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 вычислительных технологий**

**КУРСОВАЯ РАБОТА**

**ИМУННЫЙ АЛГОРИТМ ДЛЯ РЕШЕНИЯ ЗАДАЧИ РАСПОЗНАНИЯ РУКОПИСНОГО ТЕКСТА**

Работу выполнила \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. Р. Миленченко

(подпись)

Направление подготовки 02.03.02 — «Фундаментальная информатика и\_\_\_\_\_

(код, наименование)

информационные технологии»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ курс\_\_\_\_\_3\_\_\_\_\_

Направленность (профиль)\_\_\_\_Математическое и программное обеспечение компьютерных технологий\_\_\_\_

Научный руководитель

канд. техн. наук, доц. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е. Е. Полупанова

(подпись, дата)

Нормоконтролер

преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е. А. Нигодин

(подпись, дата)

Краснодар

2023

**РЕФЕРАТ**

Курсовая работа 24 с., 4 ч., 5 рис., 7 источн., 1 прил.

ИМУННЫЙ АЛГОРИТМ, ЗАДАЧА РАСПОЗНАНИЯ, РУКОПИСНЫЙ ТЕКСТ

Цель работы: исследование и разработка иммунного алгоритма для распознавания рукописного текста.

Объектом исследований в работе является задача распознавания рукописного текста.

Методологическая основа исследования включает в себя эмпирический метод, сбор статистической информации, графический метод, анализ полученных зависимостей, синтез выводов, аналогия.

В результате работы было разработано приложение, которое на основе иммунного алгоритма реализует решение задачи распознания рукописного текста.

Научная новизна работы заключается в разработке нового иммунного алгоритма для распознания рукописного текста, который позволяет эффективно распознавать написанный текст в приложении.

В результате решения поставленных задач был разработан иммунный алгоритм для распознания рукописного текста. Реализовано приложение, основанное на разработанном алгоритме, которое позволяет эффективно распознавать рукописный текст. Проведено тестирование разработанного алгоритма. Полученные результаты говорят об эффективности и надёжности разработанного алгоритма.

**СОДЕРЖАНИЕ**

[Введение 4](#_Toc153543114)

[1 Введение в проблему распознания рукописного текста 6](#_Toc153543115)

[2 Теоретические основы иммунного алгоритма 8](#_Toc153543116)

[2.1 Основные принципы работы иммунной системы 8](#_Toc153543117)

[2.2 Применение иммунных алгоритмов в информационных технологиях 9](#_Toc153543118)

[2.3 Моделирование иммунных алгоритмов в распознании рукописного   
 текста 10](#_Toc153543119)

[3 Разработка и тестирование иммунного алгоритма распознания рукописного текста 12](#_Toc153543120)

[3.1 Алгоритм распознания рукописного текста на основе иммунной   
 системы 12](#_Toc153543121)

[3.2 Основные сведения о программе 15](#_Toc153543122)

[3.3 Особенности реализации 17](#_Toc153543123)

[4 Анализ эффективности работы иммунного алгоритма 20](#_Toc153543124)

[Заключение 22](#_Toc153543125)

[Список использованной литературы 23](#_Toc153543126)

[Приложение А Основная программа 25](#_Toc153543127)

### ВВЕДЕНИЕ

Эта работа направлена на изучение использования иммунных алгоритмов в распознавании рукописного текста.

В современном информационном обществе, где огромное количество данных генерируется и обрабатывается каждый день, автоматическое распознавание рукописного текста становится все более востребованным. Рукописный текст является одним из основных способов коммуникации, и его эффективное распознавание имеет огромное значение для различных областей. Однако, распознавание рукописного текста остается сложной задачей из-за его вариативности и неоднозначности. Таким образом, разработка новых алгоритмов распознавания рукописного текста является актуальной задачей.

Целью данной курсовой работы является исследование и разработка иммунного алгоритма для распознавания рукописного текста. Иммунные алгоритмы являются метаэвристическими методами, основанными на принципах иммунной системы организма. Использование иммунного алгоритма может привести к улучшению точности и эффективности распознавания рукописного текста.

Для реализации поставленной цели необходимо решить следующие задачи:

* изучить теорию иммунных алгоритмов;
* разработать иммунный алгоритм для распознавания рукописного текста;
* проанализировать полученные результаты и сделать выводы о применимости иммунного алгоритма в распознавании рукописного текста.

Объектом исследования в данной работе является задача распознавания рукописного текста.

Предмет исследования иммунный алгоритм распознавания рукописного текста, и система на его основе, которая может эффективно распознавать и интерпретировать рукописный текст, используя принципы и механизмы, вдохновленные иммунной системой человека. Исследование направлено на изучение потенциала иммунных алгоритмов в повышении точности и эффективности рукописных систем распознавания текста, что в конечном итоге способствует достижениям в области распознавания и машинного обучения.

Информационной базой исследования являются учебные материалы, статьи и документация по искусственным иммунным алгоритмам. Методологическая основа исследования включает в себя эмпирический метод (многократный запуск программы для разных цифр и получение результатов её работы), синтез выводов, аналогия (механизм работы программы сравнивается с механизмом эволюции живых организмов).

Научная новизна работы заключается в разработке нового иммунного алгоритма для распознания рукописного текста, который позволяет эффективно распознавать написанный текст в приложении.

Теоретическая и практическая значимость работы состоит в широком применении алгоритма решения задачи распознания рукописного текста в различных областях.

Теоретическая значимость работы заключается в разработке и применении иммунного алгоритма для распознавания рукописного текста. Это может способствовать развитию методов машинного обучения и искусственного интеллекта в области обработки естественного языка.

Практическая значимость работы состоит в возможности создания систем распознавания рукописного текста, которые могут быть использованы в различных областях, таких как банковское дело, медицина, образование и другие. Это может улучшить процессы автоматизации и повысить эффективность работы в этих сферах.

1. **Введение в проблему распознания рукописного текста**

Проблема распознания рукописного текста существует в течение многих лет. Традиционные методы, используемые для печатных символов не дают удовлетворительных результатов, а альтернативные подходы сталкиваются с многочисленными препятствиями.

Проблемы распознания рукописного текста заключаются в том, что компьютеру сложно автоматически распознать и интерпретировать символы, написанные от руки. У рукописного текста могут быть размытые границы символов, нечеткие искажения формы букв, а также возможные ошибки и опечатки, которые могут затруднить точное распознавание текста.

Первая проблема связана с несовершенством процесса распознавания символов. Распознавание символов, написанных от руки, может быть затруднительным и неточным из-за естественной вариации в написании символов у разных людей. Компьютеру может быть трудно определить, какой конкретно символ был написан, особенно если он написан неразборчиво или способ написания отличается от обучающего набора данных.

Вторая проблема − разнообразие стилей письма. Каждый человек имеет свой уникальный стиль письма, что влияет на качество распознавания. Например, некоторые люди могут писать с наклоном, другие - с пропущенными штрихами, что может осложнить правильное распознавание символов.

Третья проблема заключается в неразборчивости почерка. Распознавание рукописного текста может быть затруднено, если символы написаны в контексте основных слов или предложений. Одна и та же буква может выглядеть по-разному в зависимости от ее положения относительно других символов. Кроме того, некоторые символы могут быть неразборчивыми или может иметь слишком маленький размер, для идентификации с высокой степенью уверенности.

Четвертая проблема − низкая скорость распознавания. Распознавание рукописного текста требует значительных вычислительных ресурсов и времени. Обработка больших объемов рукописного текста может быть затруднительной и занимать много времени.

Для решения вышеперечисленных проблем разработаны специальные алгоритмы машинного обучения, такие как нейронные сети и иммунные алгоритмы. Однако, повышению точности и эффективности распознавания рукописного текста является актуальной задачей.

### Теоретические основы иммунного алгоритма

### Основные принципы работы иммунной системы

Иммунный алгоритм (ИА) основан на принципах работы иммунной системы человека, которая защищает его от вредных воздействий. Рассмотрим основные концепции, на которых основана работа иммунного алгоритма.

Иммунная система человека состоит из различных типов клеток, таких как лимфоциты и макрофаги, которые работают вместе для обнаружения и уничтожения инфекций и других вредных воздействий. Иммунная система также обладает способностью к адаптации и запоминанию опасных агентов.

В иммунной системе антигены представляют собой внешние структуры, которые вызывают иммунный ответ. Антитела, в свою очередь, являются белками, способными связываться с антигенами и уничтожать их.

Иммунная система обладает способностью к самоорганизации и саморегуляции, что позволяет ей адаптироваться к новым условиям и изменениям в окружающей среде.

Иммунная система также способна подавлять аутоиммунные реакции, когда она ошибочно направляет свой иммунный ответ против собственных клеток организма.

Иммунный алгоритм использует эти теоретические основы для создания вычислительной модели, которая может эффективно решать задачи оптимизации, поиска и классификации. ИА моделирует процессы обнаружения и уничтожения "вредных" элементов в пространстве поиска, используя понятия антигенов, антител и иммунной памяти. В результате иммунный алгоритм может эффективно находить оптимальные решения в сложных задачах оптимизации и обучения без учителя.

### Применение иммунных алгоритмов в информационных технологиях

Иммунные алгоритмы используются в информационных технологиях для решения различных задач, таких как оптимизация параметров, обучение нейронных сетей, поиск векторов признаков и прогнозирование.

Применение иммунных алгоритмов в информационных технологиях осуществляется по следующим направлениям:

* оптимизация параметров;
* обучение нейронных сетей;
* поиск векторов признаков;
* прогнозирование;
* распознавание образов.

Иммунные алгоритмы могут использоваться для оптимизации параметров сложных систем, таких как: нейронные сети, генетические алгоритмы и многие другие. Они могут помочь найти оптимальные значения параметров для достижения максимальной производительности системы.

Иммунные алгоритмы могут использоваться для обучения нейронных сетей путем изменения их весовых коэффициентов. Это позволяет улучшить процесс обучения и повысить точность прогнозов.

Иммунные алгоритмы могут помочь в поиске оптимальных векторов признаков, которые могут быть использованы для классификации и анализа данных. Это может быть особенно полезно в задачах машинного зрения, обработке естественного языка и других областях, где требуется выделение важных признаков из больших объемов данных.

Иммунные алгоритмы могут использоваться для прогнозирования различных явлений и событий на основе имеющихся данных. Они могут помочь выявлять закономерности и тенденции в данных, что может быть полезно для принятия бизнес-решений и планирования стратегий.

Иммунные алгоритмы показывают хорошие результаты в различных задачах распознавания образов, включая рукописное распознавание текста. Они эффективно справляются с вариациями написания и сложностью почерка, адаптируя и развивая свои модели распознавания на основе итерационного процесса.

Таким образом, применение иммунных алгоритмов в информационных технологиях может помочь улучшить производительность систем, улучшить обучение и предсказание, повысить качество анализа данных, и эффективность распознавая образов.

### Моделирование иммунных алгоритмов в распознании рукописного текста

Моделирование иммунных алгоритмов в распознании рукописного текста представляет собой интересную область исследований, объединяющую методы обработки изображений, машинного обучения и биоинспирированных методов.

Когнитивно-исследовательские модели иммунной системы используются для определения типов антигенов, лимфоцитов и презентующих клеток.

Отличительные черты иммунной системы, такие как клеточные паттерны и механизмы взаимодействия, используются для создания алгоритмов, имитирующих естественные процессы при обнаружении, распознавании и уничтожении патогенов.

В контексте распознавания рукописного текста, иммунные алгоритмы могут быть применены для эффективного обучения систем распознавания символов или слов. Это позволяет создавать системы более адаптивные к различным стилям или отклонениям в написании символов.

Моделирование иммунных алгоритмов в распознавании рукописного текста может включать в себя следующие этапы:

* представление символов;
* создание паттернов антигенов и лимфоцитов;
* распознавание и классификация;
* обучение и адаптация;
* оценка качества распознавания.

Иммунные алгоритмы могут потребовать специфического представления символов или слов. Это могут быть бинарные или многомерные массивы, описывающие структуру и форму символов.

Антигены могут представлять собой фрагменты изображений рукописного текста, а лимфоциты - шаблоны для распознавания символов. Эти паттерны могут быть созданы как на основе обучающих данных, так и с использованием метрик сходства, таких как расстояние по Левенштейну.

Процесс обнаружения и классификации символов в рукописном тексте с использованием иммунных алгоритмов может включать в себя шаги вычисления аффинности между антигенами и лимфоцитами, а также алгоритмическую обработку совпадений и ошибок.

Иммунные алгоритмы обеспечивают механизмы обучения и адаптации к изменяющимся образцам и стилям рукописного текста. Это позволяет системе сохранять производительность при изменяющихся условиях.

Важной составляющей моделирования является оценка качества распознавания, что может включать в себя ряд метрик, таких как точность, полнота и F-мера.

### Разработка и тестирование иммунного алгоритма распознания рукописного текста

### Алгоритм распознания рукописного текста на основе иммунной системы

Классификация отдельных клеток и молекул является важнейшим свойством естественной иммунной сети.

В отличие от естественной иммунной системы, количество антигенов в данной задаче невелико и известно заранее. Учитывая это, каждому лимфоциту можно приписать номер антигена, на который данный лимфоцит будет реагировать. Это позволит упростить процедуру нахождения количества лимфоцитов, специфичных именно к этому антителу.

В естественной иммунной системе классификация происходит за счет химических реакций. Для искусственной иммунной системы можно предложить следующую организацию сети и представление лимфоцитов.

Монохромное изображение можно представить как матрицу размера М х N булевых значений, в которых true стоит на тех местах, которые в изображении соответствуют пикселям черного цвета, которые и формируют изображение.

Лимфоцит, в свою очередь, можно представить как массив из Р пар чисел. Каждая такая пара представляет собой координаты пикселя в изображении. Однако будем использовать немного усложненное представление лимфоцитов, в котором еще хранится информация о тех пикселях, которые должны быть белыми (незаполненными). Таким образом, лимфоцит содержит два списка координат, один из которых хранит координаты черных пикселей, а другой - координаты белых. И тогда число S можно представить в виде формулы суммы (1)

S=Si+S2 (1)

где

Si - количество черных пикселей поданного изображения (антигена), которые определяются координатами из первого списка лимфоцита;

S2- количество белых пикселей поданного изображения (антигена), которые определяются координатами из второго списка лимфоцита. Структурная модель лимфоцита изображена на рисунке 1.

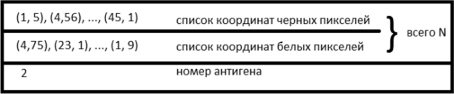


Рисунок 1- Структурная модель лимфоцита

Введем некое число S < Р. Будем говорить, что лимфоцит реагирует на данный антиген, если количество элементов матрицы антигена со значениями true, определяемых номерами строк и столбцов массива пар чисел лимфоцита, больше либо равно S. На основе данного определения можно посчитать количество лимфоцитов, реагирующих на данный антиген. Так как каждый лимфоцит имеет приписанный номер антигена (идеального или образцового), то можно узнать, на какой антиген должно реагировать большинство из этих лимфоцитов. В идеальном случае (если антиген подан без искажений и иммунная сеть хорошо обучена) именно идентификатор этого антигена и будет ответом. Вместо абсолютной величины S будем использовать относительную, которая будет представлять собой минимальный процент от числа Р, необходимый для того, чтобы лимфоцит среагировал на данный антиген. Введем меру аффинности данному антигену для каждого лимфоцита. Она будет представлять собой отношение количества элементов матрицы антигена, которые правильно определяются координатами из списков лимфоцита, т.е. аффинность будет представлена формулой (2)

*f = S/N* (2)

где

S - уже рассмотренная нами величина, а *N* - количество пикселей в обоих списках лимфоцита (количество пикселей, о которых лимфоцит хранит информацию).

Также можно отметить следующее упрощение модели естественной иммунной системы. Будем рассматривать только В-лимфоциты, причем все из них будут использоваться для классификации каждого антигена. Т-лимфоциты в данной искусственной системе рассматриваться не будут.

Таким образом, получили описание алгоритма искусственной иммунной системы. Блок-схема алгоритма приведена на рисунке 2.

Изображение выглядит как текст, диаграмма, линия, Параллельный

Автоматически созданное описание

Рисунок 2 – Блок схема предлагаемого алгоритма

### Основные сведения о программе

Программа была разработана с использованием библиотеки tkinter для создания графического интерфейса пользовательского приложения. Импортированы модули canvas, button, text и messagebox из tkinter, а также модуль partial из functools.

Результатом программы является классификация введенной пользователем матрицы данных как одного из предустановленных антигенов.

На рисунке 3 показано начальное окно программы.

Изображение выглядит как текст, Прямоугольник, снимок экрана, дизайн

Автоматически созданное описание

Рисунок 3 - Начальное состояние программы

При запуске программы в начальном окне генерируется Canvas для рисования, Text для отображения матрицы размера М х N булевых значений, в которых B стоит на тех местах, которые в изображении соответствуют пикселям черного цвета, которые и формируют изображение, и кнопки «Распознать» для подачи матрицы на классификацию.

В конечном результате программа выводит информацию о классификации антигена, который наиболее схож с введенной пользователем цифры.

Изображение конечного результата показано на рисунке 4.

При реализации данного приложения был использован язык программирования python и среда разработки PyCharm.

Изображение выглядит как текст, снимок экрана, Прямоугольник, Шрифт

Автоматически созданное описание

Рисунок 4 - Конечный результат

### Особенности реализации

Программа была разработана с использованием библиотеки tkinter для создания графического интерфейса пользовательского приложения. Импортированы модули canvas, button, text и messagebox из tkinter, а также модуль partial из functools.Программа разделена на два класса: lymphocyte и immunesystemapp.

Класс lymphocyte представляет иммунную клетку и содержит атрибуты black\_pixels и white\_pixels, которые представляют координаты черных и белых пикселей соответственно.

Также класс lymphocyte содержит метод calculate\_affinity, который вычисляет аффинность антигена на основе матрицы антигена.

Класс immunesystemapp представляет приложение и инициализируется с объектом master (главное окно приложения). Внутри класса имеются следующие компоненты:

* матрица для хранения данных о рисунке (matrix);
* канва для рисования (canvas);
* текстовое поле для отображения введенной матрицы (input\_matrix\_text);
* кнопка для подачи на распознавание (recognize\_button);
* обработчик для отслеживания рисования на канве (draw);
* лимфоциты с паттернами букв, представленные экземплярами класса lymphocyte.

Метод draw отслеживает рисование на канве и записывает информацию о рисовании в матрицу. Затем он обновляет текстовое поле с введенной матрицей и рисует черный квадрат на канве. Метод recognize выводит матрицу на консоль и классифицирует антиген на основе метода classify\_antigen. Если аффинность антигена выше заданного порога, то он классифицируется как определенный антиген и отображается всплывающее окно с результатом. В противном случае отображается окно с сообщением, что ни один антиген не классифицирован. После этого метод очищает канву и матрицу.

Метод classify\_antigen вычисляет аффинность антигена на основе переданной матрицы антигена и возвращает классифицированный антиген. Для этого он проходится по каждому лимфоциту и вычисляет аффинность с помощью метода calculate\_affinity. Если аффинность выше порога и больше текущей максимальной аффинности, то переменным max\_affinity и classified\_antigen присваиваются соответствующие значения. В конце метод возвращает классифицированный антиген.

Метод clear\_matrix очищает матрицу и обновляет текстовое поле с введенной матрицей.Метод update\_input\_matrix\_text отображает введенную матрицу в текстовом поле. Он удаляет предыдущее содержимое текстового поля и вставляет новую матрицу.

В конце кода создается главное окно и запускается основной цикл обработки событий.

Рассмотрим как выглядят входные данные:

* создаётся окно ‘master’ с названием «Искусственная Иммунная Система»;
* создаётся матрица ‘matrix’ для хранения данных о рисунке, заполненная буквой ‘W’;
* создаётся ‘Canvas’ для рисования;
* создаётся ‘Text’ для отображения введённой матрицы;
* создаётся кнопка «Распознать»(‘recognize\_button’) для запуска процесса распознавания;
* создаются объекты класса ‘Lymphocyte’ с заданными паттернами букв для последующего распознания;
* происходит привязка обработчика к ‘Canvas’ для отслеживания рисования;

Выходные данные:

* печать распознанной матрицы в окно;
* всплывающее уведомление с результатом классификации антигена;
* очистка «Canvas»;
* счистка матрицы «matrix».

1. **Анализ эффективности работы иммунного алгоритма**

Для анализа эффективности работы разработанного иммунного алгоритма использовалось два параметра – время, затраченное программой на решение задачи и числа особей.

На рисунке 5 показана зависимость времени работы иммунного алгоритма от числа особей

Изображение выглядит как текст, линия, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 5 - Зависимость времени работы алгоритмов

от числа особей

Временная сложность данного кода зависит от нескольких факторов

Числа особей (лимфоцитов) в иммунной системе: чем больше элементов в этом списке, тем больше времени потребуется на классификацию антигена. В худшем случае, временная сложность будет определятся согласно формуле (3)

*O(n)* (3)

где

n - это число особей в иммунной системе.

Исходя из предоставленных данных, можно сделать вывод о том, что время выполнения алгоритма увеличивается с увеличением числа особей. Однако коэффициент изменения времени выполнения линейный - он увеличивается линейно с увеличением числа особей.

### ЗАКЛЮЧЕНИЕ

В ходе работы был разработан иммунный алгоритм для решения задач распознавания(распознавания одиночных символов). Был проведен анализ иммунного алгоритма и изучены различные структуры, применяемые в иммунном алгоритме.

При тестировании иммунного алгоритма решения задачи распознания рукописного текста были использованы наборы символов и образцов рукописного текста для проверки правильности распознания, были сгенерированы антитела на основе образцов рукописного текста.

Полученный иммунный алгоритм показал эффективность и способность определять рукописный почерк. Он демонстрирует перспективность применения иммунных алгоритмов в разных областях жизни.

### СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. **Колоколов, А. А.** Алгоритмы искусственной иммунной системы для вариантной задачи размещения телекоммуникационных центров // А. А. Колоколов. - Текст : электронный // cyberleninka.ru : [сайт]. - 2013. - URL: https://cyberleninka.ru/article/n/algoritmy-iskusstvennoy-immunnoy-sistemy-dlya-variantnoy-zadachi-razmescheniya-telekommunikatsionnyh-tsentrov (дата обращения: 29.09.2023).
2. **Касьян, К. Н.** Разработка модифицированного метода распознавания текста на стандартизированном изображении // К. Н. Касьян, В. В. Братчиков, В. В. Шкарлупило. - Текст : электронный // cyberleninka.ru : [сайт]. - 2015. - URL: https://cyberleninka.ru/article/n/razrabotka-modifitsirovannogo-metoda-raspoznavaniya-teksta-na-standartizirovannom-izobrazhenii (дата обращения: 09.11.2023).
3. **Лобанова, В. А.** Разработка нейросетевого алгоритма распознавания надписей на изображениях реальных сцен// В. А. Лобанова, Ю. А. Иванова. - Текст : электронный // cyberleninka.ru : [сайт]. - 2022. - URL: https://cyberleninka.ru/article/n/razrabotka-neyrosetevogo-algoritma-raspoznavaniya-nadpisey-na-izobrazheniyah-realnyh-stsen (дата обращения: 09.12.2023).
4. **Мозговой, А. А.** Проблемы существующих методик распознавания рукописного текста //А. А. Мозговой. - Текст : электронный // cyberleninka.ru : [сайт]. - 2019. - URL: https://cyberleninka.ru/article/n/problemy-suschestvuyuschih-metodik-opticheskogo-rasponavaniya-rukopisnogo-teksta (дата обращения: 10.12.2023).
5. **Чернышев, Ю. О.** Искусственные иммунные системы: обзор и современное состояние . Программные продукты и системы //Ю. О. Чернышев, Г. В. Григорьев,. - Текст : электронный // cyberleninka.ru : [сайт]. - 2014. - 20 январ. - URL: https://cyberleninka.ru/article/n/iskusstvennye-immunnye-sistemy-obzor-i-sovremennoe-sostoyanie (дата обращения: 25.11.2023).
6. Описание используемой иммунной сети в задаче распознавания образов: [сайт]. - 2015. - URL: https://vuzdoc.org/242710/tehnika/opisanie\_ispolzuemoy\_immunnoy\_seti\_zadache\_raspoznavaniya\_obrazov#aftercont (дата обращения: 22.11.2023). - Текст : электронный.

### ПРИЛОЖЕНИЕ А

**Основная программа**

import tkinter as tk  
from tkinter import Canvas, Button, Text, messagebox  
from functools import partial  
  
class Lymphocyte:  
 def \_\_init\_\_(self, black\_pixels, white\_pixels):  
 self.black\_pixels = black\_pixels  
 self.white\_pixels = white\_pixels  
  
 def calculate\_affinity(self, antigen\_matrix):  
 total\_pixels = len(self.black\_pixels) + len(self.white\_pixels)  
 black\_pixels\_matched = sum(antigen\_matrix[i][j] == 'B' for i, j in self.black\_pixels)  
 affinity = black\_pixels\_matched / total\_pixels  
 return affinity  
  
class ImmuneSystemApp:  
 def \_\_init\_\_(self, master):  
 self.master = master  
 self.master.title("Искусственная Иммунная Система")  
  
 # Создаем матрицу для хранения данных о рисунке  
 self.matrix\_size = 5  
 self.matrix = [['W'] \* self.matrix\_size for \_ in range(self.matrix\_size)]  
  
 # Создаем Canvas для рисования  
 self.canvas = Canvas(self.master, width=self.matrix\_size \* 150, height=self.matrix\_size \* 140, bg="white")  
 self.canvas.pack()  
  
 # Создаем Text для отображения введенной матрицы  
 self.input\_matrix\_text = Text(self.master, height=self.matrix\_size, width=self.matrix\_size \* 2)  
 self.input\_matrix\_text.pack()  
  
 # Создаем кнопку для подачи на распознавание  
 self.recognize\_button = Button(self.master, text="Распознать", command=self.recognize)  
 self.recognize\_button.pack()  
  
 # Привязываем обработчик к Canvas для отслеживания рисования  
 self.canvas.bind("<B1-Motion>", self.draw)  
  
 # Создаем лимфоциты с паттернами букв  
 self.lymphocytes = [  
 Lymphocyte([(0, 1), (0, 2), (0, 3), (1, 1), (1, 3), (2, 1), (2, 3), (3, 1), (3, 3), (4, 1), (4, 2), (4, 3)], []),  
 Lymphocyte([(0, 2), (1, 2), (2, 2), (3, 2), (4, 2)], []),  
 Lymphocyte([(0, 1), (0, 2), (0, 3), (1, 3), (2, 1), (2, 2), (2, 3), (3, 1), (4, 1), (4, 2), (4, 3)], []),  
 Lymphocyte([(0, 1), (0, 2), (0, 3), (0, 4), (1, 4), (2, 1), (2, 2), (2, 3), (2, 4), (3, 4), (4, 1), (4, 2), (4, 3), (4, 4)], []),  
 Lymphocyte([(0, 1), (0, 3), (1, 1), (1, 3), (2, 1), (2, 2), (2, 3), (3, 3), (4, 3)], []),  
 Lymphocyte([(0, 1), (0, 2), (0, 3), (1, 1), (2, 1), (2, 2), (2, 3), (3, 3), (4, 1), (4, 2), (4, 3)], []),  
 Lymphocyte([(0, 1), (0, 2), (0, 3), (1, 1), (2, 1), (2, 2), (2, 3), (3, 1), (3, 3), (4, 1), (4, 2), (4, 3)], []),  
 Lymphocyte([(0, 1), (0, 2), (0, 3), (2, 3), (3, 3), (4, 3)], []),  
 Lymphocyte([(0, 1), (0, 2), (0, 3), (1, 1), (1, 3), (2, 1), (2, 2), (2, 3), (3, 1), (3, 4), (4, 1), (4, 2), (4, 3)], []),  
 Lymphocyte([(0, 1), (0, 2), (0, 3), (1, 1), (1, 3), (2, 1), (2, 2), (2, 3), (3, 3), (4, 1), (4, 2), (4, 3)], []),  
 ]  
  
 def draw(self, event):  
 # Определяем координаты клетки, в которой происходит рисование  
 x, y = event.x // 150, event.y // 150  
  
 # Записываем информацию о рисовании в матрицу  
 self.matrix[y][x] = 'B'  
  
 # Обновляем Text с введенной матрицей  
 self.update\_input\_matrix\_text()  
  
 # Рисуем черный квадрат на Canvas  
 self.canvas.create\_rectangle(x \* 150, y \* 150, (x + 1) \* 150, (y + 1) \* 150, fill="black")  
  
 def recognize(self):  
 # Пример: вывод матрицы на консоль  
 print("Распознанная матрица:")  
 for row in self.matrix:  
 print(row)  
  
 # Классификация антигена  
 classified\_antigen = self.classify\_antigen(self.matrix, 0.6)  
  
 if classified\_antigen is not None:  
 messagebox.showinfo("Результат", f"Антиген классифицирован как {classified\_antigen}")  
 else:  
 messagebox.showinfo("Результат", "Ни один антиген не классифицирован")  
  
 self.canvas.delete("all")  
 self.clear\_matrix()  
  
 def classify\_antigen(self, antigen\_matrix, threshold):  
 max\_affinity = 0  
 classified\_antigen = None  
  
 for antigen\_id, lymphocyte in enumerate(self.lymphocytes):  
 affinity = lymphocyte.calculate\_affinity(antigen\_matrix)  
  
 # Если аффинность выше порога и больше текущей максимальной аффинности  
 if affinity > max\_affinity and affinity >= threshold:  
 max\_affinity = affinity  
 classified\_antigen = antigen\_id  
  
 return classified\_antigen  
  
 def clear\_matrix(self):  
 # Очищаем матрицу  
 self.matrix = [['W'] \* self.matrix\_size for \_ in range(self.matrix\_size)]  
  
 # Обновляем Text с введенной матрицей  
 self.update\_input\_matrix\_text()  
  
 def update\_input\_matrix\_text(self):  
 # Отображение введенной матрицы в Text  
 self.input\_matrix\_text.delete(1.0, tk.END)  
 for row in self.matrix:  
 self.input\_matrix\_text.insert(tk.END, ' '.join(row) + '\n')  
  
# Создаем главное окно  
root = tk.Tk()  
app = ImmuneSystemApp(root)  
root.mainloop()