособие для студентов-математиков по всему миру [4].

Приложение намного раньше, в отличие от аналогов, получило способность распознавать рукописные математические задачи. Mathpix умеет решать простые математические задачи. Также здесь присутствует возможность построения графиков функций, благодаря интеграции с передовым графическим калькулятором Desmos. Этого нет ни у одного из представленных приложений, а это очень важно для решения задач алгебры и начала анализа.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 4 – Интерфейс камеры в приложении Mathpix

Программа практически в считанные секунды сканирует и считывает условие задачи. Тут же отправляет на сервер данные условия и почти мгновенно выдает ответ. Но ошибки тоже случаются, хотя довольно редко. При этом пользователю доступны инструменты для работы с задачами в режиме графика: редактирование вводных данных, добавление таблиц, заметок и дополнительных функций для нескольких графиков.

Однако приложение имеет сложный и непонятный интерфейс, который не позволяет новым пользователям начать быстро использовать калькулятор.

* 1. **Приложение MyScript Calculator**

Приложение MyScript Calculator впервые появилось в начале 2013 года. К тому же, сразу получило признание на международной выставке CES и было отмечено за инновации [5]. В приложении MyScript Calculator принцип подхода к математическим вычислениям кардинально отличается. Особенность MyScript Calculator заключается в том, что приложение работает только с рукописным вводом данных. Здесь даже отсутствуют кнопки, а все, что имеется — это чистое полотно на весь экран, имитирующее бумагу-миллиметровку. Условие математической задачи для вычисления пользователь пишет пальцем или с помощью стилуса. В данном случае предпочтительнее будет использование планшета или фаблета с цифровым пером (Рисунок 5).

Изображение выглядит как текст, доска

Автоматически созданное описание

Рисунок 5 – Пример работы приложения MyScript Calculator

Основным недостатком MyScript Calculator для Android можно считать отсутствие подробного описания решений, программа выдает только итоговый результат.

Все приведенные выше приложения обладают рядом преимуществ, благодаря которым обучения математики становится намного легче и интересней. Рассмотрим способы и методы распознавания символов, при помощи нейронной сети.

1. **Механизмы распознавания символов**

В распознавании образов широко применяются искусственные нейронные сети [6]. Нейронная сеть прямого распространения и конкурирующие нейроны могут использоваться для распознавания текстовых символов, однако при распознавании зашумленных образов (текстовых символов), используются такие нейронные сети, как Хемминга или Гроссберга. Наибольший эффект, обусловленный пластичностью, дают сети Гроссберга.

Процесс обучения сходится в зависимости от значения параметра сходства. Кроме этого, зашумленность данных может привести к ошибочным результатам, так как используемая мера сходства в ряде случаев приводит к отнесению образа не к соответствующему классу. Норма двоичного вектора в зависимости от степени зашумленности может принимать одно и тоже значение при различных зашумлениях. Наиболее просто обучение распознаванию текстовых символов осуществляется нейронными сетями прямого распространения, однако, подобные связи малоэффективны при распознавании зашумленных символов. Так как сеть прямого распространения является наиболее простой, то задача распознавания зашумленных текстовых символов этой сетью является актуальной.

Основная задача есть повышение эффективности распознавания искаженных текстовых символов за счет сегментации признакового пространства представления изображения. Для достижения цели в каждом из сегментов разбиения плоскости существования изображения ставится в соответствие значение нуля или единицы, в зависимости от наличия в нем следа изображения. При этом след изображения предлагается отождествлять с одной из базисных функций, задаваемых положением прямой в сегменте. Для этого каждый сегмент разбивается на подсегменты таким образом, чтобы при кодировании подсегмента двоичным вектором, искажение компонент не приводило бы уменьшению значений компонент, отмечающих сегмент. В качестве такой меры выбирается порог суммы компонент, отмечаемых подсегменты соответствующих сегментов. Для определения бинарных значений подсегментов необходимо определить всевозможные (характеры) функции отображения символа в сегменте посредством бинарных компонент, отмечающих подсегменты. При распознавании текстовых символов такими функциями на плоскости являются: прямая параллельная оси абсцисс, прямая параллельная оси ординат, прямая с положительным наклоном относительно оси абсцисс и прямая с отрицательным наклоном относительно оси абсцисс.

Если для обучения персептрона распознавания символа текста область задания символа разбивалась на квадраты, нумеруемые целыми положительными числами, то для обучения персептрона распознавания зашумленных символов каждый квадрат разбивается на поля, аналогично разбиению. В результате для квадрата получаем упорядоченные поля , для ([Таблица 1](#таблица1)).

Таблица 1 – Область задания символа

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

В каждом из этих полей для текстовых символов может располагаться одна из приведенных ранее функций, либо поле может быть пустым. Выбор длины двоичных векторов для кодирования области существования символов определяется количеством и топологией символа и для текстовых символов определяется, как 5x5. Такой же размер может быть принят и для полей внутри квадрата. Как и в разбиении на сегменты, 0 – округляет отсутствие следа изображения, 1 – округляет наличие следа изображений при бинарном кодировании ([Таблица 2](#таблица2)).

Таблица 2 – Пустое поле

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

Таблица 3 – Поле прямой параллельной оси абсцисс

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

Таблица 4 – Поле прямой параллельной оси ординат

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |

Таблица 5 – Поле прямой с положительным углом наклона

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |

Таблица 6 – Поле прямой с отрицательным углом наклона

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 0 | 0 | 1 |

Тогда нейронная сеть состоит из 5 нейронов (Рисунок 6) с бинарной функцией активации может одним из известных методов обучена кластеризации этих функций в каждом из сегментов. При выборе порога каждого нейрона выходной сигнал принимает единичное значение при достижении порога при весовых коэффициентах , что соответствует .

Тогда сеть из 5 нейронов будем обозначать символом и причем , если

, если двоичный вектор определяется линейным кодом, соответствующим массиву (Таблица 3);

, если двоичный вектор соответствует массиву (Таблица 4);

, если двоичный вектор соответствует массиву (Таблица 5);

, если двоичный вектор соответствует массиву (Таблица 6);

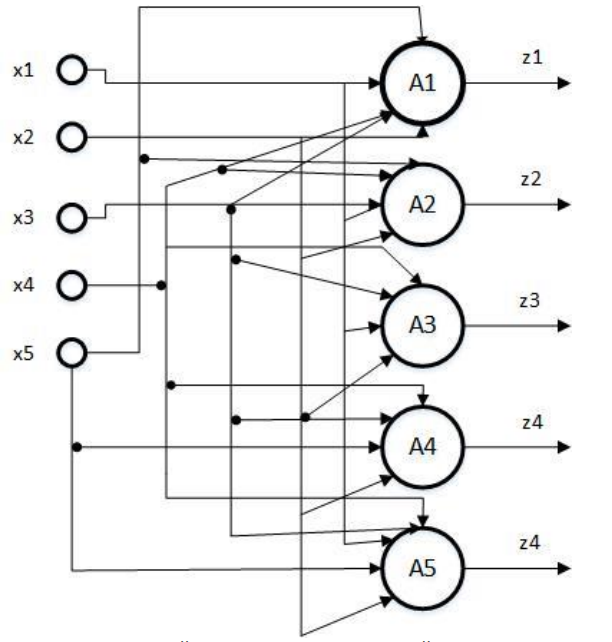


Рисунок 6 – Нейронная сеть из пяти нейронов

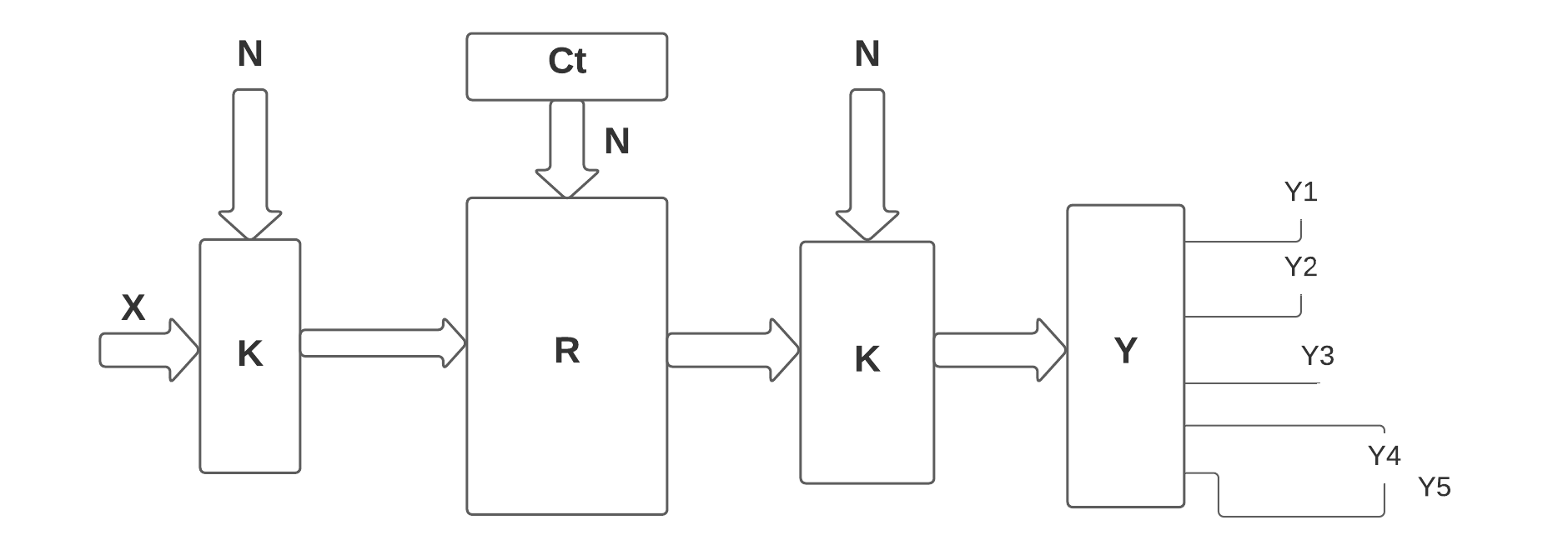


Рисунок 7 – Нейронная сеть

Сеть включает два слоя нейронов и два коммутатора , на управляющие входы которых поступают коды, задающие последовательность сегментов разбиения области задания изображения. Слой нейронов включает фиксаторы выходов слоя и нейронов по числу распознаваемых текстовых символов (Рисунок 7).

Слой фиксаторов представляет группы, каждая из которых содержит 4 за понимающих элемента, в которые заносятся единицы, отличающие соответствующего из 4 функций, изображение которой присутствует в выбранном сегменте, задаваемом кодом , формируемым счетчиком .

Слой выходных нейронов формирует единичный сигнал на выходе, определяемом истинностью логического , где – единичные выходы соответствующих фиксаторов выходов слоя .

Слой нейронов представляет сеть . Вход представляет битовую последовательность пятикомпонентного вектора. Длина последовательности определяется числом сегментов разбиения области задания изображения.

Сеть может не содержать коммутаторов, тогда слои и будут содержать группы нейронов по числу характерных функций для слоя в количестве, определяемом числом сегментов разбиения, а в слое количество нейронов определяется числом распознаваемых символов.

Допустимые искажения вектора округляются числом битов, искажения которых не приводят к переводу характеристической функции в .

Алгоритм обучения сети состоит в выделении в соответствии с задаваемым двоичным вектором в сегменте заданных характеристических функций, например, по правилу Хебба. Выделенные характеристики функции служат входами слоя , выходы которого задаются булевыми функциями входных переменных в виде таблицы истинности. Обучение слоя может быть проведено либо по правилу Хебба, либо с использованием дельта правила. При любом способе обучения сеть детерминирована, и каждому входному изображению соответствует единственный выход с сигналом равным единице.

При обучении слоя проводится модификация входного слоя для обеспечения устойчивости к искажению компонент вектора. Для чего вводится параметр сходства , который позволяет отнести искаженный входной вектор к идеальному. Если определить количество бит, искажения которых могут не учитываться при формировании изображений, то в этом случае, сеть, обученная при идеальных входных сигналах, будет давать правильную классификацию образов при допустимых искажениях [7].

Благодаря полученным знаниям о персептроне и алгоритмах, построим свою нейронную сеть распознавания математических символов.

1. **Используемый инструментарий и функционал приложения**

При построении нейронной сети использовался высокоуровневый язык программирования с динамической семантикой Python. Python лучше всего подходит для выполнения таких задач, благодаря высокой производительности при обработке данных. Изучим фреймворки Python для обучения нейронных сетей.

— TensorFlow

Применяется для продвинутых проектов, таких как создание многослойных нейронных сетей. Может использоваться для распознавания речи, лиц, объектов и изображений, а также для работы с текстом.

— PyTorch

Позволяет обучать модели быстро и эффективно. Удобный для быстрого прототипирования в исследованиях, а также небольших проектов.

— Keras

Используется для быстрого прототипирования. Хорош в кейсах, связанных с переводом, распознаванием изображений и речи.

— Tesseract

Свободная компьютерная программа, позволяющая распознавать текст с открытым исходным кодом (OCR).

Рассмотрим модуль Numpy – расширение языка Python, добавляющее поддержку больших многомерных массивов и матриц, вместе с большой библиотекой высокоуровневых математических функций для операций с этими массивам, которые позволят подбирать веса, которые соответствуют всем входам для решения поставленных задач.

Воспользуемся библиотекой Tkinter, которая содержит компоненты графического интерфейса пользователя, для написания модального окна приложения, также возьмем за основу фреймворк Tesseract, с помощью которого распознаем текст и модуль Numpy для решения системы линейных уравнений.

* 1. **Структура работы приложения**

Основная задача нашего алгоритма – выдать правильное решение математической задачи, посредством методов, взятых с линейной алгебры. Алгоритм хранит в себе как итерационные способы решения, так и прямые. Программа находит и выдает решение пользователю на экране.

По началу программа получает готовое изображение (\*.png, \*.jpg, \*.jpeg), корректирует его размер, исследует и записывает ответ в новом окне. Однако в дальнейшем планируется использовать алгоритм в мобильном приложении, где источником математической задачи будет камера смартфона.

После распознавания текста коэффициенты помещаются в промежуточную структуру данных – матрицу. Затем система линейных уравнений решается методом Гаусса, результат записывается в одномерный массив. Фрагмент Гаусса приведен ниже.

def make\_identity(matrix):  
 for nrow in range(len(matrix)-1,0,-1):  
 row = matrix[nrow]  
 for upper\_row in matrix[:nrow]:  
 factor = upper\_row[nrow]  
 upper\_row -= factor\*row  
 return matrix  
  
def gaussPivotFunc(matrix):  
 for nrow in range(len(matrix)):  
 pivot = nrow + np.argmax(abs(matrix[nrow:, nrow]))  
 if pivot != nrow:  
 matrix[[nrow, pivot]] = matrix[[pivot, nrow]]  
 row = matrix[nrow]  
 divider = row[nrow]  
 if abs(divider) < 1e-10:  
 raise ValueError(f"Матрица несовместна. Максимальный элемент в столбце {nrow}: {divider:.3g}")  
 row /= divider  
 for lower\_row in matrix[nrow+1:]:  
 factor = lower\_row[nrow]  
 lower\_row -= factor\*row  
 make\_identity(matrix)  
 global answer  
 answer = [matrix[0][3], matrix[1][3], matrix[2][3]]

В работе приложения возникла проблема распознавания. Если изображение нечеткое и сильно зашумленное, то алгоритм неправильно определит математических пример, следовательно, выдаст не правильный результат. Тогда для решения неточности распознавания, используется метод ([Рисунок 8](#рисунок8)), который проверяет условие примера и избавляется от ошибок. Однако могут встречаться изображения, которые не будут распознаны.

str = re.sub(r'\s+', '', str)  
str = str.replace('—', '-')  
str = str.replace('--', '-')  
str = str.replace('—-', '-')  
str = str.replace('t', '+')  
str = str.replace('zZ', 'z')

str = str.replace('Zz', 'z')

Рисунок 8 – Фрагмент кода для исправления неточностей

Приложение позволяет постоянно обновлять изображения с математическими примерами, при этом условия предыдущих хранятся в текстовом файле ([Рисунок 9](#рисунок9)). Результат вычисления выдается пользователю в messagebox. Интерфейс приложения отображен ниже ([Рисунок 10](#рисунок10)).



Рисунок 9 – Структура работы приложения

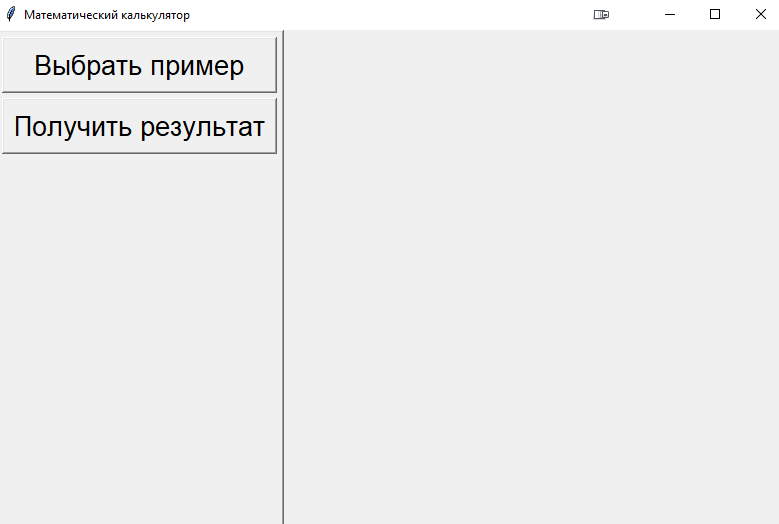


Рисунок 10 – Интерфейс приложения

* 1. **Эксперимент**

В рамках работы был проведен эксперимент по качеству распознавания символов.

Опыт 1.

На вход программы подается не зашумленное и четкое изображение ([Рисунок 11](#рисунок11)).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 11 – Опыт 1

Алгоритм правильно распознал систему линейных уравнений и выдал правильный результат ([Рисунок 12](#рисунок12)).

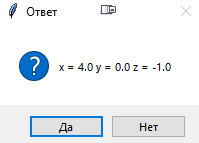


Рисунок 12 – Результат опыта 1

Опыт 2.

На вход программы подаются четкие, не зашумленные символы на темном фоне ([Рисунок 13](#рисунок13)).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 13 – Опыт 2

Несмотря на темный и блеклый фон, алгоритм справился и точно распознал систему уравнений ([Рисунок 14](#рисунок14)).

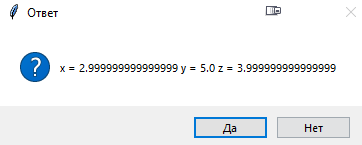


Рисунок 14 – Результат опыта 2

Опыт 3.

На вход программы подается сильно зашумленный фон и нечеткие символы ([Рисунок 15](#рисунок15)).

Изображение выглядит как текст, доска

Автоматически созданное описание

Рисунок 15 – Опыт 3

Программа неправильно распознала символы и не смогла построить матрицу коэффициентов ([Рисунок 16](#рисунок16)).

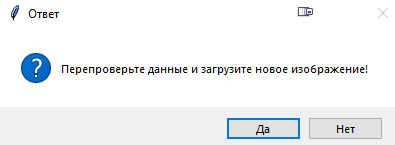


Рисунок 16 – Результат опыта 3

Опыт 4.

На вход программы подается четкий фон, но с сильно зашумленными символами ([Рисунок 17](#рисунок17)).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 17 – Опыт 4

Алгоритм не распознал математические символы, соответственно, решение этой системы не определено ([Рисунок 18](#рисунок18)).

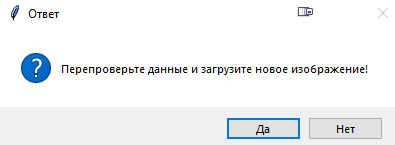


Рисунок 18 – Результат опыта 3

Итоги опытов приведены ниже ([Таблица 7](#таблица7)).

Таблица 7 – Итоги эксперимента

|  |  |  |
| --- | --- | --- |
| № Опыта | Входные данные | Результат |
| 1 | Не зашумленное, четкое изображение | распознано |
| 2 | Нечеткий фон, не зашумленные символы | распознано |
| 3 | сильно зашумленный фон и нечеткие символы | не распознано |
| 4 | четкий фон, сильно зашумленные символы | не распознано |

В ходе проведенных опытов, заметим, что программа не всегда точно распознает текст с фото, зачастую верный результат можно получить, только с не зашумленного и чистого изображения.

# **ЗАКЛЮЧЕНИЕ**

Цель курсовой работы – реализовать приложение, распознающее систему линейных уравнений и решающую ее методом Гаусса – достигнута.

В ходе курсовой работы рассмотрены принципы построения нейронных сетей и изучены их области применения. На основе академической литературы и веб-ресурсам провели анализ приложений, которые распознают и решают математические примеры.

В практической части реализовано приложение, которое решает задачу распознавания СЛАУ, заданную в печатной форме, в которой входным параметром является графическое изображения любого формата. После того как приложение распознает текст с изображения, коэффициенты помещаются в матрицу, и система решается методом Гаусса.

Интерфейс приложения реализован на языке Python с использованием библиотек Numpy, Tesseract, Tkinter.

В дальнейшем предполагается расширить область математических примеров, улучшение интерфейса и создания мобильной версии. Хранение примеров и ответов в базе данных. Добавление примеров для самостоятельного решения и отслеживания этапов решения.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Мобильное приложение Photomath : [[сайт](https://photomath.com/en/)]. – 2021. – URL: <https://photomath.com/en/> (дата обращения: 1.12.2021).
2. Мобильное и веб приложение Malmath : [[сайт](https://www.malmath.com/)]. – 2021. – URL: <https://www.malmath.com/> (дата обращения: 1.12.2021).
3. Веб приложение Mathway : [[сайт](https://www.mathway.com/ru/Algebra)]. – 2021. – URL: <https://www.mathway.com/ru/> (дата обращения: 1.12.2021).
4. Мобильное приложение Mathpix : [[сайт](https://mathpix.com/)]. – 2021 – URL: <https://mathpix.com/> (дата обращения: 1.12.2021).
5. Мобильное приложение MyScript Calculator : [сайт]. – 2021. – URL: <https://www.myscript.com/ru/calculator> (дата обращения: 1.12.2021).
6. Розенблатт, Ф. Принципы нейродинамики: Перцептроны и теория механизмов мозга / Ф. Розенблатт – М.: Мир, 1965. – 480 с.
7. Николенко, С.И. Глубокое обучение. Погружение в мир нейронных сетей / С.И. Николенко, А.А. Кадурин, Е.М. Архангельская – СПб.: Питер, 2020. – 480 с.