

ОГЛАВЛЕНИЕ

М. М. Благирев, А. О. Костыренков ИСПОЛЬЗОВАНИЕ ПРОТОКОЛОВ REST API и WEBSOCKET ДЛЯ СТРУКТУРИЗАЦИИ ТРЕХЗВЕННОГО УРОВНЯ ЭМЕРДЖЕНТНЫХ СИСТЕМ И ОТОБРАЖЕНИЯ МЕДИАСИСТЕМ	415–428
М. В. Бобырь, Н. А. Милостная, Н. И. Храпова О ПОДХОДЕ К ДЕТЕКТИРОВАНИЮ ДВИЖЕНИЯ ПЕШЕХОДОВ МЕТОДОМ ГИСТОГРАММ НАПРАВЛЕННЫХ ГРАДИЕНТОВ	429–447
И. А. Валишин ПРИМЕНЕНИЕ МЕТОДОВ КОМПЬЮТЕРНОГО ЗРЕНИЯ К РАСПОЗНАВАНИЮ СТАРОТАТАРСКОГО ТЕКСТА	448–477
Р. Ф. Давлетшин, И. С. Шахова ПРОГРАММНОЕ СРЕДСТВО ОПТИМИЗАЦИИ ПРОЦЕССОВ ВИДЕОПРОИЗВОДСТВА	478–502
А. И. Еникеева, Р. А. Бурнашев, Р. Р. Фарахов РАЗРАБОТКА НА ОСНОВЕ НЕЧЕТКОЙ ЛОГИКИ ЭКСПЕРТНОЙ СИСТЕМЫ ДЛЯ ДИАГНОСТИКИ ПНЕВМОНИИ	503–532
А. Д. Неманов, И. С. Шахова АВТОМАТИЗАЦИЯ СОРТИРОВКИ МАТЕРИАЛОВ ПО ТЕКСТУ СЦЕНАРИЯ ДЛЯ ВИДЕОМОНТАЖА	533–557
К. С. Николаев УЧЁТ СТРУКТУРЫ ДОКУМЕНТА В МЕТОДЕ АВТОМАТИЧЕСКОГО АННОТИРОВАНИЯ МАТЕМАТИЧЕСКИХ ПОНЯТИЙ В ОБРАЗОВАТЕЛЬНЫХ ТЕКСТАХ	558–577
Н. Р. Нурлыгаянов, В. В. Кугуракова ПОДХОД К СОЗДАНИЮ КОРПУСА ТЕКСТОВ ВИДЕОИГР НА ОСНОВЕ УНИВЕРСАЛЬНОЙ СТРУКТУРЫ	578–597
А. Р. Нурутдинов НЕЙРОСЕТЕВАЯ АРХИТЕКТУРА ВОПЛОЩЕННОГО ИНТЕЛЛЕКТА	598–655

А. С. Савинчева, А. А. Ференец

**АВТОМАТИЗАЦИЯ ЧТЕНИЯ СВЯЗАННЫХ ДАННЫХ
ИЗ РЕЛЯЦИОННЫХ И НЕРЕЛЯЦИОННЫХ БАЗ ДАННЫХ В КОНТЕКСТЕ
ИСПОЛЬЗОВАНИЯ СТАНДАРТА JRA**

656–678

А. Г. Уймин, В. С. Греков

**ПРИМЕНЕНИЕ АЛГОРИТМА ДУГЛАСА–ПЕККЕРА В ВОПРОСАХ ОНЛАЙН-
АУТЕНТИФИКАЦИИ ИНСТРУМЕНТОВ УДАЛЁННОЙ РАБОТЫ ПРИ ПОДГОТОВКЕ
СПЕЦИАЛИСТОВ УКРУПНЁННОЙ ГРУППЫ СПЕЦИАЛЬНОСТЕЙ 10.00.00
«ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ»**

679–694

УДК 004.4

ИСПОЛЬЗОВАНИЕ ПРОТОКОЛОВ REST API и WEBSOCKET ДЛЯ СТРУКТУРИЗАЦИИ ТРЕХЗВЕННОГО УРОВНЯ ЭМЕРДЖЕНТНЫХ СИСТЕМ И ОТОБРАЖЕНИЯ МЕДИАСИСТЕМ

М. М. Благирев¹ [0009-0008-2853-3411], А. О. Костыренков² [0009-0007-0294-694X]

^{1, 2}МИРЭА – Российский Технологический Университет, проспект Вернадского, 78, г. Москва, 119454

¹blagirevm@list.ru, ²kostyrenkov@mirea.ru

Аннотация

Проведен анализ скорости и эффективности передачи данных с использованием протоколов WebSocket и REST API. Для сравнения скорости обработки потоковых объектов и выявления более надежной технологии для разработки API-интерфейсов использованы разложения базовых функций в ряды Тейлора и Фурье. В результате выявлено, что REST API является более быстрым и доступным ресурсом для передачи информационных данных в побитовом преобразовании, а масштабируемость этого протокола преобладает в количестве обрабатываемых единиц, что позволяет расширить количество проводимых тестов.

Ключевые слова: масштабируемость, протоколирование, структуризация, REST API, WebSocket.

ВВЕДЕНИЕ

Современные веб-приложения нуждаются в эффективных методах передачи данных между клиентом и сервером. REST API и WebSocket являются двумя ключевыми подходами для решения этой задачи [1, 2].

REST API использует протокол HTTP и шаблон MVC, обеспечивая сбор, обработку и представление данных клиенту, подходит для операций CRUD (CREATE, READ, UPDATE, DELETE) и стандартизирует протоколы и форматы данных. Однако REST API требует для каждого запроса нового соединения с сервером, что может быть неэффективным при больших объемах данных.

WebSocket, напротив, устанавливает постоянное двустороннее соединение между клиентом и сервером, обеспечивая обмен данными в реальном времени без необходимости постоянных запросов, что снижает нагрузку на систему и ускоряет передачу данных. Это особенно важно для приложений с высокой нагрузкой, однако настройка WebSocket требует большего объема программного кода и может столкнуться с проблемами совместимости.

Сравнение двух названных подходов позволяет выявить наиболее эффективные решения для различных сценариев их использования в веб-приложениях.

1. REST API: РЕПРЕЗЕНТАТИВНОЕ СОСТОЯНИЕ МОДЕЛИ ПЕРЕДАЧИ И ОБРАБОТКИ ДАННЫХ

Составление и отработка протокола HTTP при обработке запроса и подхода REST позволяют обеспечить взаимосвязь между сервером и клиент-серверной частью, используя шаблон проектирования MVC при трехуровневой обработке запроса: сбор данных сервером (чтение и обучение математической модели); обработка информации (путем использования инструментов проектирования модели разработки и последующей реорганизации побитового состояния структуры в информационной системе); вывод на клиентскую часть пользователя предоставляемой обработанной информации [3].

Эмерджентные свойства обеспечиваются здесь путем взаимодействия дочерних классов и функций от основной иерархической инфографики, которые способны обеспечить синергетический эффект, связывая основную ступень проектирования и последующий этап отработки (компиляции/интерпретации) программного кода, пост- и препроцессинг для структуризации данных медиа единиц, например, передачи или получения обработанных графических объектов [4].

В настоящей работе рассмотрены использование и применение REST API методов GET, POST, PUT, DELETE для реализации основного принципа акронима CRUD баз данных (CREATE, READ, UPDATE, DELETE), а применение стека технологий позволит разработчикам модернизировать подход к обработке графических медиа с опорой на обширный спектр библиотек взаимодействия команд [5].

При оптимизации появляется возможность применить RESTful-сервисы на многофункциональных языках программирования, которые могут интерпретироваться на сторонних ресурсах или при облачном взаимодействии систем сервера с клиентской частью. Достоинство использования RESTful-сервисов обусловлено

критериями стандартизации протоколирования информационной библиотеки и статическими форматами данных [6].

В ходе проведенного исследования был использован один из немаловажных факторов ресурса REST API – масштабируемость. В контексте разработки информационной системы она способствует горизонтальному росту производительности, получая более значительные объемы запросов, а также большее количество клиентских частей пользователей, которые обращаются на сервер через протокол HTTP/HTTPS.

HTTP/HTTPS – соответственно защищенный и незащищенный протоколы передачи гипертекста между клиентом и сервером. Протокол HTTPS медленнее, чем HTTP, из-за потребления системных ресурсов для установки защищенного ресурса процессом SSL/TLS [7]. HTTP-протокол позволяет быстрее осуществлять передачу данных с сервера клиенту, чем HTTPS. RESTful требует установки нового соединения с сервисами или монолит-системами, что приводит к проблеме работоспособности в реальном времени: работа запрос–ответ не является быстропоточковой передачей данных при больших объемах количественно-качественных данных объекта. Таким образом, целесообразно рассмотреть архитектуру REST API (рис. 1) и сравнить ее с технологией WebSocket двустороннего соединения.

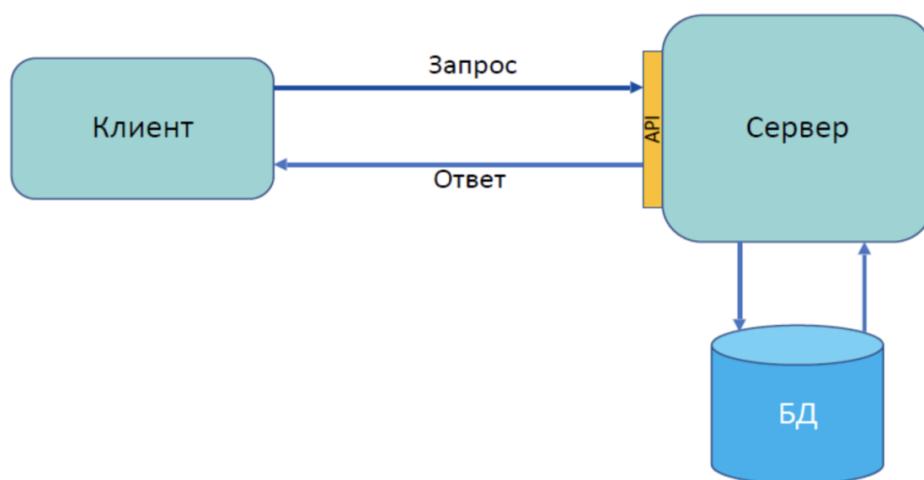


Рис. 1. Архитектура REST API.

2. WEBSOCKET – ДВУСТОРОННЕЕ СОЕДИНЕНИЕ ИНТЕРФЕЙСОВ

WebSocket является протоколом двусторонней связи между клиентом и сервером через единое постоянное соединение. В отличие от REST API, WebSocket позволяет обмениваться данными в реальном времени без необходимости постоянной отправки запросов на сервер для получения потокового ответа, что меньше нагружает систему, и быстрее напрямую отправляет объекты информационных данных на клиентскую часть.

Эффективность передачи данных поддерживается за счет непрерывного соединения между клиентом и сервером, тем самым сокращается объём передаваемых данных путем постоянной связи между клиентом и сервером, снижая нагрузку и на сеть, и на сервер. Масштабируемость для обработки большого количества одновременных подключений позволяет потоково обработать множество обращений пользователей для приложений с высокой нагрузкой [8].

Однако WebSocket, по сравнению с REST API, требует существенно большего количества описанных строк программного кода для подключения клиента к серверу по протоколу и значению ключа/токена. Многофункциональность технологии WebSocket ниже, чем у REST API, ввиду малой поддержки устаревших браузерных решений, так как сетевые прокси не всегда поддерживают соединение WebSocket, что создает пользователям весомые проблемы совместимости передачи пакетов данных.

Ключевым в WebSocket является распределение клиентских подключений в стендах, распределяющих отправку и переадресацию для обработки данных несколькими серверами для повышения стабильности, масштабируемости и надежности использования серверной составляющей [9].

Load Balancer как шаблон масштабируемости принимает входящие интернет-соединения и распределяет потоки подключения на имеющиеся сервера, равномерно распределяя нагрузку на сетевую установку. Серверы во время обработки сигнала, по которому отправляют запрос на клиентскую часть, соединяют пользователя с ответами сервера через балансировщик нагрузки и тем самым снижают нагрузку на сеть и требуют меньшее количество ресурсов ЭВМ [10].

На рис. 2 представлена архитектура WebSocket двустороннего соединения.

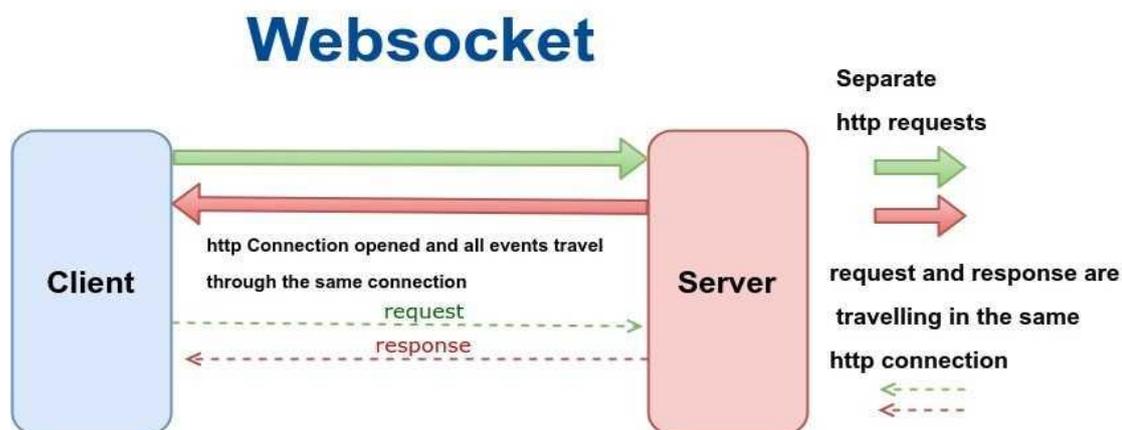


Рис. 2. Архитектура WebSocket двустороннего соединения.

Ниже на основе математического моделирования проведен анализ скорости обработки и передачи ответа протоколами двух технологий WebSocket и REST API для выявления более быстрой и надежной технологии в разработке API интерфейсов.

3. АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ И ЭФФЕКТИВНОСТИ REST API И WEBSOCKET

Для анализа производительности и эффективности REST API и WebSocket были использованы разложения базовых функций в ряды Тейлора и Фурье. Рассмотрим, как каждый из этих методов может быть применен, и проиллюстрируем это примерами вычислений [11].

Как известно, ряд Тейлора используется для аппроксимации функций в некоторой точке a и для функции $f(x)$ в окрестности этой точки имеет вид

$$f(x) = f(a) + f'(a)(x - a) + \frac{f''(a)}{2!}(x - a)^2 + \frac{f'''(a)}{3!}(x - a)^3 + \dots \quad (1)$$

Выбор точки a для аппроксимации функций рядом Тейлора влияет на точность аппроксимации вблизи этой точки: чем ближе значение x к точке a , тем точнее аппроксимация. Обычно эту точку выбирают в зависимости от области, в которой требуется высокая точность аппроксимации.

Пусть x – размер запроса в килобайтах, а t – время обработки запроса в миллисекундах. Предположим, что у нас есть функции $f(x, t)$ и $g(x, t)$ времени отклика REST API и WebSocket соответственно, зависящие как от размера запроса, так и от времени его обработки. Будем считать, что эти функции представлены в виде суммы отрезков двух рядов Фурье $b + cx + dx^2 + et + kt^2$. Коэффициенты в

этих выражениях зависят от точной реализации и характеристик системы и настраиваются на основе измерений производительности и анализа работы системы в реальных условиях. Они могут быть получены путем сбора статистики времени отклика при различных входных данных и их анализа с помощью методов регрессии или других статистических методов [12]. В итоге имеем

$$f(x, t) = 0.5 + 0.005x + 0.00015x^2 + 0.2t + 0.05t^2,$$

$$g(x, t) = 0.85 + 0.002x + 0.00001x^2 + 0.3t + 0.04t^2.$$

Ниже приведены расчеты (таблицы 1 и 2), где коэффициенты перед переменными были взяты как средние значения за время проведения 24 тестов при использовании вычислительных ресурсов системы и запуска программного обеспечения.

Таблица 1. Расчет для REST API

Коэффициенты	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>k</i>
Среднее значение	0.5	0.005	0.00015	0.2	0.05
Тест 1	0.5072	0.0035	0.00005	0.1664	0.0552
Тест 2	0.4946	0.0009	0.00014	0.2349	0.0534
Тест 3	0.5021	0.0021	0.00027	0.1957	0.0462
Тест 4	0.4956	0.0006	0.00012	0.2065	0.0533
Тест 5	0.4984	0.0034	0.00004	0.2113	0.0489
Тест 6	0.4999	0.0019	0.00024	0.1920	0.0487
Тест 7	0.5013	0.0040	0.00010	0.2068	0.0653
Тест 8	0.4985	0.0004	0.00011	0.1961	0.0563
Тест 9	0.5001	0.0037	0.00032	0.1987	0.0442
Тест 10	0.4953	0.0040	0.00023	0.2048	0.0426
Тест 11	0.4987	0.0018	0.00011	0.2193	0.0519
Тест 12	0.5002	0.0045	0.00017	0.1973	0.0564
Тест 13	0.4978	0.0010	0.00026	0.2204	0.0601
Тест 14	0.5006	0.0027	0.00020	0.1948	0.0655
Тест 15	0.4987	0.0010	0.00025	0.1991	0.0446
Тест 16	0.4982	0.0006	0.00007	0.1832	0.0556

Тест 17	0.4998	0.0043	0.00031	0.2218	0.0444
Тест 18	0.4992	0.0021	0.00011	0.2069	0.0633
Тест 19	0.5014	0.0036	0.00036	0.2252	0.0518
Тест 20	0.4983	0.0003	0.00012	0.1941	0.0518
Тест 21	0.4995	0.0046	0.00012	0.2287	0.0428
Тест 22	0.4976	0.0032	0.00006	0.2206	0.0361
Тест 23	0.4971	0.0034	0.00013	0.1954	0.0565
Тест 24	0.5034	0.0003	0.00015	0.2234	0.0578

Таблица 2. Расчет для WebSocket

Коэффициенты	b	c	d	e	k
Среднее значение	0.85	0.002	0.00001	0.3	0.04
Тест 1	0.92647443	0.00141946	0.0000084	0.307672	0.0552
Тест 2	0.93419272	0.00164982	0.0000195	0.295145	0.0534
Тест 3	0.9577212	0.00107814	0.0000077	0.301847	0.0463
Тест 4	0.77473834	0.00164851	0.0000217	0.298783	0.0533
Тест 5	0.86819741	0.00196397	0.0000158	0.298532	0.0364
Тест 6	0.84029113	0.00150823	0.0000071	0.298520	0.0487
Тест 7	0.81684921	0.00179711	0.0000192	0.293258	0.0653
Тест 8	0.94923999	0.00155853	0.0000186	0.301621	0.0536
Тест 9	0.83967457	0.00131092	0.0000216	0.293761	0.0442
Тест 10	0.92380555	0.00204906	0.0000094	0.296559	0.0426
Тест 11	0.81506841	0.00110384	0.0000212	0.298147	0.0519
Тест 12	0.87191482	0.00191879	0.0000119	0.294806	0.0564
Тест 13	0.92505202	0.00071384	0.0000124	0.303355	0.0601
Тест 14	0.84436529	0.00111471	0.0000163	0.29347	0.0655
Тест 15	0.87484921	0.00145167	0.0000116	0.301448	0.0446
Тест 16	0.86617473	0.00086182	0.0000062	0.295909	0.0556

Тест 17	0.83200251	0.00158115	0.0000213	0.295078	0.0444
Тест 18	0.93615828	0.00121738	0.0000105	0.299004	0.0633
Тест 19	0.94317695	0.00209512	0.0000094	0.304752	0.0518
Тест 20	0.80872571	0.00186102	0.0000113	0.293135	0.0518
Тест 21	0.79983723	0.00203884	0.0000141	0.294890	0.0428
Тест 22	0.87970266	0.00062239	0.0000126	0.295250	0.0361
Тест 23	0.83465939	0.00168432	0.0000067	0.301058	0.0564
Тест 24	0.89280647	0.00156213	0.0000119	0.296284	0.0538

Ниже приведено обобщение результатов, с опорой на ранее проведенные тесты (таблица 3), где представлены производные функций $f(x, t)$ и $g(x, t)$ времени отклика REST API и WebSocket.

Таблица 3. Обобщение результатов

x	t	$f(x,t)$	$f'(x,t)$	$f''(x,t)$	$g(x,t)$	$g'(x,t)$	$g''(x,t)$	REST API	WebSocket
1000	52	301.1	8.405	0.1003	135.46	4.482	0.80002	309.6	140.7
10000	143	1661.01	44.505	0.1003	1877.9	11.942	0.80002	1705.6	1889.6
50000	782	40598.3	228.41	0.1003	49781	63.682	0.80002	40726	49844.0

Таким образом, в ходе экспериментов по результатам расчетов на малых объемах данных эффективнее оказался WebSocket. На больших объемах лучше справляется с задачей REST API.

Ряд Фурье можно использовать для анализа периодических колебаний в данных, например, в передаче сообщений. Если время отклика или задержки передается периодически, можно разложить соответствующую функцию в ряд Фурье.

Представим, в качестве примера, что задержка $T(t)$ передается как 2π -периодическая четная функция $T(t)=\cos t$. Ниже приведён расчет коэффициентов для ряда Фурье [13], полученные результаты представлены в таблице 4.

Таблица 4. Расчет для ряда Фурье

REST API	
t	$T(t)$
52	0.16
143	0.057
782	0.967
234	0.047
112	0.455
7654	0.471
2134	0.653
WebSocket	
t	$T(t)$
43	0.555
164	0.804
642	0.440
154	0.998
164	0.804
5839	0.343
1782	0.754

Для исследования адаптационных алгоритмов на базе интерфейсов REST API и WebSocket была разработана система, включающая три уровня структуризации данных: сбор данных, их обработка и отображение. В качестве данных использовались текстовые потоки данных. Выполнение работы происходило в следующем порядке:

- Подготовка данных: исходные данные были преобразованы в удобный для анализа формат с использованием предобработки, включающей нормализацию и фильтрацию.
- Представление данных: преобразованные данные представляются в виде отрезков рядов Тейлора и Фурье для аппроксимации и анализа периодических составляющих.
- Интеграция с REST API и WebSocket: обработанные данные передавались через REST API для хранения и дальнейшего анализа, а также через WebSocket для реалтайм-обновлений и визуализации на клиентской стороне.
- Структуризация данных: обработанные данные структурировались в трехзвенной модели, включающей уровень начальных данных, промежуточный

уровень обработанных данных и финальный уровень визуализированных данных.

Сводка итогов применения рядов Тейлора и Фурье представлена в таблице 5.

Таблица 5. Сводка итогов применения ряда Тейлора и ряда Фурье

Метод	Преимущества	Недостатки	Применимость
Ряд Тейлора	Гладкие переходы между уровнями	Ограниченная точность при больших значениях	Аппроксимация функций
Ряд Фурье	Анализ периодических сигналов	Неэффективен для непериодических сигналов	Анализ временных рядов

Таким образом, если система в первую очередь нуждается в мгновенном обновлении данных, при этом в режиме реального времени и при эффективной передаче данных, лучше использовать WebSocket.

Если система нуждается в более простой реализации, то REST API станет отличным решением рассмотренной задачи.

ЗАКЛЮЧЕНИЕ

REST API требует нового соединения для каждого запроса, что может привести к снижению эффективности при больших объемах данных.

WebSocket обеспечивает постоянное двустороннее соединение между клиентом и сервером, что позволяет обмениваться данными в реальном времени без необходимости постоянных запросов. Это значительно снижает нагрузку на систему и ускоряет передачу данных, что особенно важно для приложений с высокой нагрузкой. Тем не менее, настройка WebSocket может потребовать большего количества программного кода и столкнуться с проблемами совместимости.

Сравнение двух подходов позволило выявить наиболее эффективные решения для различных сценариев использования в веб-приложениях. REST API подходит для структурированных, стандартизированных операций с данными, тогда как WebSocket более эффективен для приложений, требующих быстрой и постоянной передачи данных. Оптимальное применение настоящих технологий способствует оптимизации трёхуровневых звеньев по кластеризации передачи протоколов

клиент-серверной информационной системы и обработки файловых составляющих. Таким образом, использование WebSocket, в отличие от REST API, позволяет систематически быстрее преобразовать передачу данных в многопоточном режиме пользования клиентских запросов на сервер, тем самым было получено наиболее эффективное решение для отправки протоколированных пакетов в реальном времени.

СПИСОК ЛИТЕРАТУРЫ

1. REST API Tutorial // REST API. URL: <https://restfulapi.net/rest/>
2. API WebSocket (WebSockets) // MDN Web Docs.
URL: https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API
3. *Куляшова Н.М.* Программное обеспечение и алгоритмы распознавания адресных структур информационные технологии // Информационные технологии. 2021. Т. 27, № 5. С. 275–280.
4. *Oznamets V.V., Ndayishimiye D., Kovalenko N.I.* Spatial modeling when creating a coordinate base (on the example of the republic of burundi) // Slavic Forum. 2023. № 1 (39). С. 302–314.
5. *Chakraborty S., Aithal P.S.* CRUD operation on wordpress database using C# and REST API // International Journal of Applied Engineering and Management Letters. 2023. С. 130–138.
6. *Анисимов В.И., Васильев С.А., Евдокимов И.А., Тарасова О.Б.* Обзор методов автоматизации тестирования и документирования серверного интерфейса, основанного на архитектуре RESTFUL // Информационные технологии в проектировании и производстве. 2020. № 2 (178). С. 45–48.
7. *Мартыненко И.В.* Основные этапы развития криптографических протоколов SSL/TLS и IPSEC // Прикладная дискретная математика. 2021. № 51. С. 31–67.
8. *Малсугенов О.В., Чипига А.Ф., Львова А.П.* Повышение эффективности беспроводного оптического канала передачи данных в видимом диапазоне световых волн // В сборнике: Информационные технологии интеллектуальной поддержки принятия решений. Труды VII Всероссийской научной конференции (с приглашением зарубежных ученых). В 3-х томах. 2019. С. 65–70.

9. Дерябин Е.А., Готская И.Б. Анализ круговой задержки при использовании веб-протоколов HTTP И WEBSOCKET // Студенческий. 2018. № 10-2 (30). С. 32–34.
 10. Yang L., Yao H., Wang J. et al. Multi-UAV-Enabled Load-Balance Mobile-Edge Computing for IoT Networks // IEEE Internet of Things Journal. 2020. V. 7, No. 8. P. 6898–6908.
 11. Кузьмичев Н.Д. Применение рядов Тейлора–Фурье для численного и экспериментального определения производных изучаемой зависимости // Журнал Средневолжского математического общества. 2011. Т. 13, № 2. С. 70–80.
 12. Касаткина Т.И., Гречишников Е.В., Дидрих В.Е., Соловьев А.С. Математические методы моделирования и алгоритмы систем анализа и обработки информации для исследований динамики сложных систем // Вестник Воронежского института ФСИН России. 2017. № 4. С. 48–58.
 13. Романова Л.Д., Шаркунова Т.А., Елисеева Т.В. Интегральные преобразования: учеб. пособие. Пенза: Изд-во ПГУ, 2015. 80 с.
-

USE OF REST API AND WEBSOCKET INTERFACES ALGORITHMS FOR STRUCTURING THE THREE-LINK LEVEL OF EMERGENT SYSTEMS AND DISPLAYING MEDIA SYSTEMS

M. M. Blagirev¹ [0009-0008-2853-3411], A. O. Kostyrenkov² [0009-0007-0294-694X]

^{1, 2}MIREA – Russian Technological University, Vernadsky Avenue, 78, Moscow, 119454

¹blagirevm@list.ru, ²kostyrenkov@mirea.ru

Abstract

An analysis of the speed and efficiency of data transfer using the WebSocket and REST API protocols was carried out. To compare the speed of processing stream objects and identify a more reliable technology for developing APIs, expansions of basic functions in Taylor and Fourier series were used. As a result, it was revealed that the REST API is a faster and more accessible resource for transmitting information data in a bit-wise transformation, and the scalability of this protocol prevails in the number of processed units, which allows expanding the number of tests performed.

Keywords: scalability, logging, structuring, REST API, WebSocket.

REFERENCES

1. REST API Tutorial // REST API. URL: <https://restfulapi.net/rest/>
2. API WebSocket (WebSockets) // MDN Web Docs.
URL: https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API
3. *Kulyashova N.M.* Software and algorithms for recognizing address structures, information technologies // Information Technology. 2021. T. 27, no. 5. P. 275–280.
4. *Oznamets V.V., Ndayishimiye D., Kovalenko N.I.* Spatial modeling when creating a coordinate base (on the example of the republic of burundi) // Slavic Forum. 2023. No. 1 (39). P. 302–314.
5. *Chakraborty S., Aithal P.S.* CRUD operation on wordpress database using C# and REST API // International Journal of Applied Engineering and Management Letters. 2023. P. 130–138.
6. *Anisimov V.I., Vasiliev S.A., Evdokimov I.A., Tarasova O.B.* Review of methods for automating testing and documenting a server interface based on the RESTFUL architecture // Information technologies in design and production. 2020. No. 2 (178). P. 45–48.
7. *Martynenkov I.V.* Main stages of development of cryptographic protocols SSL/TLS and IPSEC // Applied discrete mathematics. 2021. No. 51. P. 31–67.
8. *Malsugenov O.V., Chipiga A.F., Lvova A.P.* Increasing the efficiency of a wireless optical data transmission channel in the visible range of light waves // In the collection: Information technologies for intelligent decision support. Proceedings of the VII All-Russian Scientific Conference (with the invitation of foreign scientists). In 3 volumes. 2019. P. 65–70.
9. *Deryabin E.A., Gotskaya I.B.* Analysis of round-trip delay when using web protocols HTTP and WEBSOCKET // Student. 2018. No. 10-2 (30). P. 32–34.
10. *Yang L., Yao H., Wang J, et al.* Multi-UAV-Enabled Load-Balance Mobile-Edge Computing for IoT Networks // IEEE Internet of Things Journal. 2020. V. 7, No. 8. P. 6898–6908.

11. *Kuzmichev N.D.* Application of Taylor-Fourier series for the numerical and experimental determination of derivatives of the studied dependence // Journal of the Middle Volga Mathematical Society. 2011. V. 13, No. 2. P. 70–80.

12. *Kasatkina T.I., Grechishnikov E.V., Diedrich V.E., Solovyov A.S.* Mathematical methods of modeling and algorithms for analysis and information processing systems for studying the dynamics of complex systems // Bulletin of the Voronezh Institute of the Federal Penitentiary Service of Russia. 2017. No. 4. P. 48–58.

13. *Romanova L.D., Sharkunova T.A., Eliseeva T.V.* Integral transformations: textbook. allowance. Penza: PSU Publishing House, 2015. 80 p.

СВЕДЕНИЯ ОБ АВТОРАХ



БЛАГИРЕВ Михаил Михайлович – ассистент кафедры инструментального и прикладного программного обеспечения Института информационных технологий МИРЭА – Российского Технологического Университета.

Mikhail Mikhailovich BLAGIREV – assistant of the department of instrumental and applied software of the Institute of Information Technologies MIREA – Russian Technological University.

email: blagirevm@list.ru;

ORCID: 0009-0008-2853-3411



КОСТЫРЕНКОВ Алексей Олегович – ассистент кафедры инструментального и прикладного программного обеспечения Института информационных технологий МИРЭА – Российского Технологического Университета.

Alexey Olegovich KOSTYRENKOV – assistant of the department of instrumental and applied software of the Institute of Information Technologies MIREA – Russian Technological University.

email: kostyrenkov@mirea.ru;

ORCID: 0009-0007-0294-694X

Материал поступил в редакцию 20 июня 2024 года

О ПОДХОДЕ К ДЕТЕКТИРОВАНИЮ ДВИЖЕНИЯ ПЕШЕХОДОВ МЕТОДОМ ГИСТОГРАММ НАПРАВЛЕННЫХ ГРАДИЕНТОВ

М. В. Бобырь¹ [0000-0002-5400-6817], Н. А. Милостная² [0000-0002-3779-9165],
Н. И. Храпова³ [0000-0001-7947-1427]

¹⁻³Юго-Западный государственный университет, Курская обл., Курск, 305040,
Россия

¹maxbobyrg@gmail.com, ²nat_mil@mail.ru, ³KhrapovaNI@yandex.ru

Аннотация

Рассмотрен подход к автоматическому распознаванию движения людей на пешеходном переходе. Он включает две основные процедуры, для каждой из которых приведены команды программного кода на языке программирования C# с помощью библиотеки компьютерного зрения EMGU. На первой процедуре с помощью комбинации методов гистограмм направленных градиентов и опорных векторов осуществляется детектирование пешеходов. Вторая процедура позволяет считывать кадры из видеопоследовательности и обрабатывать их. Данный подход позволяет детектировать движения людей на пешеходном переходе без использования специализированных нейронных сетей. При этом метод, предложенный в статье, продемонстрировал достаточную надежность распознавания движения людей, что свидетельствует о его применимости в реальных условиях.

Ключевые слова: распознавание движения пешеходов, EMGU, гистограммы направленных градиентов, метод опорных векторов

ВВЕДЕНИЕ

Детектирование движения людей на пешеходном переходе является важной задачей для повышения безопасности дорожного движения и улучшения работы систем управления дорожным движением. В рамках дуальных технологий подобные системы могут использоваться в задачах определения перемещений людей внутри помещения [1]. Существует несколько подходов и алгоритмов для решения этой задачи, которые условно делятся на следующие группы.

К первой группе относятся модели, основанные на классических методах компьютерного зрения, в частности, модели фонового вычитания [1] и модели обнаружения краев и построения на их основе контуров объектов.

Ко второй группе следует отнести методы машинного обучения [2], использующие алгоритм Хаара или метод анализа признаков гистограмм направленных градиентов (HOG – Histogram of Oriented Gradients) в сочетании с машиной опорных векторов (SVM – Support Vector Machine).

Третья группа включает такие подходы к детектированию объектов, основанные на моделях глубокого обучения, как сверточные нейронные сети (CNN) [3] или рекуррентные нейронные сети (RNN).

Алгоритмы первой группы с помощью метода фонового вычитания выделяют движущиеся объекты за счет вычитания текущего кадра из предыдущего с помощью различных фильтров, таких как приближенный медианный метод, метод скользящего среднего, адаптивная гауссовская модель [4] или экспоненциально взвешенное скользящее среднее [5]. Данная задача упрощается, если перед вычитанием фона уменьшить количество информации путем применения фильтра Канни для детектирования границ объектов [6, 7].

Алгоритмы второй группы распознавания объектов (например, лиц и тел людей) используют каскадные классификаторы, например, Хаара [8, 9]. Также для решения задач детектирования объектов используется комбинация методов гистограмм направленных градиентов с методом опорных векторов.

HOG – это дескриптор особых точек, используемый в компьютерном зрении и обработке изображений для распознавания объектов. Основная идея HOG заключается в следующем. Вместо вычисления каждого отдельного направления градиента для каждого пикселя изображения пиксели группируются в маленькие ячейки, и для каждой группы ячеек вычисляется направление градиента, показывающее ориентацию ячейки на изображении. Далее величины градиента суммируются для каждой группы ячеек, например, окна размером 3×3 пикселя. Процесс повторяется для всех групп ячеек изображения. На основе направлений градиентов делается вывод о классификации объекта.

SVM – это алгоритм машинного обучения, используемый для классификации данных. Комбинация детекторов HOG+SVM позволяет разделять объекты на

классы («пешеход» и «не пешеход»). При этом SVM стремится найти гиперплоскость, которая наилучшим образом разделяет данные в многомерном пространстве признаков.

Алгоритмы третьей группы, такие как сверточные нейронные сети, позволяют извлекать признаки на изображениях, причем свертка – это процесс применения фильтра к изображению, например, окну 3×3, позволяющая выделять границы объектов на изображении. Одним из примеров применения CNN для решения задач данного класса служит использование глубокой нейросети AlexNet [10]. Рекуррентные нейронные сети используются для анализа последовательных данных, таких как временные ряды или текст. Они предназначены для обработки последовательных данных и учитывают временные зависимости между текущим и предыдущим кадрами видео [11, 12].

Поставим задачу разработки программного кода детектирования перемещения людей на пешеходном переходе. В ходе решения этой задачи особое внимание необходимо уделить организации двух процедур, а именно, считывания всех кадров из загруженного видеофайла и обнаружения пешеходов на каждом из загруженных кадров.

Таким образом, выбор алгоритма детектирования движения людей на пешеходном переходе зависит от конкретных условий использования подобных систем, например, таких как требуемая надежность распознавания объектов и доступные вычислительные ресурсы. Традиционные методы на основе компьютерного зрения эффективны в обычных условиях, однако методы на основе глубокого обучения, хотя более сложные и ресурсоемкие, но обеспечивают повышенную точность к изменениям внешней среды.

МЕТОДОЛОГИЯ

Рассмотрим процесс детектирования движения людей на пешеходном переходе с помощью открытой библиотеки обработки изображений EMGU. Программный код для реализации этой процедуры реализован на языке программирования C#. Для обработки видеoinформации использованы следующие переменные: `capture` – для хранения захватываемых кадров изображений с видеокамеры; `frames` – для хранения количества кадров в видео и `framesCurrent` – для

хранения текущего кадра; fps используется для указания количества видеок кадров, воспроизводимых в единицу времени, например, за секунду.

Для считывания видео использован следующий код:

```
01. capture = new VideoCapture(openFileDialog1.FileName);
```

```
02. Mat m = new Mat();
```

```
03. capture.Read(m);
```

```
04. pictureBox.Image = m.Bitmap;
```

На первой строке кода создаётся объект класса `VideoCapture`, который используется для захвата видео, например, директива `openFileDialog1.FileName` представляет путь к видеофайлу, который хранится на персональном компьютере. Вторая строка кода создает объект `Mat`, который используется для хранения одного кадра видео. Третья строка кода считывает следующий кадр из видеофайла, хранящегося в объекте `capture`, и записывает его в переменную `m` класса `Mat`. Четвертая строка кода позволяет выводить считанное изображение в `pictureBox` – специальную область на экране программы для вывода распознанных изображений.

Далее необходимо получить информацию о считанном видео с помощью следующих строк программного кода:

```
05. fps = capture.GetCaptureProperty(Emgu.CV.CvEnum.CapProp.Fps);
```

```
06. frames = capture.GetCaptureProperty(Emgu.CV.CvEnum.CapProp.Frame-  
Count);
```

```
07. framesCurrent = 1;
```

Пятая строка кода получает значение частоты кадров из объекта `capture`. Шестая строка кода позволяет определить общее количество кадров в видео. И на седьмой строке кода инициализируется начальное значение переменной `framesCurrent`.

Реализация вышеуказанных семи строк программного кода позволяет открыть в окне программы только первый кадр загруженного видео (см. рис. 1).

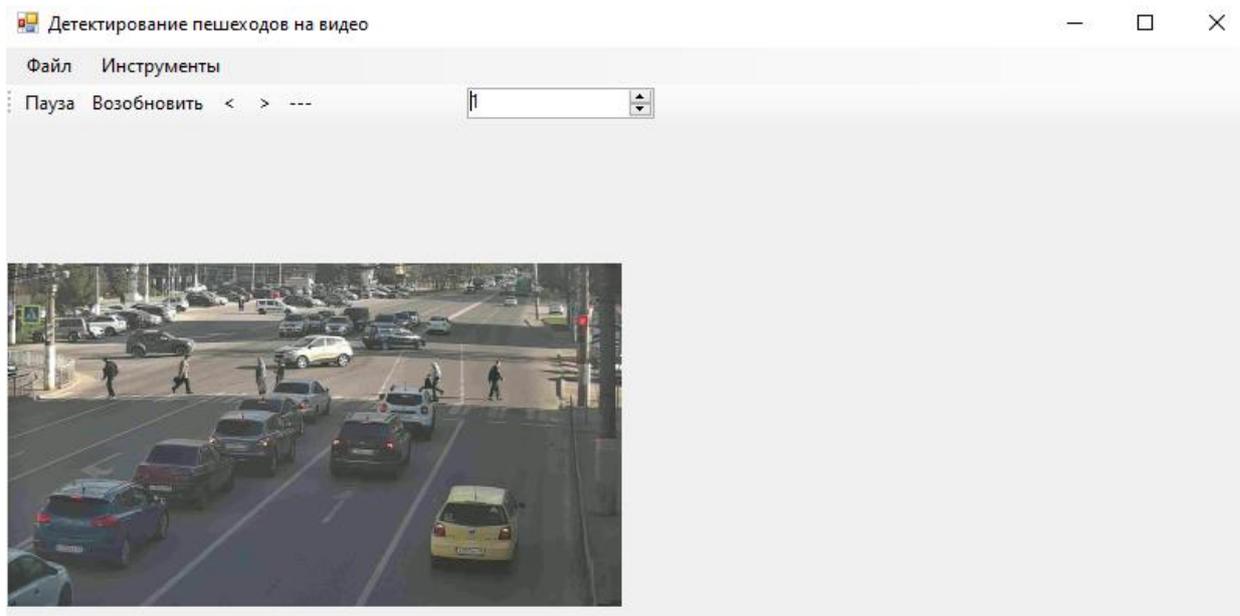


Рис. 1. Загрузка первого кадра (г. Курск, пересечение улиц Карла Маркса и Хуторской)

1.1. Процедура поиска пешеходов на перекрестке

Первоначально создадим метод Find для поиска людей на пешеходном перекрестке с помощью следующей директивы:

```
08. private Image<Bgr, byte> Find(Image<Bgr, byte> image)
```

Данный метод принимает аргумент типа Image<Bgr, byte>, который представляет собой изображение в цветовом пространстве BGR (синий, зеленый, красный) с 8-битным порогом хранения цветных пикселей и возвращает картинку того же формата с найденными пешеходами на ней. Для поиска пешеходов создаем объект regions класса MCvObjectDetection, который будет содержать информацию о распознанных объектах на изображении (координаты и их размеры)

```
09. MCvObjectDetection[] regions;
```

Затем используем метод обнаружения людей на изображении с помощью алгоритма Гистограммы направленных градиентов [13, 14]:

```
10. using(HOGDescriptor descriptor = new HOGDescriptor()) {
```

```
11.   descriptor.SetSVMDescriptor(HOGDescriptor.GetDefaultPeopleDetector());
```

```
12.   regions = descriptor.DetectMultiScale(image); }
```

В данном коде HOGDescriptor является классом библиотеки EmguCV, который реализует алгоритм HOG для обнаружения объектов на изображении. На одиннадцатой строке кода применен SVM-детектор [15, 16], который используется для обнаружения людей, при этом метод GetDefaultPeopleDetector() возвращает предварительно обученный детектор, используемый для распознавания людей. На двенадцатой строке кода в массив regions сохраняются распознанные пешеходы на изображении с помощью метода многомасштабного обнаружение объектов DetectMultiScale(image) на изображении.

Алгоритм гистограммы направленных градиентов состоит из шести этапов. На первом этапе осуществляется извлечение градиентов. Изображение разбивается на маленькие ячейки (обычно 8×8 пикселей), и для каждой из них вычисляются горизонтальные и вертикальные градиенты. Градиенты представляют направление и интенсивность изменения яркости в ячейке [7]. На втором этапе происходит создание гистограмм. Для каждой ячейки строится гистограмма направлений градиентов, содержащая бины, представляющие различные направления градиентов. И после этого гистограммы объединяются в блоки (по умолчанию 2×2 пикселя). На третьем этапе для уменьшения влияние изменений освещения применяется нормализация изображения. После этого на четвертом шаге создаётся дескриптор. При этом все нормализованные блоки объединяются в один вектор, который называется «дескриптором HOG», и этот дескриптор содержит информацию о текстуре и форме объекта. Далее осуществляется обучение классификатора, например, с помощью метода опорных векторов, который определяет, является ли область изображения искомым объектом, – в данном случае пешеходом. На заключительном, шестом этапе происходит обнаружение объектов. Если классификатор считает, что окно содержит искомый объект, то оно считается обнаруженным пешеходом и выделяется рамкой.

После обнаружения всех пешеходов на изображении выделим их на изображении с помощью красной рамки:

```
13. foreach(MCvObjectDetection pesh in regions) {  
14.   image.Draw(pesh.Rect, new Bgr(Color.Red), 3);  
15.   CvInvoke.PutText(image, "Пешеход", new Point(pesh.Rect.X, pesh.Rect.Y),  
                        Emgu.CV.CvEnum.FontFace.HersheyPlain, 1,
```

```
new MCvScalar(255, 255, 255), 2); }
```

```
16. return image;
```

Указанный код выполняет обработку массива `regions`, который содержит обнаруженные объекты – распознанных пешеходов на изображении. В цикле `foreach` (13-я строка) перебирается каждый элемент массива `regions`. Каждый элемент `resh` представляет собой распознанного пешехода. На четырнадцатой строке кода вокруг области с распознанным пешеходом рисуется прямоугольник красного цвета с толщиной линии 3 пикселя. С помощью пятнадцатой строки кода рядом с красным прямоугольником добавляется текст белого цвета, отображающий надпись «Пешеход». Шестнадцатая строка кода возвращает в переменную `image` обработанное изображение с найденными пешеходами и нарисованными вокруг них прямоугольниками красного цвета и надписями белого цвета.

Таким образом, метод `Find(Image<Bgr, byte> image)` возвращает новое изображение с распознанными пешеходами, выделенными красными прямоугольными рамками с надписями «Пешеход».

1.2. Процедура считывания кадров из видеоизображения и их обработка

Данный метод позволяет выводить исходное и обработанное изображения с помощью метода `Find` в `pictureBox` на форме компьютерного приложения. Программный код данного метода имеет следующий вид:

```
17. private async void ReadFrames(){
```

```
18.     Mat m = new Mat();
```

```
19.     while(play && framesCurrent < frames){
```

```
20.         framesCurrent += 1;
```

```
21.             capture.SetCaptureProperty(Emgu.CV.CvEnum.CapProp.PosFrames, framesCurrent);
```

```
22.             capture.Read(m);
```

```
23.             pictureBox1.Image = m.Bitmap;
```

```
24.             pictureBox2.Image = Find(m.ToImage<Bgr, byte>()).Bitmap;
```

```
25.             toolStripLabel1.Text = $"{framesCurrent} / {frames}";
```

```
26.    await Task.Delay(1000 / Convert.ToInt16(fps));}}
```

Метод `ReadFrames()` использует асинхронный режим работы, поскольку объявлен с ключевым словом `async`, и позволяет приложению выполнять другие задачи в то время, пока текущая операция ожидает завершения [17]. Логика этого метода заключается в следующем.

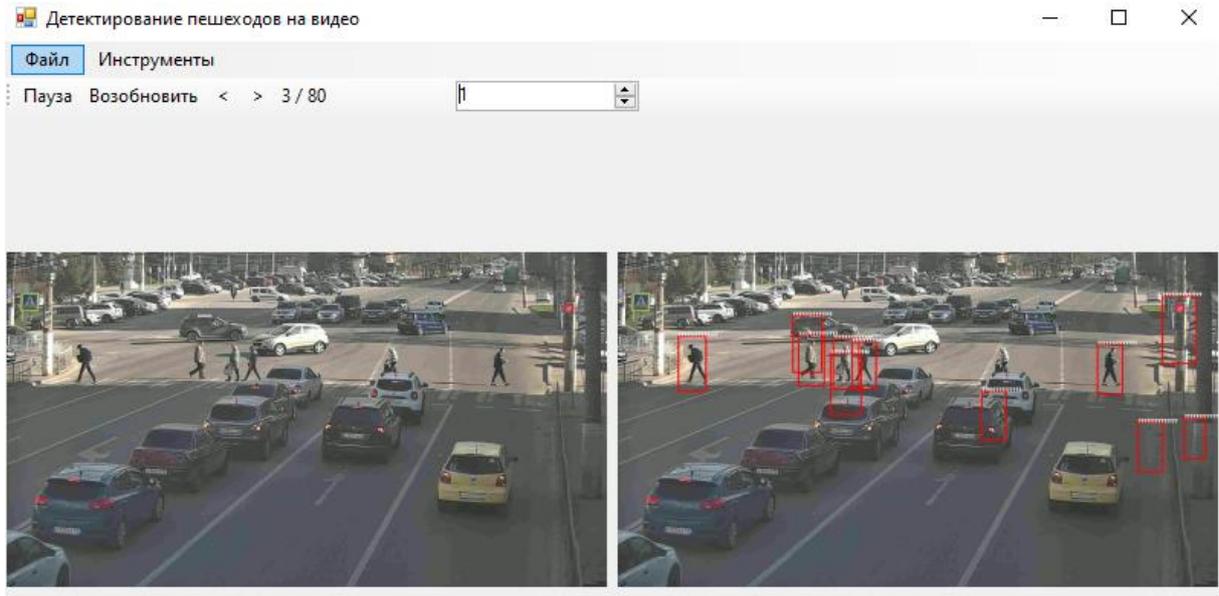
Цикл `while` перебирает все кадры введенного видеоизображения (условие `framesCurrent < frames`) или останавливается при нажатии на кнопку «Пауза» (см. рис. 1). В процессе реализации цикла `while` выполняются следующие действия. Во-первых, осуществляется увеличение счетчика текущих кадров `framesCurrent`. На двадцать первой строке кода свойство `PosFrames` получает и записывает текущую позицию (количество прочитанных кадров) в видеофайле или видеопотоке в переменную `framesCurrent`. Далее на двадцать второй строке кода осуществляется считывание кадра из видеоизображения в объект `m`. Затем на двадцать третьей и двадцать четвертой строках кода осуществляется помещение изображений в `pictureBox1` и `pictureBox2`. В `pictureBox1` хранится исходное изображение видеоизображения, в `pictureBox2` помещается обработанное видеоизображения с распознанными пешеходами на нём с помощью метода `Find`. Далее в метку `toolStripLabel1` записываются два значения номер текущего кадра `framesCurrent` и их общее количество `frames`. И на последней строке этого метода делается задержка в 1 секунду с помощью функции `await`.

МОДЕЛИРОВАНИЕ ПРОЦЕССА РАСПОЗНАВАНИЯ ЛЮДЕЙ НА ПЕШЕХОДНОМ ПЕРЕХОДЕ

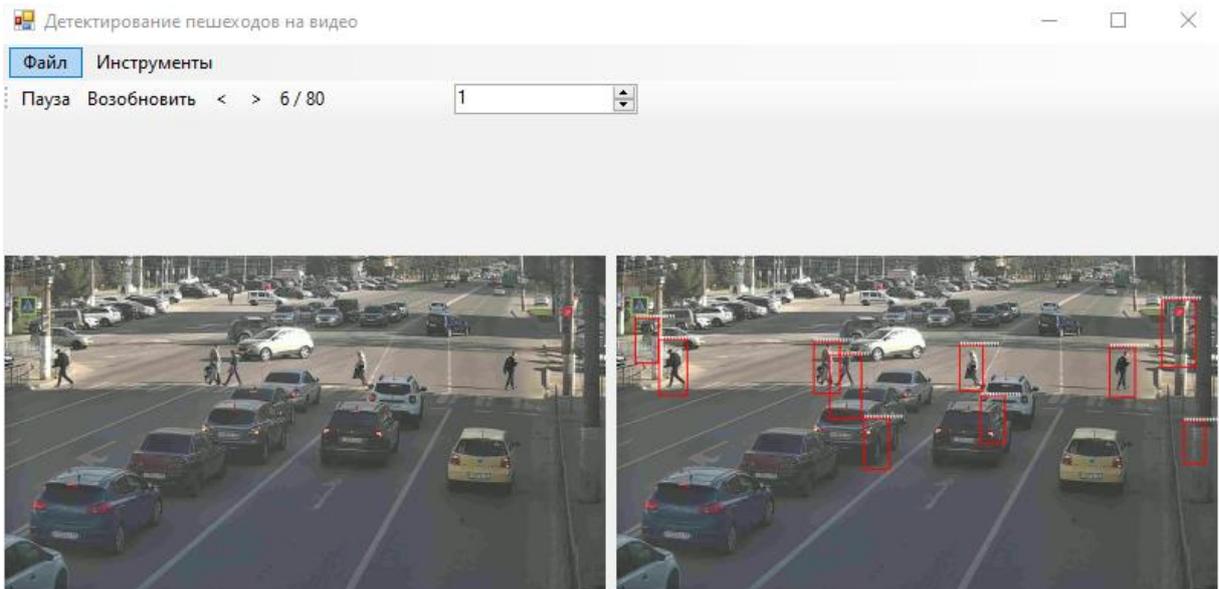
Метод детектирования перемещения людей на пешеходном переходе реализован в среде Microsoft Visual Studio на языке программирования C#, с использованием пакета Emgu CV v. 4.1.1.3497. Для экспериментов использовались персональный компьютер Intel(R) Core(TM) i5-8600K CPU 3,60 ГГц, RAM 16 ГБ, и операционная система Win10. В процессе эксперимента определялась надежность распознавания движущихся пешеходов [18].

Визуализация процесса детектирования движущихся пешеходов представлена на рисунке 2. Причем первый кадр показан на рисунке 1.

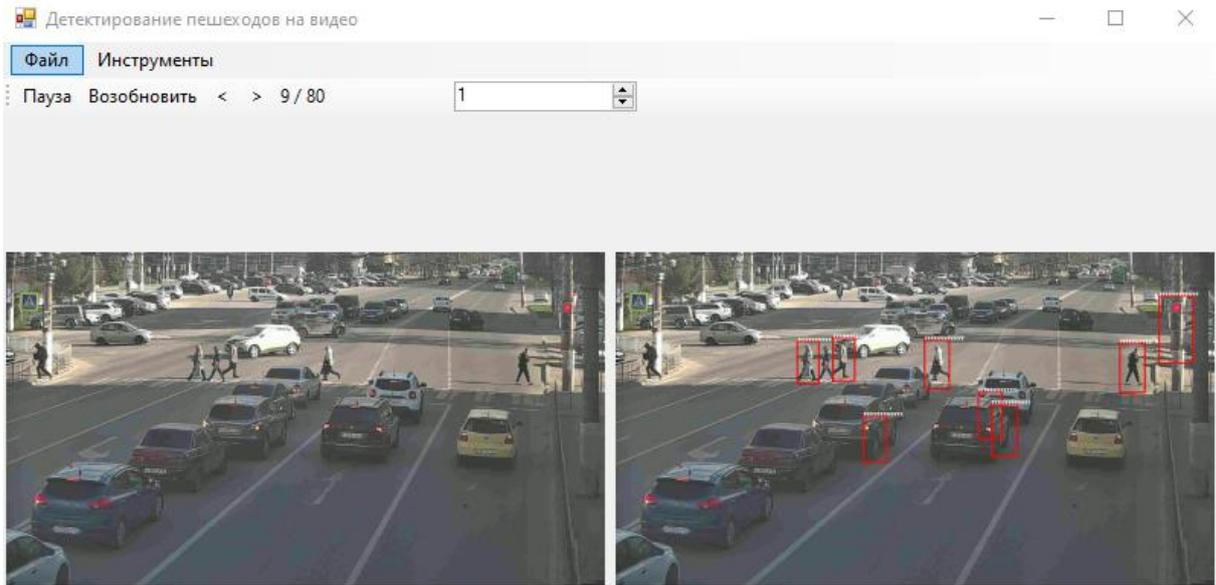
a)



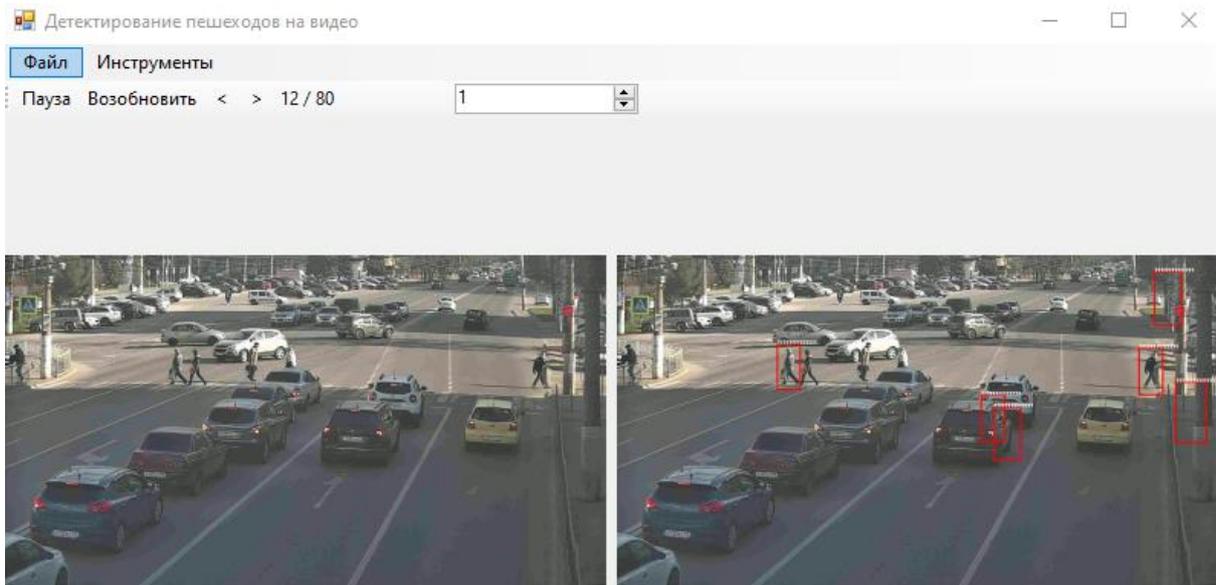
б)



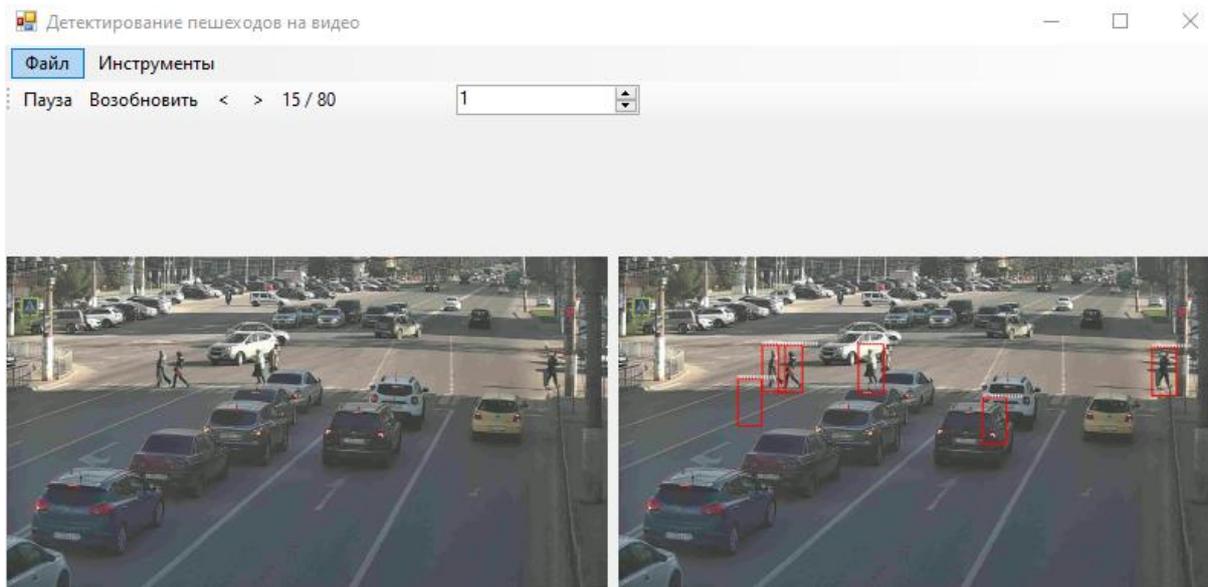
в)



г)



д)



е)

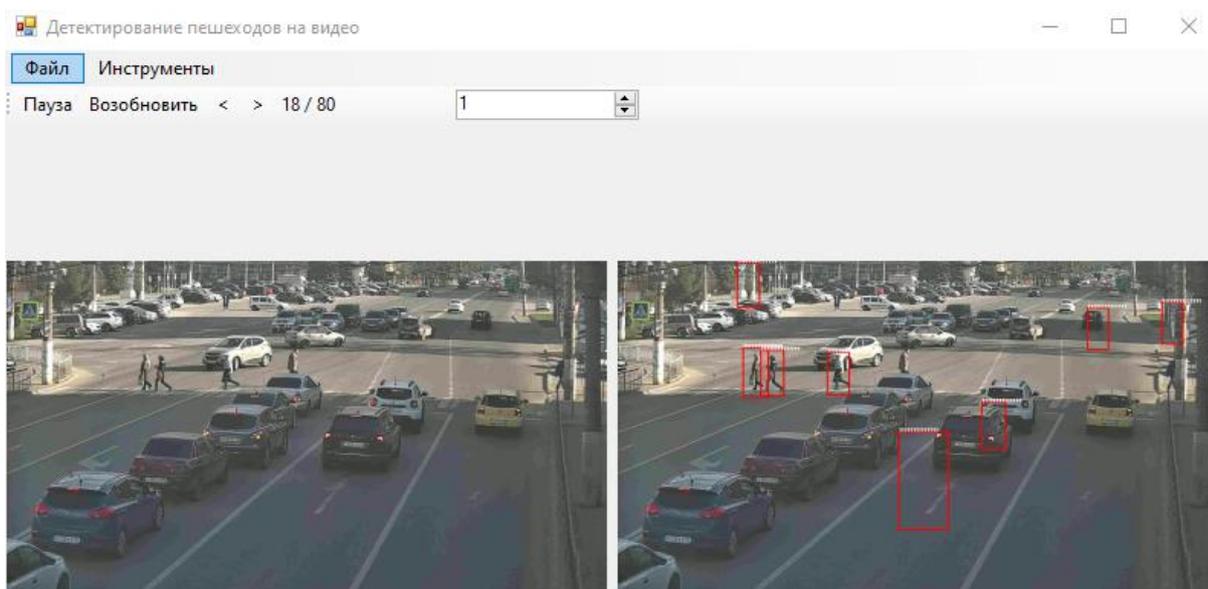


Рис. 2. Детектирование движения пешеходов: а – 3-й кадр; б – 6-й кадр; в – 9-й кадр; г – 12-й кадр; д – 15-й кадр; е – 18-й кадр

Эксперименты повторялись 100 раз для каждого из изображений, по которому необходимо было детектировать движения пешеходов. Результаты вычислительного эксперимента для шести изображений (см. рис. 2) сведены в таблицу 1.

Таблица 1. Анализ детектирования движения пешеходов

Эксперимент	Количество обнаруженных объектов		
	Правильно обнаруженные объекты	Ложно обнаруженные объекты	Пропущенные объекты
Рис. 2а	5	6	1
Рис. 2б	5	5	0
Рис. 2в	4	4	2
Рис. 2г	2	4	4
Рис. 2д	4	2	0
Рис. 2е	3	5	0
	TP	FP	FN
Сумма	23	26	7

Надежность результатов эксперимента вычислялась по формуле

$$N = \frac{TP}{TP + FP + FN},$$

где TP – количество верно обнаруженных объектов; FP – количество ложно обнаруженных объектов; FN – количество пропущенных объектов.

С учетом данных таблицы 2 надежность

$$N = \frac{23}{23 + 26 + 7} = 0.411,$$

т. е. составила 41,1%, что говорит о невысокой надежности метода распознавания движения пешеходов на основе комбинации методов гистограммы направленных градиентов и опорных векторов, что объясняется тем, что видеокамеры установлены далеко от пешеходного перехода.

ОБСУЖДЕНИЕ РЕЗУЛЬТАТОВ МОДЕЛИРОВАНИЯ И ЗАКЛЮЧЕНИЕ

Анализ моделирования процесса детектирования движения людей на пешеходном переходе показал, что:

1. Предложенный метод детектирования движения пешеходов, основанный на комбинации методов гистограммы направленных градиентов и опорных векторов, без использования специально обученных нейронных сетей или им подобных обеспечивает детектирование практически всех объектов, относящихся к классу людей.

2. Надежность рассмотренного подхода, основанного на комбинации методов гистограммы направленных градиентов и опорных векторов, составила порядка 41%.

3. В ходе экспериментальных исследований установлено, что предложенный метод распознавания движения людей на пешеходном переходе применим для обработки цветowych изображений, что делает возможным его использование в системах типа «Умный город».

4. Необходим поиск новых решений, позволяющих увеличить надежность рассмотренного подхода.

Благодарности

Работа выполнена при поддержке гранта РФФ 24-21-00055.

СПИСОК ЛИТЕРАТУРЫ

1. *Фатихов Ч.И., Григорян К.А.* Разработка системы сбора данных о перемещении людей внутри помещения // Электронные библиотеки. 2022. Т. 25, № 1. С. 87–102. <https://doi.org/10.26907/1562-5419-2022-25-1-87-102>

2. *Ямиков Р.Р., Григорян К.А.* Анализ и разработка конвейера MLOps для развертывания моделей машинного обучения // Электронные библиотеки. 2022. Т. 25, № 2. С. 177–196. <https://doi.org/10.26907/1562-5419-2022-25-2-177-196>

3. *Байрамов А.И., Фасхутдинов Т.Р., Тимергалин Д.М. [и др.].* Оценка усталости человека методом анализа фотографий лица с помощью сверточных нейронных сетей // Электронные библиотеки. 2021. Т. 24, № 4. С. 582–603. <https://doi.org/10.26907/1562-5419-2021-24-4-582-603>.

4. Поташников А.М., Власюк И.В., Аугсткальн Я.А. Анализ методов детектирования движущихся объектов различного типа на видеоизображении // *Фундаментальные проблемы радиоэлектронного приборостроения*. 2017. Т. 17, № 4. С. 1201-1204.

5. Скользящая_средняя
URL: https://ru.wikipedia.org/wiki/Скользящая_средняя (доступ свободный: дата обращения 19.06.2024)

6. *Canny J.* A computational approach to edge detection // *IEEE Trans Pattern Anal. Mach. Intell.* 1986. Т8(6). Р. 679–698.

7. Бобырь М.В., Архипов А.Е., Горбачев С.В., Цао Ц., Бхаттачарья С. Не-четко-логические методы в задаче детектирования границ объектов // *Информатика и автоматизация*. 2022. Т. 21, №2. С. 376–404.
<https://doi.org/10.15622/ia.21.2.6>

8. *Raj R., Rajiv P., Kumar P., Khari M., Verdú E., Crespo R. G., Manogaran G.* Feature based video stabilization based on boosted HAAR Cascade and representative point matching algorithm // *Image and Vision Computing*. 2020. Т101.
<https://doi.org/10.1016/j.imavis.2020.103957>

9. Бурмистров А.В. Распознавание объектов на изображениях с использованием базовых средств языка Python и библиотеки OpenCV // *Научное обозрение. Технические науки*. 2021. № 5. С. 15–19.

10. *Gou Y., Li Q., Yao R., Chen J., Zhao H., Zhang Z.* Ice accretion existence and three-dimensional shape identification based on infrared thermography detection // *Infrared Physics and Technology*. 2023. Т135.
<https://doi.org/10.1016/j.infrared.2023.104972>

11. *Zhou Z., Yu X., Chen X.* Object Detection in Drone Video with Temporal Attention Gated Recurrent Unit Based on Transformer // *Drones*. 2023. Vol. 7, No. 7.
<https://doi.org/10.3390/drones7070466>

12. Буров С.А., Тутов К.Б., Иванов А.Р. Программа прогнозирования координат движения целей на основе рекуррентной искусственной нейронной сети. Свидетельство о государственной регистрации программы для ЭВМ № 2021613317 РФ. 05.03.2021.

13. Ярошевич П.В., Богуш Р.П. Алгоритм классификации изображений парковочных мест автостоянки на основе гистограмм ориентированных градиентов и метода опорных векторов // Компьютерная оптика. 2017. Т. 41. № 1. С. 110–117. <https://doi.org/10.18287/2412-6179-2017-41-1-110-117>

14. Побережник В.И. Классификация BIM структур с помощью гистограммы направленных градиентов и метода опорных векторов // Приложение математики в экономических и технических исследованиях. 2020. Т. 10, № 1. С. 140–146.

15. Rahman M., Zhou Yu., Wang Sh., Rogers Ja. Wart Treatment Decision Support Using Support Vector Machine // International Journal of Intelligent Systems and Applications. 2020. Vol. 12, No. 1. P. 1–11. <https://doi.org/10.5815/ijisa.2020.01.01>

16. Присухина И.В., Борисенко Д.В. Машинная классификация режима работы электрической рельсовой цепи методом опорных векторов // Омский научный вестник. 2018. Т. 162, №6. С. 126–130. <https://doi.org/10.25206/1813-8225-2018-162-126-130>

17. Как на самом деле Async/Await работают в C#. Часть 6. Анализ результатов компиляции асинхронных вызовов URL: <https://habr.com/ru/articles/792782/> (доступ свободный: дата обращения 19.06.2024).

18. Бобырь М.В., Храпова Н.И., Архипов А.Е., Милостная Н.А. Программа детектирования движения объектов на пешеходном переходе. Свидетельство о государственной регистрации программы для ЭВМ № 2024660940 РФ. опубли. 14.05.2024.

ON THE APPROACH TO DETECTING PEDESTRIAN MOVEMENT USING THE METHOD OF HISTOGRAMS OF ORIENTED GRADIENTS

M. V. Bobyr¹ [0000-0002-5400-6817], N. A. Milostnaya² [0000-0002-3779-9165],

N. I. Khrapova³ [0000-0001-7947-1427]

¹⁻³Southwest State University, Kursk region, Kursk, 305040, Russia

¹maxbobyr@gmail.com, ²nat_mil@mail.ru, ³KhrapovaNI@yandex.ru

Abstract

An approach to automatically recognizing the movement of people at a pedestrian crossing presented in the article. This approach includes two main procedures, for each of which program code commands are given in the C# programming language using the EMGU computer vision library. In the first procedure, pedestrian detection is carried out using a combination of directional gradient histogram and support vector methods. The second procedure allows you to read frames from a video sequence and process them. This approach allows detecting the movements of people at a pedestrian crossing without using specialized neural networks. At the same time, the method proposed in the article demonstrated sufficient reliability of human movement recognition, which indicates its applicability in real conditions.

Keywords: *Pedestrian Motion Recognition, EMGU, Histogram of Oriented Gradients, Support Vector Machine*

REFERENCES

1. *Fatihov Ch.I., Grigorjan K.A.* Razrabotka sistemy sbora dannyh o peremeshhenii ljudej vnutri pomeshhenija // Russian digital library. 2022. T. 25, № 1. S. 87–102. <https://doi.org/10.26907/1562-5419-2022-25-1-87-102>
2. *Jamikov R.R., Grigorjan K.A.* Analiz i razrabotka konvejera MLOps dlja razvertyvanija modelej mashinnogo obuchenija // Jelektronnye biblioteki. 2022. T. 25, № 2. S. 177–196. <https://doi.org/10.26907/1562-5419-2022-25-2-177-196>
3. *Bajramov A.I., Fashutdinov T.R., Timergalin D.M.* [i dr.]. Ocenka ustalosti cheloveka metodom analiza fotografij lica s pomoshh'ju svertochnyh nejronnyh setej // Jelektronnye biblioteki. 2021. T. 24, № 4. S. 582–603. <https://doi.org/10.26907/1562-5419-2021-24-4-582-603>.

4. Potashnikov A.M., Vlasjuk I.V., Augstkal'n Ja.A. Analiz metodov detektirovaniya dvizhushhihsja ob#ektov razlichnogo tipa na videoizobrazhenii // Fundamental'nye problemy radioelektronnoho priborostroenija. 2017. T. 17, № 4. S. 1201–1204.

5. Moving average

URL: https://en.wikipedia.org/wiki/Moving_average (date of access 19.06.2024)

6. Canny J. A computational approach to edge detection // IEEE Trans Pattern Anal. Mach. Intell. 1986. Vol. 8. No. 6. P. 679–698.

7. Bobyr M.V., Arhipov A.E., Gorbachev S.V., Cao C., Bhattachar'ja S. Nechetko-logicheskie metody v zadache detektirovaniya granic obektov // Informatika i avtomatizacija. 2022. T. 21(2). S. 376–404. <https://doi.org/10.15622/ia.21.2.6>

8. Raj R., Rajiv P., Kumar P., Khari M., Verdú E., Crespo R. G., Manogaran G. Feature based video stabilization based on boosted HAAR Cascade and representative point matching algorithm // Image and Vision Computing. 2020. T101. <https://doi.org/10.1016/j.imavis.2020.103957>

9. Burmistrov A.V. Raspoznavanie ob#ektov na izobrazhenijah s ispol'zovaniem bazovyh sredstv jazyka Python i biblioteki OpenCV // Nauchnoe obozrenie. Tehniceskie nauki. 2021. № 5. S. 15–19.

10. Gou Y., Li Q., Yao R., Chen J., Zhao H., Zhang Z. Ice accretion existence and three-dimensional shape identification based on infrared thermography detection // Infrared Physics and Technology. 2023. T135. <https://doi.org/10.1016/j.infrared.2023.104972>

11. Zhou Z., Yu X., Chen X. Object Detection in Drone Video with Temporal Attention Gated Recurrent Unit Based on Transformer // Drones. 2023. Vol. 7, No. 7. <https://doi.org/10.3390/drones7070466>

12. Burov S.A., Titov K.B., Ivanov A.R. Programma prognozirovaniya koordinat dvizhenija celej na osnove rekurrentnoj iskusstvennoj nejronnoj seti. Svidetel'stvo o gosudarstvennoj registracii programmy dlja JeVM № 2021613317 RF. 05.03.2021

13. Jaroshevich P.V., Bogush R.P. Algoritm klassifikacii izobrazhenij parkovochnyh mest avtostojanki na osnove gistogramm orientirovannyh gradientov i metoda opornyh vektorov // Komp'juternaja optika. 2017. T. 41. № 1. S. 110–117. <https://doi.org/10.18287/2412-6179-2017-41-1-110-117>.

14. *Poberezhnik V.I.* Klassifikacija BIM struktur s pomoshh'ju gistogrammy napravlennyh gradientov i metoda opornyh vektorov // Prilozhenie matematiki v jekonomicheskikh i tehnicheskikh issledovanijah. 2020. T. 10, № 1. S. 140–146.

15. *Rahman M., Zhou Yu., Wang Sh., Rogers Ja.* Wart Treatment Decision Support Using Support Vector Machine // International Journal of Intelligent Systems and Applications. 2020. Vol. 12, No. 1. P. 1–11. <https://doi.org/10.5815/ijisa.2020.01.01>

16. *Prisuhina I.V., Borisenko D.V.* Mashinnaja klassifikacija rezhima raboty jektricheskoy rel'sovoj cepi metodom opornyh vektorov // Omskij nauchnyj vestnik. 2018. T. 162, №6. S. 126–130.

<https://doi.org/10.25206/1813-8225-2018-162-126-130>.

17. Kak na samom dele Async/Await rabotayut v C#. CHast' 6. Analiz rezul'tatov kompilyacii asinhronnyh vyzovov.

URL: <https://habr.com/ru/articles/792782/> (date of access 19.06.2024).

18. *Bobyр M.V., Hrapova N.I., Arhipov A.E., Milostnaja N.A.* Programma dektirovanija dvizhenija obektov na peshehodnom perehode. Svidetel'stvo o gosudarstvennoj registracii programmy dlja JeVM № 2024660940 RF. opubl. 14.05.2024.

СВЕДЕНИЯ ОБ АВТОРАХ



БОБЫРЬ Максим Владимирович – Доктор технических наук, профессор. Профессор кафедры «Программная инженерия», ФГБОУ ВО «Юго-Западный государственный университет». Области исследований: интеллектуальные системы управления, адаптивные нейро-нечеткие системы вывода, распознавание и обработка изображения, робототехнические системы.

Maxim V. BOBYR – Doctor of technical sciences, professor. Professor of the Department of Software Engineering, Southwest State University. Research areas: intelligent control systems, adaptive neuro-fuzzy inference systems, image recognition and processing, robotic systems.

email: maxbobyр@gmail.com

ORCID: 0000-0002-5400-6817



МИЛОСТНАЯ Наталья Анатольевна – Доктор технических наук. Ведущий научный сотрудник кафедры «Программная инженерия», ФГБОУ ВО «Юго-Западный государственный университет». Области исследований: методы оптимизации на основе динамического программирования, робототехнические и автоматизированные системы управления сложными объектами.

Natalya A. MILOSTNAYA – Doctor of technical Sciences. Leading Researcher, Department of Software Engineering, Southwest State University. Research areas: optimization methods based on dynamic programming, robotic and automated control systems for complex objects.

email: nat_mil@mail.ru

ORCID: 0000-0002-3779-9165



ХРАПОВА Наталья Игоревна – аспирант кафедры «Программная инженерия», ФГБОУ ВО «Юго-Западный государственный университет». Область исследования: распознавание сложных объектов с использованием нечеткой логики.

Natalia Igorevna KHRAPOVA – Postgraduate student of the Department of Software Engineering at the Southwest State University. Research area: recognition of complex objects using fuzzy logic.

email: KhrapovaNI@yandex.ru

ORCID: 0000-0001-7947-1427

Материал поступил в редакцию 15 июня 2024 года

УДК 004.891.3

ПРИМЕНЕНИЕ МЕТОДОВ КОМПЬЮТЕРНОГО ЗРЕНИЯ К РАСПОЗНАВАНИЮ СТАРОТАТАРСКОГО ТЕКСТА

И. А. Валишин^[0009-0006-6891-031X]

*Институт информационных технологий и интеллектуальных систем
Казанского (Приволжского) федерального университета, ул. Кремлевская, 35,
г. Казань, 420008*

iskander1998@list.ru

Аннотация

Разработан инструмент, распознающий строки, слова и арабские символы с отсканированного изображения. Рассмотрены возможности и перспективы применения инструмента в исследовательской деятельности. Приведены результаты экспериментов по проверке работоспособности инструмента на примере старотатарских оцифрованных произведений.

Ключевые слова: YOLO, распознавание арабских символов, нейронные сети, компьютерное зрение.

ВВЕДЕНИЕ

В последние годы отечественные библиотеки и архивы переводят свой фонд в цифровое представление. Многие исторические документы (книги, чертежи, карты, рукописи и т. п.) в силу ветхости или уникальности доступны только ограниченному кругу специалистов. Их перевод в цифровое представление обеспечивает возможность доступа к ним широкому кругу читателей [1].

Ключевое множество общественно значимых, древних данных остается за бортом развития человеческого общества. Для расшифровки подобной информации используется «ручной труд» узкопрофиллированных специалистов. Это касается и старотатарского языка, ведь зачастую процесс перевода старинных рукописей сопровождается существенными временными и человеческими затратами. Подобные исследовательские проекты становятся высоко бюджетными и тяжело исполнимыми в надлежащие сроки. Автоматизация подобных процессов позволила бы сэкономить время и ресурсы. Также появилась бы возможность помочь

специалистам в области переводов древних рукописей. Модель автоматизации может заменить большую часть работы и значительно облегчить процесс расшифровки данных.

Предметом представленного исследования являются древние рукописные и печатные тексты на старотатарском языке. Объект исследования: применение методов компьютерного зрения в задачах распознавания старотатарских текстов.

Целью проведенного исследования является разработка модели, способной с помощью методов компьютерного зрения распознать старотатарский текст. Для достижения указанной цели были поставлены следующие задачи:

- 1) Провести анализ научной литературы в предметной области;
- 2) Обнаружить или подготовить необходимый набор данных;
- 3) Разработать модель распознавания текста;
- 4) Разработать графический интерфейс модели.

1. ОБЗОР СУЩЕСТВУЮЩИХ РЕШЕНИЙ

Для достижения цели исследования был произведен поиск аналогичных решений с помощью таких систем поиска научных статей, как Google scholar, «КиберЛенинка», eLibrary.ru и ряда других. Желаемым результатом поиска было обнаружение существующих решений, которые могли бы помочь достичь поставленную цель или максимально приблизиться к ее достижению. Поиск был декомпозирован на следующие подпункты:

- 1) Поиск моделей и решений в области распознавания текста любого языка по типу ОСР (оптическое распознавание символов);
- 2) Поиск моделей и решений в области распознавания старотатарского текста;
- 3) Поиск моделей и решений в области распознавания текстов на основе арабской письменности.

Был также осуществлен поиск информации об инструментах и моделях, которые могли бы распознавать тексты на различных языках с изображения и перевести его в цифровое поле. Результаты поиска оказались удовлетворительными по причине достаточно глубокой изученности данной проблемы. Одним из самых популярных инструментов в области OCR является Tesseract, который берет свое начало с середины 1980-х годов. Эта программа разрабатывалась компанией

Hewlett-Packard, а с 2006 года разработка финансировалась компанией Google [2]. Названный инструмент дает возможность пользователям работать в различных операционных системах и хорошо справляется с распознаванием и переводом в цифровое пространство печатных и письменных текстов на изображениях [3]. Но в основном подобного рода модели натренированы на английском языке и не имеют графического интерфейса, как в случае с Tesseract, что является причиной их меньшей доступности для обычных пользователей. Исследователи также отмечают программы Microsoft Onenote, Abbyy Finereader и др. [4] [5]. Названные инструменты являются коммерческими и одними из самых популярных решений, которые представлены на рынке OCR. Анализ научной литературы, в которой описаны модели OCR, показал, что старотатарский язык отсутствует во всех решениях в качестве языка для распознавания. Присутствие арабского языка в подобных инструментах не играет существенной роли, поскольку арабица является основой письменности старотатарского языка, но слова являются татарскими.

Поиск информации об оптическом распознавании старотатарского языка результатов не дал – авторами эта проблематика не была затронута. Поэтому было решено проанализировать научную литературу об оптическом распознавании арабоязычных текстов и арабской графики. В результате было обнаружено значительное количество научных статей, которые представляют разработки в области распознавания арабоязычного текста, слов и букв (см., например, [17–20]). Имеются также публикации, в которых представлен сравнительный анализ существующих решений в рассматриваемой предметной области. В частности, установлено, что модели, в которых использовались нейронные сети (LSTM, deep CNN, CNN-RNN и др.), показали высокий уровень метрики Ассигасы (в процентном соотношении от 76.3% до 99.3%), что позволяет решать множество разнообразных задач в области распознавания арабоязычных текстов. В названных моделях использовались такие датасеты, как IFN/ENIT, HACDB, ARTI, HMBD, Hijja data set (рис. 1). В ходе поиска оптимального датасета для решения поставленной задачи был осуществлен поиск названных выше наборов данных. Было также установлено, что имеющиеся датасеты разделились на рукописные и печатные. При этом арабоязычные датасеты рукописного текста представлены в виде множества изображений различных арабских букв и слов с особенностями почерка авторов.

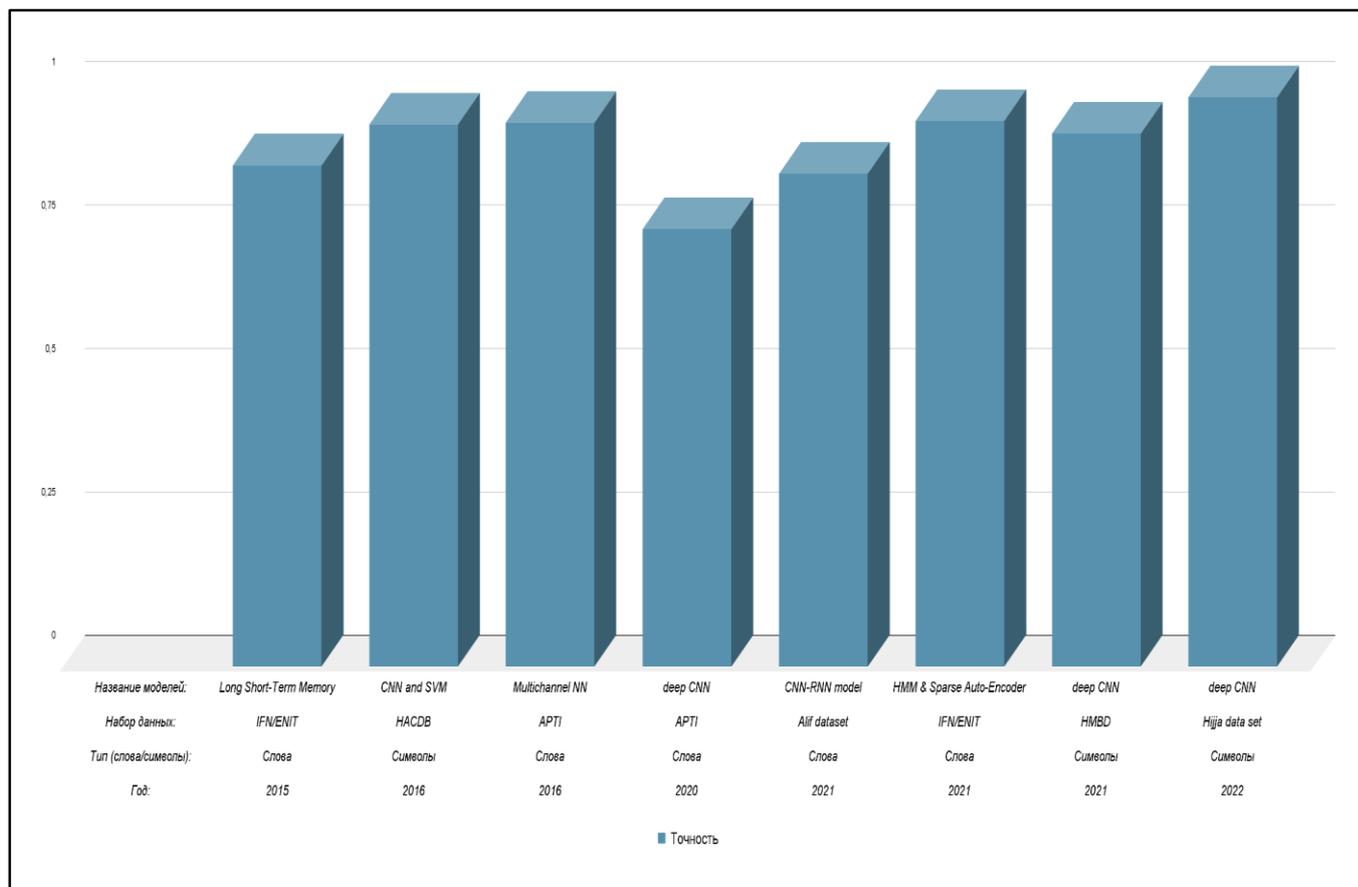


Рис. 1. Оценка качества моделей

IFN/ENIT датасет рукописного арабского текста представлен на официальном сайте, но только в демоверсии. Он содержит 2200 изображений из Тунисских городов с качеством изображения в 300 dpi. Включает в себя 26450 арабских слов и 212211 букв [6]. Полная версия набора данных предоставляется платно для коммерческих целей и бесплатно — для учебных.

Набор данных HACDB состоит из 6600 рукописных арабских букв. В этом датасете учитываются особенности арабского языка, в частности, то, что буква может быть в 4-х положениях: начальной, средней, конечной и изолированной. Этот факт делает датасет полезным для исследователей в области распознавания арабоязычных текстов, потому что буквы в нем представлены во всех возможных их проявлениях в письме [7].

Набор данных Hijja dataset включает коллекцию рукописных арабоязычных букв, которые были собраны у школьников в возрасте от 7 до 12 лет [8]. Данные

были собраны в Рияде, Саудовской Аравии в январе – апреле 2019 года. Датасет включает 47434 арабских символа разного почерка от 591 участника.

Самым упоминаемым датасетом в научных работах, которые касаются области распознавания арабоязычных текстов, является набор данных APTI. Он включает 45313600 изображений. Каждое изображение – одно арабское слово. Этот датасет был искусственно сгенерирован из 113284 слов с 10 типами шрифта и размером шрифта от 6 до 24 pt.

Перечисленные наборы данных позволяют успешно решать задачи распознавания арабоязычных текстов как в печатном, так и в рукописном видах.

Для датасета Hijja совместно с набором AHCD была разработана модель на основе CNN [9]. Модель достигает точности распознавания рукописных арабских символов 97% для Hijja и 88% для AHCD. Использование сверточных нейронных сетей демонстрирует существенные результаты в области распознавания арабских символов. Модели, которые будут представлены в следующих разделах настоящей статьи, в большинстве случаев используют CNN.

Система распознавания арабских символов AHCR-DLS показала высокую точность распознавания на тренировочных датасетах HMB1 и HMB2 от 94,9% до 97,3% [10].

Модели на основе сверточных сетей также предлагают новые подходы в классификации арабских символов с оптимизированной функцией активации ReLU. Подходы, предложенные в [11], показали точность модели 97,8% с использованием новой методики классификации.

Модель UnCNN распознавания изолированных арабоязычных символов показала конкурентоспособную точность в сравнении с приведенными выше моделями. Для оценки точности модели использовались датасеты IFHCDB, AHCD, AIA9K и HACDB [12].

Помимо обычных моделей в научной литературе также представлены и новые подходы в решении задач распознавания арабоязычных текстов, например, подход, который состоит из 4 этапов и включает state of the art модели для решения задачи. На первом этапе в YOLOv4 с помощью CNN обучается распознавание арабских печатных символов. Второй метод включает обработку перекрывающихся ограничивающих рамок, чтобы обеспечить выбор наиболее точной рамки

для каждого символа. Третий метод использует библиотеку Hunspell для проверки правописания слов и исправления ошибок. Четвертый метод использует расстояние редактирования для сравнения слов с ошибками в написании OCR с предложениями Hunspell и выбора ближайшего правильного слова. Предложенная система PAOCR достигла впечатляющей точности 82,4% для набора данных, состоящего из печатных арабских символов.

Таким образом, выше описаны датасеты арабоязычных символов (как печатных, так и письменных) для решения задачи распознавания арабских символов. Отмечены актуальные модели на основе CNN, их метрики и датасеты, которые они использовали для обучения. Указаны новые подходы в решении задачи распознавания арабских символов, что существенно помогает в распознавании старотатарского текста, который состоит из арабских символов. Выделены также коммерческие OCR, которые могли бы быть полезны для решения поставленных задач.

2. ПОДГОТОВКА НАБОРА ДАННЫХ

Решение задачи распознавания старотатарского текста требует обнаружения наиболее подходящего датасета для работы с символами и словами. В предыдущем разделе были рассмотрены наборы данных для распознавания арабских символов: IFN/ENIT, HACDB, APTI, Hijja. Эти наборы данных позволяют решить задачу распознавания арабских символов, но не затрагивают задачу распознавания слов и строк.

Для достижения поставленной цели был осуществлен выбор в пользу методики локализации символа – она предполагает локализацию объекта до тех пор, пока он не будет точно определен в качестве конечного результата поиска. В нашем исследовании предметом является старотатарский оцифрованный текст. Если рассматривать подобный текст в качестве примера, то, исходя из названной методики, нужно определить конечный результат поиска системы. Результатом будет являться арабский символ для последующей возможности собрать готовое слово из символов. Для локализации и повышения точности распознавания символа на оцифрованном изображении высокого качества необходима локализация в виде разделения текста на строки, а после – на слова. Таким образом повы-

сится точность распознавания символа, и он не затеряется среди других распознанных символов. Для решения такой задачи возникает необходимость создания набора данных для трех моделей распознавания. Первая модель будет разделять текст на строки, вторая – на слова, а третья – на символы.

В первом случае (разделения на строки) выбор был сделан в пользу старотатарского текста. Оцифрованное изображение со старотатарским текстом было размечено с одним основным классом «lines». Для работы с разметкой данных использовался сервис Roboflow, который позволяет аннотировать изображения в браузере, а затем экспортировать их для обучения модели. Было размечено 26 изображений, в них 569 аннотаций (строк) (рис. 2). Средний размер изображений составил 1466x2381 пикселей. В качестве встроенного препроцессинга изображений в сервисе Roboflow было использовано: Auto-Orient: Applied, Resize: Fit within 736x736, Grayscale: Applied, Auto-Adjust Contrast: Using Contrast Stretching. Аугментация позволила увеличить размер датасета до 37 изображений. Для аугментации использовались: Flip: Horizontal, Vertical Grayscale: Apply to 15% of images, Saturation: Between -25% и +25%, Blur: Up to 2px. Набор данных был разделен на 33 тренировочных, 3 валидационных и 1 тестовых изображений.

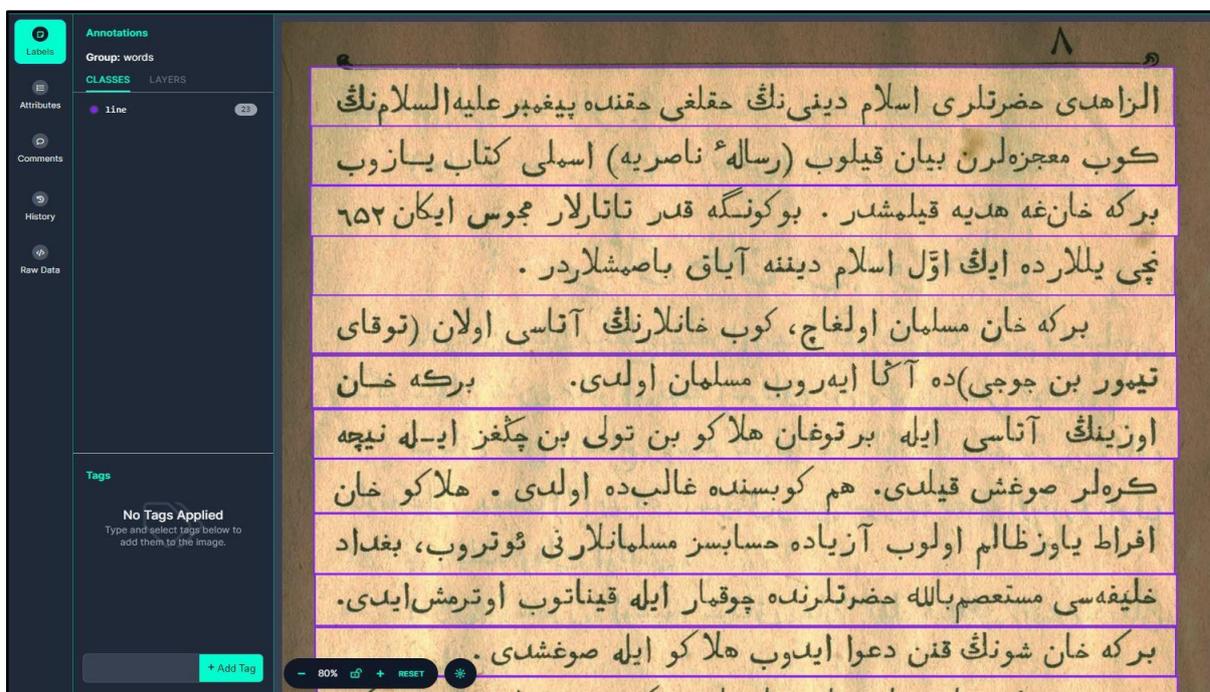


Рис. 2. Пример аннотирования изображения для модели распознавания строк в сервисе Roboflow

Следующей задачей после распознавания строк является распознавание слов внутри выделенных ранее строк. Для сбора датасета использовались вырезанные изображения строк из старотатарского оцифрованного текста. Они стали основой набора данных для обучения модели, которая может распознать старотатарское слово и выделить его в отдельное изображение. Для разметки набора данных со старотатарскими словами использовался сервис Roboflow. Единственным классом набора являлось «word». Во время разметки данных в одном изображении (строке) размечалось от 2 до 5 слов. Было размечено 100 изображений, в них 788 аннотаций (слов). Средний размер изображений составил 1458x108 пикселей. В ходе препроцессинга использовались: Auto-Orient: Applied, Resize: Fit within 640x640, Grayscale: Applied, Auto-Adjust Contrast: Using Contrast Stretching. Аугментация позволила увеличить размер датасета до 220 изображений. Для аугментации использовались: Outputs per training example: 3, Grayscale: Apply to 15% of images, Exposure: Between -13% and +13%, Blur: Up to 1.9px. Набор данных был разделен на 180 тренировочных, 25 валидационных и 15 тестовых изображений (рис. 3).



Рис. 3. Пример аннотирования изображения для модели распознавания старотатарских слов в сервисе Roboflow

Большая часть датасетов, перечисленных выше, находится в ограниченном доступе. В ходе поисков наборов данных, находящихся в открытом доступе, был обнаружен датасет арабоязычных рукописных символов HMBD v1 [13]. Этот набор данных включает 115 категорий (четыре вида написания одного символа), 54115 уникальных изображений с белым фоном и черными буквами [14]. Высота и ширина каждого изображения в датасете составляют 300 пикселей (рис. 4). В 2021 году с использованием этого датасета была создана модель deep CNN, которая показала 92,8% точности распознавания арабских рукописных символов. В наборе данных было 115 классов арабских символов, на каждый символ приходилось в среднем 4 класса для одного символа: начальное написание, между буквами, конечное написание, изолированный символ.

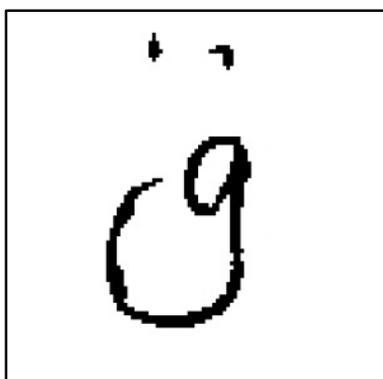


Рис. 4. Пример изображения из датасета HMBDv1 (класс Qaf_Isolated)

В старотатарском языке буквы арабские, а слова татарские. Этот факт позволил использовать датасет HMBD v1 для создания собственного набора данных, который подойдет под условия распознавания старотатарского языка, где главное условие – наличие арабских символов. Для разметки набора данных с арабскими рукописными символами был использован сервис Roboflow. Множества классов из датасета были объединены в локальные классы для одного символа. Например, классы: Kaf_End, Kaf_Isolated, Kaf_Middle, Kaf_Start, Khaa_End, Khaa_Isolated, Khaa_Start, Qaf_End, Qaf_Isolated, Qaf_Middle, Qaf_Start были объединены в один класс под названием «К». Изображения из этих классов были собраны в один класс «К» в количестве около 300 изображений и впоследствии распределены между тренировочной, валидационной и тестовой выборками в процентном соотношении 70%, 20%, 10% соответственно. Приведенное ранее решение применялось и к остальным классам. Это позволило сократить количество

классов до 22, что в 5,2 раза меньше оригинального набора данных. Также это позволило оптимизировать датасет для последующего обучения, количество изображений сократилось до 6400. Всего в созданном датасете для модели распознавания арабских (старотатарских) символов было создано 22 класса (рис. 5): A, Wau, Iy, N, K, R, L, D, B, T, M, S, Ha, G, Sh, La, Z, F, Ta, h, Zzh. Часть из них обозначает звуки или буквы из татарского языка для более точного воссоздания слова. В каждом классе набора данных было собрано от 325 до 945 изображений. Для улучшения выборки датасета было добавлено и размечено 600 изображений со старотатарскими словами. Это позволило увеличить размер датасета до 7000 изображений с размеченными арабскими символами. Аугментация позволила увеличить размер датасета до 11476 изображений. В ходе препроцессинга использовались: Auto-Orient: Applied Resize: Stretch to 320x320, Grayscale: Applied, Auto-Adjust Contrast: Using Contrast Stretching. Для аугментации использовались: Flip: Horizontal Grayscale: Apply to 15% of images, Brightness: Between -22% and +22%, Blur: Up to 4.5px. Набор данных был разделен на 8952 тренировочных, 1599 валидационных и 925 тестовых изображений.

В результате сбора данных было создано три датасета, каждый из которых определял будущую модель: модель для распознавания строк на отсканированном изображении, модель для распознавания старотатарских слов в распознанных ранее строках и модель распознавания арабских символов в распознанных ранее словах. В сумме на три датасета были размечены более 7000 изображений и 24 класса.

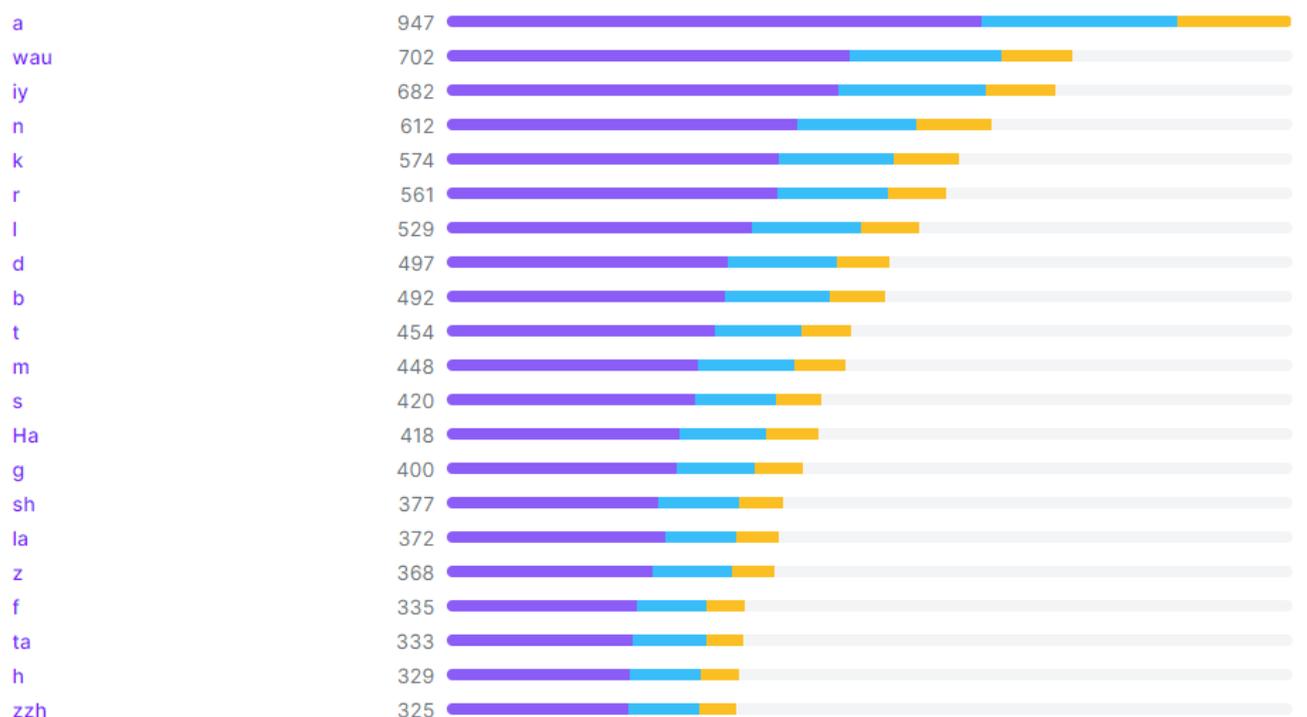


Рис. 5. Вкладка Health Check со всеми классами датасета по распознаванию арабских символов в сервисе Roboflow

РАЗРАБОТКА МОДЕЛИ РАСПОЗНАВАНИЯ ТЕКСТА

Исходя из сведений, приведенных в предыдущих двух разделах статьи, был разработан подход, который позволил оптимально решить поставленную задачу. Отсканированное изображение будет передаваться в систему распознавания старотатарского текста. Система будет делить отсканированное изображение со старотатарским текстом на строки. Далее система будет находить в строках старотатарские слова. В заключение старотатарское слово будет распознано по буквам таким образом, чтобы пользователь смог его самостоятельно определить (рис. 6).

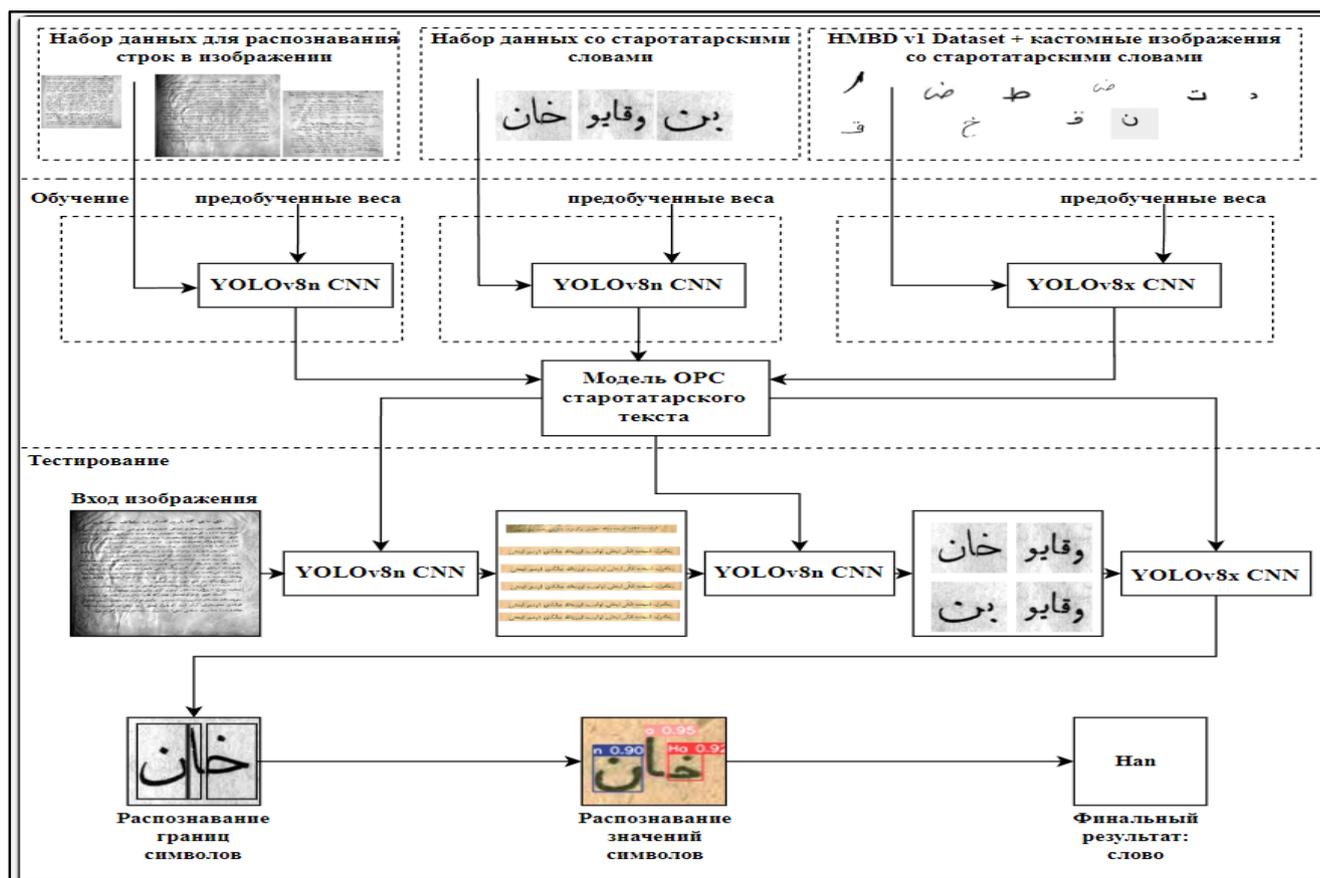


Рис. 6. Схема разработанного подхода для распознавания старотатарского текста

Всего система включает три модели и графический функциональный интерфейс: модель по распознаванию строк на основе YOLOv8n (3.2м параметров), модель по распознаванию старотатарских слов на основе YOLOv8n (3.2м параметров), модель по распознаванию арабских символов на основе YOLOv8x (68.2м параметров), графический интерфейс на основе PyQt5.

YOLO (You Only Look Once) CNN является state of the art моделью в области детекции объектов [15]. По сравнению с другими предложенными CNN моделями в распознавании арабских символов, YOLO также обладает повышенным FPS, если на вход модели подаётся видео формат. В дальнейших исследованиях возможна реализация подобного решения.

Для работы по распознаванию строк и старотатарских слов достаточно было использовать модель YOLOv8n. Эта модель обладает меньшим количеством параметров по сравнению с YOLOv8x, но больше подходит для решаемой задачи, потому что в датасетах слов и строк представлен лишь один класс.

Это позволило сократить время обучения нейронной сети и достигнуть необходимых результатов распознавания строк и слов в строках. Для модели распознавания арабских символов, напротив, использовалась модель YOLOv8x по причине того, что в наборе данных было свыше 11 тысяч изображений и 22 класса. Для большого количества классов необходима и высокая точность распознавания каждого класса. Наибольшее количество параметров и более точное распознавание объектов среди других моделей YOLOv8 позволило достичь необходимых результатов в задаче распознавания арабских символов (рис. 7).

Model	size (pixels)	mAP ^{val} ₅₀₋₉₅	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

Рис. 7. Сравнительная таблица моделей YOLOv8

Модель распознавания строк обучалась на кастомном датасете, который был описан во втором разделе статьи. За основу был взят YOLOv8n с одним классом «line» (рис. 8). Обучение происходило на 30 эпохах и позволило достигнуть результатов: Precision: 0.89, Recall: 0.88, mAP50: 0.96, mAP50-95: 0.66. (Рис. 9). Для обучения модели использовались видеокарты NVIDIA GeForce GTX 1060 6GB и NVIDIA GeForce GT 1080 8GB. Приведенные метрики в дальнейшем также использовались для оставшихся двух моделей распознавания старотатарских слов и арабских символов.

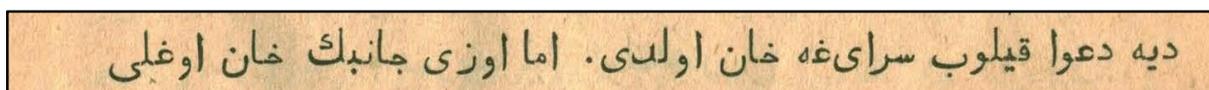


Рис. 8. Результат работы модели по распознаванию строк

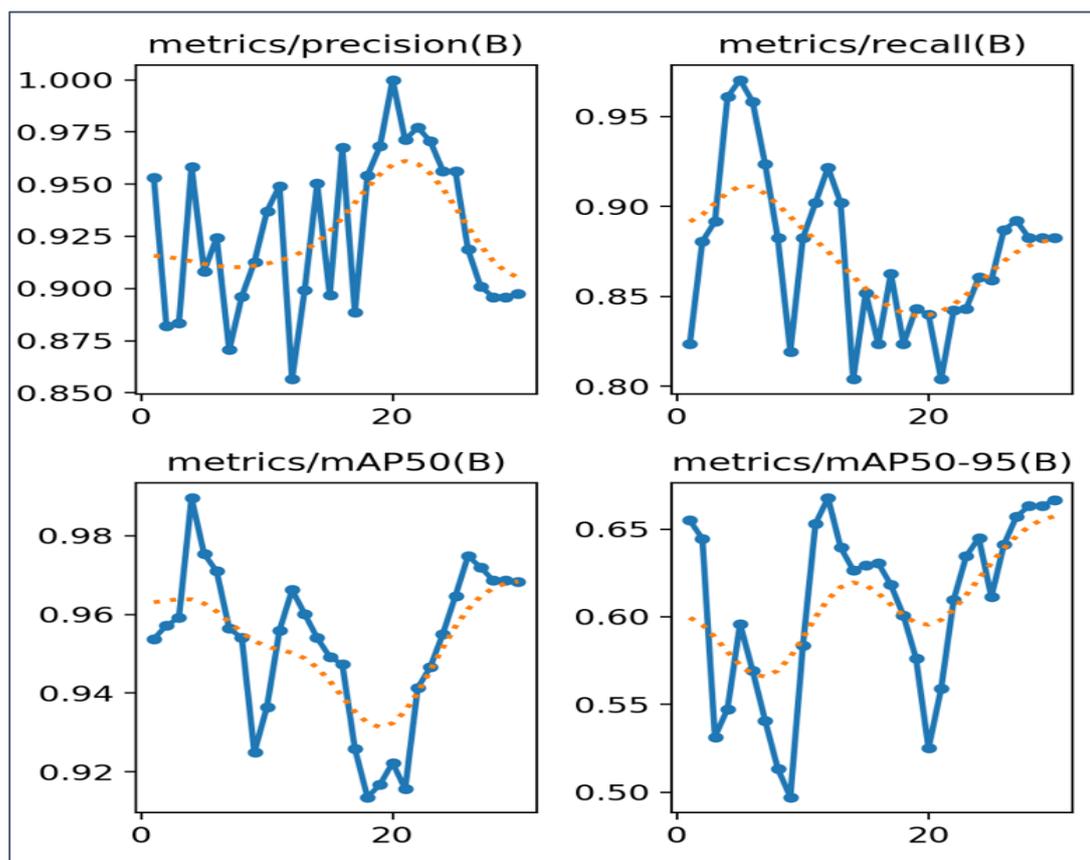


Рис. 9. Метрики, полученные в результате обучения модели распознавания строк

Модель распознавания старотатарских слов обучалась на кастомном датасете, который включал результаты работы предыдущей модели. На изображениях строк размечались старотатарские слова (рис. 10). За основу был взят YOLOv8n с одним классом «word». Обучение происходило на 50 эпохах и позволило достигнуть результатов: Precision: 0.95, Recall: 0.94, mAP50: 0.98, mAP50-95: 0.65 (рис. 11). Для обучения модели использовались видеокарты: NVIDIA GeForce GTX 1060 6GB и NVIDIA GeForce GT 1080 8GB.

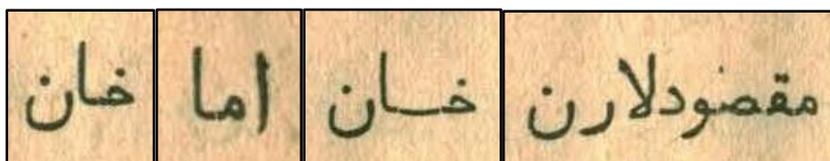


Рис. 10. Результат работы модели по распознаванию старотатарских слов из строк

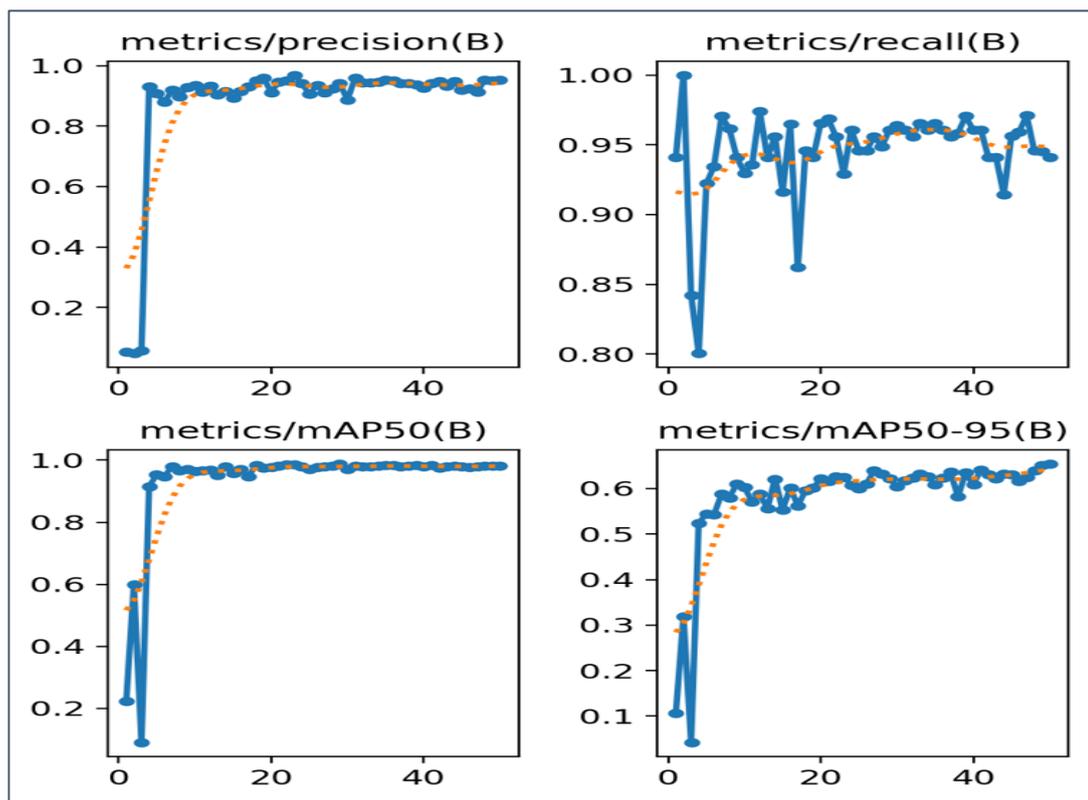


Рис. 11. Метрики, полученные в результате обучения модели распознавания старотатарских слов

Модель распознавания арабских символов обучалась на кастомном датасете изображений (результатов) из предыдущих моделей совместно с датасетом арабских рукописных символов HMBD v1. На изображениях старотатарских слов размечались арабские символы (рис. 12). За основу был взят YOLOv8x с 22 классами: A, Wau, Iy, N, K, R, L, D, B, T, M, S, Ha, G, Sh, La, Z, F, Ta, h, Zzh. Обучение происходило на 40 эпохах и позволило достигнуть результатов: Precision: 0.94, Recall: 0.93, mAP50: 0.95, mAP50-95: 0.67 (рис. 13). Для обучения модели использовались видеокарты NVIDIA GeForce GTX 1060 6GB и NVIDIA GeForce GT 1080 8GB.

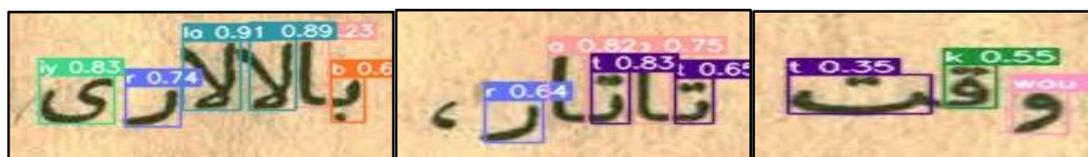


Рис. 12. Результат работы модели по распознаванию арабских символов в старотатарском слове. Слова: «Балалары», «Татар», «Вақыт».

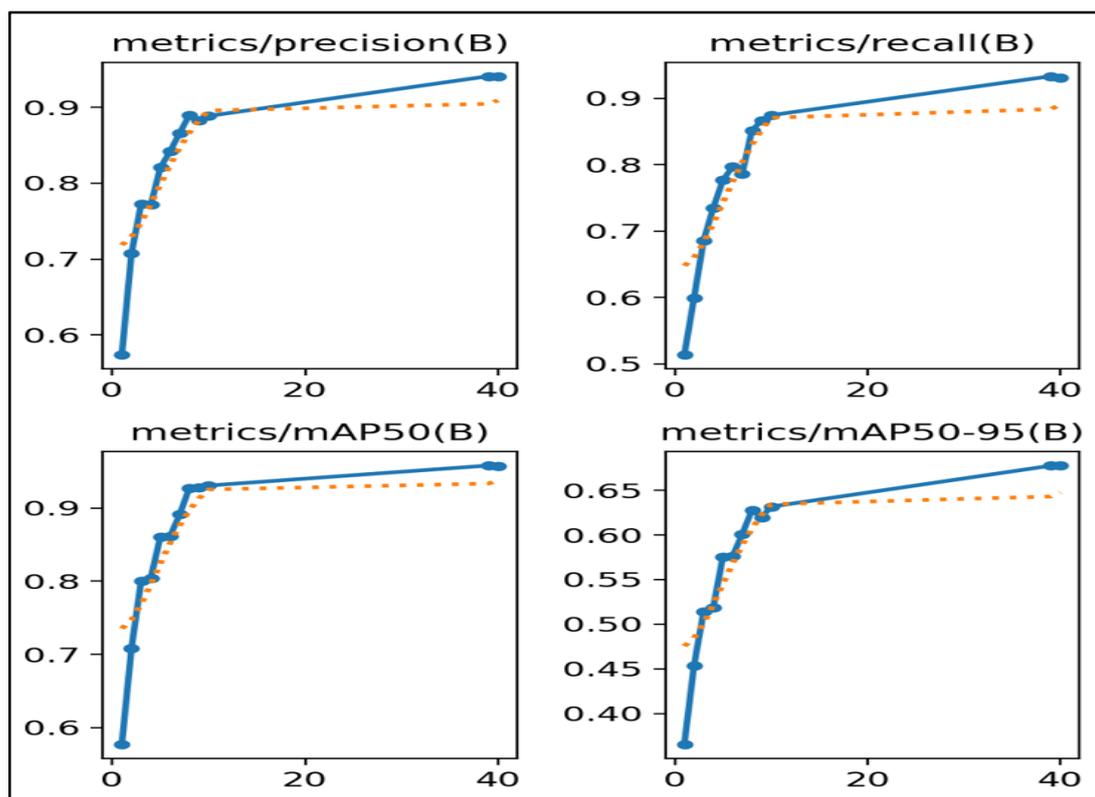


Рис. 13. Метрики, полученные в результате обучения модели распознавания арабских символов

В сумме модели выделяли строку, слово и арабский символ из отсканированного текста для последующих операций с ним. Результатом работы модели является старотатарское слово, буквы которого расшифрованы для удобного прочтения на латинице.

Итак, описаны разработанные подходы в решении задачи распознавания старотатарского текста, а также ход обучения моделей: распознавания строк, старотатарских слов и арабских символов. Представлены результаты работы моделей, гиперпараметры и результаты обучения в виде метрик.

ТЕСТИРОВАНИЕ СИСТЕМЫ

Чтобы определить точность работы системы и моделей, необходимо проверить ее на экземплярах, приближенных к реалиям. За основу тестирования системы были взяты отсканированные страницы со старотатарским текстом из произведения «Татар ханлары» (تاتار خانلاري) Мухаммадьяра Султанова, Типография братьев Каримовых, Казань, 1911 г. (рис. 14).

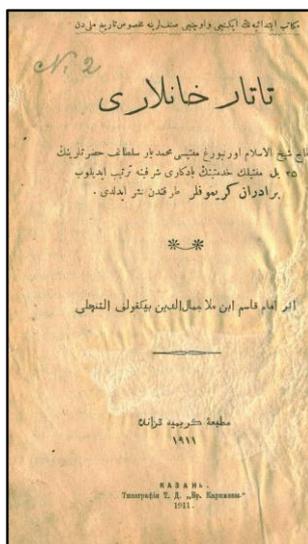


Рис. 14. Титульная страница произведения Мухаммальяра Султанова «Татар ханлары» (Татар Ханлары) на старотатарском языке, Типография братьев Каримовых, Казань, 1911 г.

Сначала система реализует модель распознавания и разделения строк. Строки необходимо разделять для дальнейших операций с распознанными изображениями в моделях распознавания слов и символов. Для тестирования модели распознавания строк было выбрано 3 примера из представленного выше произведения.

Для модели распознавания слов рассмотрено несколько примеров с использованием старотатарского текста. Изображения, которые подаются на вход модели, имеют размерность в среднем 1450 x 100 px.

Из отсканированного выше изображения на вход модели распознавания слов были поданы отрезки изображений в виде строк из текста. Модель смогла определить 95 слов из 135 (рис. 15). Это могло произойти по причинам распознавания не всех строк, также слишком близкого расстояния между словами. Такой результат возможен при установке `conf` (порог доверия) на уровень 65%. При снижении порога доверия результат становится значительно в плане результативности.

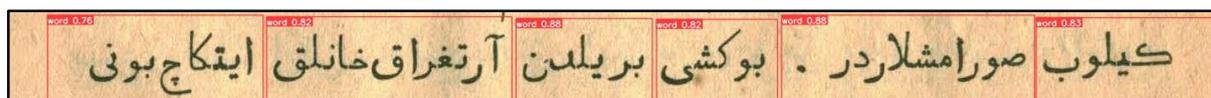


Рис. 15. Пример результата работы модели распознавания старотатарских слов

Далее на вход модели распознавания арабских символов подаются изображения слов, чтобы можно было разбить их на буквы и сделать, по возможности, транслитерацию этих слов. Из первых 10 слов примера из начала главы верно было распознано 57 арабских символов из 65 (рис. 16).

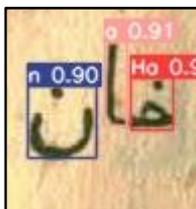


Рис. 16. Пример результата работы модели распознавания арабских символов.

Распознанное слово «Хан»

При тестировании второго случайного изображения из произведения «Татар ханлары» результатом стало 64 распознанных арабских символа из 66 на первые 10 распознанных слов.

Для более объективных результатов на вход системе подавались изображения из различных старотатарских оцифрованных произведений. Использовались в качестве примеров произведения: Мухаммадьяр Султанов «Татар ханлары» (تاتار خانلاري), Типография братьев Каримовых, Казань, 1911 г. 48 б. [21]; Шариф Камал. «Акчарлаклар» (آقچارلاقلر), Оренбург: "Вақыт" матбагасы, 1915. 88 б. [22]; «Аек бул» (آيق بول) Типо-литография Императорского университета, Казань, 1907 г. 16 б. [23]. Для тестирования системе подавалась одна случайная страница из приведенных выше произведений. В произведении «Татар ханлары» было распознано 14 из 17 строк (82.3 %), 95 слов из 135 (70.3 %), 614 символов из 644 (95.3 %), средняя точность по всем распознанным элементам в системе:

$$(82.3+70.3+95.3)/3=82.6\%.$$

В произведении «Аек бул» было распознано 6 из 9 строк (66.6 %), 16 слов из 23 (69.5 %), 50 символов из 52 (96.1 %), средняя точность по всем распознанным элементам в системе: $(82.3+70.3+95.3)/3=77.4\%$.

В произведении «Акчарлаклар» было распознано 19 из 25 строк (76 %), 145 слов из 174 (83.3 %), 854 символов из 893 (95.6 %), средняя точность по всем распознанным элементам в системе: $(82.3+70.3+95.3)/3=84.9\%$.

Результаты тестирования представлены в таблице 1.

Таблица 1. Результаты тестирования системы на основе выборки изображений из трех старотатарских оцифрованных произведений.

Произведение на старотатарском:	"Татар ханлары" 1911	"Аек бул" 1907	"Акчарлаклар" 1915
Распознанные строки	14 / 17	6 / 9	19 / 25
Распознанные слова	95 / 135	16 / 23	145 / 174
Распознанные символы	616 / 644	50 / 52	854 / 893
Средняя точность распознавания элементов	82.6 %	77.4 %	84.9 %

Результаты тестирования системы позволяют сделать вывод, что она умеет работать со старотатарским текстом: разделять изображение на строки, разделять строки на слова, слова на арабские буквы. Модель является показательной и не ставит перед собой цели получить высокую точность, но при этом может являться демонстрацией дальнейшего потенциала разработанного программного обеспечения.

РАЗРАБОТКА ГРАФИЧЕСКОГО ИНТЕРФЕЙСА

Для комфортной работы с моделями был создан графический интерфейс на основе PyQT5. Были разработаны функциональные требования для работы с моделями распознавания старотатарского текста. Перед разработкой графического интерфейса стоял выбор между Tkinter и PyQT. Выбор был сделан в пользу PyQT по причине большей производительности и более продвинутых возможностей для работы с изображениями по сравнению с Tkinter.

Графический интерфейс дает возможность пользователю осуществлять:

- Выбор изображения;
- Старт процесса распознавания выбранного изображения;
- Приближение изображения;
- Отдаление изображения;
- Подсчёт количества распознанных строк (рис. 17);
- Подсчёт количества распознанных слов (рис. 17);

- Подсчёт количества распознанных символов (рис. 17);
 - Отображение изображения распознанных строк;
 - Отображение координат распознанных строк;
 - Отображение изображений распознанных слов;
 - Отображение кириллической транслитерации распознанных слов;
- Графический интерфейс состоит из 4 окон:
- Окно с изображением распознанных строк (рис. 18);
 - Окно с прокруткой изображений распознанных слов (рис. 19);
 - Окно с транслитерацией на кириллице распознанных слов (рис. 20);
 - Окно с координатами распознанных строк (рис. 21).

РЕЗУЛЬТАТЫ:
Распознано 21 строк
Распознано 110 слов
Распознано 717 символов

Рис. 17. Вывод результатов распознавания строк, слов и символов



Рис. 18. Окно с изображением распознанных строк

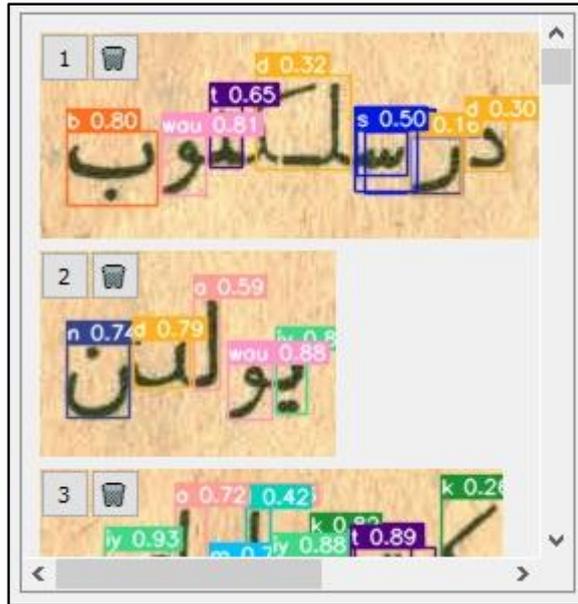


Рис. 19. Окно с прокруткой изображений распознанных старотатарских слов



Рис. 20. Окно с транслитерацией на кириллице распознанных старотатарских слов

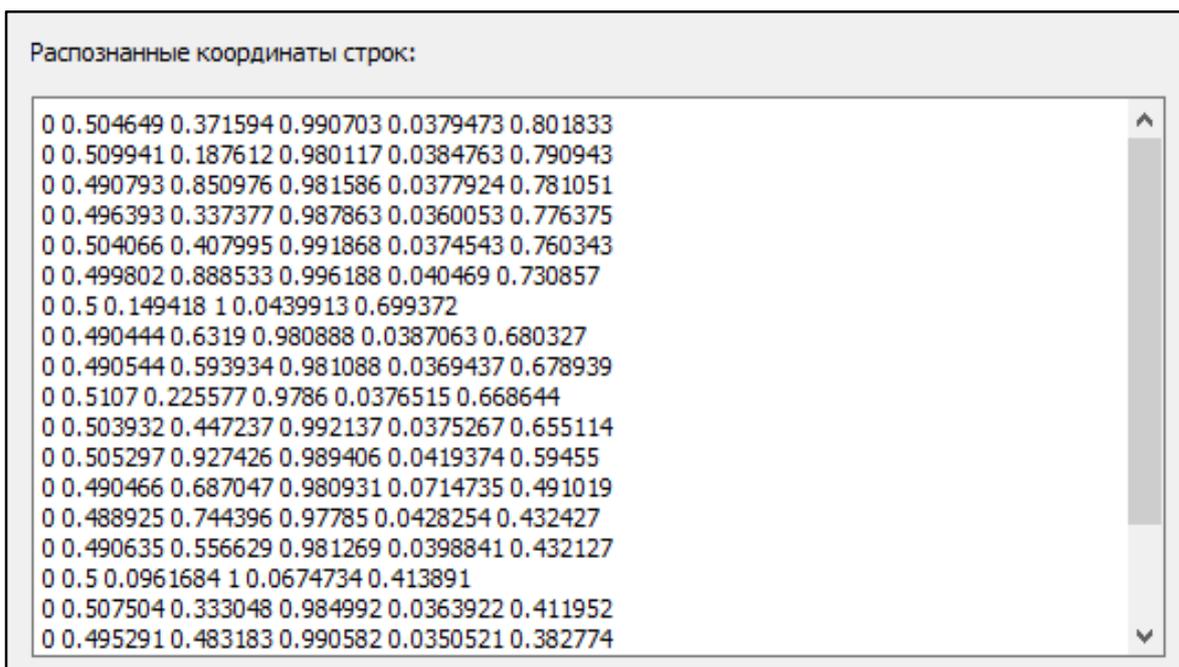


Рис. 21. Окно с координатами распознанных строк

Для удобства работы с функционалом приложения созданы кнопки:

- Выбор необходимого изображения для сканирования (рис. 22);
- Сканирование изображения (запуск процесса распознавания) (рис. 23);
- Приближение распознанного изображения (рис. 24);
- Отдаление распознанного изображения (рис. 25).

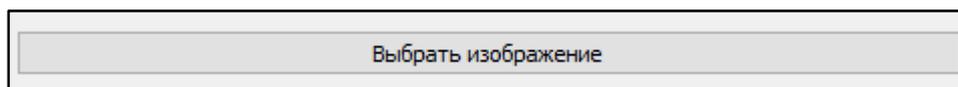


Рис. 22. Кнопка выбора необходимого изображения для сканирования

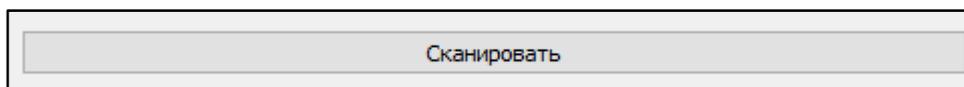


Рис. 23. Кнопка сканирования (запуска процесса распознавания текста)



Рис. 24. Кнопка приближения распознанного изображения



Рис. 25. Кнопка отдаления распознанного изображения

Чтобы новые пользователи смогли ориентироваться в приложении, была создана инструкция, которая отображается в окне справа при первом запуске программы (рис. 26).

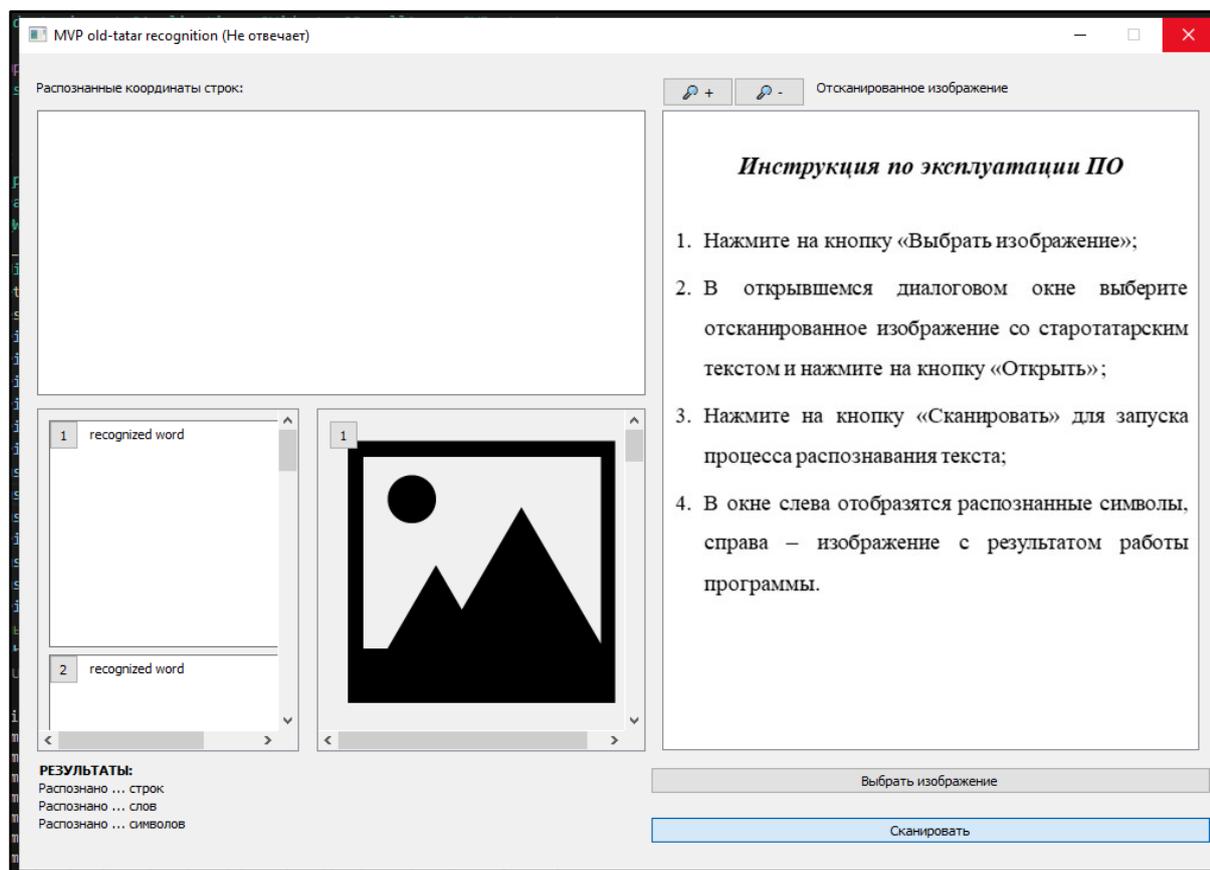


Рис. 26. Инструкция при начальном запуске программы

После выбора изображения и запуска процесса распознавания путем нажатия на кнопку «Сканировать» происходит процесс работы моделей распознавания строк, слов и символов. Отрабатываются функции транслитерации распознанных слов, подсчёта строк, подсчёта слов и подсчёта символов.

В пользовательском интерфейсе появляются результаты работы программы (рис. 27).



Рис. 27. Результаты работы программы

Разработанный графический интерфейс позволяет пользователям с удобством работать с процессом распознавания старотатарского текста. Созданные окна с транслитерацией старотатарских слов позволяют пользователям оперировать с результатами работы программы. Окна с распознанными строками и словами позволяют оперировать с текстом и находить необходимые слова.

ЗАКЛЮЧЕНИЕ

Таким образом, все поставленные задачи решены и достигнута цель исследования – разработка модели, способной с помощью методов компьютерного зрения распознать старотатарский текст.

Результатом работы стали:

- Создание набора данных для распознавания строк на старотатарском языке;
- Создание набора данных для распознавания слов на старотатарском языке;

- Создание набора данных для распознавания символов на старотатарском языке;
- Разработанная модель распознавания строк на старотатарском языке;
- Разработанная модель распознавания слов на старотатарском языке;
- Разработанная модель распознавания символов на старотатарском языке;
- Разработанный графический интерфейс модели.

Представленные разработанные методы и подходы в области распознавания старотатарского текста позволяют успешно решать также и задачи распознавания арабского текста. Модель протестирована и может быть полезна для исследователей древних рукописей, в которых использован старотатарский текст. Созданные датасеты могут быть востребованы в научной среде в задачах распознавания арабоязычных и иных текстов.

Исходные коды и файлы программы, которые были использованы в исследовательской работе, доступны по ссылке

URL: <https://github.com/iskander1998/old-tatar>

СПИСОК ЛИТЕРАТУРЫ

1. *Старовойтов В.В.* О цифровой реставрации исторических текстовых документов // Системный анализ и прикладная информатика. 2015. №1.
URL: <https://cyberleninka.ru/article/n/o-tsifrovoy-restavratsii-istoricheskikh-tekstovykh-dokumentov> (дата обращения: 24.04.2024).
2. Announcing Tesseract OCR – The official Google Code blog. URL: <https://googlecode.blogspot.com/2006/08/announcing-tesseract-ocr.html> (дата обращения: 24.04.2024).
3. *Rice S., Jenkins F., Nartker T.* The Fourth Annual Test of OCR Accuracy. 2012.
URL: https://www.researchgate.net/publication/247886491_The_Fourth_Annual_Test_of_OCR_Accuracy (дата обращения: 24.04.2024).
4. *Андрианов А.И.* Сравнение OCR-систем на основе точности анализа изображения // Бизнес-информатика. 2009. №4.
URL: <https://cyberleninka.ru/article/n/sravnenie-ocr-sistem-na-osnove-tochnosti-analiza-izobrazheniya> (дата обращения: 01.05.2024)

5. *Нестеров А.С.* Анализ рынка современных информационных систем оптического распознавания символов (OCR) // Вопросы науки и образования. 2020. №23 (107). URL: <https://cyberleninka.ru/article/n/analiz-rynka-sovremennyh-informatsionnyh-sistem-opticheskogo-raspoznavaniya-simvolov-ocr> (дата обращения: 01.05.2024).
 6. *Pechwitz M., Maddouri S.S., Märgner V., Ellouze N., Amiri H.* IFN/ENIT-database of handwritten Arabic words // Proc. of CIFED, Citeseer, 2002. P. 127–136.
 7. *Lawgali A., Angelova M., Bouridane A.* HACDB: Handwritten Arabic characters database for automatic character recognition // European Workshop on Visual Information Processing (EUVIP), 2013. P. 255–259.
 8. *Altwaijry N., Al-Turaiki I.* Arabic handwriting recognition system using convolutional neural network // Neural Comput. Appl. 2021. Vol. 33, No. 7. P. 2249–2261.
 9. *Balaha H.M., Ali H.A., Saraya M., Badawy M.* A new Arabic handwritten character recognition deep learning system (AHCR-DLS) // Neural Comput. Appl. 2021. Vol. 33, no. 11. P. 6325–6367.
 10. *Nayef B.H., Abdullah S.N.H.S., Sulaiman R., Alyasseri Z.A.A.* Optimized leaky ReLU for handwritten Arabic character recognition using convolution neural networks // Multimed. Tools Appl. 2022. Vol. 81, No. 2. P. 2065–2094.
 11. *Elkhayati M., Elkettani Y.* UnCNN: A New Directed CNN Model for Isolated Arabic Handwritten Characters Recognition // Arab J. Sci. Eng. 2022. Vol. 47, No. 8. P. 10667–10688.
 12. *Balaha H.M.* HMBD: Arabic Handwritten Characters Dataset. URL: <https://github.com/HossamBalaha/HMBD-v1> (дата обращения: 09.05.2024).
 13. *Balaha H.M., Ali H.A., Saraya M.* A new Arabic handwritten character recognition deep learning system (AHCR-DLS) // Neural Comput. Appl. 2021. Vol. 33. P. 6325–6367.
 14. *Zou Z., Chen K., Shi Z., Guo Y., Ye J.* Object Detection in 20 Years: A Survey // Proceedings of the IEEE. 2023. Vol. 111, No. 3. P. 257–276.
 15. *Закирьянов И.И., Хаялеева И.З., Валишин И.А., Курито Е.Д., Фасхутдинов А.Н.* Инструмент для распознавания языка жестов из видеопотока в режиме реального времени // Электронные библиотеки. 2023. Т. 26, № 6. URL: <https://rdl-journal.ru/article/view/804/876> (дата обращения: 01.05.2024).
-

16. Mulyana D., Rowis M. Optimization of Text Mining Detection of Tajweed Reading Laws Using the Yolov8 Method on the Qur'an // QALAMUNA: Jurnal Pendidikan, Sosial, Dan Agama. 2022. Vol. 14, No. 2. P. 1089–1110.

17. Badr Al-Badr., Sabri A.M. Survey and bibliography of Arabic optical text recognition // Signal Processing. 1995. Vol. 41, Issue 1. P. 49–77.

18. Turki H., Elleuch M., Kherallah M., Damak A. Arabic-Latin Scene Text Detection based on YOLO Models // International Conference on Innovations in Intelligent Systems and Applications (INISTA), Hammamet, Tunisia, 2023. P. 1–6.

19. Rahal N., Tounsi M., Hussain A., Alimi A.M. Deep Sparse Auto-Encoder Features Learning for Arabic Text Recognition // IEEE Access. 2021. Vol. 9. P. 18569–18584.

20. Султанов М. Татар ханлары (تاتار خانلاري) // Типография братьев Карим-вых. 1911.

URL: <https://darulkutub.com/uploads/books/820d9f6dcf1e868ee899d47e487b06189c2b816a.pdf> (дата обращения: 01.05.2024).

21. Камал Ш. Акчарлаклар (آقچار لاکلار) // "Вақыт" матбагасы. 1915.

URL: <https://miras.info/projects/mirasxane/books/425-akcharlaklar-.html> (дата обращения: 01.05.2024).

22. Аек бул (آيق بول) // Типо-литография Императорского университета. 1907.

URL: <https://darul-kutub.com/uploads/books/c3032b3c9136803dc0e38db69cd15541fb24064b.pdf> (дата обращения: 01.05.2024).

APPLICATION OF COMPUTER VISION METHODS TO OLD TATAR TEXT RECOGNITION

I. A. Valishin^[0009-0006-6891-031X]

Institute of Information Technologies and Intelligent Systems, Kazan (Volga Region) Federal University, st. Kremlevskaya, 35, Kazan, 420008

iskander1998@list.ru

Abstract

A developed tool that recognizes strings, words and Arabic characters from scanned images. The possibilities and prospects for using the tool in research activities

are considered. The results of experiments on the operational performance of the instrument are presented using the example of Old Tatar digitized images.

Keywords: YOLO, Arabic character recognition, neural networks, computer vision.

REFERENCES

1. *Starovoitov V.V.* O cifrovoj restavracii istoricheskikh tekstovykh dokumentov // *Sistemnyj analiz i prikladnaya informatika*. 2015. № 1.
URL: <https://cyberleninka.ru/article/n/o-tsifrovoy-restavratsii-istoricheskikh-tekstovykh-dokumentov>
2. Announcing Tesseract OCR – The official Google Code blog.
URL: <https://googlecode.blogspot.com/2006/08/announcing-tesseract-ocr.html>
3. *Rice S., Jenkins F., Nartker T.* The Fourth Annual Test of OCR Accuracy. 2012. URL: https://www.researchgate.net/publication/247886491_The_Fourth_Annual_Test_of_OCR_Accuracy
4. *Andrianov A.I.* Sravnenie OCR-sistem na osnove tochnosti analiza izobrazheniya // *Biznes-informatika*. 2009. № 4.
URL: <https://cyberleninka.ru/article/n/sravnenie-ocr-sistem-na-osnove-tochnosti-analiza-izobrazheniya>
5. *Nesterov A.S.* Analiz rynka sovremennykh informacionnykh sistem opticheskogo raspoznavaniya simvolov (OCR) // *Voprosy nauki i obrazovaniya*. 2020. № 23 (107).
URL: <https://cyberleninka.ru/article/n/analiz-rynka-sovremennykh-informatsionnykh-sistem-opticheskogo-raspoznavaniya-simvolov-ocr>
6. *Pechwitz M., Maddouri S.S., Märgner V., Ellouze N., Amiri H.* IFN/ENIT-database of handwritten Arabic words // *Proc. of CIFED, Citeseer*, 2002. P. 127–136.
7. *Lawgali A., Angelova M., Bouridane A.* HACDB: Handwritten Arabic characters database for automatic character recognition // *European Workshop on Visual Information Processing (EUVIP)*. 2013. P. 255–259.
8. *Altwayjry N., Al-Turaiki I.* Arabic handwriting recognition system using convolutional neural network // *Neural Comput. Appl.* 2021. Vol. 33, No. 7. P. 2249–2261.

9. *Balaha H.M., Ali H.A., Saraya M., Badawy M.* A new Arabic handwritten character recognition deep learning system (AHCR-DLS) // *Neural Comput. Appl.* 2021. Vol. 33, no. 11. P. 6325–6367.

10. *Nayef B.H., Abdullah S.N.H.S., Sulaiman R., Alyasseri Z.A.A.* Optimized leaky ReLU for handwritten Arabic character recognition using convolution neural networks// *Multimed. Tools Appl.* 2022. Vol. 81, No. 2. P. 2065–2094.

11. *Elkhayati M., Elkettani Y.* UnCNN: A New Directed CNN Model for Isolated Arabic Handwritten Characters Recognition // *Arab J. Sci. Eng.* 2022. Vol. 47, No. 8. P. 10667–10688.

12. *Balaha H.M.* HMBD: Arabic Handwritten Characters Dataset. URL: <https://github.com/HossamBalaha/HMBD-v1> (дата обращения: 09.05.2024).

13. *Balaha H.M., Ali H.A., Saraya M.* A new Arabic handwritten character recognition deep learning system (AHCR-DLS) // *Neural Comput. Appl.* 2021. Vol. 33. P. 6325–6367.

14. *Zou Z., Chen K., Shi Z., Guo Y., Ye J.* Object Detection in 20 Years: A Survey // *Proceedings of the IEEE.* 2023. Vol. 111, No. 3. P. 257–276.

15. *Zakiryaynov I.I., Khayaleeva I.Z., Valishin I.A., Kurito E.D., Faskhutdinov A.N.* Instrument dlya raspoznavaniya yazyka zhestov iz videopotoka v rezhime real-nogo vremeni // *Elektronnye biblioteki.* 2023. T. 26, № 6.

16. *Mulyana D., Rowis M.* Optimization of Text Mining Detection of Tajweed Reading Laws Using the Yolov8 Method on the Qur'an // *QALAMUNA: Jurnal Pendidikan, Sosial, Dan Agama.* 2022. Vol. 14, No. 2. P. 1089–1110.

17. *Badr Al-Badr., Sabri A.M.* Survey and bibliography of Arabic optical text recognition // *Signal Processing.* 1995. Vol. 41, Issue 1. P. 49–77.

18. *Turki H., Elleuch M., Kherallah M., Damak A.* Arabic-Latin Scene Text Detection based on YOLO Models // *International Conference on Innovations in Intelligent Systems and Applications (INISTA), Hammamet, Tunisia, 2023.* P. 1–6.

19. *Rahal N., Tounsi M., Hussain A., Alimi A.M.* Deep Sparse Auto-Encoder Features Learning for Arabic Text Recognition // *IEEE Access.* 2021. Vol. 9. P. 18569–18584.

20. *Sultanov M.* Tatar hanlary (ي ر ا ل ن ا خ ر ا ت ا ت) // *Tipografiya bratev Karimvyh.* 1911.

URL: <https://darulkutub.com/uploads/books/820d9f6dcf1e868ee899d47e487b06189c2b816a.pdf>

21. *Kamal Sh. Akcharlaklar* (دلقالراچقآ) // "Vakyt" matbagasy. 1915.

URL: <https://miras.info/projects/mirasxane/books/425-akcharlaklar-.html>

22. *Aek bul* (لوب قيا) // Tipo-litografiya Imperatorskogo universiteta. 1907.

URL: <https://darul-kutub.com/uploads/books/c3032b3c9136803dc0e38db69cd15541fb24064b.pdf>

СВЕДЕНИЯ ОБ АВТОРЕ



ВАЛИШИН Искандер Айратович – выпускник магистратур Института информационных технологий и интеллектуальных систем Казанского (Приволжского) федерального университета.

Iskander Airatovich VALISHIN – Master's at the Institute of Information Technology and Intelligent Systems, Kazan Federal University.

email: iskander1998@list.ru

ORCID: 0009-0006-6891-031X

Материал поступил в редакцию 24 июня 2024 года

ПРОГРАММНОЕ СРЕДСТВО ОПТИМИЗАЦИИ ПРОЦЕССОВ ВИДЕОПРОИЗВОДСТВА

Р. Ф. Давлетшин¹ [0009-0009-5326-3609], И. С. Шахова² [0000-0003-1591-5767]

^{1, 2}Институт информационных технологий и интеллектуальных систем,
Казанский (Приволжский) федеральный университет, г. Казань

¹rustemd02@mail.ru, ²is@it.kfu.ru

Аннотация

Предложены программные механизмы, направленные на оптимизацию процессов видеопроизводства для авторов художественных видеоматериалов – материалов, предполагающих предварительную постановочную работу. Разработан механизм создания анимированных трехмерных планов съемки (раскадровок) с использованием дополненной реальности для позиционирования и анимации перемещения актеров. С целью преодоления ограничений операционной системы iOS, связанных с доступом к сенсорам, разработан механизм отдельного захвата аудио- и видеопотоков с датчиков устройства для проведения записи, а также их последующей синхронизации по временным меткам для сохранения в память устройства. Отслеживание соблюдения правил композиционного построения и анализ качества изображения на предмет расфокусировки камеры реализованы с использованием технологий компьютерного зрения. Также представлены механизмы работы со сценарием, включающие алгоритмы обработки текста для вывода на экран в виде субтитров, а также распознавания речи актеров и сравнения её с текстом сценария.

Ключевые слова: видеопроизводство, мобильное кино, дополненная реальность, раскадровка, видеозапись, автоматизация, программное решение.

ВВЕДЕНИЕ

В последние годы отмечается тенденция к увеличению количества видеоматериалов (в том числе художественных произведений, предполагающих постановочную работу), созданных с использованием мобильных устройств [1, 2]. Данная тенденция частично обоснована регулярным совершенствованием аппаратной составляющей камер смартфонов [3], что приводит к улучшению качества изображения и его приближению к уровню изображения со специализированных камер [4], благодаря чему снижается порог входа в индустрию художественной съемки с точки зрения необходимого бюджета на кинооборудование [5].

Зачастую в процессе записи художественных видеоматериалов на любительском уровне вследствие недостатка бюджета режиссёр вынужден совмещать роли оператора, художника-постановщика и ассистента режиссёра по работе со сценарием, что может вызвать затруднения во время съёмки [6]. Так, режиссёру необходимо следить, чтобы актёры в своих действиях придерживались плана сцены; их реплики совпадали с прописанными фразами в сценарии (пренебрежение этим может вызвать логические проблемы с тавтологией [7], а заметно это станет лишь на этапе монтажа); а также следить за своими действиями как оператора: не допускать попадания в кадр элементов съёмочной площадки, соблюдать общие правила композиции кадра, и чтобы люди в кадре находились в фокусе [8].

Из вышесказанного следует, что ограниченный бюджет и невозможность расширения команды, ответственной за съёмку, не только усложняют процесс производства картины, но и негативно сказываются на итоговом результате. Решением обозначенной проблемы может стать программное средство для мобильных устройств, оптимизирующее указанные процессы. Под оптимизацией в данной работе понимается дополнение процессов отдельными сопроводительными программными механизмами, позволяющими частично автоматизировать отслеживание обозначенных ранее несоответствий плану сцены, реплик, требованиям к построению кадра.

Для анализа разработанности выявленной проблемы были рассмотрены программные решения для мобильных устройств, которые могут быть приме-

нимы на съёмочной площадке. В качестве критериев, которым должно удовлетворять решение проблемы, были сформулированы: возможность создания раскадровки и сохранения двух или более раскадровок, поддержка видеосъёмки, настройка параметров съёмки, механизмы расшифровки речи, поддержка русского языка.

Одним из программных решений, которое может быть потенциально применимо для использования в рассматриваемой предметной области, является LightSpace [9] – инструмент для отрисовки произвольных форм и линий в среде дополненной реальности, который может использоваться для зарисовки общего плана съёмки сцены. Однако в продукте отсутствует возможность анимации раскадровки, а также, ввиду особенностей продукта, нарисованные элементы будут отображаться и на итоговом видеофайле, что ограничивает использование решения в качестве инструмента для художественного кинопроизводства. StoryBoard Animator [10] предполагает создание раскадровок в виде набора зарисовок на белом холсте с возможностью их анимации. К недостаткам этого решения можно отнести отсутствие возможности видеосъёмки, а также тот факт, что созданные в приложении раскадровки ограничены пространством виртуального холста, что может вызвать затруднения, если локация съёмки окажется меньшей, чем предполагалось на этапе их планирования.

Для отслеживания несоответствия реплик актёров тексту сценария может быть использован инструмент SUFLER.PRO [11] – приложение-суфлер с возможностью видеозаписи. В сфере художественного кинопроизводства этот программный инструмент может использоваться оператором в качестве подсказки со сценарием текущей сцены, что позволит человеку за камерой визуально сравнивать фразы, прописанные в тексте, с репликами, которые в действительности произносятся актёры. Однако в этом продукте отсутствует возможность настройки таких параметров съёмки, как количество кадров в секунду, уровень баланса белого и др. (за исключением настройки разрешения записи), необходимость контроля которых в художественном кинопроизводстве отмечена в научных изданиях [12].

Аналогичная задача по выявлению несоответствий текста сценария и текста, фактически озвученного актёрами, может быть решена с использованием инструмента транскрибирования речи iOS Live Captions [13]. Он может быть применен в

процессе съёмки видео для проведения визуальной оценки наличия или отсутствия в речи актёров отклонений от изначального сценария. Недостатком этого решения является отсутствие поддержки русского языка.

Нашей целью была разработка программного средства, обладающего функциональными возможностями создания и анимации виртуальных раскадровок, осуществления видеозаписи без отображения элементов раскадровки в итоговом файле, анализа картинка на предмет нарушения композиции и расфокусировки, а также возможностью расшифровки произнесенной речи и сравнения ее со сценарием.

АРХИТЕКТУРА ПРОГРАММНОГО РЕШЕНИЯ

В связи с комплексной структурой разрабатываемого программного решения, в качестве архитектурного шаблона для программного продукта было принято решение использовать VIPER.

Архитектура программного решения, а также взаимодействие модулей приложения с прочими элементами программного кода представлены на рисунке 1.

Архитектура программного решения делится на три основные составляющие: модули программных частей решения, сервисы взаимодействия с компонентами устройства и операционной системы, а также набор сущностей.

Набор модулей включает следующие компоненты: главное меню, основной экран видеозаписи и создания раскадровки, а также экран редактирования сценария. Взаимодействие данных модулей с датчиками устройства и элементами базы данных производится при помощи соответствующих сервисов: работы с компонентами видеосъёмки, транскрибирования устной речи, работы с базой данных и механизмами персистентности. При этом для согласования единиц различных структур данных сервис видеозаписи производит их конвертацию при помощи соответствующего компонента программного решения.

Набор сущностей состоит из сущности данных об актёре в виртуальной раскадровке, метаданных непосредственно самой раскадровки, настроек видеоза-

писи, а также вспомогательных сущностей перечисления *enum*, содержащих данные о частоте кадров и скорости перемещения трехмерных моделей по раскладке.

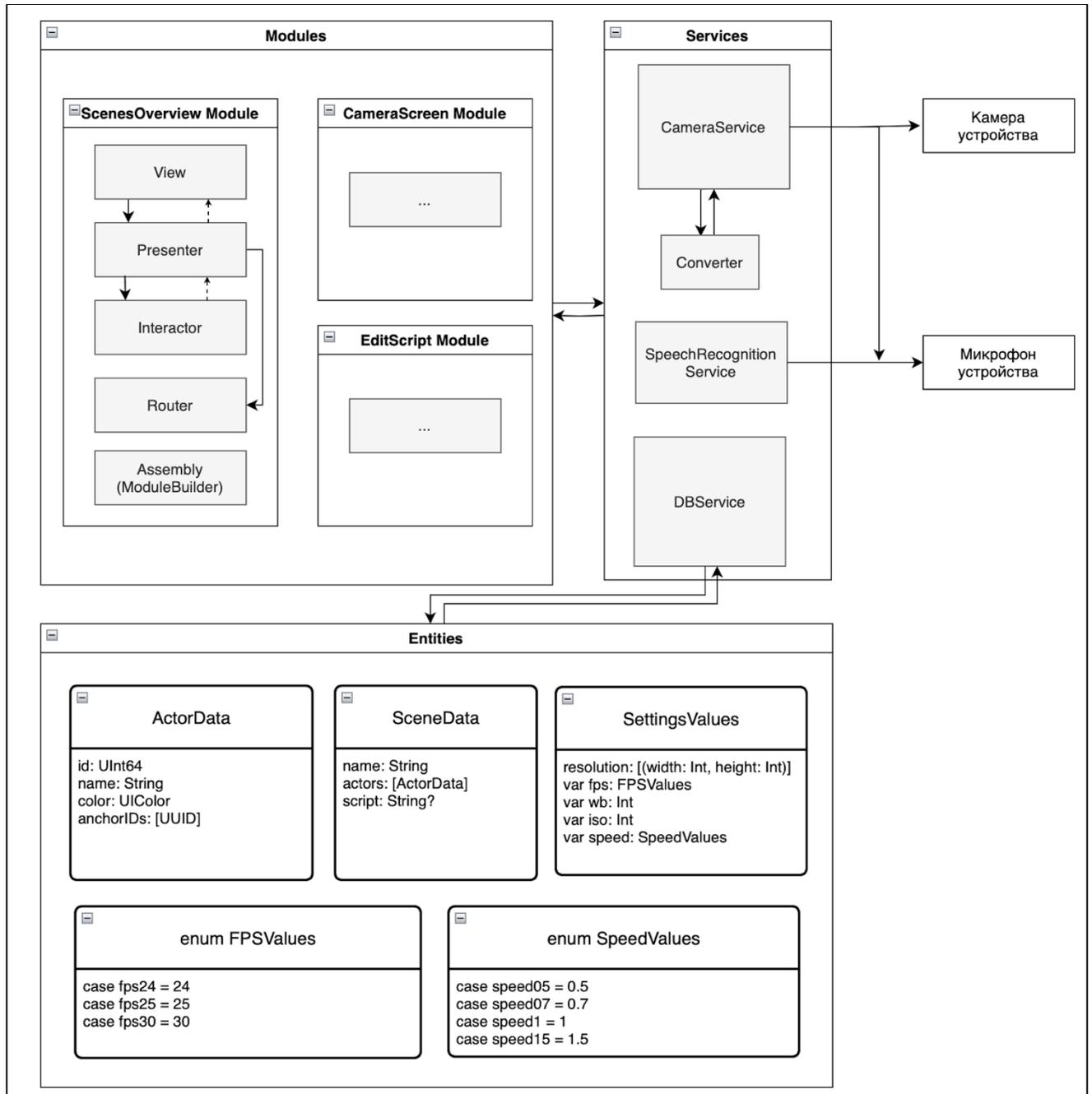


Рис. 1. Упрощенная схема архитектуры программного решения

ПРОГРАММНЫЕ МЕХАНИЗМЫ НА ОСНОВЕ ТЕХНОЛОГИИ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ

При выборе технологии для работы с дополненной реальностью были сформулированы следующие ограничения:

- целевые устройства – устройства, под управлением операционной системы iOS;
- использование программных библиотек и технологий, нативных для экосистемы Apple.

Было выбрано несколько фреймворков, удовлетворяющих поставленным ограничениям: SceneKit [14], ARKit [15], RealityKit [16]. Был проведен анализ доступных в экосистеме Apple фреймворков дополненной реальности, по результатам которого было принято решение использовать фреймворк ARKit в связке с RealityKit, поскольку ARKit уже включает многие необходимые инструменты, такие как, например, автоматическая калибровка по различным поверхностям. Однако он не включает возможность отрисовки трёхмерной графики, поэтому для рендеринга графики была использована библиотека RealityKit.

К основным компонентам используемых фреймворков ARKit и RealityKit относятся:

- **ARView** – компонент, который используется для визуального представления среды дополненной реальности на экране устройства;
- **Scene** (сцена) – контейнер, содержащий все элементы трехмерного окружения, включая 3D-модели;
- **ARAnchor** – якорная точка, которая соотносится фреймворком ARKit с определенной координатой в пространстве и к которой впоследствии прикрепляются 3D-модели;
- **ModelEntity** – виртуальный объект, представляющий собой трехмерную модель, которая размещается на сцене и прикреплена к якорной точке.

При инициации пользователем создания на сцене нового трехмерного объекта необходимо вычислить координаты формата *simd_float4x4* для создания ARAnchor на соответствующих координатах и соотношения модели с данным якорем. Вычисление координат осуществляется путем использования метода *raycast*

фреймворка RealityKit. Этот механизм направляет луч от центра экрана до ближайшей поверхности и, если поверхность найдена, возвращает результат, из которого можно получить координаты точки, на которую упал луч.

С целью визуального разделения трехмерных моделей, обозначающих актеров в виртуальной раскадровке, при создании объекта на сцене также создается и дочерний элемент, прикрепленный к данному объекту и представляющий собой однострочное текстовое представление, которое пользователь может редактировать (поле может использоваться, например, для имени актера). Так как по иерархии компонентов сцены ARKit модель имени актера является дочерним объектом модели актера, то их перемещения синхронизированы.

Путь перемещения трехмерных моделей актеров в виртуальной раскадровке задается с помощью меток путей, создаваемых пользователем. Метки путей представляют собой отдельные ModelEntity, прикрепленные к своим элементам ARAnchor. Уникальные идентификаторы меток путей модели сохраняются в виде массива в соответствующем поле, которым обладают объекты структуры данных, созданной для моделей актеров. При этом, поскольку запись происходит в порядке создания пользователем меток пути, данные метки автоматически выступают в качестве контрольных точек, задавая таким образом порядок перемещения модели актера.

Таким образом, диаграмму иерархии компонентов сцены ARView можно представить в виде, приведенном на рисунке 2.

Для наглядности пунктирной линией на рисунке представлена условная связь якорей меток пути и якорей соответствующего актера.

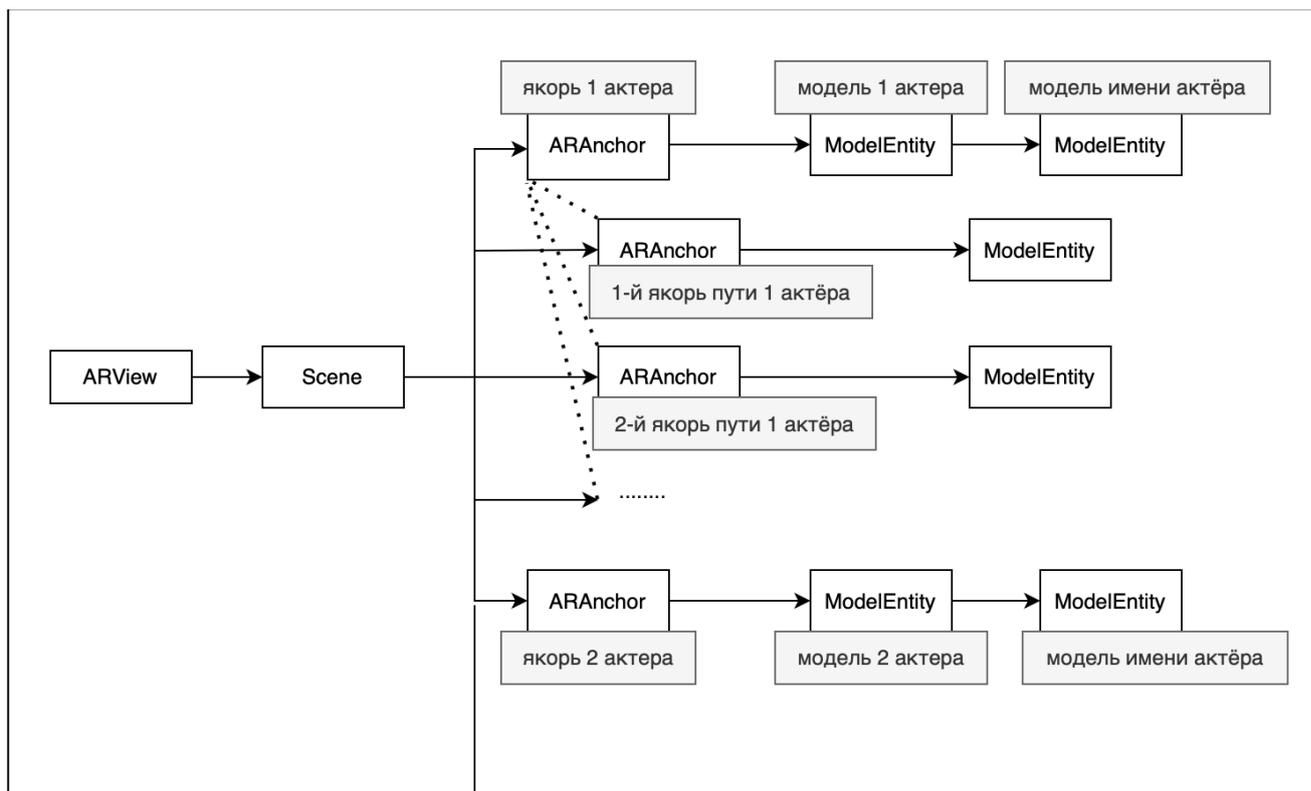


Рис. 2. Диаграмма иерархии компонентов ARView

Для перемещения трехмерных моделей по сцене в ARKit есть соответствующий метод *move()*, который принимает на вход координаты финальной точки перемещения, а также время, за которое анимация должна завершиться. При таком подходе время, за которое модель актёра в раскадровке перемещается от точки А до точки Б, было бы идентичным вне зависимости от расстояния, что, очевидно, не соответствовало бы реальной скорости перемещения актёров на площадке. Поэтому был разработан алгоритм, позволяющий динамически вычислять время перемещения трехмерной модели по сцене, исходя из расстояния между контрольными точками и заданной скорости движения.

Для работы алгоритма необходимы координаты начальной точки перемещения, а также конечной (или промежуточной к финальной, в таком случае необходимо выполнить алгоритм *n* раз, где *n* – количество промежуточных контрольных точек). Расположение трехмерных моделей в виртуальной сцене ARView описано в виде матрицы размера 4 на 4 [17] (рис. 3).

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

Рис. 3. Матрица координат трехмерных моделей в ARView

Каждый из элементов данной матрицы используется для представления соответствующей координаты объекта в трехмерном пространстве:

- a_{11}, a_{12}, a_{13} – представляют вращение вокруг оси X;
- a_{21}, a_{22}, a_{23} – представляют вращение вокруг оси Y;
- a_{31}, a_{32}, a_{33} – представляют вращение вокруг оси Z;
- a_{14}, a_{24}, a_{34} – представляют смещение объекта относительно точки начала координат по осям X, Y, Z соответственно;
- $a_{41}, a_{42}, a_{43}, a_{44}$ – координаты проецирования на поверхность изображения.

Соответствующая матрица преобразований имеется у каждого ModelEntity, прикрепленного к ARAnchor. Можно заметить, что для целей перемещения моделей по сцене используются элементы a_{14}, a_{24}, a_{34} , обозначающие смещение объекта.

После получения координат начальной и конечной точек перемещения вычисляется расстояние между соответствующими точками каждой из осей, для чего применяется алгоритм нахождения расстояния между двумя точками в n -мерном пространстве, также известного как евклидово расстояние [18]. Для вычисления времени перемещения трехмерной модели (в секундах) найденное расстояние делится на скорость, заданную пользователем. Полученное в результате значение передается в качестве входного параметра в упомянутый ранее метод `move()`.

СОХРАНЕНИЕ И ЗАГРУЗКА СЦЕНЫ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ

Фреймворк ARKit предоставляет возможность получить текущую карту трехмерного окружения с помощью встроенного метода `currentWorldMap()`, который

на выходе дает объект класса ARWorldMap, содержащий текущее состояние окружения в сессии дополненной реальности на момент вызова функции [19]. Полученный объект класса ARWorldMap сохраняется в памяти устройства компонентом класса NSKeyedArchiver. Сохранение производится по уникальному идентификатору сцены в папку в файловой системе устройства, создаваемую программным решением. По идентификатору также происходят последующая загрузка, а также удаление данных. Для сохранения данных о трехмерных моделях используется протокол Codable.

Так как объект класса ARWorldMap хранит в себе идентификаторы якорей вместе с их координатами в пространстве, а сохраненные данные о трехмерных моделях также включают в себя поле идентификатора якоря, к которому была прикреплена данная модель в сохраненной сессии, то задача загрузки сцены и восстановления предыдущего состояния сессии сводится к сопоставлению информации о моделях по совпадающим идентификаторам, повторному созданию трехмерных моделей на сцене в соответствии с хронологическим порядком расстановки их пользователем, включая прикрепление дочерних элементов к моделям, таких как элемент имени актера и метки пути, а также установке цвета трехмерным объектам.

ЗАХВАТ ВИДЕОПОТОКА С КАМЕРЫ УСТРОЙСТВА

На этапе проработки концепции программного решения предполагалось, что процесс захвата видеопотока будет происходить параллельно процессам, реализующим взаимодействие с технологиями дополненной реальности, однако после проведения ряда испытаний было установлено, что процессы, запущенные на операционной системе iOS, обладают исключительным доступом к датчикам и сенсорам устройства, что исключает возможность одновременного их использования разными приложениями и процессами.

Концепция, реализованная в программном решении, заключается в следующем: фреймворк ARKit обладает механизмом обратного вызова `session(_:didUpdate:)`, который срабатывает при каждом обновлении кадра и предоставляет возможность делегировать обработку кадра стороннему программному коду, предоставляя при этом кадры напрямую с камеры устройства,

то есть до этапа наложения трехмерной графики. Было принято решение создать алгоритм, который после начала записи покадрово принимает на вход изображения с камеры устройства, записывает их в буфер, а по окончании записи производит склейку в единый видеофайл. При этом подразумевалось, что кадры, полученные с помощью метода `session(_:didUpdate:)`, являются исключительно визуальными элементами и не содержат аудиоинформации, поэтому необходимо также одновременно производить запись аудио с помощью микрофона мобильного устройства. Это означает, что в дополнение к обозначенной выше задаче склейки отдельных кадров одним из шагов алгоритма перед выдачей готового видеофайла является объединение видеодорожки с записанным аудиопотоком. С целью исключения появления задержки между нажатием пользователем на кнопку начала записи и фактическим стартом захвата видео инициализация компонентов, необходимых для данного процесса, происходит сразу после запуска программного решения (рисунки 4 и 5).

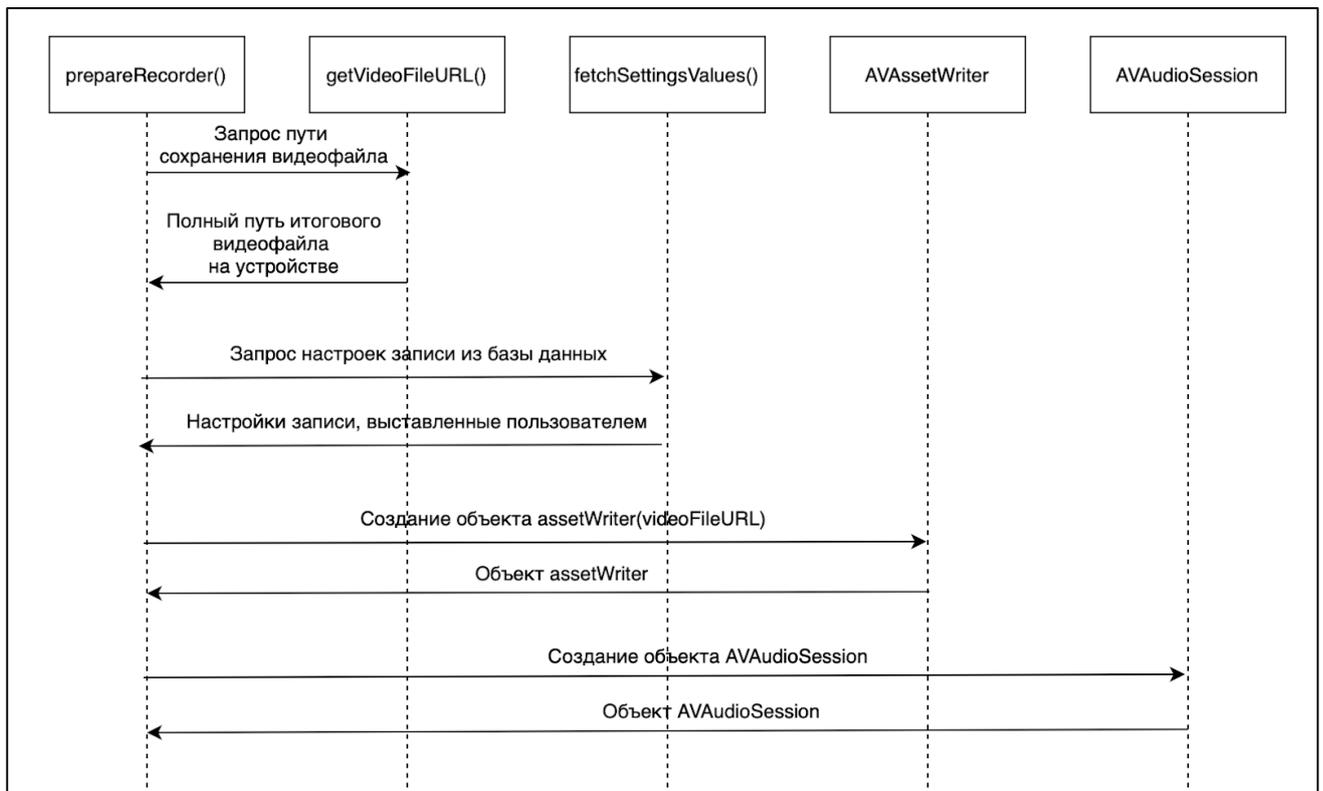


Рис. 4. Схема алгоритма инициализации компонентов видеозахвата

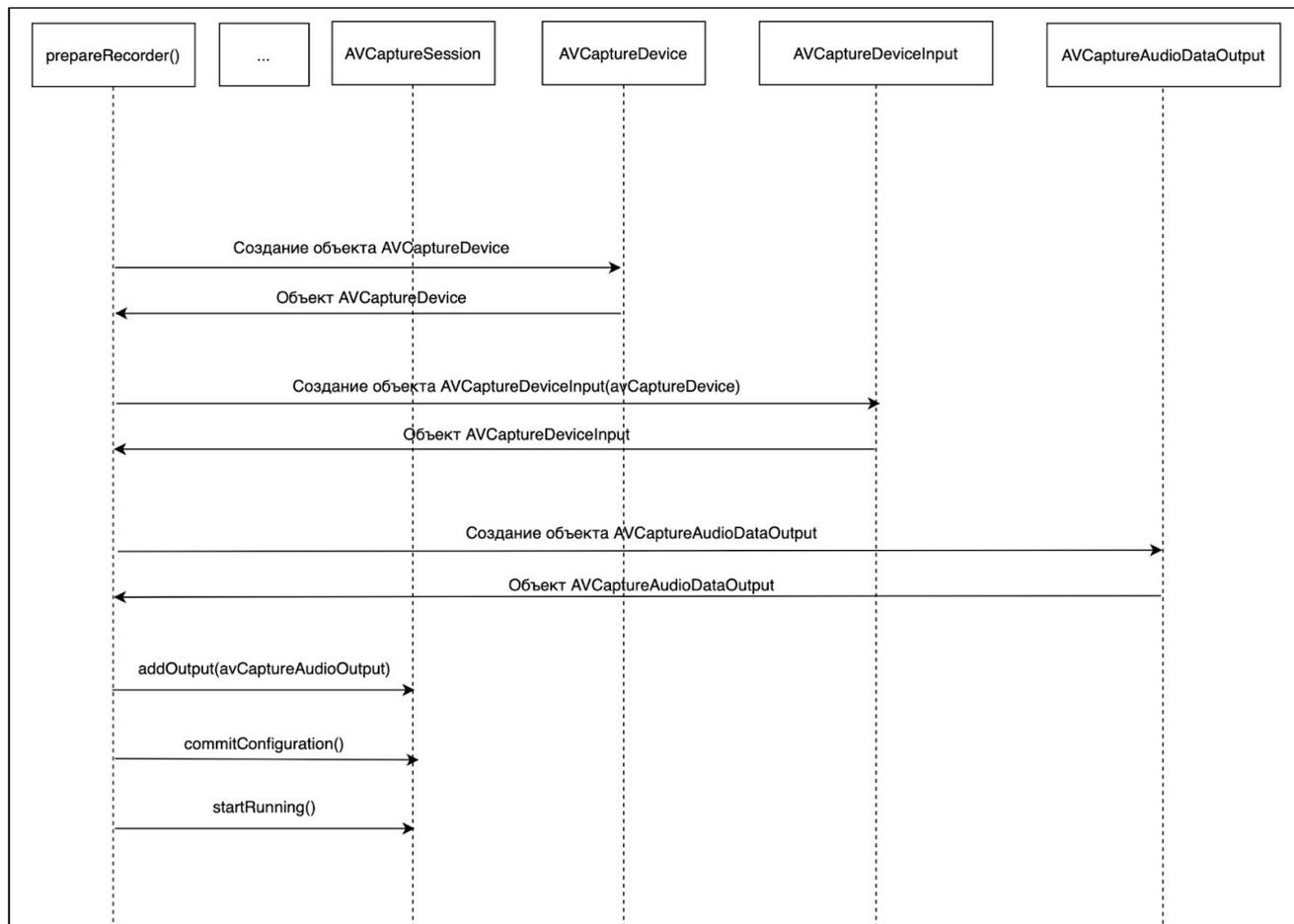


Рис. 5. Схема алгоритма инициализации компонентов видеозахвата (продолжение)

На рисунках 4 и 5 обозначены следующие методы и классы:

- *prepareRecorder()* вызывается при запуске приложения и является стартовой точкой для инициализации объектов видеозахвата;
- *getVideoFileURL()* – метод, производящий генерацию названия будущего видеофайла и возвращающий в качестве выходного параметра путь для его сохранения;
- *fetchSettingsValues()* производит запрос в сервис работы с базой данных для получения пользовательских настроек записи;
- *AVAssetWriter* – часть фреймворка *AVKit*, которая производит запись медиаданных в файл;

- *AVAudioSession* – компонент, выступающий в роли посредника между программным продуктом и механизмами, связанными с воспроизведением и записью аудио в системе iOS в целом;
- *AVCaptureSession* координирует потоки данных с устройств ввода (например, камеры или микрофона) для их записи;
- *AVCaptureDevice* является представлением физического устройства ввода;
- *AVCaptureDeviceInput* – обертка над классом *AVCaptureDevice*, которая отвечает за передачу данных от устройства ввода к *AVCaptureSession*;
- *AVCaptureAudioDataOutput* позволяет проводить аудиозапись и получать доступ к записанным звуковым данным в момент захвата.

ЗАПИСЬ ВИДЕОПОТОКА И СИНХРОНИЗАЦИЯ С ФРАГМЕНТАМИ АУДИО

При нажатии пользователем на кнопку записи начинают работу методы, связанные непосредственно с захватом и записью видеопотока в память устройства. С заранее заданными параметрами происходит создание буфера, куда записывается текущий кадр, полученный от *ARView*, а также вычисляется временная метка захвата кадра.

Вместе с этим производятся захват аудио и синхронизация с видеорядом. Алгоритм работает следующим образом:

1. сохраняется временная метка полученного буфера, содержащего фрагмент аудио длиной, эквивалентной 1 кадру;
2. вычисляется время представления путем вычитания временной метки начала записи (временной метки первого видеокadra) из временной метки буфера текущего фрагмента;
3. происходит создание нового объекта буфера путем копирования полученного и изменения его временной метки для синхронизации с видеорядом. Для решения данной задачи используется функция *CMSampleBufferCreateCopyWithNewTiming*, в качестве входных параметров принимающая:
 - *allocator* – указывает на то, какой аллокатор памяти будет использован при копировании буфера;
 - *sampleBuffer* – исходный буфер фрагмента аудио;

- *sampleTimingEntryCount* – количество элементов, подаваемых на вход, у которых необходимо изменить временную метку;
- *sampleTimingArray* – указатель на переменную, содержащую информацию о новой временной метке для фрагмента аудио;
- *sampleBufferOut* – указатель на переменную, куда будет произведено копирование исходного буфера.

4. новый объект буфера со временем представления, синхронизированным с видеорядом, записывается в *assetWriterAudioInput*.

В результате работы алгоритма записанные фрагменты аудио синхронизируются по времени с видеокадрами и вместе с ними добавляются в очередь на запись, благодаря чему видеоряд и записанный отдельно аудиоряд итогового видеофайла синхронизированы друг с другом. Визуальное представление работы алгоритма в формате “До – После” представлено на рисунках 6 и 7.

		timestamp	timestamp	timestamp	timestamp	timestamp	timestamp	timestamp
видео		0 сек. 1 кадр	0 сек. 2 кадр	0 сек. 3 кадр	0 сек. 4 кадр	...	0 сек. 25 кадр	1 сек. 1 кадр
аудио		1	2	3	4	...	25	26
		timestamp начала и завершения записи						

Рис. 6. Визуальное представление видеофайла до работы алгоритма

		timestamp	timestamp	timestamp	timestamp	timestamp	timestamp	timestamp
видео		0 сек. 1 кадр	0 сек. 2 кадр	0 сек. 3 кадр	0 сек. 4 кадр	...	0 сек. 25 кадр	1 сек. 1 кадр
аудио		1	2	3	4	...	25	26
		timestamp начала и завершения записи						

Рис. 7. Визуальное представление видеофайла после работы алгоритма

При нажатии пользователем на кнопку окончания записи *assetWriterVideoInput* и *assetWriterAudioInput* помечаются соответствующим флагом, а у *assetWriter* вызывается метод завершения работы, после чего происходит сохранение итогового видеофайла в галерею устройства с использованием нативного фреймворка Photos [20].

МЕХАНИЗМЫ ОБРАБОТКИ ВИДЕОПОТОКА

В кинематографе, как и в других областях визуальных искусств, включая фотографию и графический дизайн, применяется правило третей. Это принцип композиционного построения изображения, основанный на визуальном делении кадра на 9 частей при помощи сетки 3 на 3. Считается, что расположение основного объекта в кадре (например, актера) на пересечении линий сетки позволяет сделать кадр более визуальным сбалансированным для глаза человека [21]. В программном решении реализовано два вспомогательных инструмента, связанных с правилом третей:

1. Механизм отслеживания положения головы актера в кадре и отображения предупреждения пользователю при нарушении правила третей, основанный на функционале нативного фреймворка Vision. Для отслеживания головы в пространстве применяется компонент *VNFaceObservation*, позволяющий покадрово получать координаты лица человека. Далее выполняется проверка на текущее расположение головы человека, и в случае, если граница лица человека пересекает линию пересечения противоположной стороны (для линии пересечения с левой стороны изображения – это правая граница лица, для линии с нижней стороны – верхняя часть лица, и аналогично для остальных сторон), то предполагается, что правило третей нарушено. При обнаружении данного нарушения на экран выводится иконка, обозначающее соответствующее предупреждение для пользователя, а вокруг лица, для которого, предположительно, было нарушено правило третей, происходит отрисовка красной рамки.
2. Сетка правила третей – отрисовка происходит на этапе жизненного цикла *UIViewController viewDidAppear* и производится путем создания на экране прозрачного слоя *CALayer()* поверх основного изображения и последующего добавления на этот слой линий, являющихся объектами *UIBezierPath()*.

Также был добавлен механизм отслеживания качества изображения на предмет возможного отсутствия фокусировки камеры на лице человека. На небольшом экране устройства может быть затруднительно визуально определить, на чем сфокусирована камера в данный момент, из-за чего лицо актера может оказаться размытым, что станет заметно лишь при просмотре итогового видеофайла на большом экране. Поэтому в программное средство был добавлен вывод предупреждения, если качество картинки участка изображения с лицом человека окажется ниже заданного порогового уровня. Для реализации также был использован компонент VNFaceObservation.

МЕХАНИЗМЫ РАБОТЫ СО СЦЕНАРИЕМ

Сценарий в приложение загружается пользователем в текстовом формате, при этом предполагается, что текст, подаваемый на вход, состоит из набора отдельных реплик, имеющих вид, представленный на рисунке 8.

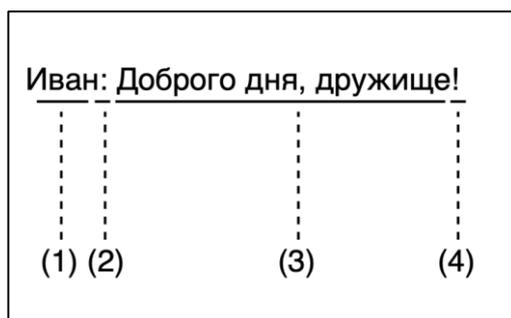


Рис. 8. Возможный вариант реплики, подаваемой на вход

Числовыми значениями на рисунке обозначены:

1. имя персонажа или актера;
2. символ двоеточия;
3. фраза (набор произвольных символов, кроме символов завершения предложения);
4. символ завершения предложения (точка, знак вопроса, восклицательный знак).

После загрузки пользователем текста сценария, удовлетворяющего шаблону, начинает работу алгоритм анализа и разделения текста на составные части. Суть

его работы заключается в следующем: алгоритм посимвольно проходится по тексту, полученному на вход. Как только следующим символом становится двоеточие, прочитанный текст записывается в переменную, отвечающую за имя. Затем происходит посимвольное чтение текста до символа завершения предложения. На данном этапе возможны 2 варианта:

1. реплика состоит из одного предложения, и то, что следует за символом завершения предложения, – следующая реплика;
2. реплика состоит из нескольких предложений, и то, что следует за символом завершения предложения, – продолжение данной реплики.

Соответственно программному решению необходимо учитывать возможность того, что реплика будет удовлетворять одному из двух вышеперечисленных вариантов, для чего алгоритм, как только следующим символом становится символ завершения предложения, предполагает, что следующий за ним текст может быть как новой репликой, так и продолжением текущей реплики. Определяется это следующим образом:

1. если реплика состоит из одного предложения, и символ завершения предложения обозначает ее завершение, то следующим за символом текстом будет имя персонажа, чья реплика идет далее по сценарию, соответственно после имени будет идти символ двоеточия;
2. если реплика состоит из нескольких предложений, и символ завершения предложения разделяет предложения внутри реплики, то следующим за символом текстом будет любой набор символов, после которого может идти любой небуквенный символ, кроме двоеточия.

Таким образом, алгоритм основан на предположении, что текст, следующий за символом завершения предложения, может быть как именем персонажа, кто говорит следующую реплику, именем внутри фразы, так и самой фразой.

МЕХАНИЗМЫ РАСШИФРОВКИ РЕЧИ

В программном решении механизмы расшифровки речи используются в тандеме с компонентом вывода субтитров на экран. При старте записи начинает свою работу инструмент распознавания речи, в основе которого лежат компоненты нативного фреймворка Speech [22] (рис. 9).

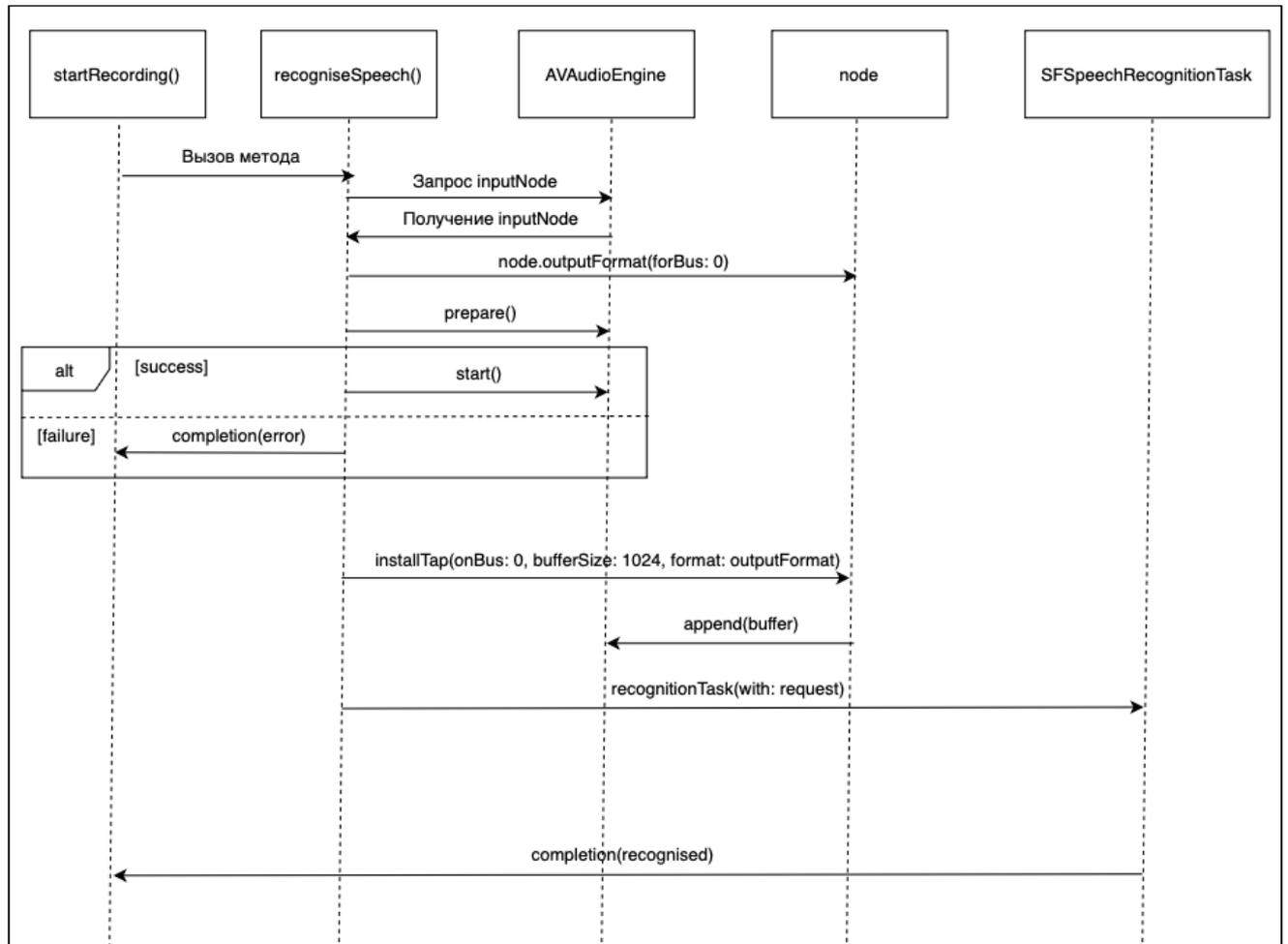


Рис. 9. Схема алгоритма транскрибирования речи

На рисунке 9 обозначены следующие методы и классы:

- *startRecording()* – метод, который вызывается при инициации пользователем начала записи. Является входной точкой компонента распознавания речи;
- *recogniseSpeech()* принимает замыкание *completion()*, которое вызывается при успешном распознавании или возникновении ошибки;
- *AVAudioEngine()* – аудиодвижок, отвечающий за получение звукового сигнала от узлов (*node*);
- *node / inputNode* – узел, производящий захват аудиосигнала;
- *SFSpeechRecognitionTask* – объект задачи распознавания речи;
- *outputFormat(forBus: 0)* – задание у узла формата записи;
- *prepare()* подготавливает аудиодвижок к запуску;

- *start()* производит запуск аудиодвижка;
- *installTap(onBus: 0, bufferSize: 1024, format: recordingFormat)* производит установку обработчика, захватывающего данные на узле *inputNode*;
- *append(buffer)* добавляет захваченные данные в буфер для распознавания;
- *recognitionTask(with: request)* запускает задачу распознавания речи;
- *completion(recognised)* вызывает замыкание *completion()* с распознанным текстом.

Результатом работы алгоритма является объект *recognised*, содержащий весь транскрибированный текст с момента начала записи. С целью проверки идентичности транскрибированного текста и текста фразы данные элементы обрезаются до двух последних слов, для чего применяются стандартные инструменты работы со строками языка Swift, и уже они проверяются на идентичность. Учитывая специфику работы инструментов распознавания речи, различия в регистре текста, а также расстановка знаков препинания в нем алгоритмом не учитываются.

Предполагается, что если в результате проведенных действий алгоритм выявил сходство распознанного текста и текста, заданного в сценарии, то реплика произнесена, и в субтитре происходит переключение на следующую фразу.

ЗАКЛЮЧЕНИЕ

Разработаны программные механизмы создания и пользовательской настройки анимированных трехмерных раскадровок. Для динамического вычисления скорости перемещения трехмерных моделей в нескольких режимах создана математическая модель с использованием метрики Евклида. Сохранение и загрузка контейнера элементов трехмерного окружения, настроек записи и загруженного сценария реализованы с использованием механизмов конвертации информации о моделях в примитивные типы данных. Для преодоления ограничений операционной системы iOS, связанных с доступом к сенсорам устройства, реализован механизм перехвата кадра у фреймворка дополненной реальности до этапа наложения трехмерной графики с параллельным вычислением временных меток каждого кадра и работающие одновременно с этим механизмы записи аудиопотока и последующей синхронизации видео- и аудиодорожек для форми-

рования итогового видеофайла с сохранением его в галерею устройства. Реализованы механизмы отслеживания положения головы человека в кадре и качества изображения на предмет возможной расфокусировки, основанные на применении технологий компьютерного зрения. Предложены также механизмы работы с загруженным в приложение сценарием, включая алгоритм сериализации сплошного текста для наглядного вывода его на экран, а также компонентов транскрибирования речи, захваченной с аудиопотока микрофона устройства, и последующего сравнения расшифрованной речи с текстом сценария.

СПИСОК ЛИТЕРАТУРЫ

1. Wang H., Yang S., Dai Z. The Development of Mobile Short Video Communication in the Context of the Mobile Internet // *Science Insights*. 2022. Vol. 40. No. 1. P. 421–426. URL: <https://doi.org/10.15354/si.22.re011>.
2. The Rise of Mobile Filmmaking: A New Era of Cinematic Expression // *ECGProductions*. 2023. URL: <https://www.ecgprod.com/the-rise-of-mobile-filmmaking/>.
3. Blahnik V., Schindelbeck O. Smartphone imaging technology and its applications // *Advanced Optical Technologies*. 2021. Vol. 10, No. 3. P. 145–232. URL: <https://doi.org/10.1515/aot-2021-0023>.
4. Smartphones vs Cameras: Closing the gap on image quality // *DxOMark*. 2021. URL: <https://www.dxomark.com/smartphones-vs-cameras-closing-the-gap-on-image-quality/>.
5. Schleser M. Smartphone filmmaking: theory and practice. // *Bloomsbury Publishing USA*. 2021. P. 8–10.
6. Friis I., Hansen A. Line-item budgeting and film-production: Exploring some benefits of budget constraints on creativity // *Qualitative Research in Accounting & Management*. 2015. Vol. 12, No. 4. P. 321–345. URL: <https://doi.org/10.1108/QRAM-01-2015-0016>.
7. Balaban O., Levchenko O., Krupskyy I., Medvedieva A., Mykhalov V. Script Structures in Modern Audio-Visual Art // *Studies in Media and Communication*. 2021. Vol. 9, No. 2. P. 45–50. URL: <https://doi.org/10.11114/smc.v9i2.5386>.

8. *Brown B.* Cinematography: Theory and Practice. 2016. P. 28–44.
9. LightSpace – 3D painting in air.
URL: <https://apps.apple.com/ru/app/lightspace-3d-painting-in-ar/id1274597316>.
10. Storyboard Animator.
URL: <https://apps.apple.com/ru/app/storyboard-animator/id1326518944>.
11. SUFLER.PRO.
URL: <https://apps.apple.com/ru/app/sufler-pro/id1480258675>.
12. *Shaner P.* Digital Filmmaking: An Introduction. // Mercury Learning and Information. 2011. P. 34–50.
13. Get live captions in real time on iPhone.
URL: <https://support.apple.com/en-in/guide/iphone/iphe0990f7bb/ios>.
14. Providing 3D virtual content with SceneKit.
URL: https://developer.apple.com/documentation/arkit/arscnview/providing3dvirtualcontentwith_scenekit.
15. ARKit Documentation.
URL: <https://developer.apple.com/documentation/arkit/>.
16. Improving the performance of a RealityKit app.
URL: <https://developer.apple.com/documentation/realitykit/improving-the-performance-of-a-realitykit-app>.
17. SIMD Float4x4.
URL: https://developer.apple.com/documentation/accelerate/simd_float4x4.
18. *Saito T., Toriwaki J.I.* New algorithms for euclidean distance transformation of an n-dimensional digitized picture with applications // Pattern Recognition. 1994. Vol. 27, No. 11. P. 1551–1565.
19. ARWorldMap.
URL: <https://developer.apple.com/documentation/arkit/arworldmap>.
20. PhotoKit. URL: <https://developer.apple.com/documentation/photokit>.
21. Cinematography Techniques: The Different Types.
URL: <https://rudafilms.com/gallery/Cinematography%20Techniques%20The%20Different%20Types.pdf>.
22. Speech. URL: <https://developer.apple.com/documentation/speech>.

SOFTWARE TOOL FOR VIDEO PRODUCTION OPTIMISATION

R. F. Davletshin¹ [0009-0009-5326-3609], I. S. Shakhova² [0000-0003-1591-5767]

^{1,2}*Institute of Information Technology and Intelligent Systems, Kazan Federal University, Kazan*

¹rustemd02@mail.ru, ²is@it.kfu.ru

Abstract

The paper proposes software mechanisms aimed at enhancing video production processes for the authors of artistic video materials. We propose a mechanism for creating animated three-dimensional shooting plans (storyboards) using augmented reality to position and animate the movement of actors. In order to overcome the limitations of the iOS operating system related to access to sensors, we developed a mechanism for separately capturing audio and video streams from device sensors for recording and their subsequent synchronization by timestamps for saving to device memory. Computer vision technologies are used to ensure compliance with the rules of compositional construction and image quality analysis. The paper also presents mechanisms for working with the script, including text processing algorithms for displaying subtitles on the screen, and speech recognition algorithms for comparing speech recognition of actors with the text of the script.

Keywords: *video production, movie making, mobile cinema, augmented reality, storyboard, video recordings, automation, software solution.*

REFERENCES

1. Wang H., Yang S., Dai Z. The Development of Mobile Short Video Communication in the Context of the Mobile Internet // *Science Insights*. 2022. Vol. 40. No. 1. P. 421–426. URL: <https://doi.org/10.15354/si.22.re011>.
2. The Rise of Mobile Filmmaking: A New Era of Cinematic Expression // *ECGProductions*. 2023. URL: <https://www.ecgprod.com/the-rise-of-mobile-filmmaking/>.
3. Blahnik V., Schindelbeck O. Smartphone imaging technology and its applications // *Advanced Optical Technologies*. 2021. Vol. 10, No. 3. P. 145–232.

URL: <https://doi.org/10.1515/aot-2021-0023>.

4. Smartphones vs Cameras: Closing the gap on image quality // DxOMark. 2021.

URL: <https://www.dxomark.com/smartphones-vs-cameras-closing-the-gap-on-image-quality/>.

5. *Schleser M.* Smartphone filmmaking: theory and practice. // Bloomsbury Publishing USA. 2021. P. 8–10.

6. *Friis I., Hansen A.* Line-item budgeting and film-production: Exploring some benefits of budget constraints on creativity // *Qualitative Research in Accounting & Management*. 2015. Vol. 12, No. 4. P. 321–345.

URL: <https://doi.org/10.1108/QRAM-01-2015-0016>.

7. *Balaban O., Levchenko O., Krupskyy I., Medvedieva A., Mykhalov V.* Script Structures in Modern Audio-Visual Art // *Studies in Media and Communication*. 2021. Vol. 9, No. 2. P. 45–50. URL: <https://doi.org/10.11114/smc.v9i2.5386>.

8. *Brown B.* Cinematography: Theory and Practice. 2016. P. 28–44.

9. LightSpace – 3D painting in air.

URL: <https://apps.apple.com/ru/app/lightspace-3d-painting-in-ar/id1274597316>.

10. Storyboard Animator.

URL: <https://apps.apple.com/ru/app/storyboard-animator/id1326518944>.

11. SUFLER.PRO.

URL: <https://apps.apple.com/ru/app/sufler-pro/id1480258675>.

12. *Shaner P.* Digital Filmmaking: An Introduction. // Mercury Learning and Information. 2011. P. 34–50.

13. Get live captions in real time on iPhone.

URL: <https://support.apple.com/en-in/guide/iphone/iphe0990f7bb/ios>.

14. Providing 3D virtual content with SceneKit.

URL: https://developer.apple.com/documentation/arkit/arscnview/providing3dvirtualcontentwith_scenokit.

15. ARKit Documentation.

URL: <https://developer.apple.com/documentation/arkit/>.

16. Improving the performance of a RealityKit app.

URL: <https://developer.apple.com/documentation/realitykit/improving-the-performance-of-a-realitykit-app>.

17. SIMD Float4x4.

URL: https://developer.apple.com/documentation/accelerate/simd_float4x4.

18. *Saito T., Toriwaki J.I.* New algorithms for euclidean distance transformation of an n-dimensional digitized picture with applications // Pattern Recognition. 1994. Vol. 27, No. 11. P. 1551–1565.

19. ARWorldMap.

URL: <https://developer.apple.com/documentation/arkit/arworldmap>.

20. PhotoKit. URL: <https://developer.apple.com/documentation/photokit>.

21. Cinematography Techniques: The Different Types.

URL: <https://rudafilms.com/gallery/Cinematography%20Techniques%20The%20Different%20Types.pdf>.

22. Speech. URL: <https://developer.apple.com/documentation/speech>.

СВЕДЕНИЯ ОБ АВТОРАХ



ДАВЛЕТШИН Рустем Фаридович – бакалавр, выпускник Института информационных технологий и интеллектуальных систем Казанского федерального университета, г. Казань.

Rustem Faridovich DAVLETSHIN – Bachelor's degree graduate from the Institute of Information Technology and Intelligent Systems at Kazan Federal University, Kazan.

email: rustemd02@mail.ru

ORCID: 0009-0009-5326-3609



ШАХОВА Ирина Сергеевна – старший преподаватель кафедры программной инженерии Института информационных технологий и интеллектуальных систем Казанского федерального университета, г. Казань.

Irina Sergeevna SHAKHOVA – Senior Lecturer at the Department of Software Engineering, Institute of Information Technology and Intelligent Systems, Kazan Federal University, Kazan.

email: is@it.kfu.ru

ORCID: 0000-0003-1591-5767

Материал поступил в редакцию 18 июля 2024 года

РАЗРАБОТКА НА ОСНОВЕ НЕЧЕТКОЙ ЛОГИКИ ЭКСПЕРТНОЙ СИСТЕМЫ ДЛЯ ДИАГНОСТИКИ ПНЕВМОНИИ

А. И. Еникеева¹ [0009-0002-9192-952x], Р. А. Бурнашев² [0000-0002-1057-0328],

Р. Р. Фарахов³ [0000-0002-7993-3182]

¹⁻³Казанский (Приволжский) федеральный университет

¹adisen24@yandex.ru, ²r.burnashev@inbox.ru, ³rus-wing-dark@mail.ru

Аннотация

Статья посвящена разработке экспертной системы для диагностики пневмонии. Эта система основана на нечеткой логике и реализована с использованием алгоритма Мамдани. Рассмотрены основные этапы разработки системы, включая фаззификацию входных данных, определение нечетких правил на основе медицинских экспертных знаний, агрегирование нечетких выводов и их дефаззификацию для получения окончательного диагностического результата. Веб-интерфейс системы реализован с использованием фреймворка Django, что обеспечивает удобство взаимодействия для пользователей. Применение медицинской экспертной системы для диагностики пневмонии может сократить время, необходимое для установления диагноза, и улучшить качество диагностики за счет интеграции опыта медицинских экспертов и современных информационных технологий.

Ключевые слова: экспертная система, нечеткая логика, медицинская диагностика, алгоритм Мамдани, система поддержки принятия решений.

ВВЕДЕНИЕ

Пневмония остается одним из наиболее распространенных и опасных заболеваний дыхательной системы, поэтому точная и быстрая диагностика играет важную роль в успешном лечении. Своевременная диагностика пневмонии имеет решающее значение для эффективного лечения, однако традиционные методы диагностики требуют значительных ресурсов и экспертной интерпретации, что может быть недоступно в условиях ограниченных ресурсов. В связи с этим разработка медицинской экспертной системы, способной повысить эффективность и точность диагностики, становится актуальной задачей.

Экспертная система на основе нечеткой логики моделирует процесс рассуждения медицинских специалистов, предоставляя ценные инструменты для поддержки принятия решений, а также является эффективным инструментом современной медицины. Система позволяет повысить точность и скорость диагностики пневмонии, что в свою очередь способствует раннему началу лечения и улучшению прогноза для пациентов.

Целью системы является предоставление систематических и надежных средств оценки пневмонии путем интеграции различных клинических параметров в единую диагностическую структуру. Работа системы основана на сравнении введенных данных с базой правил, содержащей информацию о признаках пневмонии.

1. ОБЗОР МЕТОДОВ НЕЧЕТКОЙ ЛОГИКИ

Нечеткая логика является обобщением классической теории множеств и формальной логики и обеспечивает эффективные средства отображения, неопределенностей и неточностей реального мира [1]. В рамках нечеткой логики переменные могут принимать любые значения на интервале $[0, 1]$. На основе этого понятия вводятся различные логические операции над нечеткими множествами и формулируется понятие лингвистической переменной, в качестве значений которой выступают нечеткие множества [2].

Экспертные системы, основанные на нечеткой логике, используют различные алгоритмы нечеткого вывода для обработки данных [3]. Основными алгоритмами являются алгоритмы Мамдани, Сугено и Ларсен [4].

Для разработки экспертной системы был выбран алгоритм Мамдани, который является одним из наиболее популярных методов нечеткого вывода и отличается простотой реализации. Алгоритм примечателен тем, что он работает по принципу «черного ящика». На вход поступают количественные значения, на выход – они же. На промежуточных этапах используется аппарат нечеткой логики и теория нечетких множеств [5].

База нечетких правил Мамдани содержит лингвистические правила, использующие функции принадлежности для описания применяемых концепций [6].

Алгоритм Мамдани поддерживает добавление и изменение правил без значительных изменений в структуре системы [7]. Это особенно важно для медицинской диагностики, где постоянно появляются новые знания и методы лечения. В случае

необходимости новые правила могут быть легко добавлены, что обеспечивает долгосрочную актуальность и адаптивность системы.

В общем случае механизм логического вывода включает четыре этапа: введение нечеткости (фазификация), нечеткий вывод, композиция и приведение к четкости, или дефазификация (см. рисунок 1) [8].

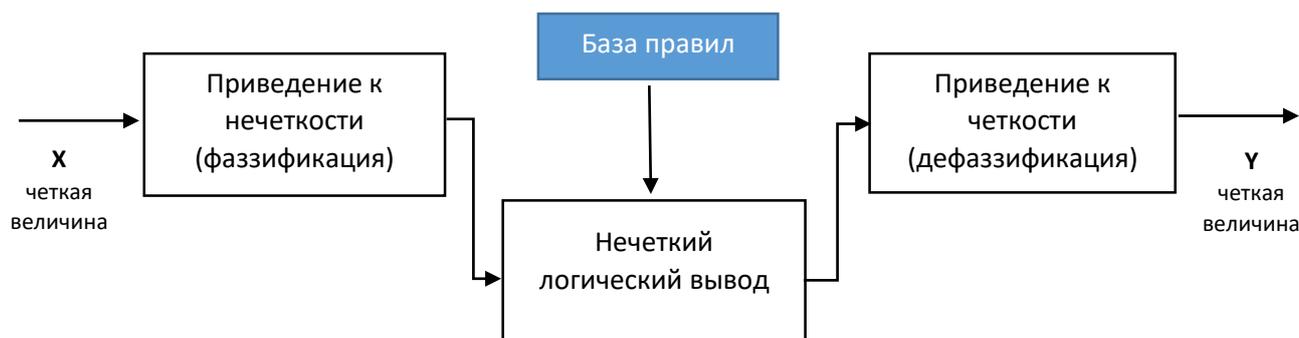


Рис. 1. Процесс обработки нечеткой логики

Алгоритм Сугено [9] является альтернативой алгоритму Мамдани и также широко используется в нечетких системах. Однако он чаще применяется в задачах управления и оптимизации, где выходные значения – линейные или полиномиальные функции входных переменных. Для медицинской диагностики, где важны интерпретируемость и простота правил, алгоритм Мамдани предпочтительнее.

Алгоритм Ларсена [10] схож с алгоритмом Мамдани, но отличается использованием произведения вместо минимума при агрегации правил. Это может привести к более сложным вычислениям и менее понятным результатам, что делает алгоритм Мамдани более подходящим для медицинских систем.

2. МОДУЛЬ НЕЧЕТКОЙ ЛОГИКИ

2.1. ОПРЕДЕЛЕНИЕ КЛИНИЧЕСКИХ ПАРАМЕТРОВ

Для разработки медицинской экспертной системы были выбраны следующие ключевые клинические параметры, указывающие на наличие пневмонии [11, 12]:

1. Возраст;
2. Кашель;
3. Одышка;
4. Температура;

5. Боль в грудной клетке;
6. Усталость;
7. Уровень кислорода в крови;
8. Уровень лейкоцитов в крови;
9. Уровень тромбоцитов в крови.

2.2. ФОРМИРОВАНИЕ БАЗЫ ПРАВИЛ

Основой для проведения операции нечеткого логического вывода является база правил, построенная на основе экспертных знаний. Каждое правило формулируется в виде утверждений «ЕСЛИ–ТО» и функции принадлежности для соответствующих лингвистических термов. Оценка условий каждого правила происходит с использованием логических операций над степенями принадлежности [13].

Процесс формирования базы правил включает следующие этапы:

1. Определение термов нечетких переменных. Для каждой входной и выходной переменных были определены термы на основе медицинских источников [11, 12] (см. таблицы 1 и 2).
2. Формулировка правил. На основе термов были сформулированы правила, отражающие причинно-следственные связи между симптомами и вероятностью пневмонии (см. таблицу 2).

Таблица 1. Определение термов нечетких входных переменных.

Входные значения	Лингвистические термы
Возраст (Age)	молодой (young)
	средний (middle)
	пожилой (elderly)
Кашель (Cough)	отсутствует (asymptomatic)
	легкий (mild)
	умеренный (moderate)
	сильный (severe)
Одышка (Shortness of breath)	отсутствует (asymptomatic)
	легкая (mild)
	умеренная (moderate)
	сильная (severe)

Температура (Temperature)	низкая (low)
	нормальная (normal)
	повышенная (elevated)
	высокая (high)
Боль в грудной клетке (Chest pain)	отсутствует (asymptomatic)
	легкая (mild)
	умеренная (moderate)
	сильная (severe)
Усталость (Fatigue)	отсутствует (asymptomatic)
	легкая (mild)
	выраженная (severe)
Уровень кислорода в крови (Saturation)	норма (normal)
	гипоксия (hypoxia)
	критическое состояние (critical)
Лейкоциты (Leukocytes)	пониженные (low)
	норма (normal)
	высокие (high)
Тромбоциты (Platelets)	пониженные (low)
	норма (normal)
	высокие (high)

Таблица 2. Определение термов нечетких выходных переменных.

Выходные значения	Выходные значения
Риск (Risk)	Низкий (low)
	Умеренный (moderate)
	Высокий (high)
	Очень высокий (very high)

Таблица 3. Пример правил экспертной системы.

	Правила
RULE 1:	IF Age IS Young AND Cough IS Asymptomatic AND Shortness of Breath IS Asymptomatic AND Temperature IS Normal AND Chest pain IS Asymptomatic AND Fatigue IS Asymptomatic AND Saturation IS Normal AND Leukocytes IS Normal AND Platelets IS Normal THEN Risk IS LOW
RULE 2:	IF Age IS Middle AND Cough IS Asymptomatic AND Shortness of breath IS Asymptomatic AND Temperature IS Normal AND Chest pain IS Asymptomatic AND Fatigue IS Asymptomatic AND Saturation IS Normal AND Leukocytes IS Normal AND Platelets IS Normal THEN Risk IS Low
RULE 3:	IF Age IS Elderly AND Cough IS Asymptomatic AND Shortness of breath IS Asymptomatic AND Temperature IS Normal AND Chest pain is asymptomatic AND Fatigue IS Asymptomatic AND Saturation IS Normal AND Leukocytes IS Normal and Platelets IS Normal THEN Risk IS Moderate
RULE 4:	IF Age is Middle AND Cough IS Moderate AND Shortness of breath IS Mild THEN Risk IS High
RULE 5:	IF Age is Elderly AND Cough IS Moderate AND Shortness of breath IS Mild THEN Risk IS Very high
RULE 5:	IF Age IS Middle AND Temperature IS High AND Chest pain IS Mild THEN Risk IS High
RULE 6:	IF Age IS Elderly AND Temperature IS High AND Chest pain IS Mild THEN Risk IS Very high
RULE 7:	IF Age IS Middle AND Fatigue IS Severe AND Saturation IS Hypoxia THEN Risk IS High
RULE 8:	IF Age IS Elderly AND Fatigue IS Severe AND Saturation IS Hypoxia THEN Risk IS Very high
RULE 9:	IF Age IS Middle AND Leukocytes IS High AND Platelets IS Low THEN Risk IS High
RULE 10:	IF Age IS Elderly AND Leukocytes IS High AND Platelets IS Low THEN Risk IS Very high
RULE 11:	IF Age IS Young AND Cough IS Severe AND Shortness of breath IS Severe THEN Risk IS Very high
	...
RULE 54:	IF Saturation IS Critical THEN Risk IS Very high
RULE 55:	IF Temperature IS Elevated THEN Risk IS Moderate
RULE 56:	IF Chest pain IS Mild THEN Risk IS Moderate
RULE 57:	IF Fatigue IS Mild THEN Risk IS Moderate

2.3. ФАЗЗИФИКАЦИЯ

Фаззификация – процесс преобразования четких входных значений в нечеткие множества. Это важный этап в работе экспертных систем, основанных на нечеткой логике, так как он позволяет системе учитывать неопределенность и варьирование данных [14]. В контексте диагностики пневмонии фаззификация помогает обработать симптомы и диагностические показатели, которые могут иметь размытые границы между различными состояниями.

В процессе фаззификации четкие значения входных переменных преобразуются в степень принадлежности к одному или нескольким нечетким множествам [15]. Для этого используются функции принадлежности, которые определяют, насколько каждое четкое значение принадлежит соответствующему нечеткому множеству.

Рассмотрим пример фаззификации температуры тела. Предположим, что у пациента температура составляет 38 градусов Цельсия. Процесс фаззификации заключается в определении степени принадлежности этого значения к каждому из нечетких множеств: «низкая», «нормальная», «повышенная» и «высокая». Функции принадлежности для переменной “Temperature” задаются в виде уравнений (1)–(4). Графическое представление данных функций изображено на рисунке 2.

$$Temp_{low}(x) = \begin{cases} 1, & x \leq 35, \\ \frac{37-x}{2}, & 35 < x < 37, \\ 0, & x \geq 37. \end{cases} \quad (1)$$

$$Temp_{normal}(x) = \begin{cases} 0, & x \leq 35, \\ \frac{x-35}{2}, & 35 < x < 37, \\ 38-x, & 37 < x < 38, \\ 0, & x \geq 38. \end{cases} \quad (2)$$

$$Temp_{elevated}(x) = \begin{cases} 0, & x \leq 37, \\ x-37, & 37 < x < 38, \\ 39-x, & 38 < x < 39, \\ 0, & x \geq 39. \end{cases} \quad (3)$$

$$Temp_{high}(x) = \begin{cases} 0, & x \leq 38, \\ x-38, & 38 < x < 39, \\ 1, & x \geq 39. \end{cases} \quad (4)$$

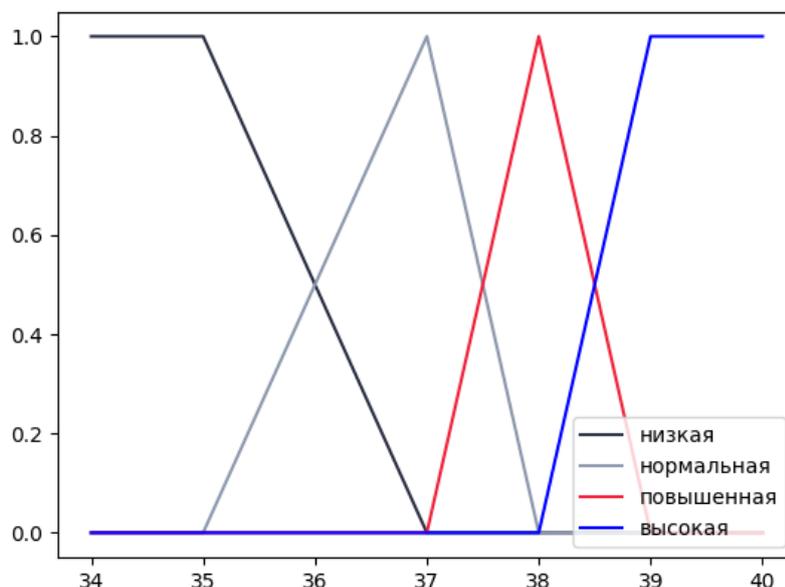


Рис. 2. Графическое представление функций принадлежности переменной “Temperature”

Реализация фаззификации переменной “Temperature” изображена на рисунке 3.

```
class Temperature(Fuzzification):
    def __init__(self):
        super().__init__('Temperature')

    def fuzzify(self, temperature_value):
        low = self.trapezoidal_membership(temperature_value, 0, 0, 35, 37)
        normal = self.triangular_membership(temperature_value, 35, 37, 38)
        elevated = self.triangular_membership(temperature_value, 37, 38, 39)
        high = self.trapezoidal_membership(temperature_value, 38, 39, 41, 42)
        return {'Low': low, 'Normal': normal, 'Elevated': elevated, 'High': high}
```

Рис. 3. Реализация фаззификации

Фаззификация переменной “Age” заключается в определении степени принадлежности значения к каждому из нечетких множеств: «молодой», «средний» и «пожилой». Функции принадлежности для этой переменной задаются в виде уравнений (5)–(7). Графическое представление данных функций представлено на рисунке 3.

$$Age_{young}, Y(x) = \begin{cases} 1, & 0 \leq x \leq 20, \\ \frac{35-x}{15}, & 20 < x < 35, \\ 0, & x \geq 35. \end{cases} \quad (5)$$

$$Age_{middle}, M(x) = \begin{cases} 0, & x \leq 20, \\ \frac{x-20}{15}, & 20 < x < 35, \\ 1, & 35 \leq x \leq 45, \\ \frac{60-x}{15}, & 45 < x < 60, \\ 0, & x \geq 60. \end{cases} \quad (6)$$

$$Age_{elderly}, E(x) = \begin{cases} 0, & x \leq 45, \\ \frac{x-45}{15}, & 45 < x < 60, \\ 1, & x \geq 60. \end{cases} \quad (7)$$

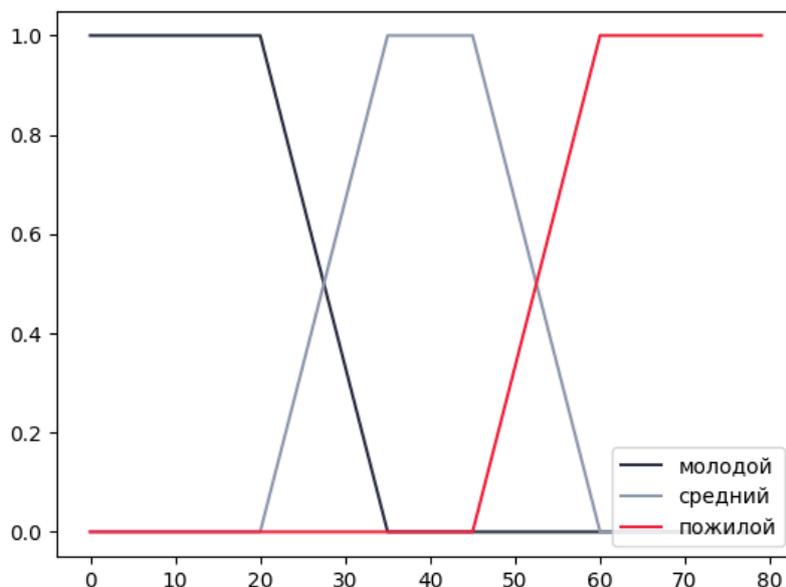


Рис. 4. Графическое представление функций принадлежности переменной “Age”

Фаззификация переменной “Cough” заключается в определении степени принадлежности значения к каждому из нечетких множеств: «отсутствует», «легкий», «умеренный» и «сильный». Функции принадлежности для этой переменной задаются в виде уравнений (8)–(11). Графическое представление данных функций изображено на рисунке 4.

$$Cough_{asympt}(x) = \begin{cases} 1, & x \leq 1, \\ \frac{3-x}{2}, & 1 < x < 3, \\ 0, & x \geq 3. \end{cases} \quad (8)$$

$$Cough_{mild}(x) = \begin{cases} 0, & x \leq 1, \\ \frac{x-1}{2}, & 1 < x < 3, \\ \frac{5-x}{2}, & 3 < x < 5, \\ 0, & x \geq 5. \end{cases} \quad (9)$$

$$Cough_{moderate}(x) = \begin{cases} 0, & x \leq 3, \\ \frac{x-3}{2}, & 3 < x < 5, \\ \frac{7-x}{2}, & 5 < x < 7, \\ 0, & x \geq 7. \end{cases} \quad (10)$$

$$Cough_{severe}(x) = \begin{cases} 0, & x \leq 5, \\ \frac{x-5}{2}, & 5 < x < 7, \\ 1, & x \geq 7. \end{cases} \quad (11)$$

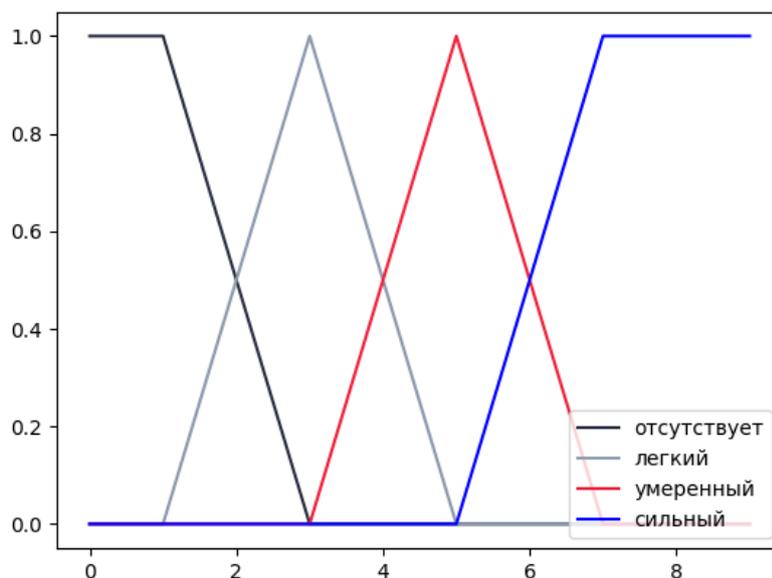


Рис. 5. Графическое представление функций принадлежности переменной "Cough"

Фаззификация переменной “Shortness of breath” заключается в определении степени принадлежности значения к каждому из нечетких множеств: «отсутствует», «легкая», «умеренная» и «сильная». Функции принадлежности для этой переменной задаются в виде уравнений (12)–(15). Графическое представление данных функций изображено на рисунке 5.

$$SoB_{asympt}(x) = \begin{cases} 1, & x \leq 1, \\ \frac{3-x}{2}, & 1 < x < 3, \\ 0, & x \geq 3. \end{cases} \quad (12)$$

$$SoB_{mild}(x) = \begin{cases} 0, & x \leq 1, \\ \frac{x-1}{2}, & 1 < x < 3, \\ \frac{5-x}{2}, & 3 < x < 5, \\ 0, & x \geq 5. \end{cases} \quad (13)$$

$$SoB_{moderate}(x) = \begin{cases} 0, & x \leq 3, \\ \frac{x-3}{2}, & 3 < x < 5, \\ \frac{7-x}{2}, & 5 < x < 7, \\ 0, & x \geq 7. \end{cases} \quad (14)$$

$$SoB_{severe}(x) = \begin{cases} 0, & x \leq 5, \\ \frac{x-5}{2}, & 5 < x < 7, \\ 1, & x \geq 7. \end{cases} \quad (15)$$

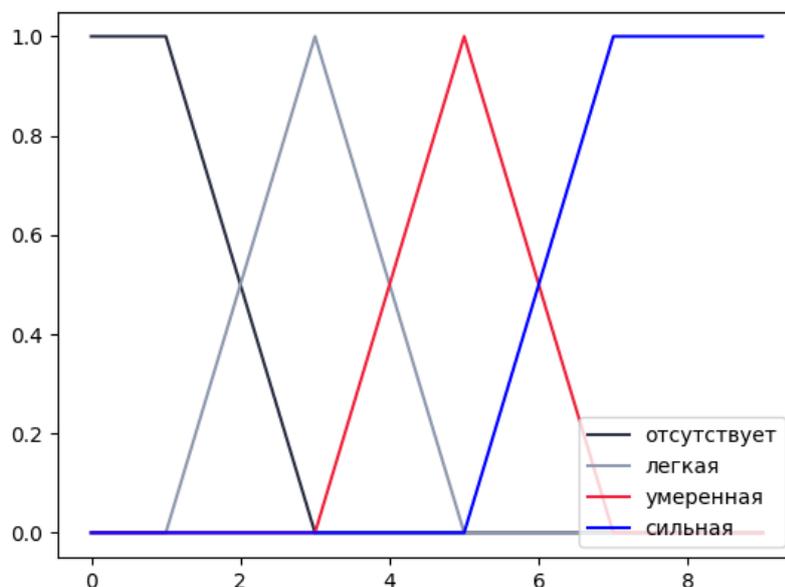


Рис. 6. Графическое представление функций принадлежности переменной «Shortness of breath»

Фаззификация переменной «Chest pain» заключается в определении степени принадлежности значения к каждому из нечетких множеств: «отсутствует», «легкая», «умеренная» и «сильная». Функции принадлежности для этой переменной задаются в виде уравнений (16)–(19). Графическое представление данных функций изображено на рисунке 6.

$$CP_{asympt}(x) = \begin{cases} 1, & x \leq 1, \\ \frac{3-x}{2}, & 1 < x < 3, \\ 0, & x \geq 3. \end{cases} \quad (16)$$

$$CP_{mild}(x) = \begin{cases} 0, & x \leq 1, \\ \frac{x-1}{2}, & 1 < x < 3, \\ \frac{5-x}{2}, & 3 < x < 5, \\ 0, & x \geq 5. \end{cases} \quad (17)$$

$$CP_{moderate}(x) = \begin{cases} 0, & x \leq 3, \\ \frac{x-3}{2}, & 3 < x < 5, \\ \frac{7-x}{2}, & 5 < x < 7, \\ 0, & x \geq 7. \end{cases} \quad (18)$$

$$CP_{severe}(x) = \begin{cases} 0, & x \leq 5, \\ \frac{x-5}{2}, & 5 < x < 7, \\ 1, & x \geq 7. \end{cases} \quad (19)$$

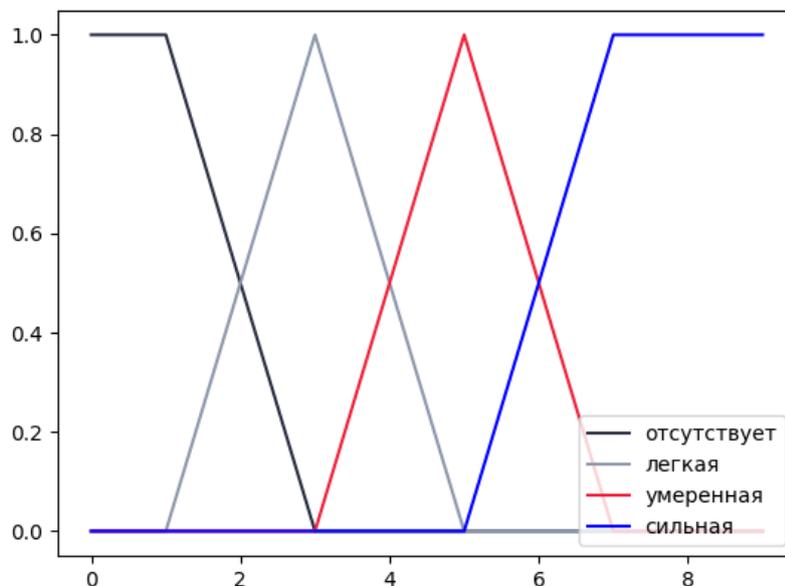


Рис. 7. Графическое представление функций принадлежности переменной “Chest pain”

Фаззификация переменной “Fatigue” заключается в определении степени принадлежности значения к каждому из нечетких множеств: «отсутствует», «легкая» и «выраженная». Функции принадлежности для этой переменной задаются в виде уравнений (20)–(22). Графическое представление данных функций изображено на рисунке 8.

$$Fatigue_{asympt}(x) = \begin{cases} 1, & x \leq 1, \\ \frac{3-x}{2}, & 1 < x < 3, \\ 0, & x \geq 3. \end{cases} \quad (20)$$

$$Fatigue_{mild}(x) = \begin{cases} 0, & x \leq 1, \\ \frac{x-1}{2}, & 1 < x < 3, \\ \frac{5-x}{2}, & 3 < x < 5, \\ 0, & x \geq 5. \end{cases} \quad (21)$$

$$Fatigue_{severe}(x) = \begin{cases} 0, & x \leq 4, \\ x - 4, & 4 < x < 5, \\ 1, & x \geq 5. \end{cases} \quad (22)$$

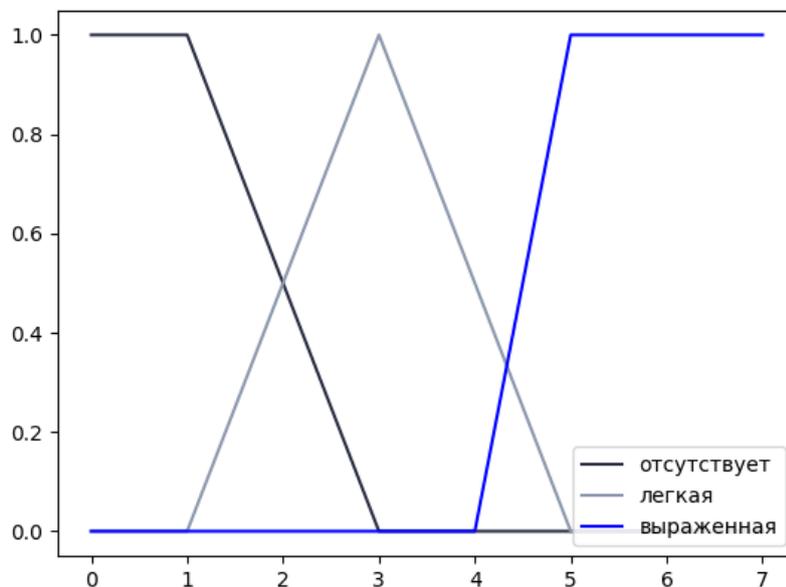


Рис. 8. Графическое представление функций принадлежности переменной "Fatigue"

Фаззификация переменной "Saturation" заключается в определении степени принадлежности значения к каждому из нечетких множеств: «норма», «гипоксия» и «критическое состояние». Функции принадлежности для этой переменной задаются в виде уравнений (23)–(25). Графическое представление данных функций изображено на рисунке 9.

$$Saturation_{normal}(x) = \begin{cases} 0, & x \leq 94, \\ x - 94, & 94 < x < 95, \\ 1, & 95 < x < 100, \\ 0, & x \geq 100. \end{cases} \quad (23)$$

$$Saturation_{hypoxia}(x) = \begin{cases} 0, & x \leq 89, \\ x - 89, & 89 < x < 90, \\ 95 - x, & 90 < x < 94, \\ 0, & x \geq 94. \end{cases} \quad (24)$$

$$Saturation_{critical}(x) = \begin{cases} 1, & x \leq 89, \\ 90 - x, & 89 < x < 90, \\ 0, & x \geq 90. \end{cases} \quad (25)$$

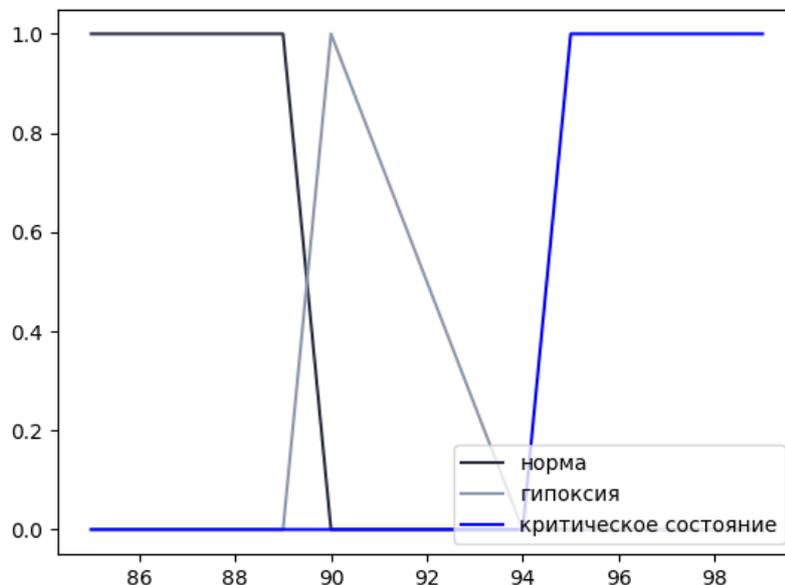


Рис. 9. Графическое представление функций принадлежности переменной “Saturation”

Фаззификация переменной “Leukocytes” заключается в определении степени принадлежности значения к каждому из нечетких множеств: «пониженные», «норма» и «повышенные». Функции принадлежности для этой переменной задаются в виде уравнений (26)–(28). Графическое представление данных функций изображено на рисунке 10.

$$Leukocytes_{low}(x) = \begin{cases} 1, & x \leq 4.5, \\ \frac{5-x}{0.5}, & 4.5 < x < 5, \\ 0, & x \geq 5. \end{cases} \quad (26)$$

$$Leukocytes_{normal}(x) = \begin{cases} 0, & x \leq 4.5, \\ \frac{x-4.5}{0.5}, & 4.5 < x < 5, \\ 1, & 5 \leq x \leq 10.5, \\ \frac{11.3-x}{0.8}, & 10.5 < x < 11.3, \\ 0, & x \geq 11.3. \end{cases} \quad (27)$$

$$\text{Leukocytes}_{high}(x) = \begin{cases} 0, & x \leq 10.5, \\ \frac{x-10.5}{0.8}, & 10.5 < x < 11.3, \\ 0, & x \geq 11.3. \end{cases} \quad (28)$$

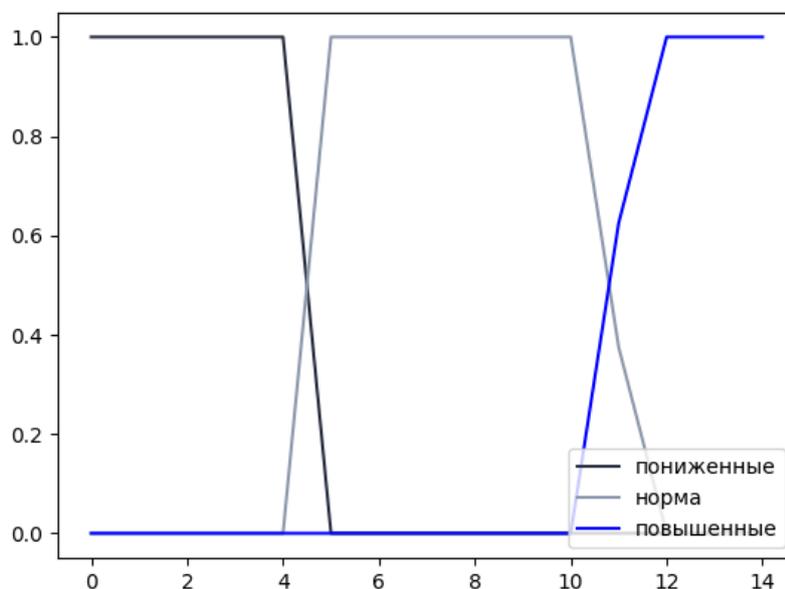


Рис. 10. Графическое представление функций принадлежности переменной "Leukocytes"

Фаззификация переменной "Platelets" заключается в определении степени принадлежности значения к каждому из нечетких множеств: «пониженные», «норма» и «повышенные». Функции принадлежности для этой переменной задаются в виде уравнений (29)–(31). Графическое представление этих функций изображено на рисунке 11.

$$\text{Platelets}_{low}(x) = \begin{cases} 1, & x \leq 180, \\ \frac{200-x}{20}, & 180 < x < 200, \\ 0, & x \geq 200. \end{cases} \quad (29)$$

$$\text{Platelets}_{normal}(x) = \begin{cases} 0, & x \leq 180, \\ \frac{x-180}{20}, & 180 < x < 200, \\ 1, & 200 \leq x \leq 300, \\ \frac{320-x}{20}, & 300 < x < 320, \\ 0, & x \geq 320. \end{cases} \quad (30)$$

$$\text{Platelets}_{high}(x) = \begin{cases} 0, & x \leq 300, \\ \frac{x-300}{20}, & 300 < x < 320, \\ 1, & x \geq 320. \end{cases} \quad (31)$$

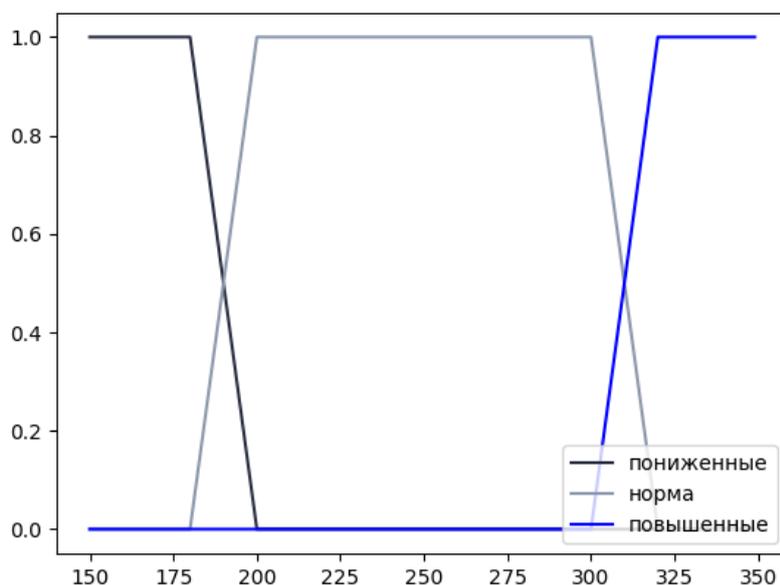


Рис. 11. Графическое представление функций принадлежности переменной “Age”

Фаззификация переменной “Risk” заключается в определении степени принадлежности значения к каждому из нечетких множеств: «низкий», «средний», «высокий» и «очень высокий». Функции принадлежности для этой переменной задаются в виде уравнений (32)–(35). Графическое представление этих функций изображено на рисунке 12.

$$\text{Risk}_{low}(x) = \begin{cases} 1, & x \leq 1, \\ 2 - x, & 1 < x < 2, \\ 0, & x \geq 2. \end{cases} \quad (32)$$

$$\text{Risk}_{moderate}(x) = \begin{cases} 0, & x \leq 1, \\ \frac{x-1}{0.5}, & 1 < x < 1.5, \\ \frac{2-x}{0.5}, & 1.5 < x < 2, \\ 0, & x \geq 2. \end{cases} \quad (33)$$

$$Risk_{high}(x) = \begin{cases} 0, & x \leq 2, \\ \frac{x-2}{0.5}, & 2 < x < 2.5, \\ \frac{3-x}{0.5}, & 2.5 < x < 3, \\ 0, & x \geq 3. \end{cases} \quad (34)$$

$$Risk_{very\ high}(x) = \begin{cases} 0, & x \leq 3, \\ x - 3, & 3 < x < 4, \\ 1, & x \geq 4. \end{cases} \quad (35)$$

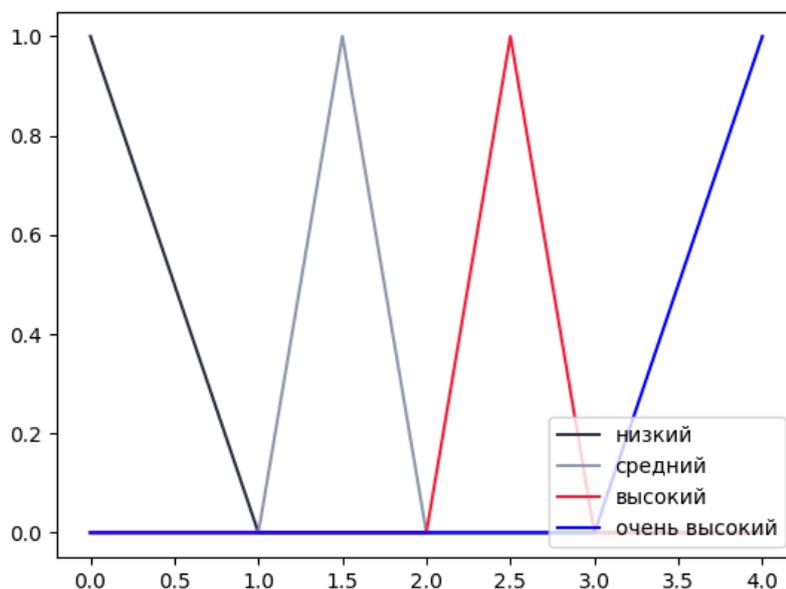


Рис. 12. Графическое представление функций принадлежности переменной “Risk”

2.4. ПРИМЕНЕНИЕ НЕЧЕТКИХ ПРАВИЛ

В рамках нечеткой логики, особенно в алгоритме Мамдани, применение нечетких правил является одним из ключевых этапов. Этот этап включает использование набора заранее определенных нечетких правил для оценки степени выполнения каждого правила на основе фаззифицированных данных [16]. Применение нечетких правил позволяет учитывать множество условий и выводить соответствующие нечеткие результаты, что является основой для принятия решений в экспертных системах.

Для каждого условия правила определяется степень принадлежности входного значения к соответствующему нечеткому множеству. Эта степень принадлежности

указывает, насколько входное значение соответствует данному нечеткому множеству.

Если правило содержит несколько условий, необходимо комбинировать степени выполнения всех условий для определения общей степени выполнения правила [17]:

- Для комбинации “AND” используется минимум значений степеней принадлежности.
- Для комбинации “OR” используется максимум значений степеней принадлежности.

Реализация применения нечетких правил представлена на рисунке 3, где:

- ‘fuzzy_inputs’ – словарь фаззифицированных значений входных переменных.
- ‘rules’ – список правил, где каждое правило состоит из условия и заключения.
- Для каждого правила вычисляется степень выполнения условий, которая определяется как минимальное значение степеней принадлежности условий правила.
- Результаты всех правил возвращаются в виде списка степеней выполнения правил и соответствующих заключений.

```
def apply_fuzzy_rules(fuzzy_inputs, rules):
    rule_strengths = []
    for condition, conclusion in rules:
        strengths = []
        for cond in condition:
            variable, label = cond.split(' IS ')
            strengths.append(fuzzy_inputs[variable][label])
        rule_strengths = min(strengths)
        rule_strengths.append((rule_strengths, conclusion))
    return rule_strengths
```

Рис. 13. Реализация применения нечетких правил

2.5. АГРЕГИРОВАНИЕ

Агрегирование является одним из ключевых этапов алгоритма Мамдани, который выполняется после применения нечетких правил. На этом этапе происходит комбинация (агрегация) всех активированных правил для каждой выходной переменной, чтобы определить общий нечеткий вывод системы [18].

Цель агрегирования состоит в том, чтобы объединить результаты нескольких правил, которые были активированы на предыдущем этапе. Каждое правило имеет свою степень выполнения, которая отражает вклад этого правила в результирующий нечеткий вывод системы [19].

После применения нечетких правил получаем список или словарь, содержащий степень выполнения каждого активированного правила и соответствующее заключение (выходная переменная и ее нечеткое множество). Результаты правил группируются по выходным категориям. Например, все правила, активированные для выходной категории "low", объединяются в одну группу.

Для каждой выходной переменной суммируются степени выполнения всех правил, активированных для этой переменной. Обычно используется операция «максимум» из степеней выполнения для каждого множества принадлежности выходной переменной. Это позволяет учесть различные аспекты, представленные различными правилами [20].

В результате агрегирования получаем агрегированные значения для каждого множества, связанного с выходной переменной. Эти значения представляют собой нечеткий вывод системы после учета всех активированных правил.

Реализация агрегирования приведена на рисунке 14, где:

- Функция принимает список кортежей 'rule_strengths', где каждый кортеж состоит из степени выполнения правила и его вывода.
- Степень выполнения 'strength' – значение от 0 до 1, показывающее, насколько сильно выполнено правило.
- Функция возвращает словарь 'aggregated', где ключи – это метки выходных переменных, а значения – максимальная степень выполнения для каждой категории.

```
def aggregate(rule_strengths):
    aggregated = {}
    for strength, conclusion in rule_strengths:
        if conclusion in aggregated:
            aggregated[conclusion] = max(aggregated[conclusion], strength)
        else:
            aggregated[conclusion] = strength
    return aggregated
```

Рис. 14. Реализация агрегирования

2.6. ДЕФАЗЗИФИКАЦИЯ

Дефаззификация является заключительным этапом в алгоритме Мамдани, где нечеткий вывод преобразуется в четкое значение. На этом этапе агрегированные нечеткие выводы, полученные после применения правил и агрегирования, конвертируются в одно четкое значение, которое можно использовать для принятия решений [21, 22].

Существует несколько методов дефаззификации, но наиболее популярным является метод центра тяжести (центроидный метод).

Метод центра тяжести предполагает вычисление центра тяжести объединенного нечеткого множества. Этот метод дает наиболее сбалансированный результат и широко используется в различных приложениях нечеткой логики.

Основные шаги дефаззификации:

1. Интеграция функции принадлежности: Вычисление взвешенной суммы всех возможных значений.
2. Выбор точки центра тяжести: Деление этой суммы на суммарную функцию принадлежности.

Реализация дефаззификации показана на рисунке 15. Здесь 'numerator' представляет сумму всех значений, а 'denominator' – сумму функций принадлежности. Результат деления – четкое значение, представляющее дефаззифицированный выход.

```
def defuzzify(aggreagated, output_functions, resolution=1000):
    x = np.linspace( start: 0, stop: 1, resolution)
    numerator = 0.0
    denominator = 0.0

    for label, strength in aggreagated.items():
        membership_function = output_functions[label]
        for xi in x:
            mu = min(strength, membership_function.fuzzify(xi))
            numerator += xi * mu
            denominator += mu

    return numerator / denominator if denominator != 0 else 0
```

Рис. 15. Реализация дефаззификации

3. РЕЗУЛЬТАТЫ

Для реализации экспертной системы были выбраны фреймворк Django, предназначенный для высоконадежной и динамичной разработки веб-сайтов, и база данных PostgreSQL.

Выбор фреймворка Django обусловлен тем, что он реализует архитектурный паттерн MVC (Model – View – Controller), и такая архитектура позволяет работать с визуальным представлением и бизнес-логикой приложения отдельно.

Разработанная экспертная система состоит из следующих компонентов:

1. Веб-интерфейс: обеспечивает взаимодействие пользователя с системой.
2. Серверная часть: обрабатывает запросы пользователей, выполняет логические операции и взаимодействует с базой данных. Основные компоненты Django:
 - Модели: определяют структуру данных, хранящихся в базе данных.
 - Представления: обрабатывают запросы пользователей и возвращают ответы.
 - Шаблоны: отображают HTML-страницы, интегрируя данные из представлений.
3. База данных: хранит базу правил и результаты диагностики.

Форма для диагностики пневмонии представляет собой интерфейс, который позволяет медицинским специалистам или пользователям вводить различные клинические данные пациента для оценки вероятности наличия пневмонии (см. рисунок 16). После обработки введенных результатов система выдает результат в виде степени риска наличия пневмонии.

The image shows a web form titled "Pneumonia Diagnosis Form". It contains several input fields, each with a label and a placeholder text: "Age" (placeholder: "Enter age"), "Cough (0-10)" (placeholder: "Enter cough severity"), "Shortness Of Breath (0-10)" (placeholder: "Enter shortness of breath severity"), "Temperature (°C)" (placeholder: "Enter temperature"), "Chest Pain (0-10)" (placeholder: "Enter chest pain severity"), "Fatigue (0-10)" (placeholder: "Enter fatigue level"), "Oxygen Level (%)" (placeholder: "Enter oxygen level"), "Leukocyte Count (x10^9/L)" (placeholder: "Enter white blood cell count"), and "Platelet Count (x10^9/L)" (placeholder: "Enter platelet count"). At the bottom of the form is a blue "Submit" button.

Рис. 16. Форма диагностики

Форма для добавления правил экспертной системы позволяет эксперту добавлять, редактировать и удалять правила, используемые для диагностики пневмонии (см. рисунок 17). Правила состоят из условий (IF) и заключений (THEN), которые описываются экспертом.

Expert Rules Form

IF Condition
Enter IF condition (e.g., Age IS Middle AND Cough IS Severe)

THEN Conclusion
Enter THEN conclusion (e.g., Risk IS High)

Add Rule

Existing Rules

#	Condition	Conclusion	Actions
1	IF Age IS Middle AND Temperature IS High	THEN Risk IS High	Edit Delete

Рис. 17. Форма для ввода правил

ЗАКЛЮЧЕНИЕ

Представлена разработанная экспертная система для диагностики пневмонии. Описан и реализован алгоритм Мамдани, который включал в себя формирование базы правил, фаззификацию входных данных, применение нечетких правил, агрегирование результатов и дефаззификацию. Экспертные системы на основе нечеткой логики представляют собой инструмент для обработки и анализа нечеткой информации в медицинской практике. Разработанная система демонстрирует принципы и применения нечеткой логики в диагностике пневмонии, предоставляя врачам ценные инструменты для принятия обоснованных и информированных решений, позволяющие автоматизировать процесс диагностики.

СПИСОК ЛИТЕРАТУРЫ

1. Григорьева Д.Р., Гареева Г.А., Басыров Р.Р. Основы нечеткой логики: Учебно-методическое пособие к практическим занятиям и лабораторным работам. Набережные Челны: Изд-во НЧИ КФУ. 2018. 42 с.
2. Fuzzy logic.
URL: https://en.wikipedia.org/wiki/Fuzzy_logic, last accessed 27/05/2024.
3. Zadeh L.A. The role of fuzzy logic in the management of uncertainty in expert systems // Fuzzy sets and systems, 1983. 11(1-3), P. 199-227.
4. Negnevitsky M. Artificial Intelligence: A guide to Intelligent Systems (2nd ed.). Addison-Wesley, 2005. 435 p.

5. *Demetriou D., See L.M., Stillwell J.* Expert Systems for Planning and Spatial Decision Support // *GeoComputation*, Chapter 11, 2014. P. 257–274. <https://doi.org/10.1201/b17091-12>.
6. *Zadeh L.A.* Fuzzy sets // In *Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A Zadeh*, 1996. P. 394–432.
7. *Pedrycz W., Gomide F.* *Fuzzy Systems Engineering: Toward Human-Centric Computing* // John Wiley & Sons, 2007. 549 p.
8. *Umair Ahmed, Ghulam Rasool, Saqib Zafar, Farhan Maqbool* Fuzzy Rule Based Diagnostic System to Detect the Lung Cancer // *International conference on Computing, Electronic and Electrical Engineering (ICE Cube)*, 2018. <https://doi.org/10.1109/ICECUBE.2018.8610976>.
9. Tools To Generate Fuzzy Takagi Sugeno Method by Python. <https://medium.com/@zekkelar1337/tools-to-generate-fuzzy-takagi-sugeno-method-by-python-748f978e3673>, last accessed 27/05/2024.
10. Comparative analysis of Mamdani, Larsen and Tsukamoto methods of fuzzy inference system for students' academic performance evaluation. URL: <https://ijsra.net/sites/default/files/IJSRA-2023-0443.pdf>, last accessed 25/05/2024.
11. Pneumonia Symptoms and Diagnosis. URL: <https://www.lung.org/lung-health-diseases/lung-disease-lookup/pneumonia/symptoms-and-diagnosis>, last accessed 27/05/2024.
12. Pneumonia Symptoms. URL: <https://www.mayoclinic.org/diseases-conditions/pneumonia/symptoms-causes/syc-20354204>, last accessed 27/05/2024.
13. *Arani L.A., Sadoughi F., Langarizadeh M.* An Expert System to Diagnose Pneumonia Using Fuzzy Logic // *Acta Inform Med*, 2019. P. 103–107. <https://doi.org/10.5455/aim.2019.27.103-101>
14. *Fazel Zarandi M.H., Zolnoori M., Moin M., Taherian M.* Fuzzy Rule-Based Expert System for Evaluating Level of Asthma Control // *J Med Syst*. 2012. V. 36, no. 5. P. 2947–2958. <https://doi.org/10.1007/s10916-011-9773-3>.
15. *Azaab S., Abu Naser S., Sulisel O.* A proposed expert system for selecting exploratory factor analysis procedures // *Journal of the college of education*. 2000 . V. 4, no. 2. P. 9–26.
16. *Hasan M.A., Sher-e-Alam K., Chowdhury A.R.* Human Disease Diagnosis Using

a Fuzzy Expert System // Journal of Computing. 2010. V. 2, no. 6.. P. 66–70.
<https://doi.org/10.48550/arXiv.1006.4544>

17. *Febria Anjara, Adam Amril Jaharadak.* Expert system for disease diagnosis in living things: A narrative review // Journal of Physics: Conference series. 2019. V. 1167.
<https://doi.org/10.1088/1742-6596/1167/1/012070>.

18. *Pavate A., Nerurkar P., Ansari N., Bansode R.* Early Prediction of Five Major Complications Ascends in Diabetes Mellitus Using Fuzzy Logic // Soft Computing in Data Analytics, Advances in Intelligent Systems and Computing. 2019. P. 759–768.
https://doi.org/10.1007/978-981-13-0514-6_72

19. *Russell S., Norvig P.* Artificial Intelligence: A Modern Approach (3rd ed.). Pearson Education, 2010. 1152 p.

20. *Buchanan B.G., Shortliffe E.H.* Rule-based expert systems: The MYCIN Experiments of the Stanford Heuristic Programming Project. Addison-Wesley, 1984. 503 p.

21. *Giarratano J.C., Riley G.* Expert Systems: Principles and Programming (4th ed.) Course Technology, 2005. 868 p.

22. *Waterman D.A.* A Guide to Expert Systems. Addison-Wesley, 1986. 419 p.

23. *Burnashev R.A., Khairullin B.M., Prokopyev N.A., Farahov R.A., Bol-sunovskaya M.V., Enikeev A.I.* Design and Development of a Research Integrated Geoinformation System with a Fuzzy Expert System // 2023 IEEE 24th International Conference of Young Professionals in Electron Devices and Materials (EDM), Novosibirsk, Russian Federation, 2023. P. 1310–1313. <https://doi.org/10.1109/EDM58354.2023.10225118>.

24. *Burnashev R.* Designing a Prototype of an Adaptive Expert System Using Fuzzy Logic and a Genetic Algorithm // 2023 International Russian Automation Conference (RusAutoCon), Sochi, Russian Federation, 2023. P. 1168–1172.
<https://doi.org/10.1109/RusAutoCon58002.2023.10272949>.

DEVELOPMENT OF A FUZZY LOGIC-BASED EXPERT SYSTEM FOR PNEUMONIA DIAGNOSTICS

A. I. Enikeeva¹ [0009-0002-9192-952x], R. A. Burnashev² [0000-0002-1057-0328],

R. R. Farahov³ [0000-0002-7993-3182]

¹⁻³Kazan (Volga Region) Federal University

¹adisen24@yandex.ru, ²r.burnashev@inbox.ru, ³rus-wing-dark@mail.ru

Abstract

The article is devoted to the development of an expert system for diagnosing pneumonia. This system is based on fuzzy logic and implemented using the Mamdani algorithm. The main stages of system development are considered, including the fuzzification of input data, the definition of fuzzy rules based on medical expert knowledge, the aggregation of fuzzy inferences, and their defuzzification to obtain the final diagnostic result. The system's web interface is implemented using the Django framework, ensuring ease of interaction for users. The use of a medical expert system for pneumonia diagnostics can reduce the time required to establish a diagnosis and improve the quality of diagnosis by integrating the experience of medical experts and modern information technologies.

Keywords: *expert system, fuzzy logic, medical diagnostics, Mamdani algorithm, decision support system.*

REFERENCES

1. Grigoreva D.R., Gareeva G.A., Basyrov R.R. *Osnovy nechetkoj logiki: Uchebno-metodicheskoe posobie k prakticheskim zanyatiyam I laboratornym rabotam.* Naberezhnye Chelny: Izd-vo NChI KFU, 2018. 42 s.
2. Fuzzy logic.
URL: https://en.wikipedia.org/wiki/Fuzzy_logic, last accessed 27/05/2024.
3. Zadeh L.A. The role of fuzzy logic in the management of uncertainty in expert systems // *Fuzzy sets and systems*, 1983. 11(1-3), P. 199-227.
4. Negnevitsky M. *Artificial Intelligence: A guide to Intelligent Systems* (2nd ed.). Addison-Wesley, 2005. 435 p.
5. Demetriou D., See L.M., Stillwell J. *Expert Systems for Planning and Spatial Decision Support* // *GeoComputation*, Chapter 11, 2014. P. 257–274.

<https://doi.org/10.1201/b17091-12>.

6. Zadeh L.A. Fuzzy sets // In Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A Zadeh, 1996. P. 394–432.
7. Pedrycz W., Gomide F. Fuzzy Systems Engineering: Toward Human-Centric Computing // John Wiley & Sons, 2007. 549 p.
8. Umair Ahmed, Ghulam Rasool, Saqib Zafar, Farhan Maqbool Fuzzy Rule Based Diagnostic System to Detect the Lung Cancer // International conference on Computing, Electronic and Electrical Engineering (ICE Cube), 2018. <https://doi.org/10.1109/ICECUBE.2018.8610976>.
9. Tools To Generate Fuzzy Takagi Sugeno Method by Python. URL: <https://medium.com/@zekkelar1337/tools-to-generate-fuzzy-takagi-sugeno-method-by-python-748f978e3673>, last accessed 27/05/2024.
10. Comparative analysis of Mamdani, Larsen and Tsukamoto methods of fuzzy inference system for students' academic performance evaluation. <https://ijsra.net/sites/default/files/IJSRA-2023-0443.pdf>, last accessed 25/05/2024.
11. Pneumonia Symptoms and Diagnosis. URL: <https://www.lung.org/lung-health-diseases/lung-disease-lookup/pneumonia/symptoms-and-diagnosis>, last accessed 27/05/2024.
12. Pneumonia Symptoms. URL: <https://www.mayoclinic.org/diseases-conditions/pneumonia/symptoms-causes/syc-20354204>, last accessed 27/05/2024.
13. Leila Akramian Arani, Frahnaz Sadoughi, Mustafa Langarizadeh. An Expert System to Diagnose Pneumonia Using Fuzzy Logic // Acta Inform Med, 2019. P. 103–107. <https://doi.org/10.5455/aim.2019.27.103-101>
14. Fazel Zarandi M.H., Zolnoori M., Moin M., Taherian M. Fuzzy Rule-Based Expert System for Evaluating Level of Asthma Control // J Med Syst. 2012. V. 36, no. 5. P. 2947–2958. <https://doi.org/10.1007/s10916-011-9773-3>.
15. Azaab S., Abu Naser S., Sulisel O. A proposed expert system for selecting exploratory factor analysis procedures // Journal of the college of education. 2000. V. 4, no. 2. P. 9–26.
16. Hasan M.A., Sher-e-Alam K., Chowdhury A.R. Human Disease Diagnosis Using a Fuzzy Expert System // Journal of Computing. 2010. V. 2, no. 6.. P. 66–70.

<https://doi.org/10.48550/arXiv.1006.4544>

17. *Febria Anjara, Adam Amril Jaharadak.* Expert system for disease diagnosis in living things: A narrative review // *Journal of Physics: Conference series.* 2019. V. 1167. <https://doi.org/10.1088/1742-6596/1167/1/012070>.

18. *Pavate A., Nerurkar P., Ansari N., Bansode R.* Early Prediction of Five Major Complications Ascends in Diabetes Mellitus Using Fuzzy Logic // *Soft Computing in Data Analytics, Advances in Intelligent Systems and Computing.* 2019. P. 759–768. https://doi.org/10.1007/978-981-13-0514-6_72

19. *Russell S., Norvig P.* *Artificial Intelligence: A Modern Approach* (3rd ed.). Pearson Education, 2010. 1152 p.

20. *Buchanan B.G., Shortliffe E.H.* *Rule-based expert systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.* Addison-Wesley, 1984. 503 p.

21. *Giarratano J.C., Riley G.* *Expert Systems: Principles and Programming* (4th ed.) Course Technology, 2005. 868 p.

22. *Waterman D.A.* *A Guide to Expert Systems.* Addison-Wesley, 1986. 419 p.

23. *Burnashev R.A., Khairullin B.M., Prokopyev N.A., Farahov R.A., Bol-sunovskaya M.V., Enikeev A.I.* Design and Development of a Research Integrated Geoinformation System with a Fuzzy Expert System // *2023 IEEE 24th International Conference of Young Professionals in Electron Devices and Materials (EDM), Novosibirsk, Russian Federation, 2023.* P. 1310–1313.

<https://doi.org/10.1109/EDM58354.2023.10225118>.

24. *Burnashev R.* Designing a Prototype of an Adaptive Expert System Using Fuzzy Logic and a Genetic Algorithm // *2023 International Russian Automation Conference (RusAutoCon), Sochi, Russian Federation, 2023.* P. 1168–1172.

<https://doi.org/10.1109/RusAutoCon58002.2023.10272949>.

СВЕДЕНИЯ ОБ АВТОРАХ



ЕНИКЕЕВА Аделя Искандеровна – аспирант Института информационных технологий и интеллектуальных систем Казанского (Приволжского) федерального университета, ассистент кафедры анализа данных и технологий программирования Института вычислительной математики и информационных технологий Казанского (Приволжского) федерального университета.

Adelya ENIKEEVA – Post-graduate Student, Institute of Information Technologies and Intelligent Systems, Assistant at the Department of Data Analysis and Programming technologies, Institute of Computational Mathematics and Information Technologies, Kazan (Volga Region) Federal University. Current scientific interests: human-computer interaction, software engineering, data management systems.

email: adisen24@yandex.ru;

ORCID: 0009-0002-9192-952x



БУРНАШЕВ Рустам Арифович – доцент кафедры анализа данных и технологий программирования Института вычислительной математики и информационных технологий Казанского (Приволжского) федерального университета.

Rustam BURNASHEV – Associate Professor, Institute of Computational Mathematics and Information Technologies, Kazan (Volga Region) Federal University.

Current scientific interests: expert systems, data analysis, fuzzy logic, machine learning.

email: r.burnashev@inbox.ru;

ORCID: 0000-0002-1057-0328



ФАРАХОВ Рустам Ринатович – ассистент кафедры анализа данных и технологий программирования Института вычислительной математики и информационных технологий Казанского (Приволжского) федерального университета.

Rustam FARANOV – Assistant at the Department of Data Analysis and Programming Technologies, Institute of Computational Mathematics and Information Technologies, Kazan (Volga Region) Federal University.

Current scientific interests: expert systems, internet of things.

email: rus-wing-dark@mail.ru;

ORCID: 0000-0002-7993-3182

Материал поступил в редакцию 15 мая 2024 года

АВТОМАТИЗАЦИЯ СОРТИРОВКИ МАТЕРИАЛОВ ПО ТЕКСТУ СЦЕНАРИЯ ДЛЯ ВИДЕОМОНТАЖА

А. Д. Неманов¹ [0009-0004-1840-2347], И. С. Шахова² [0000-0003-1591-5767]

^{1,2} *Институт информационных технологий и интеллектуальных систем,
Казанский федеральный университет, г. Казань*

¹andrewoch@yandex.ru, ²is@it.kfu.ru

Аннотация

Процесс видеомонтажа включает множество трудоемких операций по сортировке и подготовке материалов, что требует значительных временных затрат. В статье описана разработка программного решения для автоматизации этих процессов с использованием технологии машинного обучения. Основное внимание уделено созданию системы, способной классифицировать и сортировать медиафайлы по тексту сценария, тем самым повышая эффективность подготовки материалов к монтажу. Система включает модули распознавания речи, классификации аудио и видео, а также алгоритмы определения соответствия сценарию. Тестирование показало, что предложенная система правильно классифицирует медиафайлы в большинстве случаев, что позволяет существенно сократить время на черновой монтаж.

Ключевые слова: *видеомонтаж, автоматизация, машинное обучение, распознавание речи, классификация аудио, классификация видео, coreml, параллельные вычисления, сценарий, soundex, tf-idf, косинусное сходство, обработка естественного языка*

ВВЕДЕНИЕ

Монтаж является неотъемлемой частью производства видео и состоит из таких процессов, как выбор и соединение подходящих видеоматериалов, коррекция цвета, добавление специальных эффектов, аудиодорожек и музыки.

В процессе съемок у авторов накапливается значительное количество материала. При этом, как правило, для улучшения качества звука голоса людей запи-

сываются на отдельные микрофоны, из-за чего в результате монтажеру предоставляются как видеофайлы, так и звуковые файлы. Среди них содержатся неудачные дубли, тестовые записи и записи с плохим качеством звука.

На раннем этапе обработки материалов видеомонтажеру необходимо провести так называемый «черновой» монтаж, то есть собрать в монтажной программе примерный макет видеопроекта и проверить, что нужно переснять, каких кадров не хватает или какие кадры не «клеятся» на монтаже [1]. Этот процесс подразумевает следующие действия: отсечь бракованный материал, отобрать все удачные дубли, найти звуковые файлы, соответствующие видеоряду, перепроверить материал на наличие технического брака и сделать заключение о готовности приступить к монтажу. Для этого необходимо вручную прослушивать и сортировать файлы, что сопровождается значительными трудозатратами.

Хотя создатели кино редко показывают статистику по итогам своих съемочных процессов, известно, что обычный съемочный день длится от 8 до 12 часов. При этом длина отснятого удачного материала, который потом попадает в итоговый вариант, составляет 2–3 минуты в кино, 6–7 минут в простых разговорных сериалах. Количество дублей на каждый кадр может различаться от 2–3 до нескольких десятков [2, 3]. При этом нужно учитывать, что запись ведется на несколько устройств и что специалисту необходимо пересмотреть материалы в среднем по 2 раза, чтобы убедиться в качестве отобранных материалов. Так, для минутного ролика черновой монтаж может длиться до 20 минут, а для полнометражного фильма может потребоваться 240 часов.

Чтобы упростить себе задачу, съемочная команда переименовывает файлы по заранее оговоренному шаблону, отмечая удачные дубли, иногда даже находясь непосредственно на месте съемок. Однако эти действия занимают больше времени, что создает риск отклониться от графика [4].

Решением обозначенной проблемы является инструмент, который позволит автоматизировать сортировку видео и аудио, а затем, следуя предоставленному сценарию, определить самые удачные дубли с точки зрения соответствия сценарию.

Для проведения обзора альтернативных решений были выделены следующие критерии: возможность сортировки по сценарию по тексту речи, звуку и описанию картинки, проверка соответствия сценарию, автоматизация сортировки.

Одним из альтернативных программных решений является программа для монтажа видео Adobe Premiere Pro¹. Она не имеет функционала автоматической сортировки файлов, но предоставляет возможность вручную организовывать иерархию файлов в проекте.

Еще одним похожим решением является программа Davinci Resolve². Она также не имеет возможности автоматизированной сортировки по сценарию, однако предоставляет модуль распознавания речи в аудиофайлах, что позволяет пользователю искать нужные файлы по тексту речи.

Самым близким к необходимому решению является AVID Media Composer. Это программа, которая специализируется на подготовке видеофрагментов к монтажу, сортировке, а также черновому монтажу [5]. В ней имеется функционал добавления сценария, расшифровки речи и поиска по тексту речи. Также файлы можно автоматически отсортировать по сценарию. Однако это работает только для фраз, по сценам экспозиции файлы сортировать нельзя.

СТАНДАРТ ОФОРМЛЕНИЯ СЦЕНАРИЯ

В киноиндустрии приняты определенные стандарты, по которым оформляется сценарий. Существует несколько стандартов с разными вариантами оформления, но их объединяет одно: сценарии состоят из логических блоков, позволяющих упростить их понимание и контролировать процесс съемок, отмечая отснятые части [6].

Среди таких блоков можно выделить два:

- 1) диалоги;
- 2) описание действия (далее сцены-экспозиции).

Блоки диалогов состоят из последовательности фраз. Фразы, в свою очередь, состоят из имени персонажа и текста речи этого персонажа. Так как в некоторых случаях съемка диалога происходит в несколько дублей или даже по одной фразе на дубль, было принято решение впоследствии работать именно с отдельной фразой.

Сцены-экспозиции состоят из описания какого-либо действия без речи персонажей. Предполагается, что текстовый файл сценария соответствует стандарту

¹ <https://www.adobe.com/products/premiere.html>

² <https://www.blackmagicdesign.com/products/davinciresolve>

киноиндустрии, чтобы имелась возможность программным путём разбить сценарий на логические блоки (рис. 1).

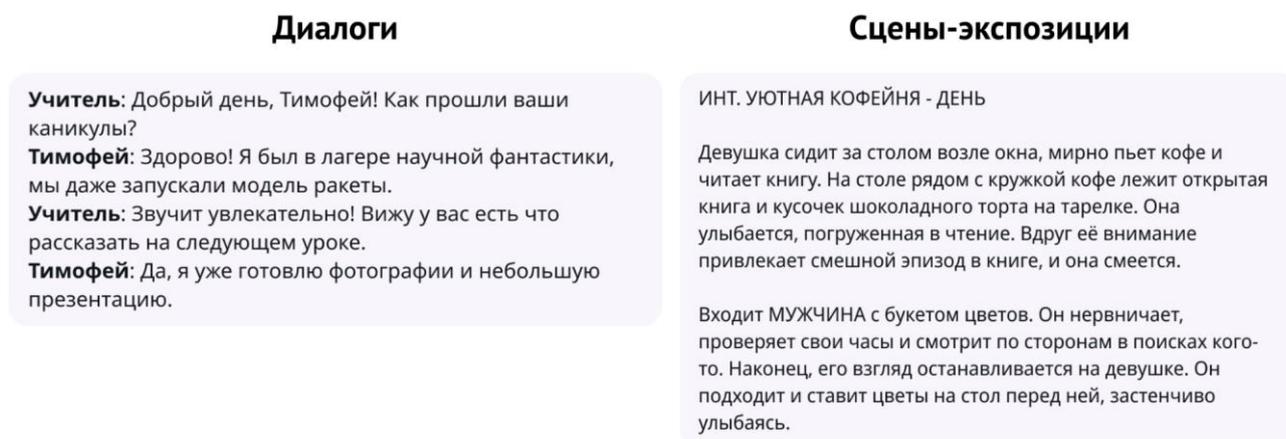


Рис. 1. Пример логических блоков

ПРОГРАММНЫЕ МЕХАНИЗМЫ СОРТИРОВКИ ДИАЛОГОВ ПО СЦЕНАРИЮ

Определить фразу, произнесенную персонажем, можно, сравнивая текст расшифровки речи в файле и текст фразы в сценарии. Однако в действительности существует ряд проблем, для решения которых требуется усложнить этот механизм:

- 1) Расшифровка речи сопровождается погрешностью в распознавании – могут распознаваться другие слова, близкие по звучанию.
- 2) В файле помимо самой фразы могут записываться и служебные разговоры. У профессионалов это может быть фраза «Камера, мотор!», а у начинающих могут быть записаны случайные диалоги за кадром, которые по длине не отличаются от самой фразы текста. Алгоритм определения фразы должен обладать возможностью игнорировать эти части при сравнении.
- 3) В одном файле может быть записано сразу несколько дублей одной сцены.

Первую проблему можно решить, преобразуя слова в расшифровке речи и тексте сценария в фонетические коды и сравнивая их. Это позволит опираться на произношение слов, а не на то, как они пишутся.

Вторая и третья проблемы решаются сегментацией аудиофайла во время расшифровки речи и определением сегментов, подходящих или неподходящих по сценарию. Это также может помочь при черновом монтаже, ведь если автоматически отметить часть файла, подходящую по сценарию, то можно обращать внимание только на эту часть, не затрачивая время на остальные.

Таким образом, порядок работы механизма сортировки диалога по сценарию выглядит следующим образом: у медиафайла производится расшифровка речи с временными отметками для каждого сегмента. Затем у этой расшифровки все слова преобразуются в фонетические коды. Такое же преобразование происходит и у текста каждой фразы в сценарии. После этого алгоритм сравнивает фонетические коды файла с кодами каждой фразы, определяя наиболее подходящую фразу и указывая, какие сегменты в файле подходят к этой фразе, а какие нет (рис. 2).

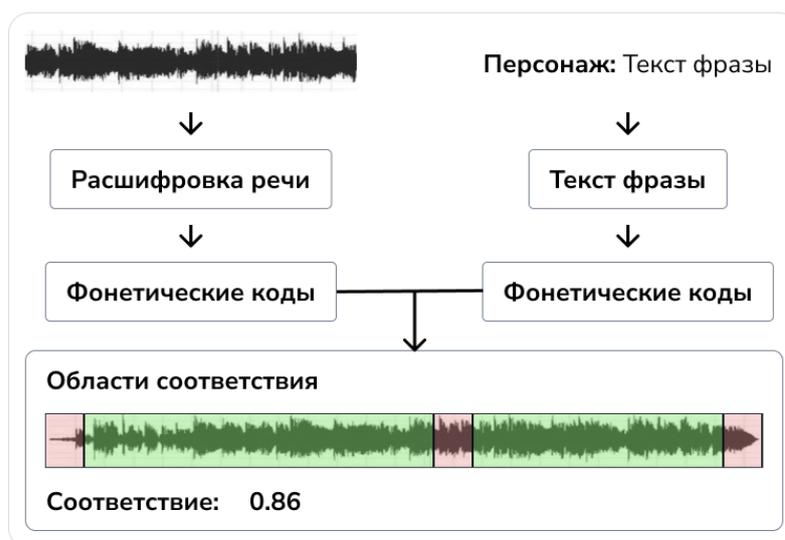


Рис. 2. Схема работы механизма сортировки диалогов по сценарию

При реализации расшифровки речи необходимо было балансировать между точностью и скоростью, так как цель предложенного инструмента — сэкономить специалисту время, которое он тратит на сортировку видео- и аудиофайлов.

Для решения данной задачи была выбрана модель расшифровки речи Whisper от OpenAI [7, 8]. За счёт архитектуры, учитывающей контекст предложения, выбранная модель показывает точность, достаточную для решения поставленной задачи даже в случае сниженного качества обрабатываемых аудиозаписей. Кроме того, она поддерживает несколько языков, что обеспечивает перспективы адаптации программного решения под эти языки. Была выбрана модель small, третья по величине.

При выборе подходящей библиотеки, использующей Whisper для расшифровки и имеющей поддержку сегментации, были рассмотрены два варианта: SwiftWhisper [9] и WhisperKit [10]. Главным преимуществом WhisperKit является то, что эта библиотека позволяет сегментировать аудиофайл по одному слову в сегменте. Благодаря этому можно избежать ситуаций, когда в один сегмент могли попасть как часть речи по сценарию, так и часть посторонних разговоров. При этом временные границы сегментов с использованием WhisperKit определяются без погрешностей точнее по сравнению со SwiftWhisper (рис. 3).

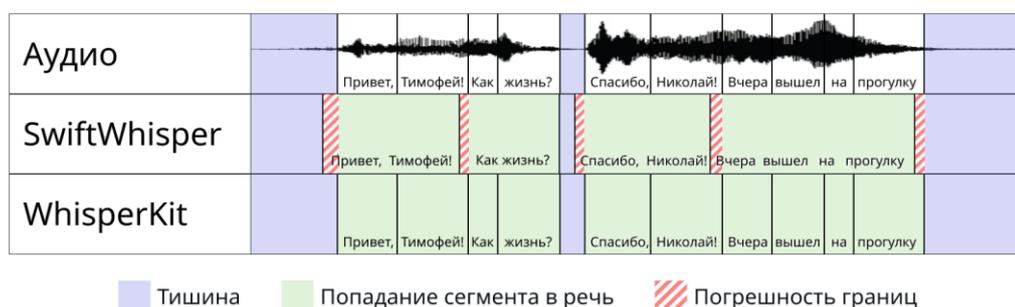


Рис. 3. Сравнение сегментации библиотек расшифровки речи

За счет оптимизации под ARM-процессоры распознавание речи с WhisperKit работает быстрее, чем реализация SwiftWhisper на C++, и, кроме того, имеется возможность распознавать несколько файлов одновременно.

Ещё одним преимуществом WhisperKit является то, что можно распознавать несколько файлов одновременно. Был разработан модуль расшифровки, который имеет собственную очередь задач DispatchQueue [11], в которые можно добавлять файлы на расшифровку. Очередь контролируется с помощью DispatchSemaphore, который пропускает на расшифровку файлы по количеству

ядер процессора устройства. Таким образом получается избежать переполнения оперативной памяти устройства, даже если в очереди находится несколько сотен файлов, ведь одновременно распознаваться будет только определенное их количество. К тому же этот модуль не блокирует работу остального приложения и позволяет добавлять файлы в очередь, даже если в ней уже находятся файлы. При добавлении в очередь файлы сортируются по длительности, чтобы более короткие распознавались первыми и быстрее покидали очередь, освободив память. Для обработки ошибок и лучшего контроля большого количества асинхронных процессов был использован Combine [12].

Перед сравнением было решено конвертировать слова в фонетические коды Soundex [13]. Был реализован модуль, который преобразует строку в массив фонетических кодов с использованием инструмента RussianSoundex из библиотеки Fonetika [14]. Для оптимизации уже преобразованные слова добавлялись в словарь, чтобы повторно использовать полученный код. Также при загрузке сценария все его блоки диалогов сразу обрабатываются, и заготовленные коды для каждой сцены сохраняются для дальнейшей работы.

Был разработан алгоритм классификации фразы, который работает следующим образом: при сравнении с каждой фразой в массиве сегментов расшифровки сразу подсчитывается, какие сегменты подряд подходят по сценарию, а какие нет. Таким образом формируется чередующаяся последовательность из наборов подходящих и неподходящих сегментов. В процессе составляется список из начальных и конечных индексов каждого набора, а также длины этого набора.

После составления списков наборов сегментов для каждой фразы выбирается список с самым длинным подходящим набором, и фраза, к которой был создан этот набор, выбирается как подходящая для файла. После этого по полученному списку индексов происходит склеивание исходных сегментов, и новый набор сегментов заменяет исходный.

С таким подходом удалось реализовать алгоритм, который для обработки одного файла имеет временную сложность $O(N \times P \times M)$, где:

- N — количество слов в расшифровке файла,
- P — количество фраз в сценарии,
- M — средняя длина фразы в сценарии.

Было проведено тестирование алгоритма, в результате которого выявилась следующая проблема: если в середине расшифровки речь распознается плохо и даже фонетические коды у слов отличаются, то в результате получается два подходящих сегмента, разделенные неподходящим сегментом из проблемных слов. Однако, поскольку общее сходство по сценарию считается как сходство самого длинного подходящего сегмента, из-за разделенных сегментов алгоритм выдаёт значение, которое отличается от фактического минимум в половину, что может привести к неправильной классификации.

Поэтому алгоритм был дополнен механизмом подсчёта сквозной длины совпадения с фразой: при нахождении подходящего сегмента определяется, на каком слове фразы этот сегмент закончился. Затем при нахождении следующего подходящего сегмента определяется слово фразы, с которого новый сегмент начался. Вычисляется разрыв между конечным словом фразы из прошлого сегмента и начальным словом фразы из нового сегмента. Если разница между этим разрывом и длиной неподходящего сегмента между подходящими не превышает два слова, то считается, что новый подходящий сегмент продолжает фразу из первого сегмента, пропустив столько слов фразы, сколько слов в неподходящем сегменте (рис. 4).

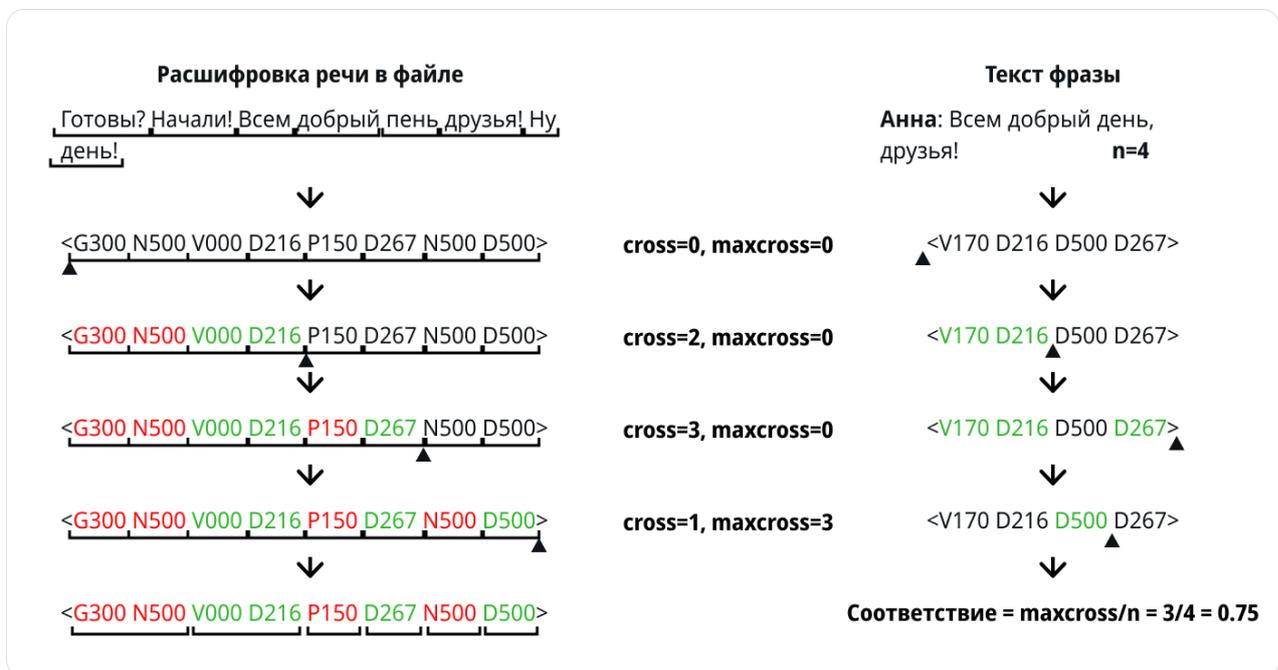


Рис. 4. Определение соответствия фразе с подсчетом сквозной длины

Общее соответствие файла сценарию считается как сумма длин сегментов, продолжающих друг друга. Теперь алгоритм при выборе подходящей фразы опирается именно на максимальную сквозную длину.

СТАНДАРТ ОФОРМЛЕНИЯ СЦЕНЫ ЭКСПОЗИЦИИ

Для сортировки по сценам-экспозициям было необходимо классифицировать изображение и звук в видеозаписи. Чтобы определить предметную область, которую необходимо покрыть классификацией, были изучены стандарты оформления сценариев. В голливудском формате сцена экспозиции выглядит следующим образом: сверху пишутся ИНТ (интерьер) либо НАТ (натура), затем название локации и время суток. После чего пишется текст сцены (рис. 5) [15].

ИНТ. УЮТНАЯ КОФЕЙНЯ - ДЕНЬ

Девушка сидит за столом возле окна, мирно пьет кофе и читает книгу. На столе рядом с кружкой кофе лежит открытая книга и кусочек шоколадного торта на тарелке. Она улыбается, погруженная в чтение. Вдруг её внимание привлекает смешной эпизод в книге, и она смеется.

Входит МУЖЧИНА с букетом цветов. Он нервничает, проверяет свои часы и смотрит по сторонам в поисках кого-то. Наконец, его взгляд останавливается на девушке. Он подходит и ставит цветы на стол перед ней, застенчиво улыбаясь.

Рис. 5. Пример сцены экспозиции

Опираясь на этот стандарт, необходимо было классифицировать медиа-файлы и тексты блоков экспозиции, а затем найти способ сравнить определенные классы и на основе этого подобрать самую подходящую сцену для файла.

ПРОГРАММНЫЕ МЕХАНИЗМЫ КЛАССИФИКАЦИИ АУДИОФАЙЛОВ

Классификация аудио необходима, чтобы определять объекты на видео, которые не присутствуют в кадре, а находятся вне зоны видимости камеры. Часто в сценариях прописывается краткое описание атмосферы сцены, в том числе через

звуки, что является возможностью для автоматизированной классификации сцены по звуку (рис. 6).

Интерьер. Кухня в загородном доме. Утро.

Сквозь большое окно льется мягкий солнечный свет, освещая деревянный стол, на котором стоит чашка горячего кофе. Слышен тихий шум работающего холодильника и щебетание птиц за окном. На фоне едва слышно журчание воды из крана. Кухонный радио воспроизводит негромкую, расслабляющую музыку.

Маша стоит у плиты, переворачивая блины на сковороде. Её движения размеренные, спокойные. В воздухе витает приятный запах свежеспеченных блинов. Маша ставит тарелку с готовыми блинами на стол и садится за него, берёт чашку кофе, делает глоток и закрывает глаза от удовольствия. В этот момент за окном слышен звук проезжающей машины.

Рис. 6. Пример описания атмосферы сцены через звуки

Для классификации аудио был выбран SoundAnalysis Framework [16] из-за своей оптимизации для языка Swift и широкого набора определяемых классов – он содержит 303 класса аудио. Был реализован модуль классификации файлов с поддержкой многопоточности. SoundAnalysis классифицирует аудио по сегментам, поэтому для общей уверенности в классе считается средняя уверенность по всем сегментам. После проведения тестов был установлен порог уверенности, равный 0,3.

Также был реализован механизм, который самостоятельно определяет, к какому типу сцены отнести файл и по какому алгоритму его сортировать. В модели классификации аудио есть класс speech, и если уверенность в нём больше заданного порога, то файл будет считаться файлом диалога и сортироваться по методу, описанному в предыдущем разделе. Иначе файл считается файлом сцены экспозиции и обрабатывается по правилам классификации таких файлов. Было проведено тестирование на наборе данных из 421 аудиофайла и 218 видеофайлов, после чего для аудиофайлов был установлен порог уверенности 30%, а для видеофайлов – 60%.

ПРОГРАММНЫЕ МЕХАНИЗМЫ ОБНАРУЖЕНИЯ ОБЪЕКТОВ НА ВИДЕОЗАПИСИ

Нахождение объектов на видеозаписи является ключевым способом классификации файлов, так как определяет, что находится в кадре, и оно по определению включается в сцену экспозиции, даже если у неё не будет описания других пунктов.

Для нахождения объектов на видео было решено выбрать оптимизированную для языка программирования Swift модель YOLOv8 [17], так как эта модель специализируется на анализе изображений со скоростью, достаточной, чтобы обнаруживать объекты в реальном времени. В разработанном программном решении была использована сконвертированная в CoreML модель YOLOv8X, которая обеспечивает лучшую точность среди предобученных моделей.

Так как модель работает только с изображениями, был разработан модуль, который с помощью библиотеки Vision выбирает кадр из каждой секунды видео, находит объекты на выбранных изображениях с использованием YOLO и потом оценивает степень уверенности относительно всех кадров.

Этот модуль также подсчитывает количество объектов одного класса в кадре, присваивая файлу дополнительные классы с количественным параметром. К примеру, пусть в видео есть два действующих лица. Модуль определяет класс «person» с уверенностью 100%, а класс «2 person» — с уверенностью 85%. Это необходимо для более точного сравнения со сценарием, чтобы фильтровать сцены с другим количеством людей, определять правильное количество объектов в кадре и так далее.

ПРОГРАММНЫЕ МЕХАНИЗМЫ КЛАССИФИКАЦИИ ДЕЙСТВИЙ НА ВИДЕОЗАПИСИ

Результаты определения объектов на видеозаписи соответствуют существительным в описании сцены экспозиции, в то время как результаты классификации действий должны охватывать глаголы. Поэтому для успешного определения сцены это также необходимый пункт.

Классификация действий на видео происходит с использованием библиотеки Visual Action Kit [18]. Однако была выявлена следующая проблема: библиотека принимает на вход и распознаёт только видео длительностью не более 300

кадров (6–12 секунд в зависимости от частоты кадров). Поэтому она была доработана механизмом разделения входного файла на сегменты длиной 300 кадров и последовательной обработки каждого из них. Это позволило распознавать видео любой длины. Уверенность в классе для всего видео определялась как средняя уверенность по всем сегментам.

ПРОГРАММНЫЕ МЕХАНИЗМЫ КЛАССИФИКАЦИИ ВРЕМЕНИ СУТОК

Все вышеперечисленные механизмы классификации нужны для непосредственной работы алгоритма сравнения, поскольку их результаты описывают действие, происходящее в сцене. Однако количество уникальных классов, по которым сможет определиться сцена экспозиции, сильно ограничено. Было необходимо дополнить это количество так, чтобы новые классы позволяли сужать круг выбора подходящей сцены, что повысило бы точность определения.

Для решения данной задачи предложен программный механизм, основанный на определении классов локаций и времени суток. При оформлении сценария по ранее упомянутому стандарту они будут у каждой сцены, и для файла будет выбираться сцена только среди тех, у которых те же время суток и локация.

Был собран датасет из 4 классов (утро/день/вечер/ночь) размером 1668 изображений (таблица 1).

Таблица 1. Описание датасета для обучения модели классификации времени суток

Класс	Тренировочные данные	Тестовые данные
day	798	38
evening	781	36
morning	790	63
night	800	51

Модель была обучена с помощью CreateML [19], ее точность на тестовом датасете показала 71% (таблица 2). На реальных данных этого оказалось достаточно, так как распознаваться в итоге будут не одно изображение, а набор кадров

из видео. И уверенность в определённом классе может повыситься, а неудачные кадры с ложным определением не окажут сильного влияния на итоговый результат.

Таблица 2. Результат обучения модели классификации времени суток

Accuracy	Precision	Recall	F1 Score
0.71	0.72	0.71	0.71

Был разработан модуль, который обрабатывает кадры в видеофайле и с помощью обученной модели определяет время суток, после чего для каждого класса времени суток вычисляет среднюю уверенность по всем кадрам.

ПРОГРАММНЫЕ МЕХАНИЗМЫ ОПРЕДЕЛЕНИЯ ЛОКАЦИИ НА ВИДЕОЗАПИСИ

Для определения локации был найден проект indoor-scene-classification, который содержал датасет из 64 классов интерьеров [20]. Было принято решение на основе этого датасета обучить новую модель, предварительно дополнив датасет классами локаций экстерьеров. Итого в датасете было собрано 124 класса локаций (64 интерьера, 60 экстерьеров). Общий размер датасета – 18680 изображений.

Все локации в наборе классов являются подклассами двух родительских классов: interior и nature. Поэтому в случаях, когда модель не сможет точно определить локацию на видео, она, по крайней мере, определит, интерьер это или экстерьер, что уже наполовину сузит потенциальный круг поиска сцены (рис. 7).



Рис. 7. Наследование классов локаций в датасете

С использованием CreateML была обучена модель классификации локаций на изображениях, тесты показали 76% точности на тестовом наборе данных (таблица 3). Был разработан модуль, который обрабатывает кадры видео и определяет его локацию, вычисляя среднюю уверенность в классе по всем кадрам.

Таблица 3. Результаты обучения модели классификации локаций

Accuracy	Precision	Recall	F1 Score
0.76	0.83	0.77	0.80

ПРОГРАММНЫЕ МЕХАНИЗМЫ КЛАССИФИКАЦИИ ТЕКСТА СЦЕНАРИЯ

В совокупности все модели классификации видео могут распознать 913 различных классов. Для сравнения файлов с текстом сценария было необходимо выделить то же множество классов из блоков текста. Но при этом нужно было учитывать, что в тексте могут использоваться синонимы слов или связанные по контексту слова или словосочетания.

С использованием механизмов обработки естественного языка wordnet из NLTK [21] для наборов классов каждой из моделей были отдельно составлены словари синонимов или связанных по контексту слов (рис. 8). Если хотя бы один из синонимов встретится в тексте блока экспозиции, то блоку присваивается класс по этому синониму.

```
day: день, дневное время, светлое время суток, деньки, среди дня, полдень, посреди дня  
evening: вечер, вечернее время, закат, сумерки, предзакатное время, к вечеру, вечером  
morning: утро, утреннее время, рассвет, заря, на заре, на рассвете, рано утром  
night: ночь, ночное время, полночь, темное время суток, глубокая ночь, посреди ночи, ночью
```

Рис. 8. Пример словаря синонимов для классов времени суток

При обработке блока сцены текст очищается от пунктуации, все слова переводятся в начальную форму. Затем проводится поиск по каждому слову текста в каждом из словарей, и выявленные классы присваиваются блоку сценария. Таким образом, алгоритм не требователен к конкретному оформлению сцены по стандарту: достаточно просто упомянуть локацию в самом описании сцены или использовать слово, близкое по контексту.

Для классов, которые определялись моделью обнаружения объектов на видео, также был добавлен механизм подсчета количества объектов одного класса. Для локаций дополнительно задавался родительский класс. Для времени суток был реализован механизм, который отыскивает в тексте блока сцены формат времени (hh:mm), чтобы определить время суток в случаях, когда сценарист указал точное время событий в сцене.

Также нужно было учесть, что в сценарии присутствуют имена собственные, которые нужно определять как людей. Так как при загрузке текста сценария определяются диалоги, в которых указаны имена персонажей, которые произносят фразу, то в приложении хранится список всех персонажей, которые найдены в сценарии. Данные имена используются как синонимы к классу person (рис. 9).

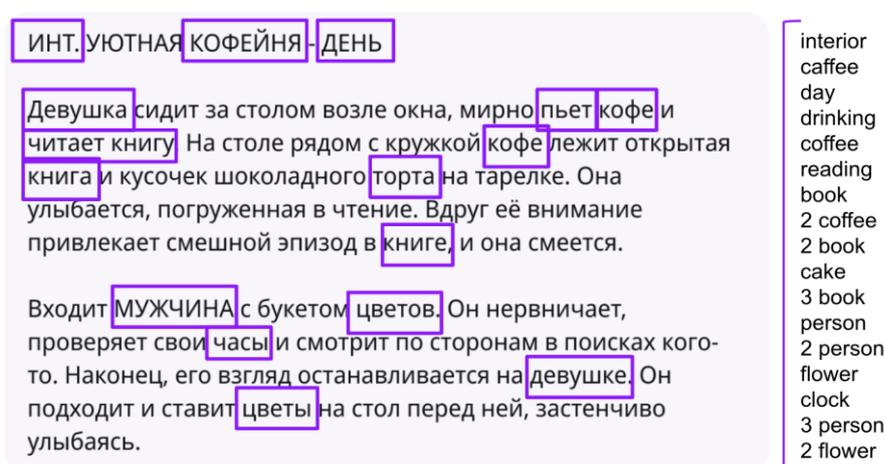


Рис. 9. Классификация блока экспозиции

ОПРЕДЕЛЕНИЕ СООТВЕТСТВИЯ ЭКСПОЗИЦИЙ СЦЕНАРИЮ

В результате разработки механизмов классификации и у файлов, и у блоков сценария потенциально могут определиться классы из полученного набора в 913 классов и дополнительных классов для обнаруженных объектов во множественном числе. Поэтому был необходим алгоритм, который сравнит классы файла с классами всех сцен и подберет лучшую сцену. При этом стоило учитывать, что некоторые классы могут повторяться в некоторых сценах чаще, чем другие.

Для решения этой задачи был выбран метод сравнения по косинусному сходству векторов, составленных с помощью списка классов для каждого файла или сцены. Вектора было решено строить из индексов TF-IDF, которые показывают наличие класса в элементе и важность этого класса относительно всех

остальных [22]. Используя такой подход, удастся лучше определять сцену для файла, опираясь на уникальные классы, которые не встречаются нигде больше или встречаются реже.

Был реализован модуль, который принимает файлы с набором определенных для них классов. Для каждого файла определяется область поиска фразы, опираясь на классы времени суток и локации, если они есть. Далее набор остальных классов каждого файла преобразуется в вектор TF-IDF индексов, и вычисляется косинусное сходство этого вектора с вектором классов каждой сцены (рис. 10). Сцена, с которой сходство наибольшее, определяется как подходящая.

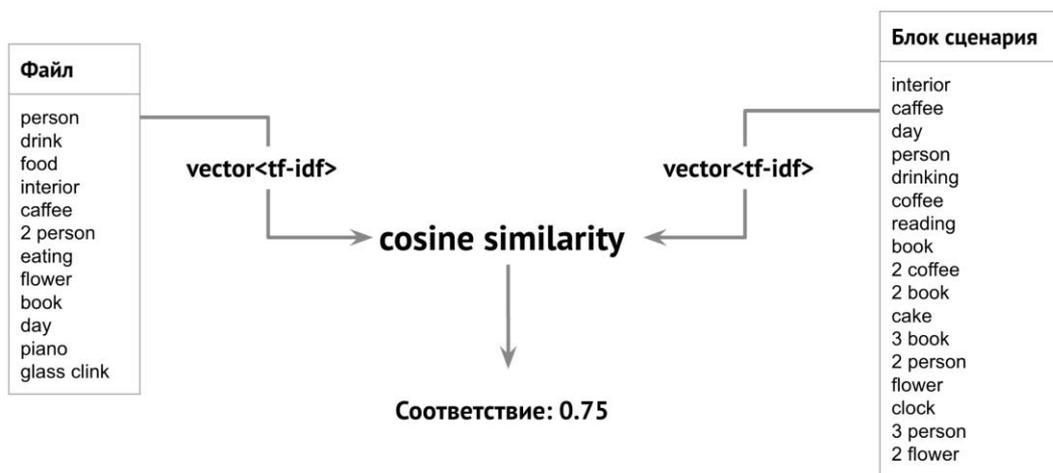


Рис. 10. Алгоритм классификации сцены экспозиции

ОЦЕНКА ТОЧНОСТИ И ПРОИЗВОДИТЕЛЬНОСТИ СИСТЕМЫ

С использованием библиотеки WhisperKit скорость расшифровки одного файла разнится от $\frac{1}{4}$ до $\frac{1}{2}$ от длины файла. Для коротких файлов расшифровка происходит немного медленнее упомянутых скоростей. Это объясняется необходимостью инициализировать модель перед расшифровкой. Инициализация занимает несколько лишних секунд, что незаметно при расшифровке длинных файлов, но видно при обработке множества коротких.

Реализованный модуль с очередью файлов был протестирован на наборе из 421 аудиофайла общей длительностью более 9 часов. Расшифровка всех файлов была завершена примерно за 50 минут.

Модель классификации аудиофайла была протестирована на этом же наборе данных. Длительность распознавания одного файла составляет 0.004 от

длительности этого файла. Все файлы были классифицированы примерно за 4 минуты.

Модели классификации видео были протестированы на наборе данных из 218 видеофайлов. Модели определения локации и времени суток обрабатывают файл практически моментально. Время обработки файла моделью нахождения объектов на видео в среднем составляет 0.15 от длительности файла, а моделью определения действий на видео – 0.29 от длительности файла.

После расшифровки речи и классификации видео и аудио каждый файл был вручную связан с фактической сценой из сценария, подходящей для него. В сценарии всего имелось 820 фраз и 407 сцен-экспозиций.

При этом из общего набора данных в 639 файлов имелось:

- 378 файлов, фактически относящиеся к фразам;
- 187 файлов, фактически относящиеся к сценам экспозициям;
- 73 сторонних медиа файла, не связанные со сценарием.

Успешным результатом работы считается то, что разработанная система правильно определит фразу или сцену-экспозицию для большинства соответствующих файлов. Чтобы минимизировать число ошибок классификации сцен, на поиск и исправление которых пользователю потребовалось бы дополнительное время, было принято решение задать определённый порог уверенности. В случаях, когда алгоритмы недостаточно уверены в правильности классификации, сцена для файла не распознается. Программа сразу отобразит все подобные файлы, чтобы пользователь смог сортировать их вручную. Порог уверенности также должен решить проблему в тех случаях, когда были импортированы сторонние файлы, не связанные со сценарием. Для таких файлов тоже не нужно определять сцену.

Был создан отладочный механизм, который при получении результатов алгоритма классификации маркирует файлы одной из пяти меток:

1. Сцена распознана правильно;
2. Сцена для файла не определена;
3. Сцена распознана неправильно;
4. Сцена для стороннего файла не определена;
5. Сцена для стороннего файла определена.

После сбора данных о соответствии всех определенных файлов с использованием python-библиотеки sklearn была построена ROC-кривая, и по ней вычислен оптимальный порог, который необходимо задать для улучшения результатов алгоритма [23]. Так был вычислен порог в 34% (рис. 11).

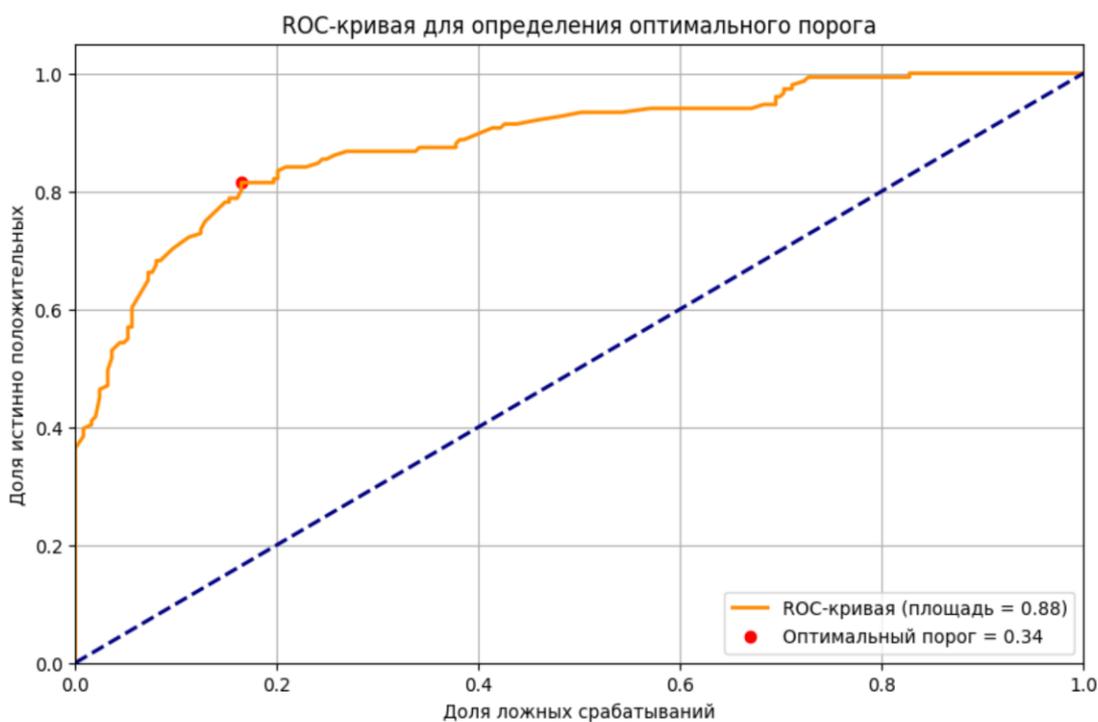


Рис. 11. Определение оптимального порога

Также были учтены случаи для сопоставления с длинными фразами. В этих случаях порог должен работать только тогда, когда у файла соответствие сценарию меньше определенного количества слов (в процессе тестирования было вычислено это количество — 8). Таким образом, если у файла с фразой сходится 8 и более слов, но соответствие ниже порога только потому, что сама фраза состоит из нескольких десятков слов, она всё равно присваивается файлу. Это необходимо в ситуациях, когда длинный монолог снимают в несколько дублей, и в одном дубле будет не вся фраза.

После исправлений в алгоритме была проведена повторная проверка, и теперь результат стал удовлетворительным: большинству файлов была правильно подобрана фраза. А у большинства файлов, для которых не удалось найти фразу правильно, фраза не определилась вовсе во избежание ошибок (рис. 12).

Материалы для монтажа (378 файлов)



Рис. 12. Оценка точности механизма определения фразы

Алгоритм классификации сцен-экспозиций также работает правильно в большинстве случаев. К тому же у него меньший процент ошибок в сравнении с классификацией фраз (рис. 13).

Материалы для монтажа (187 файлов)

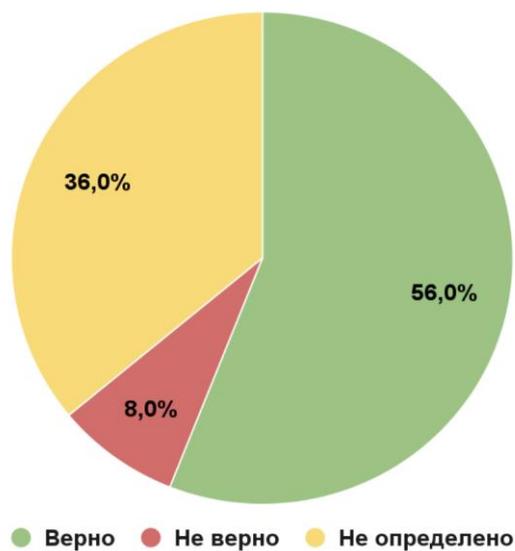


Рис. 13. Оценка точности механизма определения сцены экспозиции

Тестирование на наборе сторонних файлов также дало хороший результат: подавляющее большинство таких файлов не обрабатывается алгоритмами (рис. 14).

Сторонние файлы (74 файла)

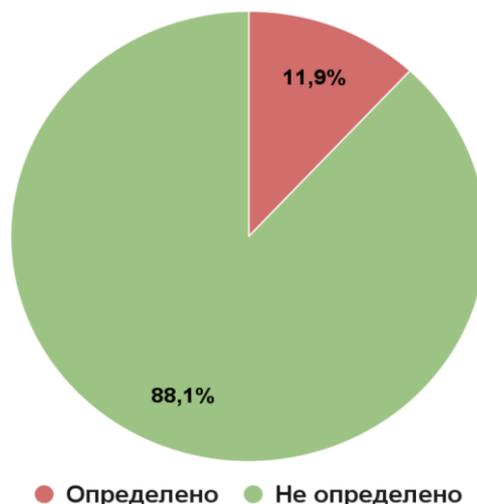


Рис. 14. Оценка точности механизма обнаружения сторонних файлов

Чаще всего оставшиеся ошибки возникают с фразами, состоящими из 1–3 слов, так как вероятность случайного произнесения хотя бы одного из этих слов высокая, и у коротких фраз порог преодолевается очень быстро. К тому же тестирование происходило на сценарии, где фраз почти вдвое больше, чем файлов. И некоторые фразы могут быть похожи друг на друга. На менее объемном сценарии точность алгоритма может быть выше.

ЗАКЛЮЧЕНИЕ

Представлено программное решение, которое позволяет автоматизировать сортировку медиа файлов и оценку соответствия их сценарию перед монтажом видеопроекта.

Программное решение было протестировано на наборе из более чем 400 аудиофайлов и 200 видеофайлов, в результате чего правильно определило 64% файлов для сцен-диалогов и 56% файлов для сцен-экспозиций. Таким образом, за

счет автоматизации удалось сократить время, затрачиваемое на сортировку, на 50%.

СПИСОК ЛИТЕРАТУРЫ

1. Монтаж | Теория кино // Студия “Кинокафе”, 2019.
URL: <https://www.kinocafe.ru/theory/?tid=1221>.
2. Сколько времени длится съемка сцены? // Celebrity.fm, 2020.
URL: <https://celebrity.fm/ru/how-long-does-filming-a-scene-take>.
3. Кинопродюсерство в кинематографии // Allbest, 2020.
URL: https://revolution.allbest.ru/culture/01202211_2.html#text.
4. Создание документального фильма: как работают монтажеры – победители «Эмми» // Sayhi, 2019.
URL: <https://say-hi.me/24-kadra/sozdanie-dokumentalnogo-filma-kak-rabotayut-montazhyory-pobediteli-emmi.html>.
5. Media Composer // AVID, 2023.
URL: <https://ch.avid.com/media-composer>.
6. Как правильно оформить сценарий – правила с примерами // BandBand, 2023. URL: <https://bandband.ru/blog/kak-oformit-scenarij>.
7. Whisper // OpenAI, 2022. URL: <https://openai.com/research/whisper>.
8. OpenAI Whisper // GitHub, 2022.
URL: <https://github.com/openai/whisper>.
9. SwiftWhisper // GitHub, 2023.
URL: <https://github.com/exPHAT/SwiftWhisper>.
10. WhisperKit // GitHub, 2024.
URL: <https://github.com/argmaxinc/WhisperKit>.
11. An Effective Review of Phonetics Algorithms // ResearchGate, 2023.
URL: https://www.researchgate.net/publication/375967141_An_Effective_Review_of_Phonetics_Algorithms.
12. Russian Soundex // GitHub, 2022.
URL: https://github.com/roddar92/russian_soundex.
13. Про многопоточность 2. GCD // Habr, 2021.
URL: <https://habr.com/ru/articles/578752/>.
14. Combine // Apple Developer Documentation, 2024.

URL: <https://developer.apple.com/documentation/combine>.

15. Как правильно оформить сценарий – правила с примерами | Основные сценарные разделы // BandBand, 2023.

URL: <https://bandband.ru/blog/kak-oformit-scenarij#dirs>.

16. Sound Analysis // Apple Developer, 2022.

URL: <https://developer.apple.com/documentation/soundanalysis>.

17. YOLOv8 // Ultralytics, 2023.

URL: <https://docs.ultralytics.com/ru/models/yolov8>.

18. VisualActionKit // GitHub, 2020.

URL: <https://github.com/lukereichold/VisualActionKit>.

19. Creating an Image Classifier Model // Apple Developer, 2021.

URL: <https://developer.apple.com/documentation/createml/creating-an-image-classifier-model>.

20. keremberke/indoor-scene-classification // Hugging Face, 2022.

URL: <https://huggingface.co/datasets/keremberke/indoor-scene-classification>.

21. Sample usage for wordnet // NLTK, 2023.

URL: <https://www.nltk.org/howto/wordnet.html>.

22. Извлечение признаков из текстовых данных с использованием TF-IDF // Habr, 2023. URL: https://github.com/roddar92/russian_soundex.

23. Selecting the Optimal Probability Threshold for a Classification Model, ROC Curve Analysis and KS Score // DataScienceByExample, 2023.

URL: <https://www.datasciencebyexample.com/2023/04/19/choose-threshold-for-classification-model-with-ROC-analysis>.

AUTOMATION OF FOOTAGES SORTING BY SCREENPLAY TEXT FOR VIDEO EDITING

A. D. Nemanov¹ [0009-0004-1840-2347], **I. S. Shakhova**² [0000-0003-1591-5767]

^{1, 2}*Institute of Information Technology and Intelligent Systems, Kazan Federal University, Kazan;*

¹andrewoch@yandex.ru, ²is@it.kfu.ru

Abstract

The video editing process involves numerous labor-intensive operations for sorting and preparing footages, requiring significant time investment. This article describes the development of a software solution that uses machine learning technology to automate these processes.

The primary focus is on creating a system capable of classifying and sorting media files according to the screenplay text, thereby increasing the efficiency of material preparation for editing. The system includes modules for speech recognition, audio and video classification, and algorithms for determining screenplay compliance.

Testing showed that the proposed system correctly classifies media files in most cases, significantly reducing rough-cut editing time.

Keywords: *video editing, automation, machine learning, speech recognition, audio classification, video classification, coreml, parallel computing, screenplay, soundex, tf-idf, cosine similarity, natural language processing.*

REFERENCES

1. Montazh | Theory of Cinema // Studio "Kinocafe", 2019.
URL: <https://www.kinocafe.ru/theory/?tid=1221>.
2. Skol'ko vremeni dlitsya syemka stseny? // Celebrity.fm, 2020.
URL: <https://celebrity.fm/ru/how-long-does-filming-a-scene-take>.
3. Kinoproduyserstvo v kinematografii // Allbest, 2020.
URL: https://revolution.allbest.ru/culture/01202211_2.html#text.
4. Sozdanie dokumental'nogo fil'ma: kak rabotayut montazhery-pobediteli "Emmi" // Sayhi, 2019.
URL: <https://say-hi.me/24-kadra/sozdanie-dokumentalnogo-filma-kak-rabotayut-montazhyory-pobediteli-emmi.html>.
5. Media Composer // AVID, 2023.
URL: <https://ch.avid.com/media-composer>.
6. Kak pravil'no oformit' stsenaariy – pravila s primerami // BandBand, 2023.
URL: <https://bandband.ru/blog/kak-oformit-scenarij>.
7. Whisper // OpenAI, 2022. URL: <https://openai.com/research/whisper>.
8. OpenAI Whisper // GitHub, 2022.
URL: <https://github.com/openai/whisper>.

9. SwiftWhisper // GitHub, 2023.
URL: <https://github.com/exPHAT/SwiftWhisper>.
 10. WhisperKit // GitHub, 2024.
URL: <https://github.com/argmaxinc/WhisperKit>.
 11. An Effective Review of Phonetics Algorithms // ResearchGate, 2023.
URL: https://www.researchgate.net/publication/375967141_An_Effective_Review_of_Phonetics_Algorithms.
 12. Russian Soundex // GitHub, 2022.
URL: https://github.com/roddar92/russian_soundex.
 13. Pro mnogopotchnost' 2. GCD // Habr, 2021.
URL: <https://habr.com/ru/articles/578752/>.
 14. Combine // Apple Developer Documentation, 2024.
URL: <https://developer.apple.com/documentation/combine>.
 15. Kak pravil'no oformit' stsensariy – pravila s primerami | Osnovnye stsensarnye razdely // BandBand, 2023.
URL: <https://bandband.ru/blog/kak-oformit-scenarij#dirs>.
 16. Sound Analysis // Apple Developer, 2022.
URL: <https://developer.apple.com/documentation/soundanalysis>.
 17. YOLOv8 // Ultralytics, 2023.
URL: <https://docs.ultralytics.com/ru/models/yolov8>.
 18. VisualActionKit // GitHub, 2020.
URL: <https://github.com/lukereichold/VisualActionKit>.
 19. Creating an Image Classifier Model // Apple Developer, 2021.
URL: <https://developer.apple.com/documentation/createml/creating-an-image-classifier-model>.
 20. keremberke/indoor-scene-classification // Hugging Face, 2022.
URL: <https://huggingface.co/datasets/keremberke/indoor-scene-classification>.
 21. Sample usage for wordnet // NLTK, 2023.
URL: <https://www.nltk.org/howto/wordnet.html>.
 22. Izvlechenie priznkov iz tekstovykh dannyykh s ispol'zovaniem TF-IDF // Habr, 2023. URL: https://github.com/roddar92/russian_soundex.
 23. Selecting the Optimal Probability Threshold for a Classification Model, ROC Curve Analysis and KS Score // DataScienceByExample, 2023.
-

URL: <https://www.datasciencebyexample.com/2023/04/19/choose-threshold-for-classification-model-with-ROC-analysis>.

СВЕДЕНИЯ ОБ АВТОРАХ



НЕМАНОВ Андрей Дмитриевич – бакалавр программной инженерии Института информационных технологий и интеллектуальных систем Казанского федерального университета.

Andrey Dmitrievich NEMANOV – Bachelor of Software Engineering, Institute of Information Technology and Intelligent Systems, Kazan Federal University.

email: andrewoch@yandex.ru

ORCID: 0009-0004-1840-2347



ШАХОВА Ирина Сергеевна – старший преподаватель кафедры программной инженерии Института информационных технологий и интеллектуальных систем Казанского федерального университета, г. Казань.

Irina Sergeevna SHAKHOVA – Senior Lecturer at the Department of Software Engineering, Institute of Information Technology and Intelligent Systems, Kazan Federal University, Kazan.

email: is@it.kfu.ru

ORCID: 0000-0003-1591-5767

Материал поступил в редакцию 19 июля 2024 года

УЧЁТ СТРУКТУРЫ ДОКУМЕНТА В МЕТОДЕ АВТОМАТИЧЕСКОГО АННОТИРОВАНИЯ МАТЕМАТИЧЕСКИХ ПОНЯТИЙ В ОБРАЗОВАТЕЛЬНЫХ ТЕКСТАХ

К. С. Николаев^[0000-0003-3204-238X]

*Лаборатория суперкомпьютерного моделирования Отделения суперкомпьютерных систем и параллельных вычислений ФГБУ «Национальный исследовательский центр «Курчатовский институт»», г. Москва;
Казанский (Приволжский) федеральный университет, г. Казань.*

konnikolaeff@yandex.ru

Аннотация

Обогащение образовательных текстов семантическим содержимым (в частности, дополнение документа гиперссылками на страницы сервиса, отображающего подробную информацию о понятиях, используемых в тексте) способствует повышению эффективности усвоения материала обучающимися. Существующие методы семантической разметки образовательных текстов не учитывают структурные особенности таких документов, что приводит к избыточному распознаванию понятий.

В статье описано развитие метода автоматического аннотирования математических понятий в образовательных математических текстах путем добавления функционала для учета структуры образовательного документа. Основное назначение метода заключается в обработке образовательных материалов курса дистанционного образования «Технология решения планиметрических задач». Соблюдение единого шаблона при создании страниц курса позволяет применить анализ веб-разметки страниц и ключевых слов, примененных создателями курса. Основной задачей в данном процессе является определение типа ячеек таблицы, в которых находятся текстовые фрагменты образовательных материалов. В соответствии с рекомендациями создателей курса, определения необходимо выделять в ячейках, содержащих постановку задачи, а также в тех блоках, где указаны входные данные задачи. Определение типа ячеек таблиц производится с помо-

щью анализа их атрибутов и поиска ключевых слов в их содержимом. Такое ограничение распознаваемых фрагментов текста позволяет улучшить восприятие страниц курса учеником и повысить качество усвоения учебного материала.

***Ключевые слова:** семантический анализ, математическая онтология, дидактические отношения, математическое образование, разметка документа.*

ВВЕДЕНИЕ

Применение онтологий в образовательных приложениях является актуальной задачей, так как применение классических образовательных материалов без семантического содержимого не позволяет в полной мере вовлечь учащегося в образовательный процесс и снижает общую эффективность обучения. Также онтологии позволяют студенту сформировать более цельную картину изучаемой дисциплины, изучить отношения и закономерности между отдельными понятиями в проблемной области.

Онтологии и семантические технологии часто применяются в задачах улучшения образовательного процесса. В работе [1] авторы предложили онтологию для поддержки и унификации процесса составления учебной программы. В работе [2] применены технологии Semantic Web для персонализации обратной связи в образовательных средах. В [3] использована методология MOnto для разработки многоязычных приложений в образовательной среде. Онтологии также активно применяются в обучении анатомии и медицины в целом. В [4] представлен подход к персонализации программ образовательных курсов с помощью онтологии. В [5] онтологии и семантические технологии применены для корректировки и персонализации назначения заданий студентам. Авторы [6] предложили использовать онтологии в качестве основы рекомендационной системы.

Для использования преимуществ, предлагаемых семантическими технологиями при обработке и представлении математического контента, была предложена экосистема OntoMath^{Edu} [7]. Она содержит набор инструментов для применения в задачах математического образования.

Основным источником понятий и отношений между понятиями названной экосистемы служит онтология OntoMath^{Edu} [8–18]. Она подробно отражает предметную область планиметрии и предназначена для включения в программу курса по планиметрии, который разрабатывается в Казанском федеральном университете для общеобразовательных средних школ. В онтологии OntoMath^{Edu} выделены три уровня:

1. Предметный уровень, включающий понятия, которые не зависят от языка и входят в программу обучения в средней школе.
2. Лингвистический уровень, содержащий текстовые представления понятий предметного уровня онтологии.
3. Уровень онтологии верхнего уровня, предоставляющий метаонтологические аннотации.

Отличительной особенностью онтологии OntoMath^{Edu} от других предметных онтологий является наличие дидактических отношений между понятиями: образовательные системы, образовательные уровни и пререквизиты [19–21]. Образовательные системы отражают различия в подходе к изучению понятий планиметрии в различных странах [22] (на данный момент в онтологии представлены образовательные системы РФ, Великобритании и США). Образовательные уровни позволяют разделить множество понятий на подмножества, содержащие понятия, изучаемые на том или ином образовательном уровне (например, в образовательной системе РФ для программы планиметрии: 7 класс, 8 класс, профильный 8 класс и т. д.). Пререквизиты отражают последовательность изучения понятий в соответствии с образовательной системой [23].

В состав экосистемы OntoMath^{Edu} входит набор взаимосвязанных сервисов, обеспечивающих обработку, генерацию и визуализацию математических документов [24, 25]. Этот набор содержит сервисы, приведенные ниже (см. также рис. 1).

1. Сервис обогащения онтологии, формирующий набор рекомендаций для экспертов по улучшению структуры онтологии путем поиска контекстуально близких понятий, между которыми отсутствуют прямые связи в онтологии.
 2. Сервис генерации тестовых заданий, формирующий тестовые задания с вводом ответа и интерактивные задания по восстановлению иерархии понятий образовательной онтологии [26].
-

3. Сервис семантического аннотирования формул в pdf-документах, формирующий семантическое представление встроенных формул на основе включенных в них компонентов для формирования коллекции документов, применяемых в семантическом поисковике формул [27].
4. Сервис выделения математических понятий в документах.
5. Сервис визуализации подграфов семантических сетей [28].

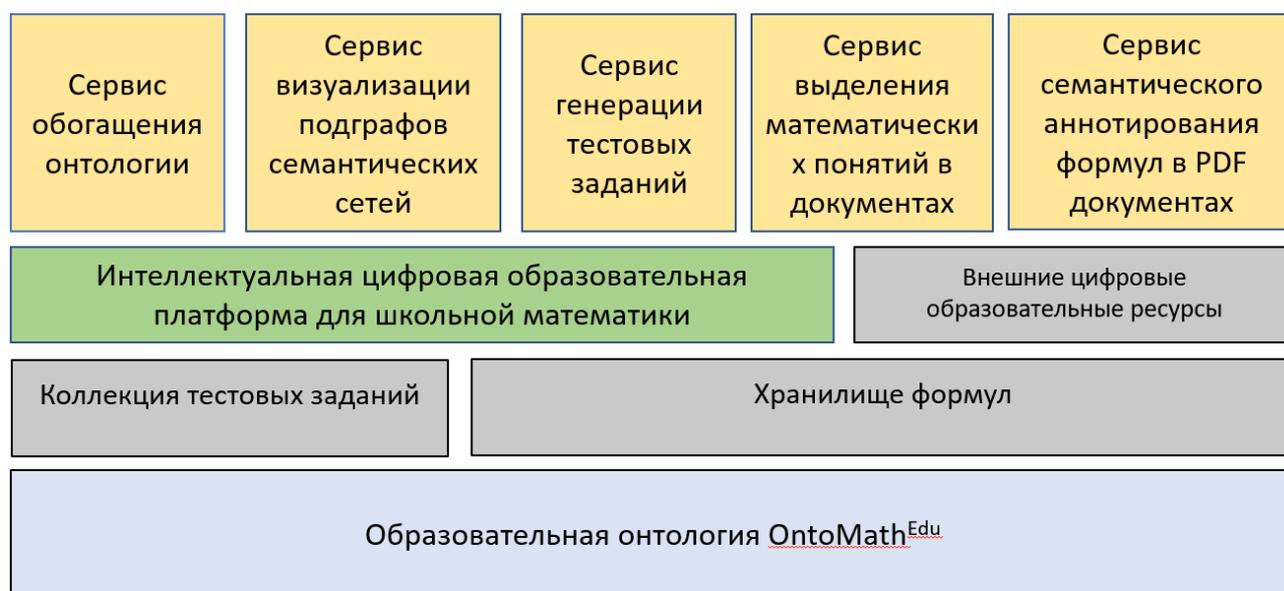


Рис. 1. Структура экосистемы OntoMath^{Edu}

Помимо перечисленных сервисов в состав экосистемы входят наборы данных, необходимые для работы сервисов и являющиеся результатами работы сервисов. Хранилище формул содержит набор математических утверждений, представленных на естественном языке, в нотации LaTeX и в формализации OpenMath [29]. Коллекция тестовых заданий содержит множество таких заданий, сформированных на основе математических утверждений в формате OpenMath из хранилища формул, а также множество интерактивных заданий на восстановление фрагментов иерархии понятий онтологии [OntoMath^{Edu}](https://github.com/CLLKazan/OntoMathEdu) (<https://github.com/CLLKazan/OntoMathEdu>). Ссылки на внешние образовательные ресурсы интегрированы в концепты онтологии и содержат информацию о концептах на других ресурсах в интернете.

Основой сервиса выделения математических понятий в документах является метод автоматического аннотирования математических понятий в образовательных математических текстах. Сервис осуществляет поиск математических понятий из онтологии *OntoMath^{Edu}* в образовательных текстах в формате HTML и производит обогащение документа ссылками на сервис визуализации подграфов семантических сетей, отображающий подробную информацию о математическом понятии и понятиях, связанных с ним. В следующем разделе описаны метод автоматического аннотирования математических понятий в образовательных математических текстах и его улучшение с помощью учета разметки документов.

МЕТОД АВТОМАТИЧЕСКОГО АННОТИРОВАНИЯ МАТЕМАТИЧЕСКИХ ПОНЯТИЙ В ОБРАЗОВАТЕЛЬНЫХ МАТЕМАТИЧЕСКИХ ТЕКСТАХ

Блок-схема метода автоматического аннотирования математических понятий в образовательных текстах представлена на рис. 2. Далее перечислены основные шаги указанного метода.

1. Предобработка документа, включающая разбор HTML-документа на теги, извлечение текстового содержимого, нормализация всех слов в предложениях.

2. Выбор значимых слов (существительные, прилагательные в полной и краткой формах) в пределах каждого предложения. Формирование цепочек значимых слов с длинами от 1 до N, где N – количество значимых слов в предложении.

3. Предобработка концептов онтологии *OntoMath^{Edu}*: нормализация всех слов в названиях концептов (например, концепт «Вершина многоугольника» преобразуется в «вершина многоугольник») и выделение значимых слов.

4. Сравнение цепочек слов в предложениях с множеством значимых слов в концептах онтологии, фильтрация и сохранение результатов разметки.

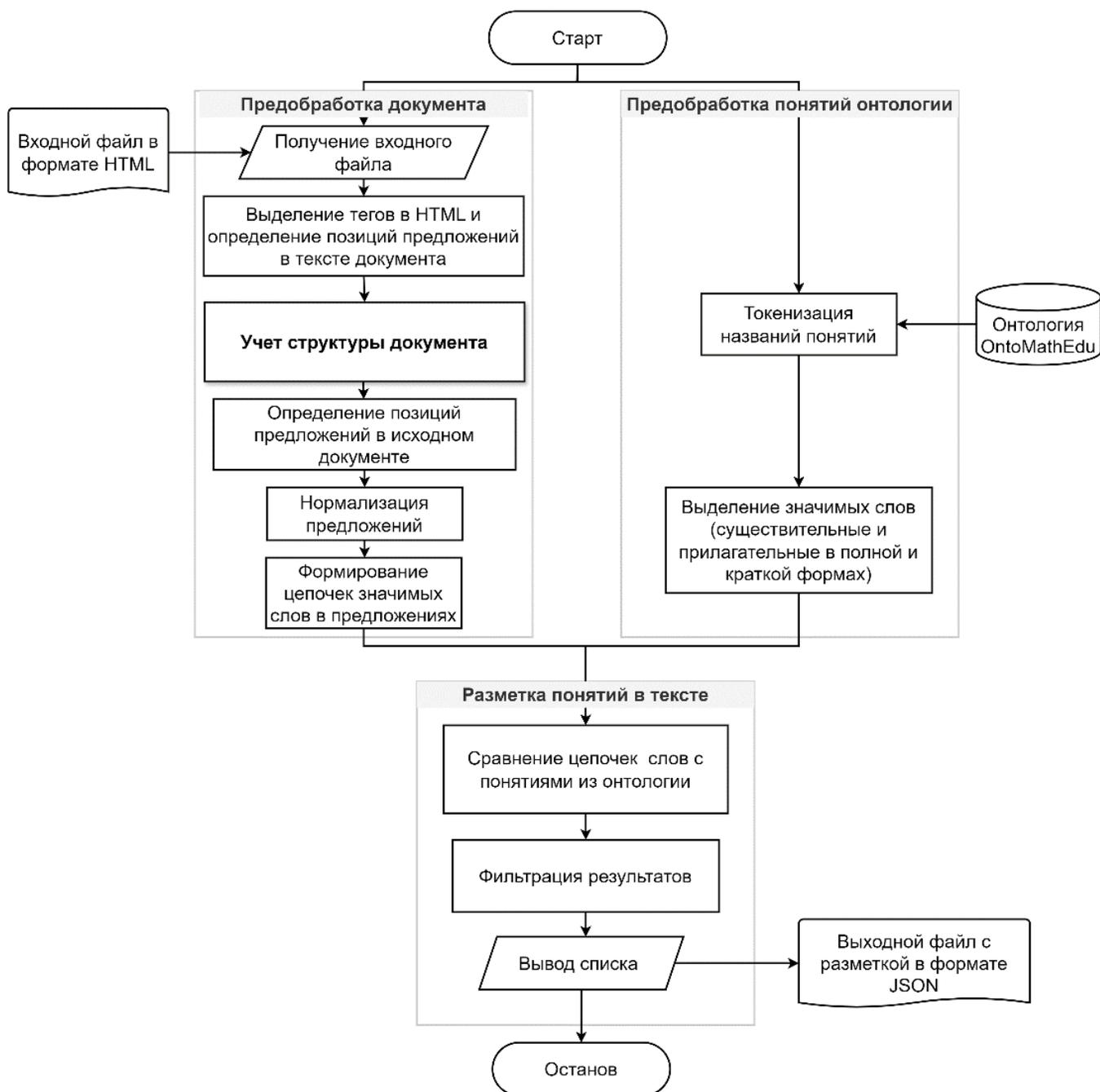


Рис. 2. Блок-схема метода автоматического аннотирования математических понятий в образовательных текстах

При полном совпадении сочетания слов и названия концепта каждому слову в сочетании присваивается ссылка на концепт онтологии. Если слово оказывается связанным с двумя и более концептами, приоритет отдается понятию с наибольшим количеством слов. Например, в тексте «В прямоугольном треугольнике проведена высота из угла А» на слово «треугольнике» будут претендовать

понятия «Треугольник» и «Прямоугольный треугольник», и в результате понятие «Треугольник» будет игнорироваться.

Для управления процессом аннотирования и редактирования результатов было разработано программное обеспечение, позволяющее в интерактивном режиме производить разметку документов в формате HTML с визуализацией текущего состояния документа и производить привязку концептов онтологии OntoMath^{Edu} к фрагментам текста документа. На рис. 3 приведен интерфейс этого программного обеспечения.

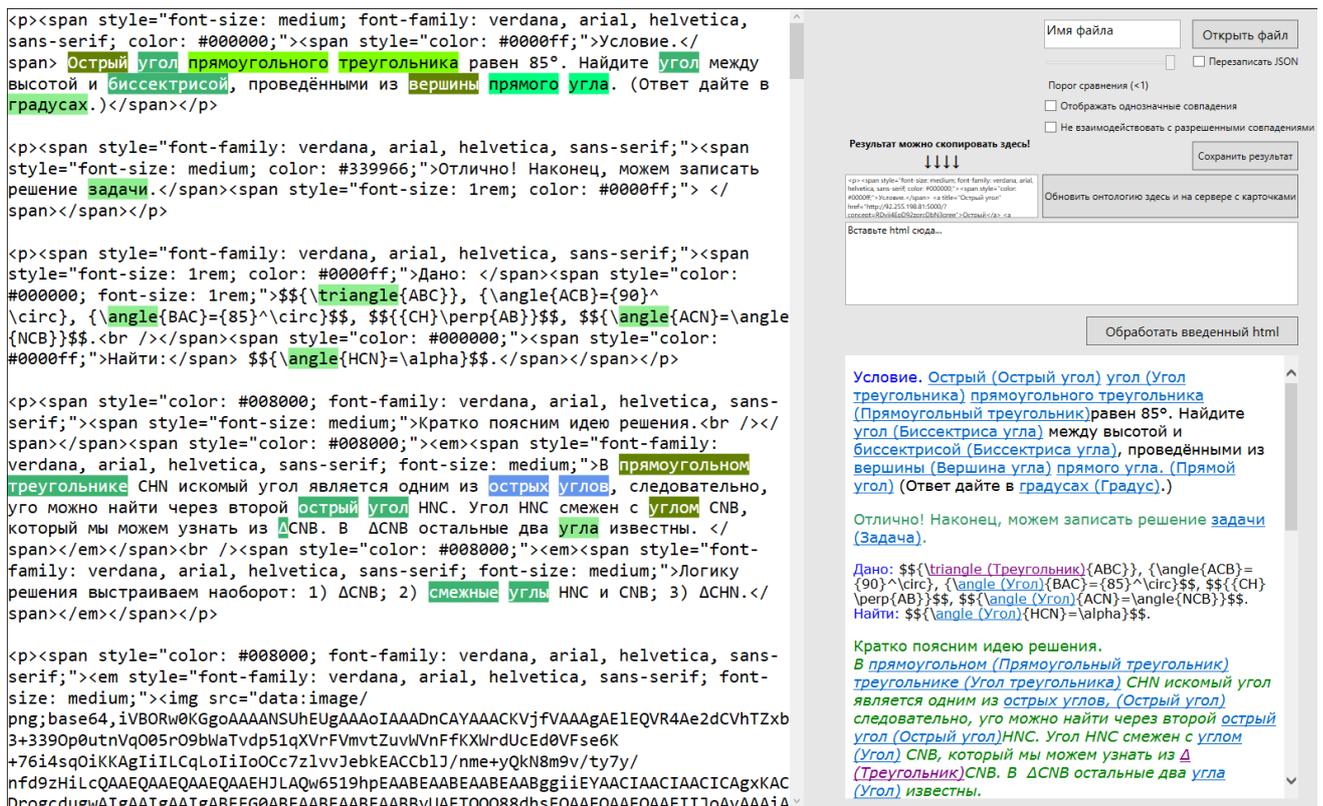


Рис. 3. Интерфейс программного обеспечения для управления методом аннотирования математических понятий в образовательных текстах

Названное программное обеспечение применено при разработке материалов курса дистанционного образования «Технология решения планиметрических задач». С учетом отзывов авторов курса возникла необходимость улучшения процесса аннотации математических понятий с помощью учета структуры образовательных документов для исключения фрагментов документа из процесса аннотации. Модификация указанного программного обеспечения описана в следующем разделе.

УЧЁТ ОГРАНИЧЕНИЙ В МЕТОДЕ АВТОМАТИЧЕСКОГО АННОТИРОВАНИЯ МАТЕМАТИЧЕСКИХ ПОНЯТИЙ В ОБРАЗОВАТЕЛЬНЫХ МАТЕМАТИЧЕСКИХ ТЕКСТАХ

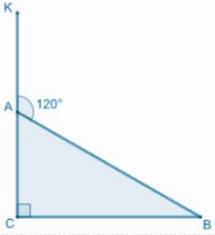
Учебные материалы дистанционного курса «Технологии решения планиметрических задач» представлены в виде таблицы, в отдельных ячейках которых размещены условие задачи, вспомогательная информация и блоки с вариантами заданий. В процессе аннотирования должны участвовать только ячейки с условием задачи и блоки с вариантами заданий. На рис. 4 приведен фрагмент HTML-разметки, содержащей условие задачи. Жирным шрифтом выделены теги, на основе которых происходит фильтрация ячеек таблицы (`<table>` `<tbody>` `<td>`, производится поиск самой широкой ячейки). Кроме того, на этом рисунке жирным шрифтом выделен пример ключевой фразы, по которой дополнительно фильтруются ячейки таблицы («Условие задачи», «Дано»). Набор ключевых фраз будет расширяться в зависимости от разнообразия форматов образовательных документов в дистанционном курсе.

```
<table border="0">
  <tbody>
    <tr>
      <td colspan="3">
        <p><span
            style="color: #ff6600;"><strong>Условие задачи</strong>
        </span>В <a title="Прямоугольный треугольник"
          href="http://84.39.245.172:5000">прямоугольном
          треугольнике</a> <em>ABC</em> с <a title="Прямой угол"
          href="http://84.39.245.172:5000">прямым
          углом</a> <em>C</em> <a
            title="Внешний угол треугольника"
            href="http://84.39.245.172:5000">внешний
            угол</a> при вершине <em>A</em> равен  $120^\circ$ , <em>AC
          </em>+ <em>AB</em> = 18. Найти <em>AC</em> и
          <em>AB</em>.</p>
        </td>
      </tr>
    </tbody>
  </table>
```

Рис. 4. Результат аннотирования математических понятий без учета структуры документа

На рис. 5 представлено сравнение результатов автоматического аннотирования до и после добавления учета разметки документа. Как видно из рис. 5 а, без учета разметки метод аннотирования производит разметку также во вспомогательных блоках (фрагменты текста с фиолетовым цветом шрифта). После исключения блоков, не содержащих ключевые фразы, метод аннотирования затрагивает только значимые блоки (рис. 5 б).

Условие задачи. В прямоугольном треугольнике ABC с прямым углом C внешний угол при вершине A равен 120° , $AC + AB = 18$. Найти AC и AB .



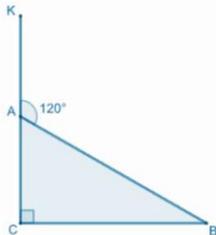
На продолжении стороны AC отметим точку K и составим краткую запись.

Кстати, давай договоримся обозначать углы одной буквой там, где угол определен однозначно, например, углы при вершинах B и C будем обозначать $\angle B$ и $\angle C$. При вершине A имеются два угла, их мы обозначим тремя буквами – $\angle BAC$ и $\angle KAB$.

Давай проверим, сможешь ли ты распознать краткую запись, соответствующую требованиям.

Ано: ΔABC , $\angle C = 90^\circ$, $\angle KAB$ – внешний с $\angle A = 120^\circ$, $AC + AB = 18$ см.	Ано: ΔABC , $\angle C = 90^\circ$, $\angle KAB$ – внешний с $\angle BAC = 120^\circ$, $AC + AB = 18$ см.	Ано: ΔABC , $\angle C = 90^\circ$, $\angle A$ – внешний с $\angle BAC = 120^\circ$, $AC + AB = 18$ см.
Н йти: AC, AB	Н йти: AC, AB	Н йти: AC, AB

Условие задачи. В прямоугольном треугольнике ABC с прямым углом C внешний угол при вершине A равен 120° , $AC + AB = 18$. Найти AC и AB .



На продолжении стороны AC отметим точку K и составим краткую запись.

Кстати, давай договоримся обозначать углы одной буквой там, где угол определен однозначно, например, углы при вершинах B и C будем обозначать $\angle B$ и $\angle C$. При вершине A имеются два угла, их мы обозначим тремя буквами – $\angle BAC$ и $\angle KAB$.

Давай проверим, сможешь ли ты распознать краткую запись, соответствующую требованиям.

Ано: ΔABC , $\angle C = 90^\circ$, $\angle KAB$ – внешний с $\angle A = 120^\circ$, $AC + AB = 18$ см.	Ано: ΔABC , $\angle C = 90^\circ$, $\angle KAB$ – внешний с $\angle BAC = 120^\circ$, $AC + AB = 18$ см.	Ано: ΔABC , $\angle C = 90^\circ$, $\angle A$ – внешний с $\angle BAC = 120^\circ$, $AC + AB = 18$ см.
Н йти: AC, AB	Н йти: AC, AB	Н йти: AC, AB

а)
б)

Рис. 5. Результат аннотирования математических понятий. а) без учета структуры документа, б) с учетом структуры документа

ЗАКЛЮЧЕНИЕ

Представлено развитие метода автоматического аннотирования математических понятий в образовательных текстах с помощью учета структуры документов в формате HTML, позволяющего исключить из разметки фрагменты текста, содержащие вспомогательную информацию, и уменьшить объем ручной работы для авторов дистанционного курса «Технология решения планиметрических задач». В будущем планируется улучшить учет разметки образовательных документов.

Укажем основные направления планируемого расширения функциональных возможностей разметки документов.

1. Условия фильтрации фрагментов документа, представленные выше, специфичны для стиля оформления учебных материалов в конкретном дистанционном курсе. Необходима разработка более универсальных параметров фильтрации блоков документа, подходящих для большого количества учебных материалов.

2. Расширение функционала для настройки анализа разметки документа в программное обеспечение для визуализации и редактирования разметки.

Благодарности

Статья подготовлена в рамках государственного задания Федерального государственного бюджетного учреждения «Национальный исследовательский центр «Курчатовский институт» (тема FNEF-2024-0014).

СПИСОК ЛИТЕРАТУРЫ

1. *Zouri M., Ferworn A.* An Ontology-Based Approach for Curriculum Mapping in Higher Education // 2021 IEEE 11th Annu. Comput. Commun. Work. Conf. 2021. P. 141–147.

2. *Cheniti Belcadhi L.* Personalized feedback for self-assessment in lifelong learning environments based on semantic web // Comput. Human Behav. 2016. Vol. 55. P. 562–570.

3. *Florrence M.* Building a multilingual ontology for education domain using monto method // Comput. Sci. Inf. Technol. 2020.

4. *Ibrahim M.E., Yang Y., Ndzi D.L., Yang G., Al-Maliki M.* Ontology-Based Personalized Course Recommendation Framework // IEEE Access. 2019. Vol. 7. P. 5180–5199.

5. *Radovic M., Petrovic N., Tosic M.* An Ontology-Driven Learning Assessment Using the Script Concordance Test // Appl. Sci. 2022. Vol. 12, No. 3.

6. *Obeid C., Lahoud I., El Khoury H., Champin P.A.* Ontology-based Recommender System in Higher Education // Web Conf. 2018 – Companion World Wide Web Conf. WWW 2018. 2018. P. 1031–1034.

7. *Nikolaev K.S., Nevzorova O.A., Falileeva M.V.* Developing the OntoMathEdu ecosystem for educational applications // CEUR Workshop Proc. 2021. Vol. 2910. P. 81–87.

8. *Невзорова О.А., Фалилеева М.В., Липачёв Е.К., Шакирова Л.Р., Дюпина А.Э., Кириллович А.В.* Образовательная онтология OntoMathEdu: проблемы онтологического инжиниринга // XX национальная конференция по искусственному интеллекту с международным участием, КИИ-2022. Труды конференции. В 2 т. Том 1. М.: Издательство МЭИ, 2022. С. 177–188.

9. *Nevzorova O.A., Falileeva M.V., Kirillovich A.V., Lipachev E.K., Shakirova L.R., Dyupina A.E.* OntoMathEdu Educational Ontology: Problems of Ontological Engineering // *Pattern Recognit. Image Anal.* 2023. Vol. 33. P. 460–466.

<https://doi.org/10.1134/S1054661823030367>

10. *Dyupina A., Falileeva M.* Problems of Designing Polylingual Ontology OntoMathEdu // *Communications in Computer and Information Science.* 2020. Vol. 1242. P. 179–190. https://doi.org/10.1007/978-3-030-65218-0_14

11. *Shakirova L., Falileeva M., Kirillovich A., Lipachev E., Nevzorova O., Nevzorov V.* Modeling and Evaluation of the Mathematical Educational Ontology // *SSI-2019 Scientific Services & Internet Proceedings of the 21st Conference on Scientific Services & Internet (SSI-2019) Novorossiysk-Abrau, Russia, September 23–28, 2019.* CEUR Workshop Proceedings. 2019. Vol. 2543. P. 305–319.

12. *Шакирова Л.Р., Фалилеева М.В., Кириллович А.В., Липачёв Е.К., Невзорова О.А., Невзоров В.Н.* Образовательная математическая онтология OntoMathEdu: структура и отношения // *Научный сервис в сети Интернет: труды XXI Всероссийской научной конференции (23–28 сентября 2019 г., г. Новороссийск).* М.: ИПМ им. М.В. Келдыша. 2019. С. 653–661.

13. *Nevzorova O., Shakirova L., Falileeva M., Kirillovich A., Lipachev E., Nevzorov V.* OntoMathEdu Educational Mathematical Ontology: Annotation of Concepts // *Selected Contributions of the "Russian Advances in Artificial Intelligence" Track at RCAI 2020 co located with 18th Russian Conference on Artificial Intelligence (RCAI 2020).* Moscow, Russia, October 10–16, 2020. Vol. 2648. P. 157–168.

14. *Kirillovich A., Nevzorova O., Falileeva M., Lipachev E., Shakirova L.* OntoMathEdu: A Linguistically Grounded Educational Mathematical Ontology // *Intelligent Computer Mathematics. Series: Lecture Notes in Computer Science.* 2020. Vol. 12236. P. 157–172. <https://doi.org/10.1007/978-3-030-53518-6>

15. *Kirillovich A., Nevzorova O., Falileeva M., Lipachev E., Shakirova L.* OntoMathEdu: Towards an Educational Mathematical Ontology // *Edwin Brady et al. (Eds.). Workshop Papers at 12th Conference on Intelligent Computer Mathematics (CICM-WS 2019), Prague, Czech Republic, 8–12 July 2019.* CEUR Workshop Proceedings. 2020. Vol. 2634.

16. *Shakirova L., Falileeva M., Kirillovich A., Lipachev E., Nevzorova O., Nevzorov V.* Modeling and Evaluation of the Mathematical Educational Ontology // Proceedings of the 21st Conference on Scientific Services & Internet (SSI-2019). CEUR Workshop Proceedings. Vol. 2543. P. 305–319.

17. *Шакирова Л.Р., Фалилеева М.В., Кириллович А.В., Липачев Е.К., Невзорова О.А., Невзоров В.Н.* Образовательная математическая онтология OntoMathEdu: структура и отношения // Научный сервис в сети Интернет. 2019. Том 21. С. 653–661. <https://doi.org/10.20948/abrau-2019-84>

18. *Шакирова Л.Р., Фалилеева М.В., Кириллович А.В., Невзорова О.А.* Проектирование образовательной математической онтологии: проблемы и методы решения на примере курса планиметрии // Труды международной конференции по компьютерной и когнитивной лингвистике: Материалы XV международной конференции по компьютерной и когнитивной лингвистике TEL-2018 в 2-х томах, Казань. 2018. С. 393–405. EDN SQGRVY.

19. *Falileeva M., Kirillovich A., Shakirova L., Nevzorova O., Lipachev E., Dyupina A.* OntoMathEdu Educational Mathematical Ontology: Prerequisites, Educational Levels and Educational Projections // CEUR Workshop Proceedings. 2020. Vol. 2784. P. 346–351.

20. *Фалилеева М.В., Кириллович А.В., Невзорова О.А., Шакирова Л.Р., Липачёв Е.К., Дюпина А.Э.* Системы образовательных проекций, уровней и пререквизитов математической онтологии OntoMathEdu // Электронные библиотеки. 2021. Т. 24. № 3. С. 505–530. <https://doi.org/10.26907/1562-5419-2021-24-3-505-530>

21. *Nevzorova O.A., Falileeva M.V., Kirillovich A.V., Shakirova L.R., Lipachev E.K., Dyupina A.E.* Modeling of didactic relationships in the OntoMathEdu educational mathematical ontology // CEUR Workshop Proceedings. 2021. Vol. 2910. P. 11–21.

22. *Фалилеева М.В., Кириллович А.В., Невзорова О.А., Шакирова Л.Р., Липачёв Е.К., Дюпина А.Э.* Разработка образовательных проекций математической онтологии OntoMathEdu // Научный сервис в сети Интернет. 2020. Т. 22. С. 604–611. <https://doi.org/10.20948/abrau-2020-26>. EDN WBNWIC

23. *Kirillovich A., Falileeva M., Nevzorova O., Lipachev E., Dyupina A., Shakirova L.* Prerequisite Relationships of the OntoMathEdu Educational Mathematical On-

tology // J.C. Figueroa-García et al. (Eds.). Proceedings of the 8th Workshop on Engineering Applications (WEA 2021). Communications in Computer and Information Science. 2021. Vol. 1431. P. 517–524. https://doi.org/10.1007/978-3-030-86702-7_44

24. *Nevzorova O., Shakirova L., Falileeva M., Nikolaev K.* Towards Ontology-Based Software Services for Teaching School Mathematics // L. Gómez Chova et al. (Eds.). Proceedings of the 15th International Technology, Education and Development Conference (INTED 2021), March 8–9th. 2021. P. 5288–5297. <https://doi.org/10.21125/inted.2021.1085>

25. *Невзорова О.А., Липачёв Е.К., Николаев К.С.* Семантические сервисы цифровой экосистемы OntoMath для математического образования // *Электронные библиотеки*. 2023. Т. 26. № 4. С. 538–564. <https://doi.org/10.26907/1562-5419-2023-26-4-538-569>

26. *Nikolaev K.S., Kirillovich A.V., Nevzorova O.A.* A corpus-based approach to elementary geometry knowledge test generation. // INTED2020 Proceedings. 2020. P. 6342–6348. <https://doi.org/10.21125/inted.2020.1710>

27. *Невзорова О.А., Николаев К.С.* Семантическое аннотирование математических формул в PDF-документах // *Электронные библиотеки*. 2022. Т. 25. № 6. С. 616–639. <https://doi.org/10.26907/1562-5419-2022-25-6-616-639>

28. *Николаев К.С.* Сервис генерации учебных карточек математических понятий для дистанционного курса по геометрии // *Электронные библиотеки*. 2023. Т. 26. № 3. P. 365–377. <https://doi.org/10.26907/1562-5419-2023-26-3-365-377>

29. *Nevzorova O.A., Kirillovich A.V., Nikolaev K.S., Galiaskarova K.R.* Towards a Semantically Annotated Corpus of Educational Mathematical Texts in Russian (short paper) // International Conference on Data Analytics and Management in Data Intensive Domains. 2019. P. 299–305. CEUR Workshop Proceedings. 2019. Vol. 2523. P. 299–305.

TAKING INTO ACCOUNT THE STRUCTURE OF THE DOCUMENT IN THE METHOD OF AUTOMATIC ANNOTATION OF MATHEMATICAL CONCEPTS IN EDUCATIONAL TEXTS

K. S. Nikolaev^[0000-0003-3204-238X]

*Supercomputer Modeling Laboratory of the Department of Supercomputer Systems and Parallel Computing of the Federal State Budgetary Institution "National Research Center "Kurchatov Institute"", Moscow;
Kazan (Volga Region) Federal University, Kazan.*

konnikolaeff@yandex.ru

Abstract

The enrichment of educational texts with semantic content (in particular, adding hyperlinks to the pages of the service that displays detailed information about concepts in the text) helps to increase the efficiency of students' assimilation of the material. The existing methods of semantic markup of educational texts do not take into account the structural features of such documents, which leads to excessive recognition of concepts. This article describes the development of the method of automatic annotation of mathematical concepts in educational mathematical texts by adding functionality to account for the structure of an educational document. The main purpose of the method is to process educational materials of the distance education course "Technology for solving planimetric problems". Following a single template when creating course pages allows you to apply an analysis of the web page markup and keywords used by the course creators. The main task in this process is to determine the type of table cell containing text fragments of educational materials. In accordance with the recommendations of the course creators, definitions should be highlighted in the cells containing the task statement, as well as in those blocks where the input data of the task is indicated. The type of table cells is determined by analyzing their attributes and searching for keywords in their contents. This limitation of recognizable text fragments will improve the student's perception of the course pages and improve the quality of learning.

Keywords: *semantic analysis, mathematical ontology, didactic relations, mathematical education, document markup.*

REFERENCES

1. *Zouri M., Ferworn A.* An Ontology-Based Approach for Curriculum Mapping in Higher Education // 2021 IEEE 11th Annu. Comput. Commun. Work. Conf. 2021. P. 141–147.
2. *Cheniti Belcadhi L.* Personalized feedback for self-assessment in lifelong learning environments based on semantic web // Comput. Human Behav. 2016. Vol. 55. P. 562–570.
3. *Florrence M.* Building a multilingual ontology for education domain using monto method // Comput. Sci. Inf. Technol. 2020.
4. *Ibrahim M.E., Yang Y., Ndzi D. L., Yang G., Al-Maliki M.* Ontology-Based Personalized Course Recommendation Framework // IEEE Access. 2019. Vol. 7. P. 5180–5199.
5. *Radovic M., Petrovic N., Tosic M.* An Ontology-Driven Learning Assessment Using the Script Concordance Test // Appl. Sci. 2022. Vol. 12, No. 3.
6. *Obeid C., Lahoud I., El Khoury H., Champin P. A.* Ontology-based Recommender System in Higher Education // Web Conf. 2018 – Companion World Wide Web Conf. WWW 2018. 2018. P. 1031–1034.
7. *Nikolaev K.S., Nevzorova O.A., Falileeva M.V.* Developing the OntoMathEdu ecosystem for educational applications // CEUR Workshop Proc. 2021. Vol. 2910. P. 81–87.
8. *Nevzorova O.A., Falileeva M.V., Lipachev E.K., Shakirova L.R., Dyupina A.E., Kirillovich A.V.* Obrazovatel'naya ontologiya OntoMathEdu: problemy` ontologicheskogo inzhiniringa // XX nacional'naya konferenciya po iskusstvennomu intellektu s mezhdunarodny`m uchastiem, KII-2022. Trudy` konferencii. V 2 t. Tom 1. M.: Izdatel'stvo ME`I, 2022. S. 177–188.
9. *Nevzorova O.A., Falileeva M.V., Kirillovich A.V., Lipachev E.K., Shakirova L.R., Dyupina A.E.* OntoMathEdu Educational Ontology: Problems of Ontological Engineering // Pattern Recognit. Image Anal. 2023. Vol. 33. P. 460–466.
<https://doi.org/10.1134/S1054661823030367>
10. *Dyupina A., Falileeva M.* Problems of Designing Polylingual Ontology OntoMathEdu // Communications in Computer and Information Science. 2020. Vol. 1242. P. 179–190. https://doi.org/10.1007/978-3-030-65218-0_14

11. *Shakirova L., Falileeva M., Kirillovich A., Lipachev E., Nevzorova O., Nevzorov V.* Modeling and Evaluation of the Mathematical Educational Ontology // SSI-2019 Scientific Services & Internet Proceedings of the 21st Conference on Scientific Services & Internet (SSI-2019) Novorossiysk-Abrau, Russia, September 23–28, 2019. CEUR Workshop Proceedings. 2019. Vol. 2543. P. 305–319.

12. *Shakirova L.R., Falileeva M.V., Kirillovich A.V., Lipachev E.K., Nevzorova O.A., Nevzorov V.N.* Obrazovatel'naya matematicheskaya ontologiya OntoMathEdu: struktura i otnosheniya // Nauchny`j servis v seti Internet: trudy` XXI Vserossijskoj nauchnoj konferencii (23–28 sentyabrya 2019 g., g. Novorossiysk). M.: IPM im. M.V. Keldy`sha. 2019. S. 653–661.

13. *Nevzorova O., Shakirova L., Falileeva M., Kirillovich A., Nevzorov V., Lipachev E.* OntoMathEdu Educational Mathematical Ontology: Annotation of Concepts // Selected Contributions of the "Russian Advances in Artificial Intelligence" Track at RCAI 2020 co-located with 18th Russian Conference on Artificial Intelligence (RCAI 2020). Moscow, Russia, October 10–16, 2020. Vol. 2648. P. 157–168.

14. *Kirillovich A., Nevzorova O., Falileeva M., Lipachev E., Shakirova L.* OntoMathEdu: A Linguistically Grounded Educational Mathematical Ontology // Intelligent Computer Mathematics. Series: Lecture Notes in Computer Science. 2020. Vol. 12236. P. 157–172. <https://doi.org/10.1007/978-3-030-53518-6>

15. *Kirillovich A., Nevzorova O., Falileeva M., Lipachev E., Shakirova L.* OntoMathEdu: Towards an Educational Mathematical Ontology // Edwin Brady et al. (Eds.). Workshop Papers at 12th Conference on Intelligent Computer Mathematics (CICM-WS 2019), Prague, Czech Republic, 8–12 July 2019. CEUR Workshop Proceedings. 2020. Vol. 2634.

16. *Shakirova L., Falileeva M., Kirillovich A., Lipachev E., Nevzorova O., Nevzorov V.* Modeling and Evaluation of the Mathematical Educational Ontology // Proceedings of the 21st Conference on Scientific Services & Internet (SSI-2019). CEUR Workshop Proceedings. Vol. 2543. P. 305–319.

17. *Shakirova L.R., Falileeva M.V., Kirillovich A.V., Lipachev E.K., Nevzorova O.A., Nevzorov V.N.* Obrazovatel'naya matematicheskaya ontologiya OntoMathEdu: struktura i otnosheniya // Nauchny`j servis v seti Internet. 2019. № 21. S. 653–661. <https://doi.org/10.20948/abrau-2019-84>

18. *Shakirova L.R., Falileeva M.V., Kirillovich A.V., Nevzorova O.A.* Proektirovanie obrazovatel'noj matematicheskoy ontologii: problemy` i metody` resheniya na primere kursa planimetrii // Trudy` mezhdunarodnoj konferencii po komp'yuternoj i kognitivnoj lingvistike: Materialy` XV mezhdunarodnoj konferencii po komp'yuternoj i kognitivnoj lingvistike TEL-2018 v 2-x tomax, Kazan. 2018. S. 393–405. EDN SQGRVY.

19. *Falileeva M., Kirillovich A., Shakirova L., Nevzorova O., Lipachev E., Dyupina A.* OntoMathEdu Educational Mathematical Ontology: Prerequisites, Educational Levels and Educational Projections // CEUR Workshop Proceedings. 2020. Vol. 2784. P. 346–351.

20. *Falileeva M., Kirillovich A., Hevzorova O.A., Shakirova L.R., Lipachev E.K., Dyupina A.E.* Educational projection systems, levels and prerequisites of mathematical ontology OntoMathEdu // Russian Digital Libraries. 2021. V. 24, No. 3. P. 505–530. <https://doi.org/10.26907/1562-5419-2021-24-3-505-530>

21. *Nevzorova O.A., Falileeva M.V., Kirillovich A.V., Shakirova L.R., Lipachev E.K., Dyupina A.E.* Modeling of didactic relationships in the OntoMathEdu educational mathematical ontology // CEUR Workshop Proceedings. 2021. V. 2910. P. 11–21.

22. *Falileeva M.V., Kirillovich A.V., Nevzorova O.A., Shakirova L.R., Lipachyov E.K., Dyupina A.E.* Razrabotka obrazovatel'ny`x proekcij matematicheskoy ontologii OntoMathEdu // Nauchny`j servis v seti Internet. 2020. № 22. S. 604–611. <https://doi.org/10.20948/abrau-2020-26>. EDN WBNWIC

23. *Kirillovich A., Falileeva M., Nevzorova O., Lipachev E., Dyupina A., Shakirova L.* Prerequisite Relationships of the OntoMathEdu Educational Mathematical Ontology // J.C. Figueroa-García et al. (Eds.). Proceedings of the 8th Workshop on Engineering Applications (WEA 2021). Communications in Computer and Information Science. 2021. Vol. 1431. P. 517–524. https://doi.org/10.1007/978-3-030-86702-7_44

24. *Nevzorova O., Shakirova L., Falileeva M., Nikolaev K.* Towards Ontology-Based Software Services for Teaching School Mathematics // L. Gómez Chova et al. (Eds.). Proceedings of the 15th International Technology, Education and Development Conference (INTED 2021), March 8–9, 2021. P. 5288–5297. <https://doi.org/10.21125/inted.2021.1085>

25. *Nevzorova O.A., Lipachev E.K., Nikolaev K.S.* Semantic services of the digital ecosystem OntoMath for mathematical education // *Russian Digital Libraries*. 2023. Vol. 26. No. 4. P. 538–564. <https://doi.org/10.26907/1562-5419-2023-26-4-538-569>

26. *Nikolaev K.S., Kirillovich A.V., Nevzorova O.A.* A corpus-based approach to elementary geometry knowledge test generation. // *INTED2020 Proceedings*. 2020. P. 6342–6348. <https://doi.org/10.21125/inted.2020.1710>

27. *Nevzorova O.A., Nikolaev K.S.* Semantic Annotation of Mathematical Formulas in PDF-Documents // *Russian Digital Libraries*. 2022. Vol. 25. No. 6. P. 616–639. <https://doi.org/10.26907/1562-5419-2022-25-6-616-639>

28. *Nikolaev K.S.* Service for generating educational cards of mathematical concepts for a distance learning course in geometry // *Russian Digital Libraries*. 2023. Vol. 26. No. 3. P. 365–377. <https://doi.org/10.26907/1562-5419-2023-26-3-365-377>

29. *Nevzorova O.A., Kirillovich A.V., Nikolaev K.S., Galiaskarova K.R.* Towards a Semantically Annotated Corpus of Educational Mathematical Texts in Russian (short paper) // *International Conference on Data Analytics and Management in Data Intensive Domains*. 2019. P. 299–305. *CEUR Workshop Proceedings*. 2019. Vol. 2523. P. 299–305.

СВЕДЕНИЯ ОБ АВТОРЕ



НИКОЛАЕВ Константин Сергеевич – к. т. н., младший научный сотрудник Лаборатории суперкомпьютерного моделирования Отделения суперкомпьютерных систем и параллельных вычислений ФГБУ «Национальный исследовательский центр «Курчатовский институт»». Основные направления научных исследований: обработка естественного языка, искусственный интеллект.

Konstantin Sergeevich NIKOLAEV – Cand. Sc. (Technology), Junior Researcher at the Supercomputer Modeling Laboratory of the Department of Supercomputer Systems and Parallel Computing of the Federal State Budgetary Institution "National Research Center "Kurchatov Institute"". Major fields of scientific research are Natural Language processing, artificial intelligence.

email: konnikolaeff@yandex.ru

ORCID: 0000-0003-3204-238X

Материал поступил в редакцию 1 марта 2024 года

Переработанная версия поступила в редакцию 11 июня 2024 года

УДК 004.415.25

ПОДХОД К СОЗДАНИЮ КОРПУСА ТЕКСТОВ ВИДЕОИГР НА ОСНОВЕ УНИВЕРСАЛЬНОЙ СТРУКТУРЫ

Н. Р. Нурлыгаянов¹ [0000-0002-7969-0221], В. В. Кугуракова² [0000-0002-1552-4910]

^{1, 2}Институт информационных технологий и интеллектуальных систем
Казанского федерального университета, ул. Кремлёвская, 35, г. Казань, 420008

¹nikita25@mail.ru, ²vlada.kugurakova@gmail.com

Аннотация

Рассмотрена проблема высокой и увеличивающейся стоимости разработки видеоигр, для её решения предложено применить процедурную генерацию контента, что позволит снизить затраты на разработку.

Работа является частью масштабного исследования по автоматическому созданию прототипов видеоигр и посвящена обработке игровых сценариев, то есть текстов на естественном языке. Предложено выделять из сценариев необходимые сущности и передавать их дальнейшим шагам алгоритма, который по текстовым описаниям будет генерировать игровые ресурсы.

Существует несколько публикаций, посвящённых обработке игровых текстов, в которых предложено несколько различных структур хранения выделенной информации. В настоящей статье предложен универсальный формат, который подойдёт для обработки текста любой видеоигры и позволит создать корпус текстов для использования в дальнейших исследованиях и автоматической генерации игровых прототипов.

Ключевые слова: PCG, NLP, разработка видеоигр.

ВВЕДЕНИЕ

В индустрии видеоигр существует проблема, связанная с тем, что бюджеты игр¹ быстро увеличиваются, поэтому разработка огромных и инновационных игр становится слишком дорогой (рис. 1) [1].



Рис. 1. Быстрое повышение стоимости разработки игр [1]

В последние годы игровая индустрия претерпела значительные изменения в своих бизнес-модели и подходах к разработке. Эти изменения во многом обусловлены экспоненциальным ростом затрат на создание игр, особенно в сегменте AAA²-проектов. Отчёт Управления по конкуренции и рынкам Великобритании [2] свидетельствует о том, что затраты на разработку новой видеоигры *Call of Duty*

¹ Перечень видеоигр, упоминаемых в этом исследовании, приведён в разделе Лудография.

² AAA (произносится как «трипл-эй») — неофициальный термин, используемый в игровой индустрии для обозначения игр с самыми высокими бюджетами разработки и маркетинга. Происхождение термина не имеет однозначного объяснения, но предполагается, что он возник по аналогии с системой оценки облигаций, где AAA обозначает наивысший рейтинг. В контексте видеоигр AAA указывает на наивысшее качество, масштаб и ожидаемый коммерческий успех проекта. AAA-игра характеризуется крупными инвестициями, франшизами, включающими несколько названий, множеством персонажей и длинными сюжетными линиями, а также большой базой поклонников.

достигли 300 миллионов долларов, а бюджет ожидаемой в 2025 году новой версии из известной серии³ видеоигр *Grand Theft Auto VI* (или *GTA6*) составляет минимум 250 миллионов долларов. Это резко контрастирует с бюджетом типичных AAA-игр пять лет назад, составлявшим от 50 до 150 миллионов долларов.

Чтобы справиться с затратами игровой разработки [1], растущими экспоненциально, игровые компании прибегают к различным стратегиям, таким как изменение моделей монетизации.

В последние годы наблюдается тенденция к фрагментации игрового контента: вместо полноценных расширений, как это было ранее (например, дополнения для *Half-Life*), сейчас чаще встречаются небольшие, короткие по игровому времени пакеты DLC⁴. Яркий пример такого подхода демонстрирует видеоигра *Payday 2*.

Кроме того, даже видеоигры, продающиеся по полной цене, часто включают микротранзакции [3] — операции во внутриигровом магазине для покупки цифровых товаров [4]. К числу спорных изменений последних лет относятся введение NFT⁵, а также повышение базовой цены на копию AAA-игры с \$60 до \$70 [5–8].

Чтобы снизить эти затраты, некоторые компании обращаются к методам машинного обучения для оптимизации процесса разработки. Например, в 2023 году компания Ubisoft⁶ представила *Ghostwriter* — инструмент, который генерирует контекстно-зависимые фразы для персонажей, сокращая время, которое нарративные дизайнеры тратят на стандартные диалоги, позволяя им сосредоточиться на важных диалогах [9]. Интересно отметить, что один из его создателей

³ GTA — серия видеоигр в жанре экшен-приключений от компании Rockstar Games.

⁴ DLC, сокр. от англ. Downloadable Content — дополнительный загружаемый контент.

⁵ NFT, сокр. от англ. Non-Fungible Token — невзаимозаменяемый токен, уникальный цифровой актив, представляющий право собственности на определённый виртуальный или физический объект, зарегистрированный в блокчейне. NFT позволяет подтвердить подлинность и уникальность цифровых предметов, таких как произведения искусства, игровые предметы или коллекционные объекты.

⁶ Ubisoft — французская компания, занимающаяся разработкой, изданием и дистрибуцией видеоигр. Основана в 1986 году пятью братьями Гийемо. Ubisoft известна своими популярными игровыми сериями, такими как, например, *Assassin's Creed*. Компания является одной из крупнейших в игровой индустрии и имеет множество студий и офисов по всему миру, занимаясь разработкой игр для различных платформ, включая консоли, ПК и мобильные устройства.

ранее работал над *AI Dungeon* — текстовой игрой, в которой используются модели GPT, в том числе GPT-2 и GPT-3 — для создания уникальных сюжетных линий на основе действий игрока [10]. Этот опыт демонстрирует преимущество в развитии технологий искусственного интеллекта (ИИ) для игровой индустрии — переход от экспериментальных проектов, к более специализированным инструментам.

Такая эволюция подчёркивает растущую роль ИИ в автоматизации различных аспектов игрового дизайна, от генерации диалогов до создания целых сюжетных линий.

Это, в свою очередь, открывает новые возможности для процедурной генерации контента (ПГК, PCG), позволяя разработчикам создавать более разнообразный и динамичный игровой опыт при потенциальном снижении затрат на разработку. В [11] рассматривалась возможность автоматического создания прототипа видеоигры на основе набора входных данных. Сделан вывод, что использование компьютерной генерации позволит сократить время, которое необходимо потратить специалистам в предметных областях, значит, и расходы на оплату их труда.

Настоящая статья посвящена описанию первого шага алгоритма автоматической генерации игровых прототипов, а именно, сбора сценариев видеоигр и разметки их для создания корпуса, который позже может быть использован для процедурной генерации контента.

БЛИЗКИЕ РАБОТЫ

Ван Стегерен и Тёне [12] предложили качественные критерии для корпусов видеоигровых текстов, описали источники игровых текстов и представили три корпуса с примерами использования. Корпус — хорошо организованный набор данных, предназначенный для того, чтобы исследовать определённый языковой признак (набор признаков) или обучать инструмент NLP [13].

Критериями качества набора данных обозначены:

- богатство — наборы данных должны содержать игровой текст и информацию о внутриигровом контексте,
- репрезентативность — тексты должны быть написаны профессионалами и получены из хорошо известных (коммерческих) игр,

- разнообразие — наборы данных должны отражать разнообразие видеоигр,
- переносимость — наборы данных должны храниться в текстовом формате, не требующем специальных инструментов для чтения и редактирования.

Мы предполагаем, что за счёт интеграции этих критериев и использования предложенной файловой структуры сценарии различных игр можно будет обрабатывать с использованием единого набора алгоритмов. Этот подход призван облегчить разработку методов процедурной генерации контента для видеоигр, если использовать методы обработки естественного языка для улучшения генерации уровней, персонажей и других игровых элементов.

В контексте создания корпусов для исследований в области видеоигр стоит отметить работу [14], в которой представлен значительный набор данных. Авторы этой статьи создали корпус, содержащий информацию о 755 играх вместе с их метаданными. Этот набор данных был разработан специально для содействия исследованиям по применению ИИ в сфере видеоигр. Важно отметить, что игры в этом корпусе были разработаны студентами, что отражено в названии набора данных. Это обстоятельство ставит под вопрос репрезентативность корпуса для всей индустрии видеоигр. Авторы статьи, однако, обосновывают ценность своего набора данных тем, что он отражает реальный опыт студентов в разработке игр. Тем не менее, для целей нашего исследования, которое сфокусировано на аспектах разработки видеоигр с точки зрения информатики (а не на психологических аспектах), вопрос репрезентативности остаётся критически важным. В рамках данной работы предпринимается попытка создать корпус, который бы более точно отражал общие тенденции и практики в индустрии видеоигр в целом.

Исследование [15] демонстрирует потенциал использования ИИ для генерации игрового контента. Авторы показали, что игровые задания со структурой **<название, задача, описание>** могут быть сгенерированы при помощи GPT-2, обученной на специально подготовленном наборе данных с заданиями. Для оценки качества сгенерированного контента был проведён онлайн-опрос. Результаты показали, что хотя сгенерированные задания не могут быть использованы в игре сразу после генерации, они представляют собой ценную основу, значи-

тельно упрощающую работу авторов игровых заданий. Важно отметить, что данное исследование было сфокусировано на одной игре — *World of Warcraft*, что ограничивает возможность прямого обобщения результатов на другие игры.

Работа [16] продолжает [15], в ней предложено генерировать базовые варианты заданий для ролевых игр в общем, а модели обучать на текстах пяти игр (*Baldur's Gate* части I и II, *The Elder Scrolls* части IV и V, *Torchlight* часть II). В [16] также предложен новый формат данных, ориентированный на задания.

Вернёмся к работе [12], рассмотрим предложенные в ней три набора данных — *Torchlight II quests*, *Knights of the Old Republic dialogue*, *The Elder Scrolls documents* (два из которых представлены в формате CSV и один в формате JSON, что не ограничивает их понимания, так как такие файлы можно открывать и редактировать с помощью любых привычных текстовых редакторов).

- *Torchlight II quests* имеет сложную иерархическую структуру (см. рис. 2), можно отметить, что имя вложенного поля в структуре описания диалога содержит его тип.

```
[
  {
    "name": "A2-after-Search_for_mana_guardian",
    "questrankmin": "21",
    "questrankmax": "35",
    "dialogpriority": "8",
    "autoassign": "true",
    "questcontrollercompletes": "true",
    "quest_guid": "3243555067944249791",
    "requirements": {
      "questscomplete": {
        "quest": "A2-SEARCH_FOR_MANA_GUARDIAN"
      }
    },
    "dialog": {
      "passive": [
```

Рис. 2. Пример *Torchlight II quests*

Этот набор данных предлагается использовать для сравнительного изучения основных и второстепенных заданий, исследования «вкусного (flavor)» текста

и, так как цель задания — это его резюме, для оценивания техник суммаризации в контексте видеоигр [17].

- Knights of the Old Republic dialogue состоит из следующих полей:
 - id;
 - speaker;
 - listener;
 - text;
 - animation;
 - comment;
 - next;
 - previous;
 - source_dlg;
 - audiofile.

Этот набор данных предлагается использовать для изучения диалогов и анализа тональности.

- Структура The Elder Scrolls documents:
 - url:
 - game;
 - url;
 - description;
 - author;
 - title;
 - text.

Первую версию этого набора данных использовали для адаптации анализатора тональности англоязычных текстов для учёта характерного для данной игровой серии лексикона [18].

Каждый из этих трёх наборов данных использует свою собственную структуру и хранит тексты для применения в различных целях. Более того, многие поля избыточны или затрудняют использование собранных данных. Например, поле `animation` избыточно, так как в игре может не быть анимации.

Таким образом, в [12] описаны три набора данных, которые нельзя считать сведёнными в корпус, так как они не имеют единой структуры. Для каждого

набора данных даны примеры применений, но они не обязательно совпадают или пересекаются.

ПРЕДЛАГАЕМЫЙ ПОДХОД

Мы предлагаем другую структуру (хранящуюся в файле формата JSON) для корпуса видеоигровых текстов, содержащую только данные, необходимые для ПГК и обобщённых исследований. Каждая запись внутриигрового текста рассматривается как часть диалога.

- id:
 - id_dialogue;
 - id_entry;
 - requires;
 - opens;
 - speaker;
 - text;
 - comment.

id — неотрицательное число, идентифицирующее определённую часть текста,

id_dialogue — неотрицательное число, идентифицирующее определённый диалог (отличие этого поля от предыдущего в том, что записи с уникальными id могут иметь одинаковый id_dialogue),

id_entry — неотрицательное число, идентифицирующее определённую часть диалога,

requires — список id: какие части текста игрок должен открыть, чтобы разблокировать данную часть,

opens — список id: какие части текста игрок разблокирует, открыв данную часть,

speaker — неотрицательное число, идентифицирующее персонажа (игрового или неигрового),

text — строка, содержащая «реплику»,

comment — внутриигровой контекст для данной записи.

Такая структура подходит для любого типа внутриигрового текста. Например, для текста на текстовых объектах (надписи, вывески, плакаты и т. п.), размещённых в игровом окружении, большая часть полей может быть оставлена пустой, в зависимости от игры задание может являться частью невырожденного диалога. Вырожденный диалог — любой текст, записанный в предложенном формате, не являющийся репликами между персонажами.

Также эта структура позволяет объединять тексты из любых разновидностей видеоигр. Применение предложенного подхода приведёт к созданию обобщённого корпуса, который можно использовать в дальнейших исследованиях.

Поля text и comment соответствуют критерию богатства [19], критерий переносимости [20] удовлетворяется хранением обработанных данных в файлах с форматом JSON.

АПРОБАЦИЯ ПРЕДЛАГАЕМОГО ПОДХОДА

Мы обработали тексты из игры *James Bond 007: From Russia with Love* (версии для PSP⁷), которая удовлетворяет критерию репрезентативности. Мы использовали тексты диалогов с контекстными комментариями [21] и видео прохождения [22] для разделения диалогов.

Несмотря на то, что экшен⁸-игры обычно содержат меньшее количество текста по сравнению с ролевыми, на этапе обработки сценария мы столкнулись с неожиданной сложностью в анализе и структурировании имеющегося текстового контента.

Анализируя игру после её прохождения и обладая полным представлением о её сюжете и персонажах, мы можем установить, что для всех реплик определённого персонажа следует использовать одно и то же значение speaker, даже

⁷ PSP— PlayStation Portable, портативный аналог домашней консоли.

⁸ Экшен (от англ. action — действие) — процесс активного взаимодействия игрока с игровым миром, включающий интенсивные действия, такие как сражения, погони или преодоление препятствий, часто являющийся ключевым элементом геймплея в играх различных жанров. (термин также используется для жанра, характеризующимся динамичным игровым процессом, требующим от игрока быстрой реакции и координации)

если в тексте этот персонаж упоминается под разными именами или псевдонимами. Разработка алгоритма для автоматического выполнения этого процесса идентификации и унификации говорящих является ключевым моментом с точки зрения применимости всего подхода к анализу игровых сценариев.

Был также проведён эксперимент со сценарием игры *Ведьмак 3* [23]. Производилось сопоставление текста происходящему в игре. Когда находилась значимая информация, она переносилась на диаграмму (рис. 3). Уже на этом, начальном, этапе эксперимента мы столкнулись с несколькими сложностями:

1. Неполная информация. Некоторые фразы, описывающие локацию и произносимые главным героем, отсутствовали в сценарии.
2. Вариации в именовании сущностей. Несколько названий могут называть одну и ту же сущность.

Ранее такое встречалось во время обработки *James Bond 007: From Russia with Love*. Дополнительные имена не обязательно могут быть краткими формами основного имени. Например, «*one of the sorceress' conjured crustacean-like creatures*» и «*crab*» — это одно и то же существо.

3. Недостаточное понимание контекста. Без знания игры некоторые части сценария могут быть истолкованы неправильно.

Например, после первого видео на локации Kaer Morhen игрок получает контроль над персонажем. Можно взаимодействовать с объектами в первой комнате или вступить в диалог с другим персонажем. В сценарии представлено краткое описание «*If Geralt inspects things in Yen's room:*». Далее идут только реплики, которыми обмениваются персонажи. Информация о названиях объектов или том, что персонаж делает в данный момент, отсутствует.

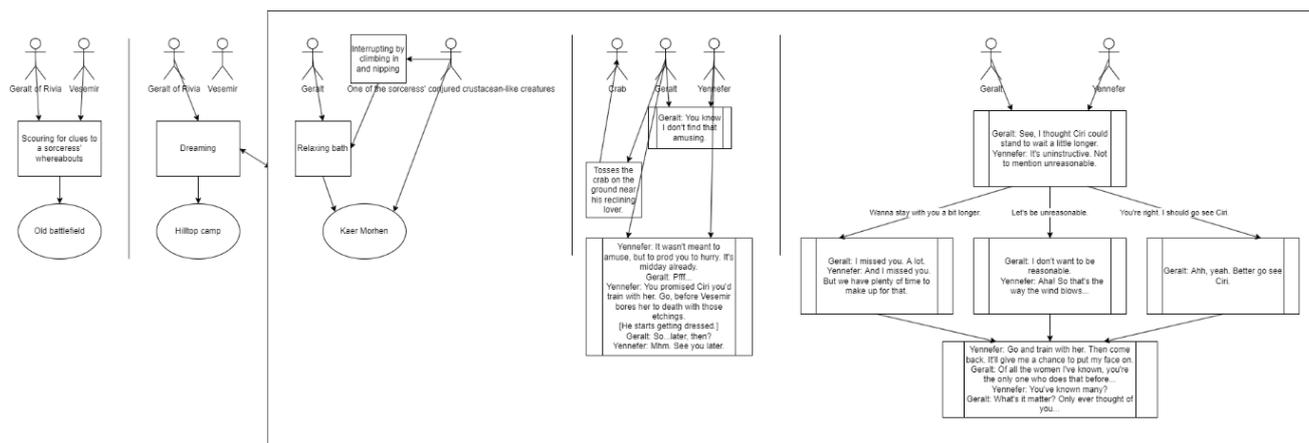


Рис. 3. Схема начала сценария игры *Ведьмак 3: Дикая Охота*

Перечисленные трудности подчёркивают важность понимания контекста и потребность в устойчивых алгоритмах, которые способны разбираться в вариациях наименований сущностей и извлекать точную информацию из сценариев игр. Несмотря на выявленные сложности, предложенный подход показал свою перспективность для создания универсального корпуса текстов видеоигр.

Однако для его успешной реализации требуется разработка дополнительных алгоритмов и методов обработки текста. Для преодоления выявленных трудностей необходимо:

- разработать алгоритмы для автоматической идентификации и унификации именования персонажей и других игровых сущностей,
- создать методы извлечения контекстной информации из игровых сценариев,
- исследовать возможности применения методов машинного обучения для улучшения понимания контекста игры.

Апробация предложенного подхода на реальных игровых сценариях позволила выявить ключевые проблемы в создании универсального корпуса текстов видеоигр. Полученные результаты демонстрируют необходимость дальнейшего совершенствования методов обработки игровых текстов и подчеркивают важность разработки специализированных алгоритмов для работы с игровым контентом. Преодоление выявленных трудностей откроет новые возможности для автоматизации процессов разработки игр и проведения исследований в области игровой индустрии.

ЗАКЛЮЧЕНИЕ

В рамках продолжающегося исследовательского проекта (которому посвящены работы [11], [24–26] и ряд других) ведётся работа над решением общей задачи — генерацией видеоигровых прототипов на основе набора данных, в том числе текстовых (сценариев компьютерных игр). Эти прототипы должны быть такими, чтобы их можно было оценить и изменить специалистам.

В частности, в [11] было показано, что возможно генерировать раскадровки с использованием текста в качестве входных данных, поэтому некоторые из следующих логичных шагов — это создание коллекции видеоигровых текстов и обработка их для создания корпуса.

Будущие исследования будут в первую очередь сосредоточены на разработке алгоритма, который станет начальной стадией запланированного конвейера создания прототипов. Этот алгоритм должен отвечать следующим требованиям:

1. Принимать сценарии видеоигр в качестве входных данных.
2. Создавать структурированные файлы JSON, содержащие все внутриигровые тексты, отформатированные в соответствии с предложенной нами структурой файлов.
3. Включать такие методы, как распознавание именованных сущностей (Named Entity Recognition), связывание именованных сущностей (Named Entity Linking) и нормализация именованных сущностей (Concept Normalization), для решения выявленных в ходе эксперимента проблем. Однако важно отметить, что этого может оказаться недостаточно, поэтому мы признаём необходимость дальнейших исследований и поиска альтернативных решений.

После разработки алгоритма исследование будет продолжено в нескольких направлениях:

1. расширение корпуса добавлением обработанного текста из игр разных жанров для удовлетворения критерия разнообразия,
2. использование корпуса для ПГК,
3. извлечение информации из поля комментариев (comment) для получения информации о локациях, предметах и событиях.

Предложенный подход и дальнейшие исследования в этой области имеют потенциал значительно упростить и ускорить процесс разработки видеоигр, что особенно актуально в контексте растущей сложности и стоимости игровых проектов.

Благодарности

Работа выполнена за счёт средств Программы стратегического академического лидерства Казанского (Приволжского) федерального университета («ПРИОРИТЕТ-2030»).

ЛУДОГРАФИЯ⁹

Bethesda Game Studios. (2007)[2006.] *The Elder Scrolls IV: Oblivion*. [видеоигра][загрузка][Windows, Xbox 360, PlayStation 3]. Bethesda Softworks, 2K, Ubisoft.

Bethesda Game Studios. (2022)[2011.] *The Elder Scrolls V: Skyrim*. [видеоигра][загрузка][Windows, Xbox 360, PlayStation 3, Xbox One, PlayStation 4, Switch, Xbox Series X|S, PlayStation 5]. Bethesda Softworks.

BioWare Edmonton. (2021) [2003.] *Star Wars: Knights of the Old Republic*. [видеоигра][загрузка][Xbox, Windows, macOS, iOS, Android, Switch]. LucasArts.

BioWare, Overhaul Games. (2019) [1998.] *Baldur's Gate*. [видеоигра][загрузка][Windows, macOS, iOS, Android, Linux, Switch, PlayStation 4, Xbox One]. Black Isle Studios, Interplay Entertainment, GraphSim Entertainment.

BioWare. (2019) [2000.] *Baldur's Gate II: Shadows of Amn*. [видеоигра][загрузка][Windows, macOS, iOS, Android, Linux, Switch, PlayStation 4, Xbox One]. Black Isle Studios, Interplay Entertainment.

⁹ Лудография (от лат. *ludus* — «игра» и греч. *graphō* (γράφω) — «пишу») — список игр, упоминаемых в тексте, который отделён от библиографии. В лудографию должны быть включены все игры, упомянутые в тексте, даже если они не являются основным объектом исследования. Здесь описание каждой игры представлено в формате: <Авторский корпус. (год публикации)[год первой публикации]. *название игры* (версия).[тип игры][поддержка][платформа], издатель.> Формат выбран, исходя из рекомендаций <http://semionerd.blogspot.com/p/how-to-write-ludography.html>. Отметим, что использование термина может варьироваться в зависимости от конкретного исследования и контекста. В некоторых работах «ludography» может использоваться аналогично библиографии, но для списка игр, в то время как в других он может относиться к более глубокому анализу игрового процесса.

Blizzard Entertainment. (2024) [2004.] *World of Warcraft*. [видеоигра][загрузка][Windows, macOS]. Blizzard Entertainment.

CD Projekt Red. (2022) [2015.] *Ведьмак 3: Дикая Охота*. [видеоигра][загрузка][Windows, PlayStation 4, Xbox One, Switch, PlayStation 5, Xbox Series X|S]. CD Projekt.

Latitude. (2021) [2019.] *AI Dungeon*. [видеоигра][онлайн][браузер]. Latitude.

Rebellion Developments. (2006) [2006.] *James Bond 007: From Russia with Love*. [видеоигра][UMD][PlayStation Portable]. Electronic Arts, MGM Interactive.

Rockstar Games. (2025) [2025.] *Grand Theft Auto VI*. [видеоигра][загрузка][PlayStation 5, Xbox Series X|S]. Rockstar Games.

Runic Games. (2019) [2012.] *Torchlight II*. [видеоигра][загрузка][Windows, macOS, Linux, Switch, PlayStation 4, Xbox One]. Runic Games.

Starbreeze Studios, Overkill Software, Sumo Digital. (2023) [2013.] *Payday 2*. [видеоигра][загрузка][PlayStation 4, Switch, PlayStation 3, Xbox 360, Xbox One, Windows, Linux, GeForce Now]. Starbreeze Studios, 505 Games.

Treyarch, Raven Software. (2024) [2024.] *Call of Duty: Black Ops 6*. [видеоигра][загрузка][Windows, PlayStation 5, Xbox Series X|S, PlayStation 4, Xbox One]. Activision.

Valve Corporation, Gearbox Software. (2004) [1998.] *Half-Life*. [видеоигра][загрузка, CD][Windows, PlayStation 2, Linux, macOS]. Valve Corporation, Sierra Studios.

СПИСОК ЛИТЕРАТУРЫ

1. Art Direction Bootcamp: Building a Creative Future with Artificial Intelligence. URL: <https://youtu.be/9FAXAgRrOSE?t=1737>.

2. Hu J., Wu C. Microsoft's Acquisition of Activision Blizzard: A Case Study // Highlights in Business, Economics and Management. 2023. Vol. 7. P. 363–371.

3. Behuria S. Microtransactions as a Business Model in Video Gaming Industry: Its Comparison with Traditional Model and Effect on Other Industries // Available at SSRN 4418003. 2022.

4. *Mahendra F. A., Jayadi R., Halim E.* Examining Drivers of Urge to Purchase in Video Game Microtransactions // Journal of System and Management Sciences. 2024. Vol. 14. No. 5. P. 131–147.

5. *Tavares R. et al.* Gamers' Reaction to the Use of NFT in AAA Video Games // Procedia Computer Science. 2023. Vol. 219. P. 606–613.

6. Exclusive: Ubisoft CEO speaks.

URL: <https://www.axios.com/newsletters/axios-gaming-534001ff-b32e-4a65-92c7-afb79f52dc13.html>.

7. Microsoft Raising Prices on New, First-Party Games Built for Xbox Series X|S to \$70 in 2023.

URL: https://www.ign.com/articles/microsoft-raising-prices-new-first-party-games-xbox-series-70-2023-redfall-starfield?utm_source=facebook,twitter.

8. 2K breaks gaming's de facto \$60 ceiling, asks \$70 for next-gen NBA 2K21.

URL: <https://arstechnica.com/gaming/2020/07/will-nba-2k21s-70-price-set-a-new-next-gen-console-standard/>.

9. The Convergence of AI and Creativity: Introducing Ghostwriter.

URL: <https://news.ubisoft.com/en-us/article/7Cm07zbBGy4Xm16WgYi25d/the-convergence-of-ai-and-creativity-introducing-ghostwriter>.

10. AI Dungeon. URL: <https://aidungeon.com/>.

11. *Sahibgareeva G.F., Bedrin O., Kugurakova V.* Visualization Component for the Scenario Prototype Generator as a Video Game Development Tool // CEUR Workshop Proceedings. 2020. Vol. 2784. P. 267–282.

12. *van Stegeren J., Theune M.* Fantastic strings and where to find them: The quest for high-quality video game text corpora // 12th Intelligent Narrative Technologies Workshop, INT 2020. — CEUR, 2021.

13. *Mitkov R. (ed.)*. The Oxford handbook of computational linguistics. Oxford university press, 2022. Vol. 1. 329 p.

14. *Anderson B.R., Smith A.M.* The Undergraduate Games Corpus: A Dataset for Machine Perception of Interactive Media // Proceedings of the AAAI Conference on Artificial Intelligence. 2021. Vol. 35. No. 1. P. 3–11.

15. *van Stegeren J., Myśliwiec J.* Fine-tuning GPT-2 on annotated RPG quests for NPC dialogue generation // Proceedings of the 16th International Conference on the Foundations of Digital Games. 2021. P. 1–8.

16. *Värtinen S., Hämmäläinen P., Guckelsberger C.* Generating Role-Playing Game Quests with GPT Language Models // IEEE Transactions on Games. 2022.

17. *Summerville A. et al.* Procedural Content Generation via Machine Learning (PCGML) // IEEE Transactions on Games. 2018. Vol. 10. No. 3. P. 257–270.
<https://doi.org/10.1109/TG.2018.2846639>.

18. *Bergsma T., van Stegeren J., Theune M.* Creating a Sentiment Lexicon with Game-Specific Words for Analyzing NPC Dialogue in The Elder Scrolls V: Skyrim // Workshop on Games and Natural Language Processing. 2020. P. 1–9.

19. *Слива М.В.* Стратегии адаптивного построения в компьютерных играх // Компьютерные инструменты в образовании. 2013. № 3. P. 31–43.
<https://doi.org/10.32603/2071-2340-2013-0-3-31-43>

20. *Суворова Т.Н., Rogozin К.И.* Подходы к оценке обучающих компьютерных игр // Вестник Оренбургского государственного педагогического университета. Электронный научный журнал. 2020. № 4. С. 380–394.
<https://doi.org/10.32516/2303-9922.2020.36.26>

21. From Russia with Love – Game Script – PSP – By ironyisntdead.
URL: <https://gamefaqs.gamespot.com/psp/929602-from-russia-with-love/faqs/75458>.

22. James Bond 007: From Russia with Love PSP Playthrough - With The Best Bond Ever, Sean Connery. URL: <https://youtu.be/dWckl9yd71M>.

23. Game Script (PS4) by Shotgunnova.
URL: <https://gamefaqs.gamespot.com/ps4/702760-the-witcher-3-wild-hunt/faqs/73353>.

24. *Sahibgareeva G.F., Kugurakova V.V.* Branched Structure Component for a Video Game Scenario Prototype Generator // CEUR Workshop Proceedings. 2021. Vol. 3066. P. 101–111.

25. *Сахибгареева Г.Ф., Кугуракова В.В.* Редактор интерактивной структуры для инструмента генерации сценарных прототипов // Электронные библиотеки. 2022. Т. 24, № 6. С. 1184–1202.
<https://doi.org/10.26907/1562-5419-2022-24-6-1184-1202>

26. Сахибгареева Г.Ф., Кугуракова В.В., Большаков Э.С. Инструменты балансирования игр // Электронные библиотеки. 2023. Т. 26, № 2. С. 225–251.

<https://doi.org/10.26907/1562-5419-2023-26-2-225-251>.

A NEW APPROACH TO CREATING A CORPUS OF VIDEO GAME TEXTS

N. R. Nurlygaianov¹ [0000-0002-7969-0221], V. V. Kugurakova² [0000-0002-1552-4910]

^{1,2}*Institute of Information Technologies and Intelligent Systems, Kazan (Volga Region) Federal University, Kremlevskaia st., 35, Kazan, 420008*

¹nikita25@mail.ru, ²vlada.kugurakova@gmail.com

Abstract

The problem of high and increasing cost of video game development is considered, and to solve it is proposed to apply procedural content generation, which will reduce development costs.

The work is a part of a large-scale research on automatic prototyping of video games and is devoted to the processing of game scenarios, i.e. natural language texts. It is proposed to extract the necessary entities from the scripts and pass them to further steps of the algorithm, which will generate game resources based on the textual descriptions.

There are several publications devoted to game text processing, in which several different structures for storing the extracted information are proposed. In this paper we propose a universal format that is suitable for processing the text of any video game and allows to create a corpus of texts for use in further research and automatic generation of game prototypes.

Keywords: PCG, NLP, game development.

REFERENCES

1. Art Direction Bootcamp: Building a Creative Future with Artificial Intelligence. URL: <https://youtu.be/9FAXAgRrOSE?t=1737>.
2. Hu J., Wu C. Microsoft's Acquisition of Activision Blizzard: A Case Study // Highlights in Business, Economics and Management. 2023. Vol. 7. P. 363–371.

3. *Behuria S.* Microtransactions as a Business Model in Video Gaming Industry: Its Comparison with Traditional Model and Effect on Other Industries // Available at SSRN 4418003. 2022.

4. *Mahendra F. A., Jayadi R., Halim E.* Examining Drivers of Urge to Purchase in Video Game Microtransactions // Journal of System and Management Sciences. 2024. Vol. 14. No. 5. P. 131–147.

5. *Tavares R. et al.* Gamers' Reaction to the Use of NFT in AAA Video Games // Procedia Computer Science. 2023. Vol. 219. P. 606–613.

6. Exclusive: Ubisoft CEO speaks.
URL: <https://www.axios.com/newsletters/axios-gaming-534001ff-b32e-4a65-92c7-afb79f52dc13.html>.

7. Microsoft Raising Prices on New, First-Party Games Built for Xbox Series X|S to \$70 in 2023.

URL: https://www.ign.com/articles/microsoft-raising-prices-new-first-party-games-xbox-series-70-2023-redfall-starfield?utm_source=facebook,twitter.

8. 2K breaks gaming's de facto \$60 ceiling, asks \$70 for next-gen NBA 2K21.
URL: <https://arstechnica.com/gaming/2020/07/will-nba-2k21s-70-price-set-a-new-next-gen-console-standard/>.

9. The Convergence of AI and Creativity: Introducing Ghostwriter.
URL: <https://news.ubisoft.com/en-us/article/7Cm07zbBGy4Xml6WgYi25d/the-convergence-of-ai-and-creativity-introducing-ghostwriter>.

10. AI Dungeon. URL: <https://aidungeon.com/>.

11. *Sahibgareeva G.F., Bedrin O., Kugurakova V.* Visualization Component for the Scenario Prototype Generator as a Video Game Development Tool // CEUR Workshop Proceedings. 2020. Vol. 2784. P. 267–282.

12. *van Stegeren J., Theune M.* Fantastic strings and where to find them: The quest for high-quality video game text corpora // 12th Intelligent Narrative Technologies Workshop, INT 2020. — CEUR, 2021.

13. *Mitkov R. (ed.)*. The Oxford handbook of computational linguistics. Oxford university press, 2022. Vol. 1. 329 p.

14. *Anderson B.R., Smith A.M.* The Undergraduate Games Corpus: A Dataset for Machine Perception of Interactive Media // Proceedings of the AAAI Conference on Artificial Intelligence. 2021. Vol. 35. No. 1. P. 3–11.

15. *van Stegeren J., Myśliwiec J.* Fine-tuning GPT-2 on annotated RPG quests for NPC dialogue generation // Proceedings of the 16th International Conference on the Foundations of Digital Games. 2021. P. 1–8.

16. *Värtinen S., Hämäläinen P., Guckelsberger C.* Generating Role-Playing Game Quests with GPT Language Models // IEEE Transactions on Games. 2022.

17. *Summerville A. et al.* Procedural Content Generation via Machine Learning (PCGML) // IEEE Transactions on Games. 2018. Vol. 10. No. 3. P. 257–270.
<https://doi.org/10.1109/TG.2018.2846639>.

18. *Bergsma T., van Stegeren J., Theune M.* Creating a Sentiment Lexicon with Game-Specific Words for Analyzing NPC Dialogue in The Elder Scrolls V: Skyrim // Workshop on Games and Natural Language Processing. 2020. P. 1–9.

19. *Sliva M.V.* Adaptive Construction Strategies in Computer Games [Strategii adaptivnogo postroeniya v komp'yuternykh igrakh]. Computer Tools in Education [Komp'yuternye instrumenty v obrazovanii]. 2013. № 3. S. 31–43. (In Russian).
<https://doi.org/10.32603/2071-2340-2013-0-3-31-43>

20. *Suvorova T.N., Rogozin K.I.* Approaches to the Evaluation of Educational Computer Games [Podkhody k otsenke obuchayushchikh komp'yuternykh igr]. Bulletin of Orenburg State Pedagogical University. Electronic Scientific Journal [Vestnik Orenburgskogo gosudarstvennogo pedagogicheskogo universiteta. Elektronnyy nauchnyy zhurnal]. 2020. №. 4. S. 380–394 (In Russian).
<https://doi.org/10.32516/2303-9922.2020.36.26>

21. From Russia with Love – Game Script – PSP – By ironyisntdead.
URL: <https://gamefaqs.gamespot.com/psp/929602-from-russia-with-love/faqs/75458>.

22. James Bond 007: From Russia with Love PSP Playthrough – With The Best Bond Ever, Sean Connery. URL: <https://youtu.be/dWckl9yd71M>.

23. Game Script (PS4) by Shotgunnova.
URL: <https://gamefaqs.gamespot.com/ps4/702760-the-witcher-3-wild-hunt/faqs/73353>.

24. *Sahibgareeva G.F., Kugurakova V.V.* Branched Structure Component for a Video Game Scenario Prototype Generator // CEUR Workshop Proceedings. 2021. Vol. 3066. P. 101–111.

25. *Sahibgareeva G.F., Kugurakova V.V.* Interactive Structure Editor for a Scenario Prototype Generation Tool [Redaktor interaktivnoy struktury dlya instrumenta generatsii stsenarnykh prototipov]. Electronic Libraries [Elektronnye biblioteki]. 2022. Vol. 24, No. 6. P. 1184–1202. (In Russian). DOI: 10.26907/1562-5419-2022-24-6-1184-1202

26. *Sahibgareeva G.F., Kugurakova V.V., Bolshakov E.S.* Game Balancing Tools [Instrumenty balansirovaniya igr]. Electronic Libraries [Elektronnye biblioteki]. 2023. Vol. 26, No. 2. P. 225–251 (In Russian).

<https://doi.org/10.26907/1562-5419-2023-26-2-225-251>.

СВЕДЕНИЯ ОБ АВТОРАХ



НУРЛЫГАЯНОВ Никита Рамильевич — ассистент кафедры программной инженерии ИТИС КФУ. Сфера научных интересов — автоматизация процесса разработки компьютерных игр.

Nikita Ramilevich NURLYGAIANOV — Assistant of the Software Engineering Department of ITIS KFU. Research interest is automation of the computer game development process.

email: nikita25@mail.ru

ORCID: 0000-0002-7969-0221



КУГУРАКОВА Влада Владимировна — доцент, к. т. н., и. о. зав. кафедрой индустрии разработки видеоигр ИТИС КФУ. Сфера научных интересов — различные аспекты проектирования игр.

Vlada Vladimirovna KUGURAKOVA — Associate Professor, acting head of the Video Game Development Industry Chair of ITIS KFU. Research interests include various aspects of game design.

email: vlada.kugurakova@gmail.com

ORCID: 0000-0002-1552-4910

Материал поступил в редакцию 10 июня 2024 года

УДК 004.8

НЕЙРОСЕТЕВАЯ АРХИТЕКТУРА ВОПЛОЩЕННОГО ИНТЕЛЛЕКТА

А. Р. Нурутдинов^[0009-0007-0918-0879]

ПАО "Таттелеком", ул. Ершова, 57, г. Казань, 420061

ayrat.nurutdinov@gmail.com

Аннотация

В последние годы достижения в области искусственного интеллекта (ИИ) и машинного обучения обусловлены успехами в разработке больших языковых моделей (LLM) на основе глубоких нейронных сетей. В то же время, несмотря на существенные возможности, LLM имеет такие принципиальные ограничения, как спонтанная недостоверность в фактах и суждениях; допущение простых ошибок, диссонирующих с высокой компетентностью в целом; легковерие, проявляющееся в готовности принимать за истину заведомо ложные утверждения пользователя; отсутствие сведений о событиях, произошедших после завершения обучения.

Вероятно, ключевой причиной является то, что обучение биологического интеллекта происходит через усвоение неявных знаний воплощенной формой интеллекта, позволяющей решать интерактивные физические задачи реального мира. Биоинспирированные исследования нервных систем организмов позволяют рассматривать мозжечок, координирующий движения и поддерживающий равновесие, в качестве главного кандидата для раскрытия методов реализации воплощенного физического интеллекта. Его простая повторяющаяся структура и способность управлять сложными движениями дают надежду на возможность создания аналога адаптивным нейронным сетям.

В настоящей работе изучается биоинспирированная архитектура мозжечка как форма аналоговых вычислительных сетей, способная моделировать сложные физические системы реального мира. В качестве простого примера представлена реализация воплощенного ИИ в виде многокомпонентной модели щупальца осьминога, демонстрирующей потенциал в создании адаптивных физических систем, обучающихся и взаимодействующих с окружающей средой.

Ключевые слова: Искусственные нейронные сети, большие языковые модели, неявное обучение, мозжечок, аналоговые компьютеры, воплощенный интеллект, мягкие роботы, осьминоги.

1. ВВЕДЕНИЕ

В последние годы в области искусственного интеллекта (AI) и машинного обучения значительные достижения были получены благодаря развитию больших языковых моделей (LLM), построенных на глубоких нейронных сетях. Исследователи из Microsoft Research пришли к выводам, что ChatGPT-4o можно рассматривать как раннюю версию системы общего искусственного интеллекта (AGI) [1]. В то же время, несмотря на существенные возможности, LLM имеет значимые ограничения [2], такие как: спонтанная недостоверность в фактах и суждениях; допущение простых ошибок, диссонирующих с высокой компетентностью в целом; легкое доверие, проявляющееся в готовности принимать за истину заведомо ложные утверждения пользователя; отсутствие сведений о событиях, произошедших после завершения обучения. Однако названные часто упоминаемые недостатки не так фундаментальны, как следующие четыре фактора:

1. Обучение наиболее продвинутых LLM требует чрезвычайно объема вычислительной мощности. По данным, предоставленным OpenAI, для обучения GPT-3 с параметрами 175B потребовалось 3640 petaFLOP/s-days [3]. Согласно сведениям проекта TOP500 [4], ранжирующего 500 самых мощных нераспределенных компьютерных систем в мире, суперкомпьютер Eagle компании Microsoft, предоставляющей облачную платформу Azure для обучения моделей OpenAI, имеет вычислительную мощность R_{max} , равную 561,2 petaFlop/s. Таким образом, для разового GPT-3 обучения необходимо шесть дней непрерывной работы Eagle только над этой задачей. Если говорить о модели GPT-4, количество параметров которой не разглашается, но явно на порядок больше, можно предположить, что необходимы месяцы работы третьего по производительности суперкомпьютера в мире. Очевидно, что увеличение параметров еще на порядок потребует годы на создание следующего поколения LLM.

2. Дальнейшее развитие языковых моделей требует все большего размера очищенных текстовых данных, качественный объем которых ограничен. Согласно опубликованным данным, для обучения GPT-3 уже использовался

чрезвычайно большой набор данных, основная часть которого была получена из Common Crawl – базы, собравшей за последние 17 лет более 250 млрд страниц. В обучающую выборку также были добавлены известные корпуса высокого качества, включая расширенную версию набора данных WebText, коллекции книг в интернете (Books1 и Books2) и англоязычную «Википедию» (см. Таблица 1).

Таблица 1.

<i>Набор данных</i>	<i>Количество (токены)</i>	<i>Вес в обучении</i>
Common Crawl (фильтрованный)	410 миллиардов	60 %
WebText2	19 миллиардов	22 %
Books1	12 миллиардов	8 %
Books2	55 миллиардов	8 %
Wikipedia	3 миллиарда	3 %
ВСЕГО	499 миллиардов	

С увеличением размеров моделей искусственного интеллекта, таких как GPT-4, возрастает и потребность в большем объеме обучающих данных. При этом остро встает вопрос очистки данных, так как имеющиеся масштабные наборы часто содержат существенную часть избыточной или нерелевантной информации. Кроме того, все больше возникает претензий о нарушении авторских прав. Например, The New York Times предъявила иск, в котором утверждает, что компания OpenAI использовала миллионы статей этой газеты для обучения, что привело к уменьшению числа подписчиков издания [5].

Также следует отметить, что многочисленные исследования подтверждают значительное влияние социальных сетей на процесс обмена информацией [6]. Эти сети играют важнейшую роль в накоплении и хранении знаний, а также в предоставлении возможности находить и получать к ним доступ. Хотя точные цифры не раскрываются, можно предположить, что данные из социальных сетей составляют значимую, но не главную часть обучающей информации для моделей GPT, а основная масса данных по-прежнему вынуждено берется из более традиционных источников, таких как веб-страницы и научные статьи. При этом стоит отметить, что большая часть информации на таких платформах, как

Facebook, Instagram и Telegram, вероятно, носит приватный характер и не может быть получена и использована сторонними лицами для обучения.

Все это приводит к тому, что расширение обучающих данных осуществляется за счет мультимедиа контента, который имеет гораздо более низкую плотность информации, и это явно требует существенного роста как размера моделей, значит, так и затрат на их обучение.

3. Неспособность к непрерывному обучению на своем опыте. LLM не адаптирует и не изменяет свое поведение на основе взаимодействия с конкретными пользователями или их данными, то есть генерирует ответы, основываясь исключительно на обучающем наборе данных и внутренних алгоритмах, но не обновляет свои знания или не учитывает новую информацию от пользователей после завершения обучения. Способность к дообучению определяется только размером *контекстного окна (Input Context Window)*, соответствующего количеству токенов, которые модель может рассматривать одновременно при выполнении задачи. Окно является ограничением области входных данных, на которой модель может сосредоточить внимание, чтобы вывести нужные закономерности или сгенерировать ответ. В таблице 2 представлено максимальное количество токенов в запросе, заявленное разработчиками.

Таблица 2.

Модель	Дата выпуска	Контекстное окно
GPT-3.5 Turbo	28.11.2022	4k токенов
GPT-4	14.03.2023	8k токенов
GPT-4o	13.05.2024	128k токенов
Gemini 1.5 Pro	15.02.2024	8k токенов
Claude 3 Haiku	13.03.2024	200k токенов
Llama 3 70B Instruct	18.04.2024	8k токенов

Мы видим стремление разработчиков максимально увеличить размер контекстного окна, однако даже текущие размеры крайне ограничены; для примера, оценочно роман «Война и мир» Л. Толстого содержит около 750k токенов. Кроме того, обучение модели изменяет фундаментальные параметры модели –

вес, а контекстное окно – только входные параметры, на основе которых генерируется ответ, то есть увеличение контекстного окна нельзя рассматривать как полноценную, пусть даже локальную, замену дообучения.

Эти ограничения значительно сужают спектр возможного использования LLM, ограничивая их в основном энциклопедическими знаниями общего характера. В отличие от LLM, традиционные поисковые системы, несмотря на их меньшую структурированность, предоставляют более актуальную информацию о текущих событиях. В результате LLM не могут в большинстве сфер интеллектуальной деятельности конкурировать с человеком, который способен непрерывно обновлять свои знания в реальном времени.

4. Исходный состав данных для обучения не содержит огромный пласт неявных знаний. Модели GPT используют архитектуры трансформеров, которые позволяют им эффективно анализировать и обрабатывать тексты, экстрагируют знания. Процесс обучения включает в себя обработку огромного количества текстов, что позволяет этим моделям накапливать знания, по сути схожие с базами знаний в традиционных экспертных системах, ранее основанных на алгоритмах. Одной из ключевых характеристик LLM является их SelfIE (Self-Interpretation of Embeddings) – способность строить внутренние представления и логические связи между различными концепциями [7]. В процессе обучения модели не только запоминают информацию, но и создают внутренние сложные иерархии данных, что позволяет им делать выводы на основе контекста и генерировать осмысленные тексты [8]. Этот процесс аналогичен тому, как экспертные системы используют правила логического вывода для принятия решений. Таким образом, LLM представляют собой своего рода мощные экспертные системы, самонастраивающиеся в процессе обучения, которые не только накапливают знания на основе огромных объемов предоставленных текстовых данных, но и формируют внутренние алгоритмы логических связей между различными концепциями.

Однако использование для обучения предварительно уже сконцентрированных и структурированных знаний, сформированных исходя из представления определенных групп человечества об их значимости, порождает труднопреодолимые проблемы. В частности, неравномерное представление демографических и культурных групп, доминирование определенных языков и идеологий, а

также существующие стереотипы и эмоциональные ассоциации в обучающих данных порождают алгоритмическую предвзятость, приводящую к наследованию и усилению предвзятостей, присутствующих в данных, на которых эти модели обучаются [9].

Гораздо большим ограничением является проблема неявных знаний, то есть знаний, которые люди используют интуитивно и не могут объяснить или формализовать [10]. Такие знания включают интуитивные навыки и опыт, которые сложно передать ИИ. В первую очередь это физические и интерактивные способности к взаимодействию с окружающим миром, то есть те навыки, которые формируются у человека в силу биологических особенностей или в первые годы жизни и плохо или совсем не осознаются им. Это приводит к тому, что LLM не обладают истинным пониманием мира. В результате, несмотря на способность генерировать правдоподобные тексты, модели LLM спонтанно дают нелогичные или бессмысленные ответы, выходящие за пределы их контекстного шаблона.

Безусловно, LLM сейчас активно осваивают существенную нишу в ландшафте интеллектуальных систем – интеллектуальных агентов, способных проявлять высшие когнитивные функции в области языковых способностей, абстрактного мышления и аккумуляции знаний и фактов, накопленных в текстовом виде. Однако они не могут полноценно заменить или воспроизвести эмоциональные, социальные, моторные и метакогнитивные системы, требующие глубокого уровня самосознания, адаптивности, межличностного взаимодействия и базирующиеся на глубоко персонализированной информации. Еще большей сложностью сталкиваются попытки использования LLM для создания кибернетических интеллектуальных систем реального физического мира, изначально максимально соответствующих нервным системам организмов, решающих базовые задачи адаптации и выживания и способных к непрерывному естественному обучению. Нахождению возможных путей решения этой задачи посвящена настоящая статья.

2. НЕЯВНЫЕ ЗНАНИЯ И ФОРМЫ ПРОЯВЛЕНИЯ ИНТЕЛЛЕКТА

Современные исследования утверждают, что неявное обучение является фундаментальным и непрерывным процессом в ходе познания человеком окружающего мира [11]. Понятие «*обучение без осознания*» имеет решающее значение для автономных интеллектуальных систем широкого назначения, поскольку неотъемлемым их качеством является непрерывное пожизненное обучение, через которое происходит постоянная адаптация по изучению и изменению окружающей среды.

Примерами неявного обучения являются способность ездить на велосипеде, управлять автомобилем, плавать, ходить, кататься на коньках и лыжах. Каждое из этих умений хранится как неявная процедурная память, которая может быть вызвана без сознательного усилия.

Одно из главных различий между явной и неявной памятью заключается в том, что в *неявной памяти хранятся неосознаваемые воспоминания о навыках и умениях. В явной памяти хранятся факты и события, которые можно вспомнить с помощью сознательного усилия.* То есть память – это не единая система с сознанием, а несколько систем, имеющих разные принципы работы [12]:

Эксплицитная память – декларативная память, содержащая воспоминания о фактах, событиях и информационных материалах.

Имплицитная память – обучение навыкам и формирование привычек. В имплицитной памяти опыт изменяет поведение без какого-либо осознанного содержания или переживания того, что память используется. Имплицитная память измеряется через результаты деятельности. Она не является воспоминанием, но оказывает существенное влияние на черты личности или даже создает их.

Эти две системы работают параллельно друг с другом. Ниже приведена таблица, характеризующая оба вида памяти.

Таблица 3.

Фактор	Явная (эксплицитная) память	Неявная (имплицитная) память
Процесс в памяти	Сознательная, целенаправленная	Неосознанная, автоматическая
Структура памяти	Гиппокамп и височная доля	Неокортекс, мозжечок и др.
Вид информации	Факты, вербальные, семантические, операционные и процедурные описания	Эмоциональные, условные, сенсорные, автоматические и процедурные навыки

Наиболее тесно неявные знания связаны с важнейшей когнитивной функцией – *праксисом*, способностью выполнять целенаправленные движения и действия, которые требуют координации и последовательности. Этот термин охватывает широкий спектр моторных навыков, начиная от базовых действий, таких как одевание и чистка зубов, до сложных профессиональных и бытовых умений, таких как вождение автомобиля или приготовление пищи. Нарушение этой способности вызывает *апраксию* – неврологическое состояние, при котором пациент теряет способность выполнять знакомые движения или решать задачи, несмотря на сохранение желаний и физических возможностей для этого. К часто описываемым типам *апраксии* относят идеомоторную, концептуальную, конечностно-кинестическую и орофациальную [13]. С учетом акцента настоящих исследований на ограничениях больших языковых моделей следует подчеркнуть, что *апраксия* может возникать и при отсутствии каких-либо языковых нарушений, несмотря на близость областей в коре головного мозга, вовлеченных в *праксис* и обработку языка [14], что доказывает важность изучения интеллекта, ответственного за физическое взаимодействие.

Таким образом, интеллект представляет собой совокупность разнообразных способностей и навыков, которые могут быть весьма разнородными по своей природе и использовать различные механизмы реализации. Исторические и современные определения интеллекта включают в себя как общие способности к логическому рассуждению, абстрактному мышлению и адаптации к

новым условиям, так и конкретные навыки, такие как память, числовое понимание и пространственная визуализация. В интеллектуальном ландшафте каждый из этих навыков играет свою уникальную роль, формируя комплексную картину человеческого интеллекта.

Н. Gardner [15] в попытке охватить основные формы интеллекта предложил концепцию множественности его категорий. Анализ классификации различных форм позволил выделить наиболее обсуждаемые категории: IQ (intelligence quotient), EQ (emotional quotient) и PI (physical intelligence). IQ относится к классическому логическому мышлению и эрудиции, EQ отражает эмоциональное понимание, эмпатию, способность взаимодействовать с другими людьми, а PI определяет способность тела к физическому взаимодействию с реальным миром, включая управление частями тела и органами, а также адаптацию к окружающей среде. Следует отметить, что в отношении последней формы интеллекта существует множество направлений исследований и интерпретаций [16], таких как физический интеллект (physical quotient – PQ), воплощенный интеллект (embodied intelligence – EI), воплощенное познание (embodied cognition – EC) и телесно-кинестетический интеллект. Перечисленные направления скорее дополняют, чем конкурируют друг с другом, но общим для них является развитие концепции о ключевой роли физической активности, моторных навыках и сенсорном восприятии в нашем мышлении и обучении. Далее будет использован весь их спектр в зависимости от контекста.

В контексте существующего ИИ большие языковые модели демонстрируют значительные успехи в областях, связанных с IQ, предлагая правильные ответы на основе анализа огромных объемов текстовой и визуальной информации. Однако они сталкиваются с ограничениями при решении задач, требующих глубокого эмоционального понимания человека (EQ), и еще большие проблемы возникают в части непосредственного физического взаимодействия с реальным миром (PI). Прогресс в области робототехники привел к широкому использованию роботов в промышленности, но ввиду низкого уровня PI они все еще далеки от универсальности, автономии и адаптивности, сопоставимой даже с возможностями простейших организмов.

С экономической точки зрения анализ соответствия форм интеллекта и существующих для них рынков труда констатирует, что виды деятельности, связанные с IQ в чистом виде, занимают только 6–7 % рынка, доля физического воплощенного интеллекта (PI) среди востребованных профессий – более 33 % и свыше 60 % – в сочетании с EQ [17]. Большая часть рынка труда требует непосредственного антропоморфного взаимодействия с окружающими физическими объектами в реальном времени, что соответствует биологическим нервным системам, сформировавшимся в процессе эволюции путем непосредственного контакта с реальным миром. В то же время существующие реализации ИИ создаются на основе огромных объемов опосредованной информации, представленной в языковой форме, и предполагают диалоговую форму взаимодействия, преимущественно через текстовые или преобразованные в текст голосовые сообщения.

Следует отметить, что данную ситуацию прекрасно осознает и крупнейший производитель решений ИИ: «*The next wave of AI is **physical AI**. AI that understands the laws of physics. AI that can work among us*» («Следующая волна ИИ – это физический ИИ. ИИ, понимающий законы физики. ИИ, который может работать среди нас»), – спрогнозировал CEO Nvidia Дженсен Хуанг [18]. Исследователи отмечают, что происходит новый сдвиг парадигмы от эпохи «интернет-ИИ» к «воплощенному ИИ». В предыдущей парадигме алгоритмы и агенты ИИ обучались на наборах данных, состоящих из изображений или текста, которые в основном были получены из интернета. Этот подход, хотя и позволил достичь значительных успехов в области машинного обучения и компьютерного зрения, имел свои ограничения, связанные с контекстуальностью и применимостью знаний в реальных условиях. В новой парадигме PI-акцент смещается на обучение через непосредственное взаимодействие с окружающей средой, в том числе в симуляторах. Это взаимодействие осуществляется посредством эгоцентрического восприятия, которое напоминает человеческое восприятие мира. Воплощенные агенты ИИ могут двигаться, чувствовать и реагировать на стимулы в окружающей среде, что позволяет им лучше понимать контекст и применять полученные знания на практике [19].

Воплощенное познание – это концепция, предполагающая глубокую зависимость интеллекта от особенностей физического агента, то есть, когда аспекты

тела за пределами центрального мозга играют значительную причинную или физически конститутивную роли в когнитивной обработке [20]. *Воплощенный интеллект* – это подход к разработке и пониманию разумного поведения агентов через учет жесткой связи между ними и их окружением (*situatedness*), ограничениями собственного физического тела, перцептивной и моторной системы и мозга агента (*embodiment*). Возникновение области воплощенного интеллекта тесно связано с параллельным развитием вычислительной информатики и робототехники, где основное внимание уделяется морфологическим вычислениям и сенсорно-моторной координации в моделях эволюционной робототехники, а также нейро- и когнитивных наук [21].

Существующая и доминирующая парадигма «интернет-ИИ» привела к формированию соответствующих архитектур нейронных сетей, эффективно решающих задачи классификации, генерации текста и визуализации информации. Однако эпоха «воплощенного ИИ» требует глубокого пересмотра используемых архитектур нейронных сетей, в том числе с использованием биоинспирированного подхода. Этот подход предполагает создание архитектурных аналогов биологических сетей, которые являются результатом полумиллиарда лет эволюции, способны адаптироваться и обучаться в реальном времени через взаимодействие с физическим миром. Кроме того, учитывая эволюционную природу биологического воплощенного интеллекта, важнейшей и, вероятно, ключевой стадией формирования его искусственного аналога является обучение агента воплощенного интеллекта, которое предполагает по аналогии с живыми организмами три следующих этапа.

Генетический эволюционный отбор – формирование физического тела агента, приобретающего на этом этапе наиболее оптимальную форму, сегментацию с устойчивыми параметрами. Процесс аналогичен естественному отбору в природе.

Первичное обучение в среде – начальное обучение агента в среде, подобное тому, как это происходит с животными в их первые годы жизни. В его ходе формируются параметры, соответствующие как специфике среды, так и особенностям конкретного экземпляра, что позволяет агенту адаптироваться к окружающей среде и приобрести базовые навыки, необходимые для выживания и функционирования.

Отладка непрерывного обучения – тестирование и настройка процесса непрерывного обучения и адаптации агента к изменяющимся условиям среды.

Для моделирования всех названных этапов необходимы симуляторы среды и конкуренции, которые позволяют существенно ускорить соответствующие процессы и минимизировать материальные затраты, фактически являясь генераторами неявных знаний.

3. МОЗЖЕЧОК КАК РЕЗУЛЬТАТ НЕЙРОСЕТЕВОЙ ЭВОЛЮЦИИ

Эволюция многоклеточных организмов началась, когда некоторые клетки утратили свою индивидуальность, став частями крупных систем. Это привело к появлению различных биологических ветвей (царств) – растений, грибов и животных. У животных развилась нервная система, позволившая быстро и целенаправленно взаимодействовать между клетками, улучшая координацию и управление движениями через специализированные клетки – нейроны [22].

Исторически происхождение нервных систем обсуждается в свете двух концептуально различных моделей – «ввод–вывод» (*input–output model: IO*) и «внутренняя координация» (*internal coordination model: IC*). Эти модели определяют две разные функции нервной системы как устройства управления. Согласно модели *IO*, основная роль нервной системы заключается в *получении сенсорной информации и ее обработке для обеспечения необходимой двигательной реакции*. Модель *IC* основана на концепции о центральной роли ранних нервных систем в *стимулировании и координации внутренней активности крупных многоклеточных организаций* (рисунки 1).

Отличие между моделями на абстрактном уровне заключается в двух принципиально различных видах координации. Системы, основанные на *IO*, обеспечивают согласованность действия организма с окружающей средой, то есть координацию между действием и состояниями, где действие является выбором самого организма, а целью – приведение внутренних параметров к требуемым величинам, исходя из внешних факторов. Целью системы, основанной на *IC*, *напротив*, является координация различных аспектов того, что делает сам организм. В этом случае он приводит микродействия частей системы к требуемым параметрам, необходимым для макродействия в целом, но с учетом состояния внешней среды.

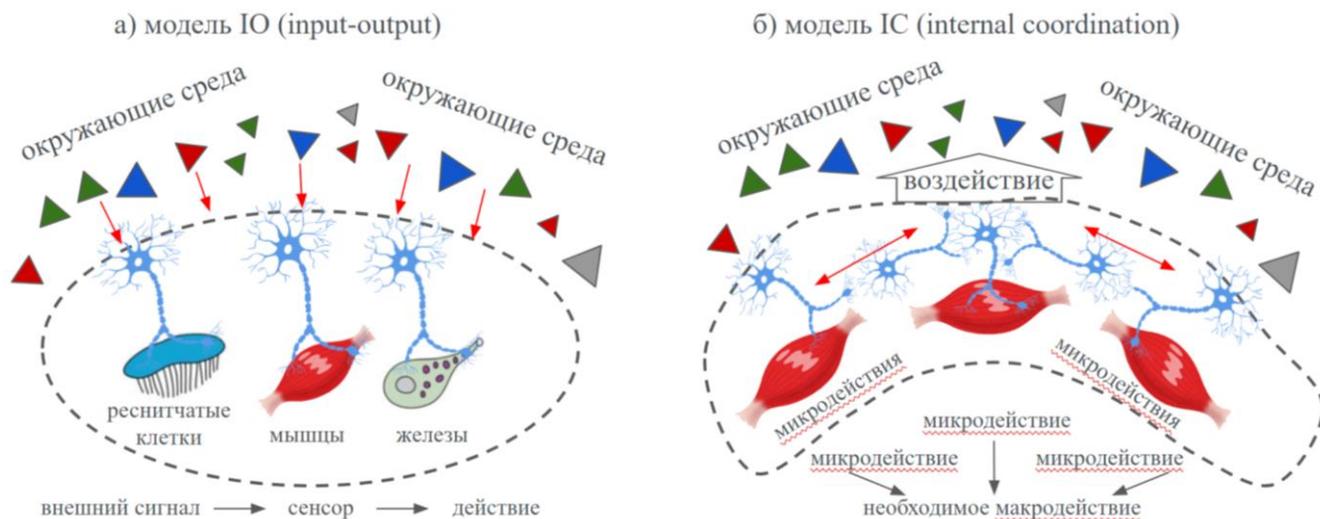


Рис. 1. Две концептуальные модели нервных систем: а) модель ввода–вывода; б) модель внутренней координации

Таким образом, в то время как модель *IO* предполагает эффекторную систему и определяет ее алгоритмическую реакцию, модель *IC* подчеркивает эволюционный сдвиг, связанный с созданием новых многоклеточных эффекторов. В частности, важным эволюционным достижением стала способность использования организмами большого количества сократительных тканей (мышц) – до 700 у позвоночных и до 1500 – у насекомых. При этом общая масса мышц существенно больше, чем масса других органов, и может достигать более половины веса тела. В то же время организация скоординированного сократительного движения сотнями мышц – очень сложная задача, которую не следует воспринимать как нечто само собой разумеющееся, что иногда случается при обсуждении концепции *IO*.

За прошедшие пятьсот миллионов лет эволюции от первых децентрализованных нейронных сетей, появившихся у кишечнополостных и гребневиков, до центральной нервной системы человека нервные системы претерпели множество трансформаций, результатом которых стало возникновение специализированных отделов головного мозга животных, позволяющих осуществлять управление сложными системами. Поставленная задача по определению архитектуры, способной решать задачи внутренней координации с помощью биоинспирированного подхода, может быть выполнена через нахождение и исследование отделов

нервной системы, присущих большинству животных, отвечающих за координацию движения частей тела и их регулярных структур, анализ которых позволит понять принципы построения и масштабирования.

В ходе изучения и перебора различных областей нервных систем биологических организмов можно предположить, что наиболее подходящим органом для первичной реализации выбранного подхода является мозжечок. Он традиционно рассматривается как полнофункциональный орган, осуществляющий контроль и координацию движений, поддержание позы, равновесия и мышечного тонуса. Имеются данные, показывающие участие мозжечка в реализации широкого спектра моторных функций, таких как обучение, планирование, временное согласование. При опухолях мозжечка у 95,5 % больных выявляются статокINETические нарушения [23]. При этом повреждения мозжечка не приводят к полной утрате способности двигаться, но вызывают нарушенную координированность движений, что, в частности, проявляется в виде тремора, которые напоминают колебания неисправной системы управления.

Особенностью мозжечка является удивительно правильная и простая клеточная организация, которая повторяется во всей его коре [24]. Его топография позволяет предположить, что он функционально неоднороден, а его повторяющиеся модули (microcomplexes) обрабатывают различную информацию, зависящую от местоположения корковых входных сигналов. Ito M предположил, что вклад мозжечка в когнитивные функции аналогичен его роли в контроле движения. Согласно этой теории, мозжечок формирует внутреннюю модель взаимодействия посредством повторяющихся действий и обратной связи, что в случае знакомых сценариев позволяет мозжечку эффективно выполнять скоординированные движения, обратная связь которых имеет существенную задержку [25]. Предполагается, что аналогичные процессы происходят в широком спектре умственных операций.

Эти особенности свидетельствуют о возможности существования простой, но чрезвычайно универсальной и эффективной архитектуры нейронных сетей, позволяющей формировать сложные физические модели как частей тела самого организма, так и объектов внешнего мира.

4. НЕЙРОАРХИТЕКТУРА И ОБЩЕПРИНЯТЫЕ МОДЕЛИ МОЗЖЕЧКА

Цитоархитектура мозжечка состоит из трех слоев. Под ними находится белое вещество, содержащее миелинизированные нервные волокна (рисунок 2).



Рис. 2. Слои коры мозжечка и белое вещество под ними

Кора мозжечка состоит из следующих слоев:

– *молекулярного*, являющегося внешним, содержащим огромное количество параллельных волокон и уплощенные *дендритные деревья* клеток Пуркинье и Гольджи. Также здесь находятся два типа ингибирующих интернейронов: звездчатые клетки и корзинчатые клетки;

– *слоя Пуркинье*, представляющего собой узкую зону, содержащую тела клеток Пуркинье, которые являются одними из самых крупных нейронов мозга. Дендритные деревья клеток Пуркинье сильно уплощены, что позволяет им пересекаться с параллельными волокнами;

– *зернистого* – нижнего слоя, содержащего зернистые клетки, являющиеся самыми многочисленными нейронами мозга. Зернистые клетки получают возбуждающие входы от моховидных волокон и передают их через параллельные волокна к клеткам Пуркинье. В этом слое находятся интернейроны, такие как клетки Гольджи, которые модулируют активность зернистых клеток.

Белое вещество мозжечка расположено под корой и состоит преимущественно из миелинизированных нервных волокон, которые обеспечивают связь между корой мозжечка и другими частями центральной нервной системы.

В коре мозжечка выделяют шесть классических типов нейронов, более 99 % из них – это клетки видов, приведенных в таблице 4 [26].

Таблица 4.

	<i>Granule cells</i> гранулярные клетки	<i>Golgi cells</i> клетки Гольджи	<i>Purkinje cells</i> клетки Пуркинье	<i>Stellate cells</i> звездчатые клетки
Человек	$1.01 \cdot 10^{11}$	$8.1 \cdot 10^7$	$3 \cdot 10^7$	$1.6 \cdot 10^9$
Кошка	$2 \cdot 10^9$	10^6	10^6	$3 \cdot 10^7$
Крыса	$2.65 \cdot 10^8$	$6.4 \cdot 10^5$	$6.1 \cdot 10^5$	$6.9 \cdot 10^6$

В свою очередь, преобладающими элементами в нейронной сети коры мозжечка являются гранулярные клетки (*granule cells*), которые, по некоторым оценкам, составляют до 99 % всех нейронов мозжечка.

Согласно современному представлению, существует три основные модели мозжечка, каждая из которых играет ключевую роль в понимании функций этой структуры мозга [27]. Модели предсказания (*Forward Models*), внутренней обратной связи (*Internal Feedback Models*) и оптимального обратного контроля (*Optimal Feedback Control Models*) дают представление о современном уровне понимания участия мозжечка в моторном контроле и обучении.

Имеющиеся интерпретации функционального назначения мозжечка отталкиваются от необходимости решения важнейшей проблемы в биологическом управлении моторикой – неизбежной временной задержки афферентных сенсорных сигналов при достижении центральной нервной системы. Мозг всегда получает информацию о теле и окружающей среде с некоторым опозданием. Например, визуальные сигналы достигают первичной зрительной коры через 30 мс, а теменной коры – через 80 мс после начала стимула. В целом время задержки составляет от 10 мс у мелких животных и до 100 мс у крупных, что сопоставимо с временными масштабами быстрых движений.

Соответственно, существующее объяснение работы мозжечка строится на предположении, что он решает проблему задержки сенсорной обратной связи путем вычисления будущего состояния тела на основе текущей оценки тела и афферентного сигнала управления движением (рисунок 3). Предполагается, что

предсказательное вычисление эмулирует или моделирует фактическое движение тела, по сути, решая уравнение движения тела во времени и предвосхищая будущее положение, отсюда и название – внутренняя прямая модель [28].

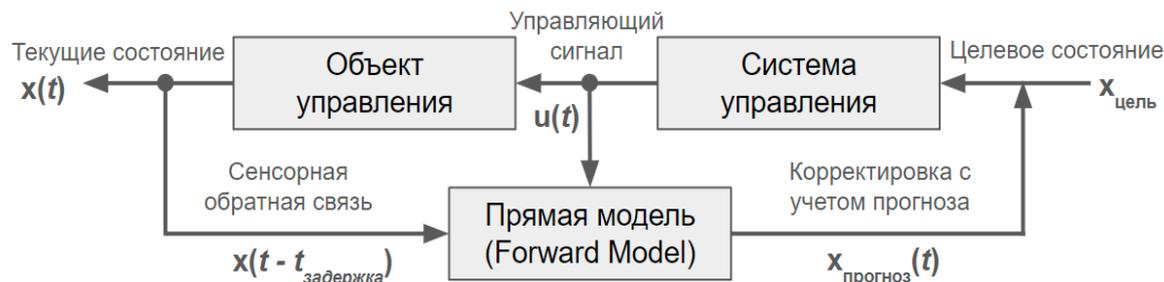


Рис. 3. Внутренняя прямая модель

Внутренняя прямая модель предсказывает состояние тела во времени, которое затем используется контроллером обратной связи, тем самым обеспечивая быстрые, стабильные и точные движения. Управление обратной связью, основанное на прогнозировании, является внутренней обратной связью. Существует ряд доказательств, подтверждающих гипотезу предсказательной прямой модели и внутренней обратной связи, полученных в исследованиях нейровизуализации, исследованиях неинвазивной стимуляции и психофизических исследованиях на людях. В целом мозг предсказывает последствие своего действия, сравнивает предсказание с фактическим последствием и улучшает следующее действие как в двигательном обучении на основе ошибок, так и в обучении с подкреплением на основе подкрепления.

Предсказания внутренних моделей, называемых прямыми моделями, потому что они преобразуют двигательные команды в сенсорные последствия, могут использоваться как для создания калиброванных движений на протяжении всей жизни, так и для улучшения способности сенсорной системы оценивать состояние тела и окружающего мира. Прямые модели полезны только в том случае, если они производят непредвзятые прогнозы. Имеющиеся данные показывают, что прямые модели остаются калиброванными посредством двигательной адаптации: обучения, обусловленного ошибками сенсорного прогнозирования [29].

Кроме того, предполагается, что прямая модель, в дополнение к прогнозным вычислениям, обеспечивает расчет матриц усиления как для контроллера

обратной связи, так и для *фильтрации Калмана*. Модель оптимального управления с обратной связью (*optimal feedback control – OFC*) формирует единую структуру, которая объединяет различные вычислительные процессы в обучении и управлении акторами. Основным элементом модели OFC является вычисление двух матриц усиления (матрицы усиления Калмана и матрицы усиления обратной связи).

Как итог, внутренняя прямая модель обеспечивает три ключевые роли в обучении и управлении моторикой: (1) *прогнозирование состояния для компенсации задержки сенсорной обратной связи*; (2) *прогнозирование состояния для вычисления ошибки между прогнозируемым и фактическим результатами* и (3) *вычисление усиления фильтра Калмана и обратной связи*. Множественность вычислительных функций мозжечка подтверждается двигательными дисфункциями при нарушении его работы, в совокупности известных как мозжечковая атаксия, при этом пациенты с мозжечковыми нарушениями страдают от неспособности к двигательной адаптации и обучению.

5. МОЗЖЕЧКОВАЯ НЕЙРОСЕТЬ ФИЗИЧЕСКОГО ИНТЕЛЛЕКТА

У нас есть способность контролировать движения в различных ситуациях, от простых движений конечностей до сложнейших гимнастических упражнений. И выполнение даже простых движений глаз и рук является чрезвычайно сложным с теоретической точки зрения, так как требует координации множества мышц. При этом мышцы быстро устают, изменяя свои реакции от одного движения к другому, а передача сигналов по нервам происходит значительно медленнее, чем требуется. Однако, несмотря на эти ограничения, нервная система способна адаптироваться и компенсировать их, обеспечивая точные движения на протяжении всей жизни. Мозжечок постоянно корректирует ошибки движений, сравнивая запланированные и фактические результаты [29].

Современные исследователи, основываясь на формальном соответствии между экспериментально полученными данными и уравнениями фильтра Калмана, предполагают следующие вычислительные шаги в мозжечковых цепях (рисунок 4) [28, 30].

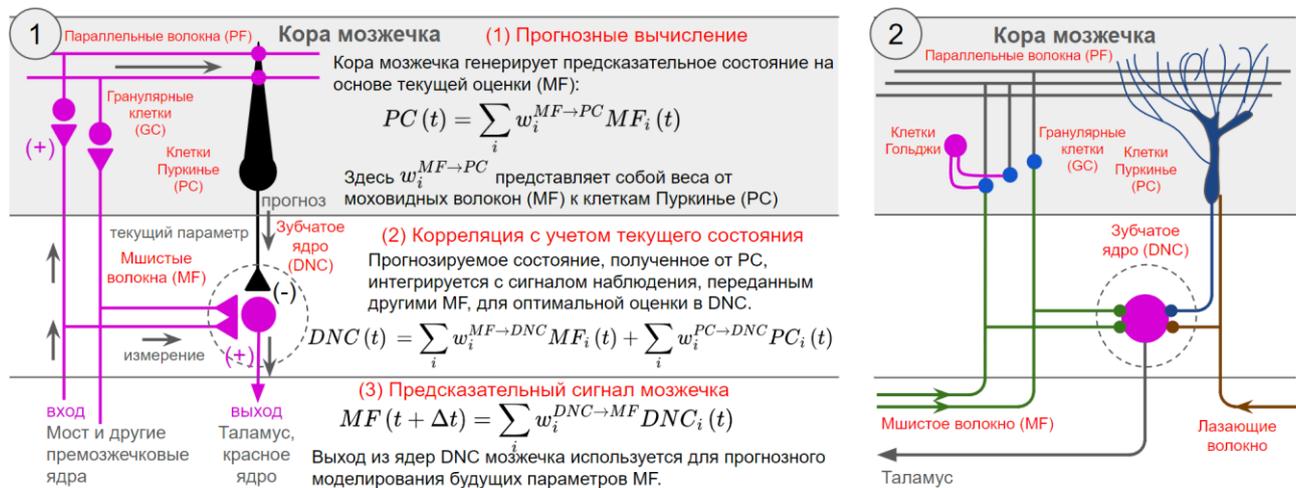


Рис. 4. Схемы мозжечковых контуров и соответствующие вычислительные шаги из разных источников.

(1) клетки Пуркинью (PCs) вычисляют предсказательное состояние из текущей оценки, передаваемой моховидными (мшистыми) волокнами (MFs) (предсказательные вычисления);

(2) активность глубоких ядер мозжечка (DNC) объединяет предсказанное состояние от активности PCs и сенсорную обратную связь от активности MFs (фильтрационные вычисления);

(3) активность DNC предсказывает будущий вход для MFs (предсказание мозжечка).

Предполагается, что данная модель мозжечка выполняет не только внутреннее предсказание, но и оптимальную интеграцию предсказанного состояния и сенсорных сигналов обратной связи.

Как упоминалось ранее, мы имеем аналогию с фильтром Калмана, являющимся эффективным рекурсивным фильтром, основанным на прогнозной оценке вектора состояния динамической системы, с использованием ряда неполных и зашумленных измерений ее состояния. Этот фильтр, введенный Рудольфом Калманом в 1960 году, широко применяется в обработке сигналов, инженерных и эконометрических моделях.

Фильтр, так же, как мозжечок, работает в формате предсказания-коррекции, состоящем из этапов обновления по времени и измерениям. Математическая теория фильтра Калмана и его различных интерпретаций широко представлена в научной литературе [31], здесь только отметим, что идея алгоритма Калмана состоит в нахождении наилучшего приближения к истинной координате x_{k+1} при комбинировании предсказанного значения $\hat{x}_{k+1|k}$ и показания сенсора z_{k+1} (рисунок 5).

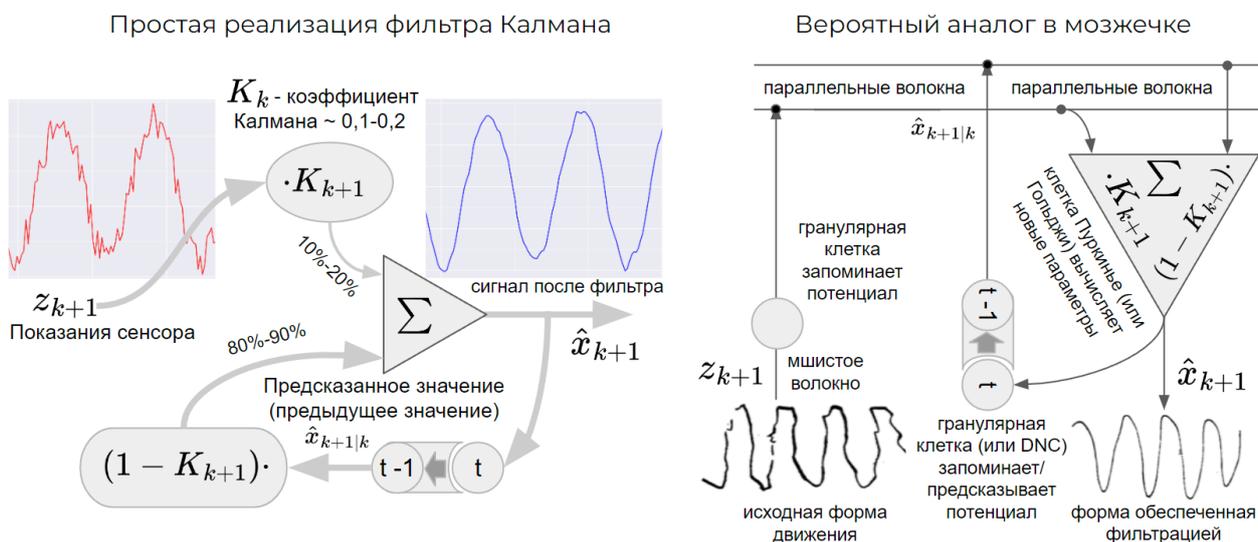


Рис. 5. Простейшая реализация фильтра Калмана и ее возможная интерпретация в структуре мозжечка

Назначаются веса для этих двух источников информации с помощью коэффициента Калмана K_{k+1} . Таким образом, оценка координаты x_{k+1} на очередном шаге вычисляется как $\hat{x}_{k+1} = K_{k+1}z_{k+1} + (1 - K_{k+1})\hat{x}_{k+1|k}$, где $\hat{x}_{k+1|k}$ является предсказанием на основе предыдущего шага. Коэффициент Калмана K_{k+1} выбирается так, чтобы минимизировать среднеквадратичную ошибку между оценочной \hat{x}_{k+1} и истинной координатой x_{k+1} . Если сенсор точный, K_{k+1} будет ближе к единице, иначе вес предсказанного значения будет больше. В результате фильтр Калмана эффективно снижает влияние шума и улучшает точность оценки состояния системы.

На рисунке 5 также показана возможная интерпретация фильтра как нейробиологической структуры, содержащей рассмотренный мозжечковый контур. С целью демонстрации состоятельности данной гипотезы (рисунок 6) автором настоящей работы подготовлены код на языке программирования Python, демонстрирующий работу вышеприведенной версии линейного фильтра Калмана для синусоиды со случайными помехами (1–4), и иллюстрация попытки пациента с заболеванием мозжечка воспроизвести верхнюю синусоидальную кривую (А, В) [32].

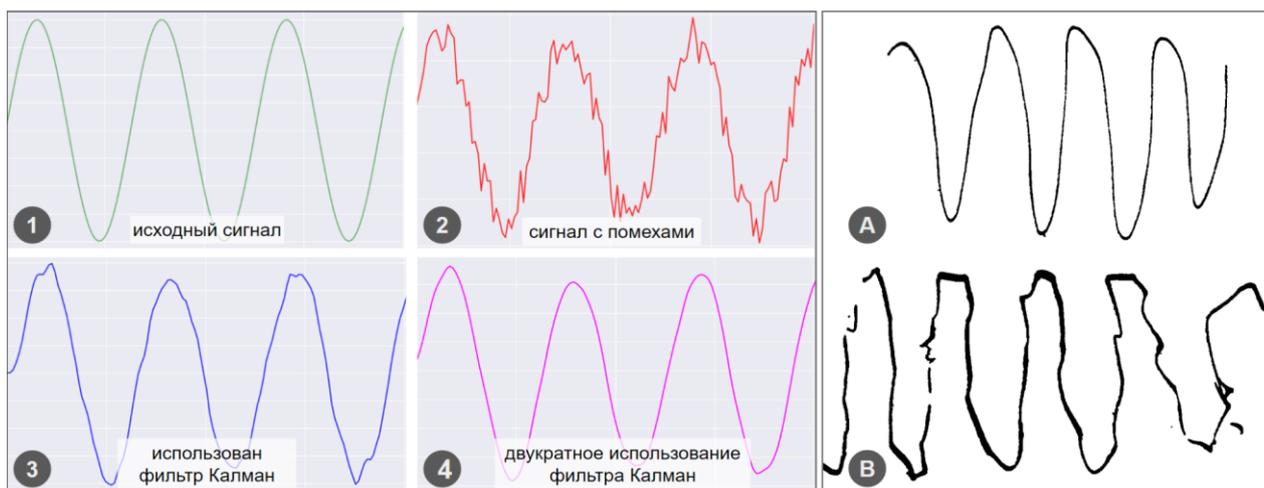


Рис. 6. Программная реализация линейного фильтра Калмана (1–4) и рисунок пациента с заболеванием мозжечка

Однако следует отметить, что предлагаемые современными исследователями схемы межклеточного взаимодействия (рисунок 4) не в полной мере отвечают требованиям построения фильтра Калмана, так как для возникновения полноценного контура необходима передача сигнала как $DNC \Rightarrow MF \Rightarrow GC$.

6. РАЗВИТИЕ АРХИТЕКТУРЫ МОДЕЛИ ФИЗИЧЕСКОГО ИНТЕЛЛЕКТА

Мозжечок является одним из древнейших разделов мозга, он имеется у представителей всех позвоночных и содержит более половины всех нейронов мозга млекопитающих [33]. Очевидно, что сводить его функционал к простой рекуррентной фильтрации – чрезмерное упрощение. Более того, сам процесс рекуррентного предсказания быстро усложняется при росте глубины прогноза и кинематической сложности частей тела организма. В то же время известно, что

искусственные нейронные сети с рекуррентными связями способны аппроксимировать любые динамические системы [34]. Вследствие этого двигательная последовательность может быть смоделирована нейросетями как процесс, который развивается в течение определенного периода. Следовательно, мозжечок должен содержать архитектуру, которая позволяет из простых повторяющихся сегментов сформировать сложнейшую динамическую модель всего многоэлементного тела организма, способную к адаптации к внешнему миру через непрерывное обучение. Далее предложена к рассмотрению гипотеза о механизме построения такой системы.

Кинестетический интеллект для достижения желаемого результата управляемым им организмом должен опираться на прогнозные модели взаимодействия частей тела между собой и окружающей средой. Учитывая, что сами живые существа находятся в реальном физическом мире, такие модели должны строиться на основе законов классической механики, позволяющих, в свою очередь, описать любую механическую систему с помощью дифференциальных уравнений. Ньютон отразил эту взаимосвязь в виде анаграммы, смысл которой в современных терминах можно вольно передать так: «*Законы природы выражаются дифференциальными уравнениями*» [35].

Покажем, каким образом может функционировать биологический аналоговый мозжечковый компьютер на примере нахождения вычислительного решения классического дифференциального уравнения гармонических колебаний:

$$\ddot{x} = -w^2 x \quad \text{where : } \dot{x} = \frac{dx}{dt}; \quad \ddot{x} = \frac{d^2x}{dt^2} = \frac{d\dot{x}}{dt}$$

где x – осциллирующий параметр, \dot{x} – скорость и \ddot{x} – ускорение его изменения. Схема простого аналогового компьютера, обеспечивающего численное решение данного уравнения методом Эйлера, приведена на рисунке 7.

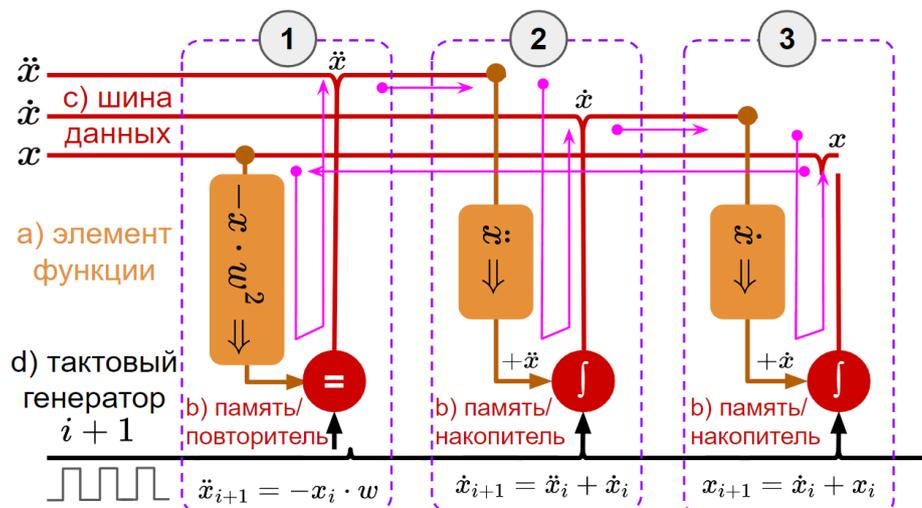


Рис. 7. Схема аналогового компьютера, решающего уравнение $\ddot{x} = -w^2 x$.

В нем три сегмента (1–3), состоящие из типовых элементов классического аналогового компьютера: операционный блок (а), использующий параметры шины данных (с) и осуществляющий над ними простые арифметические операции, а также элемент памяти (b), сохраняющий или интегрирующий входящий сигнал и поддерживающий уровень параметра в шине данных.

Краткое описание элементов компьютера и выполняемых ими операций:

(1) сегмент – усиливает/аттенюирует сигнал x в $-w^2$ раз, сохраняет и поддерживает уровень в шине \ddot{x} ;

(2) сегмент – передает параметр без изменений на интегрирующий элемент, а также поддерживает уровень в шине \dot{x} ;

(3) сегмент – сходный с предыдущим, то есть передает параметр без изменений на интегрирующий элемент, а также поддерживает уровень в шине x .

Таким образом, сегменты образуют полноценный замкнутый контур, в котором процессы соответствуют модели гармонических колебаний и который решает в реальном времени дифференциальное уравнение.

Возвращаясь к цитоархитектуре мозжечка, мы можем выявить в его структуре два схожих вида сегмента, показанных на рисунке 8.

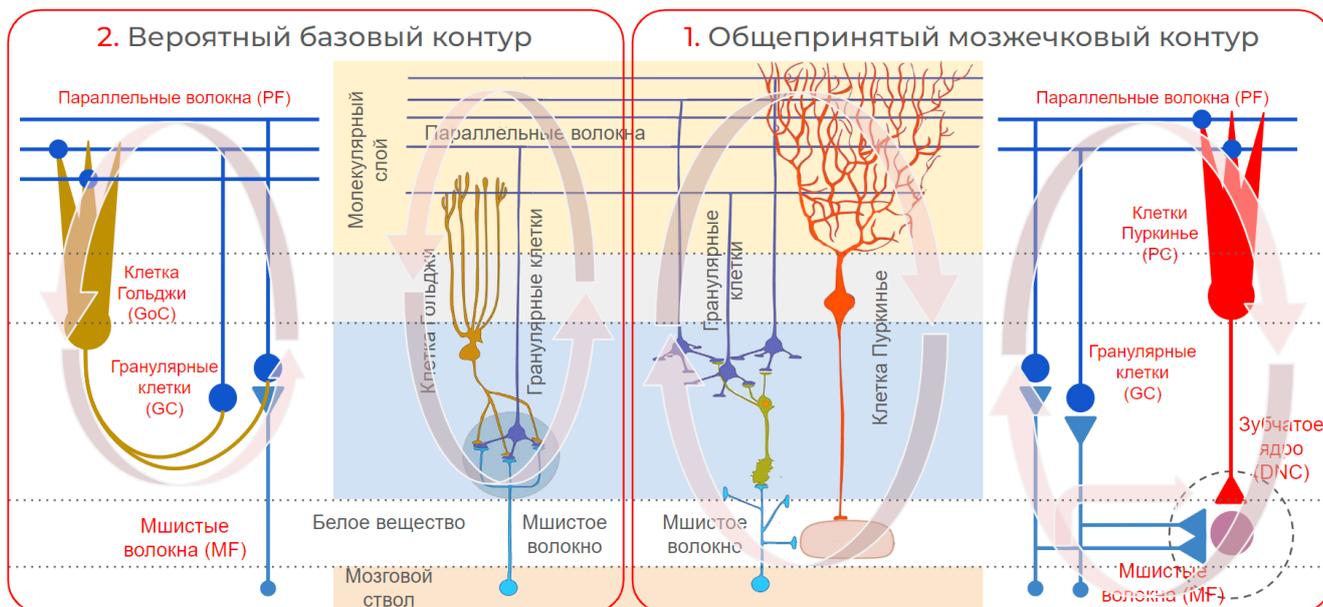


Рис. 8. Два вероятных базовых сегмента биологического аналогового компьютера

При отображении биологической архитектуры была использована структура коры мозжечка взрослого человека, состоящая из четырех отдельных слоев:

(1) поверхностно-молекулярного слоя, бедного клетками структуры, богатого аксонами – параллельными волокнами гранулярных (звездчатых) клеток (GC), дендритами клеток Пуркинье (PC);

(2) узкого слоя PC, включающего монослой сомат PC.

Под слоем PC находится

(3) гранулярный слой, плотно упакованный соматами GC и клетками Гольджи (GoC).

Три клеточных слоя покрывают

(4) слой белого вещества, который включает кортикальные афферентные и эфферентные аксонные пути [36].

По мнению автора, гораздо более вероятным выглядит второй вариант слева, где роли в сегменте распределены следующим образом:

(a) *клетка Гольджи (GoC)* выполняет роль операционного блока, способного принимать параметры из параллельных волокон и проводить над ними простые арифметические операции, что в принципе согласуется с классическим представлением о работе нейрона;

(b) *гранулярная клетка (GC)* получает от аксонов GoC параметры и выполняет роль элемента памяти, *сохраняющего или интегрирующего* полученные параметры и поддерживающего их уровень в параллельных волокнах (внутренняя шина данных).

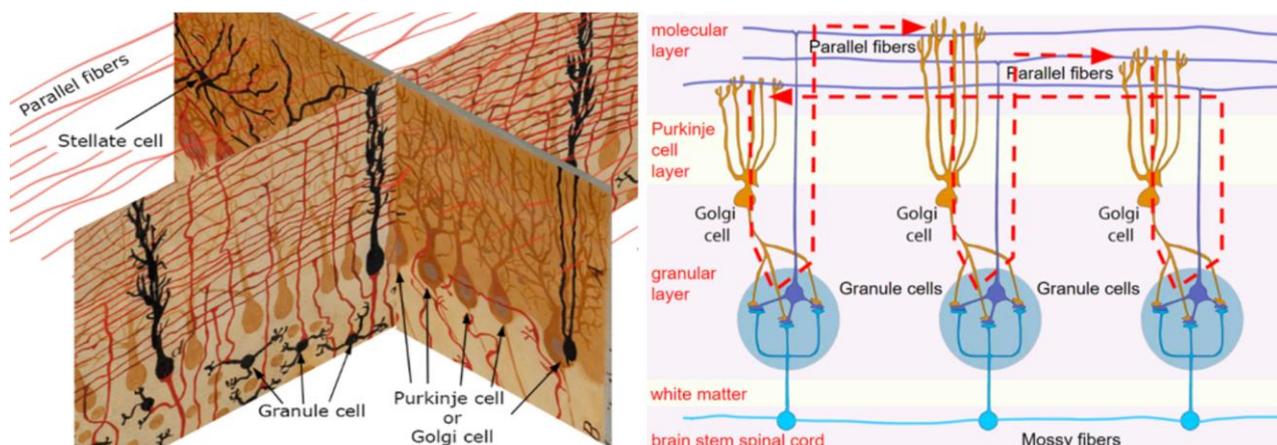


Рис. 9 – Изометрическая проекция и схема биологического аналогового компьютера, решающего дифференциальное уравнение

На рисунке 9 справа показана изометрическая проекция коры мозжечка с использованием рисунков Santiago Ramon y Cajal [37], отображающих основные нервные клетки и их повторяющиеся элементы в архитектуре, а слева – возможная модель биологического аналогового компьютера, решающего дифференциальное уравнение гармонических колебаний, построенная на базе предложенной концепции.

В обоснование приведенной концепции можно привести утверждение, что на уровне молекулярного слоя клетки Гольджи принимают асимметричные синаптические контакты от параллельных волокон и участвуют в механизмах *feed-forward inhibition* и *feedback inhibition loops*, индуцируя синхронную ритмичную активность в гранулярных клетках (рисунок 10) [38]. Исследователи продемонстрировали ингибирующую природу клеток Гольджи и показали, что организация цепей зернистого слоя основана на характерных петлях ингибирования прямой и обратной связи, направленных к дендритам зернистых клеток. При

этом клетки Гольджи являются крупнейшими и наиболее многочисленными интернейронами зернистого слоя, который содержит одну клетку Гольджи на каждые несколько сотен или тысяч зернистых клеток (~6000 у кошек; ~1200 у людей, ~400 у крыс) [39].

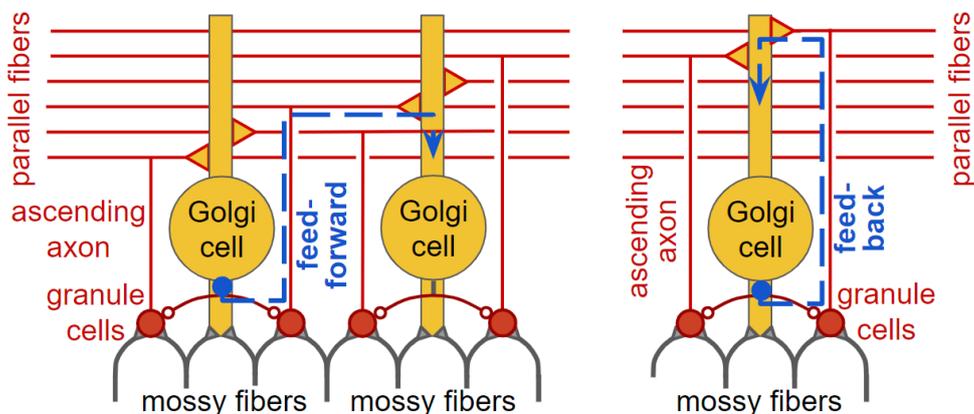


Рис. 10. Схема связей интернейронов при формировании механизма возвратного торможения в коре мозжечка: прямая петля и петля обратной связи.

Предлагаемая архитектура может быть легко масштабирована до метода Рунге–Кутты не только классического четвертого, но и более высоких порядков. Это приводит к росту количества используемых нейронов и замедлению процесса прогнозирования, но одновременно и к значительному повышению точности. Так как для многих животных точность движения является крайне критическим параметром естественного отбора, вероятно, для части прогнозных процессов сформированы именно такие сегменты.

7. ОСОБЕННОСТИ ЦИФРОВОГО МОДЕЛИРОВАНИЯ ОРГАНИЧЕСКИХ НЕЙРОННЫХ СЕТЕЙ

Экспериментальные результаты нейробиологии привели исследователей к определению трех поколений моделей нейронных сетей [40]. Третье поколение в качестве вычислительных элементов использует спайковые (импульсные) нейроны (spiking neurons or integrate-and-fire neurons), сеть из которых (spiking neural networks – SNN) гораздо более точно имитирует естественные органические нейронные сети. В SNN нейроны не передают информацию в виде вычис-

ляемого постоянного потенциала в каждом цикле распространения и для каждого нейрона (как это происходит с типичными многослойными перцептронными сетями), а направляют сигнал только тогда, когда мембранный потенциал достигает определенного значения, называемого порогом (the threshold). В этот момент нейрон срабатывает и генерирует потенциал действия, или спайк – кратковременное изменение мембранного потенциала, которое распространяется по длинному аксону.

Как следствие, создание точной модели аналогового биокомпьютера на базе архитектуры мозжечка с учетом реализации spiking neural networks требует создания принципиально новых нейроморфных процессоров, радикально отличающихся от компьютерной архитектуры фон Неймана, доминирующей в IT-отрасли последние полвека. В этой сфере ведутся активные разработки: Akida™ от BrainChip, Neurogrid от Stanford University, SpiNNaker от University of Manchester, IBM TrueNorth chip.

По этой причине, на первый взгляд, между предложенной архитектурой мозжечка и спайковыми нейронными сетями имеются фундаментальные противоречия, так как каждая из переменных в дифференциальном уравнении, описывающая предложенную архитектуру, определяется некоторым потенциалом, непрерывно изменяющимся во времени. В свою очередь, спайковая модель оперирует импульсами, точнее, частотой импульсов, то есть мозжечковая модель в большей степени соответствует второму поколению искусственных нейронных сетей. Однако spiking может рассматриваться как разновидность частотной модуляции (frequency modulation), то есть в качестве одного из видов аналоговой модуляции, при которой модулирующий сигнал управляет частотой несущего колебания, а амплитуда остается относительно постоянной.

В качестве доказательства состоятельности предложенной гипотезы рассмотрим динамическую модель биологической активности нейрона на примере модели Izhikevich, которая относится к классу моделей с дискретным временем и представляет собой точечные отображения. Модель Izhikevich изначально разработана в непрерывном времени, но ее дискретизация с временным шагом 1 мс с использованием метода Эйлера может быть описана следующим выражением [41]:

$$\begin{aligned} & \text{if } v_n < 30 \begin{cases} v_{n+1} = 0.04 \cdot v_n^2 + 6 \cdot v_n + 140 - u_n + I \\ u_{n+1} = 0.004 \cdot v_n + 0.98 \cdot u_n \end{cases} \\ & \text{otherwise} \begin{cases} v_{n+1} = c \\ u_{n+1} = u_n + d \end{cases} \end{aligned}$$

где v – мембранный потенциал, u – медленная восстанавливающая переменная, параметр c определяет напряжение при переустановке; I – ионный ток, инжектированный в нейрон, благодаря которому и происходит изменение мембранного потенциала.

Медленная подсистема имеет особенности: когда генерируется спайк, медленная переменная испытывает резкий скачок, равный значению параметра d . Быстро-медленные свойства модели позволяют ей воспроизводить разнообразные типы колебательной активности, в частности, спайк-бёрстовые колебания.

На рисунке 11 показана зависимость частоты импульса от величины тока с использованием модели Izhikevich.

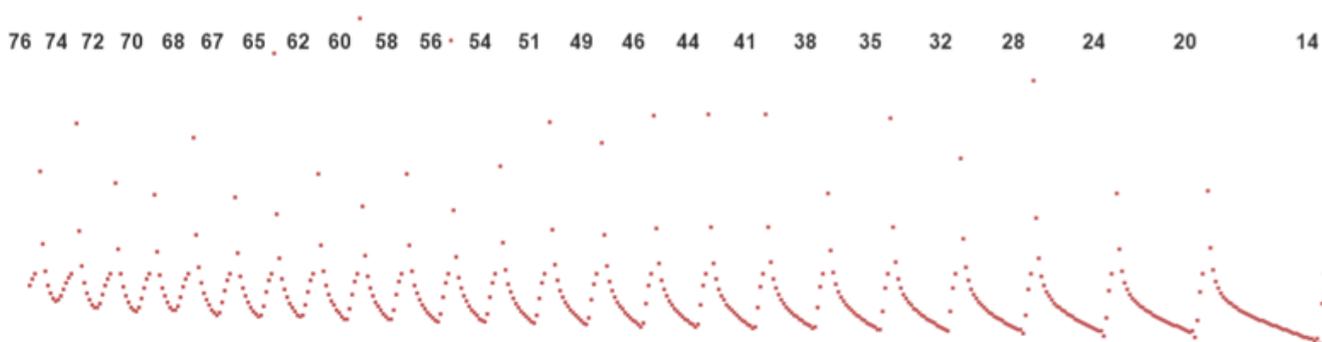


Рис. 11. Частотная модуляция в spiking neural networks. Сверху показана величина тока I . Частота всплесков пропорциональна величине тока

Изображение быстрых изменений есть не что иное, как классическая модель «накопления и сброса». Hodgkin, основоположник бифуркаций в нейронной динамике, стимулировал аксон кальмара импульсами различной амплитуды и выявил три вида нейронной возбудимости:

Class I – в зависимости от силы подаваемого тока потенциалы действия могут генерироваться с низкими частотами;

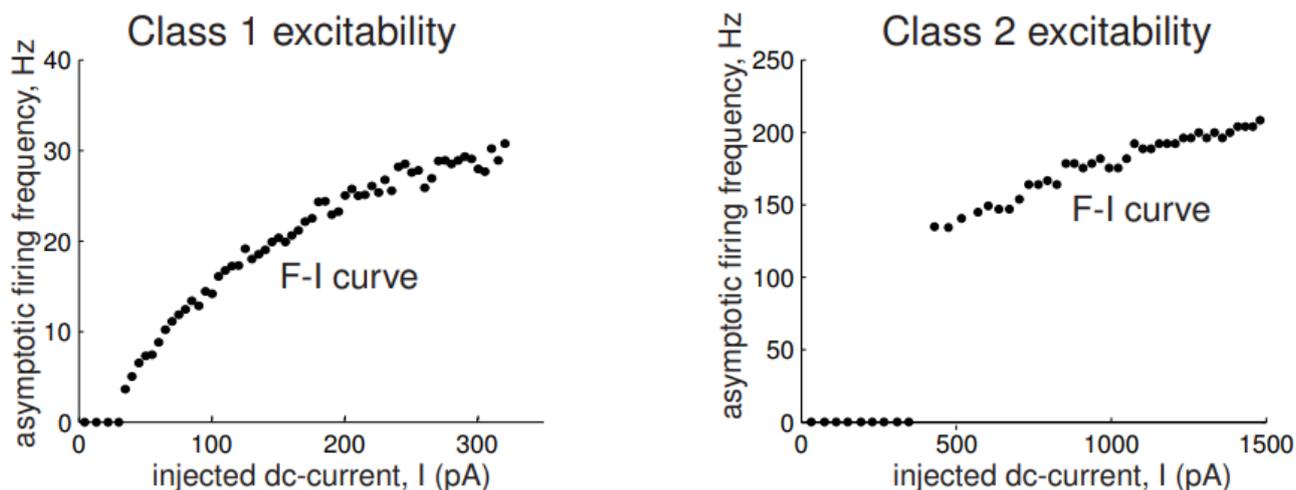


Рис. 12. Frequency-current (F-I) curves of cortical pyramidal neuron and brainstem mesV neuron

Class II – потенциал действия генерируется в определенной полосе частот;

Class III – откликом на импульсный ток является генерация одного потенциала действия.

На рисунке 12 изображены frequency-current (F-I) curves активности нейрона для I и II классов возбудимости [42].

Таким образом, учитывая, что потенциал v служит источником итогового тока I , биологические нейроны определяются как частотное представление амплитудной модели нейронных сетей. При этом оно включает в свою операционную работу понятие времени.

По всей видимости, эволюционная природа выбранного естественным отбором спайкового метода основано на гораздо большей помехоустойчивости. Здесь уместно сравнение с частотной модуляцией (FM) – технологией, используемой в телекоммуникациях, радиовещании и обработке сигналов. Она гораздо менее восприимчива к радиочастотным помехам, чем сигнал амплитудной модуляции (AM) равной мощности. Однако искусственные нейронные сети гораздо легче и эффективнее реализовать на компьютерах архитектуры фон Неймана, непосредственно использовав амплитудное представление. К тому же абсолютную точность и помехоустойчивость обеспечивает принцип хранения и передачи уровня сигнала в данной архитектуре – через его двоичное кодирование.

Работа биологического аналогового компьютера носит непрерывный характер. Даже если мы имеем дело с частотной модуляцией, то и плавность изменения частот непрерывна. В то же время архитектура фон Неймана изначально содержит дискретный итерационный способ вычислений. По этой причине на ранее приведенном схематическом изображении цитоархитектуры мозжечка, решающего дифференциальное уравнение гармонических колебаний, одному из моховидных волокон (*mossy fiber*) были приданы свойства тактового генератора. В реальности *mossy fiber* являются основным и универсальным источником афферентных сигналов, поступающих в мозжечок.

Согласно теореме Котельникова (*The Nyquist–Shannon sampling theorem*), «любую функцию $F(t)$, состоящую из частот от 0 до f , можно непрерывно передавать с любой точностью при помощи чисел, следующих друг за другом менее чем через $1/(2f)$ секунды». То есть переход от сигналов, непрерывных во времени, к сигналам, дискретным во времени, может быть осуществлен без потерь при условии необходимой частоты дискретности. Таким образом, в случае принятия этих двух допущений биоинспирированный алгоритм построения нейронных сетей на базе архитектуры мозжечка, позволяющего находить вычислительные решения систем дифференциальных уравнений, может быть легко реализован в классических цифровых компьютерах и языках их программирования, при этом такое упрощение соответствует концепции построения биологического эталона.

8. МОЗЖЕЧОК КАК АНАЛОГОВЫЙ КОМПЬЮТЕР ОБЩЕГО НАЗНАЧЕНИЯ

Модель мозжечка в полной мере может соответствовать принципам построения классических аналоговых компьютеров, длительное время являвшихся единственным инструментом научных и военных расчетов. Согласно существующим определениям [43], название аналоговых вычислительных машин возникло как отражение принципа построения таких систем, основанного на подборе физических компонентов (биологических, химических, электрических, механических, оптических и т. п.), процессы в которых протекают аналогичным образом. Так как в реальном мире большинство динамических моделей описы-

вают системами дифференциальных уравнений, подбирается физический компонент, процессы в котором представлены теми же математическими выражениями. Таким образом, создается полная аналогия моделируемого процесса.

Теорию аналогового компьютера *общего назначения* (general purpose analog computer – GPAC) впервые предложил К. Шеннон в рамках доказательства теоремы, согласно которой самую общую систему обыкновенных дифференциальных уравнений

$$f_k(x, y_1, y_1' \dots y_1^m; y_2, y_2' \dots y_2^m \dots y_n, y_n' \dots y_n^m) = 0; \text{ где } k = 1, 2 \dots n$$

m -го порядка от n -зависимых переменных можно решить на дифференциальном анализаторе с использованием конечного числа только двух типов устройств:

– *интеграторов*, обеспечивающих выходной сигнал пропорционально интегралу (обычно по времени) от входного сигнала. То есть устройство получает параметры u и v , выдает $w = \int_{v_0}^v (u + a)dv$, где a – произвольная константа;

– *сумматоров*, обеспечивающих выходной сигнал, равный $w = u + v$ (в более комплексной формулировке определение выглядит как объединение сумматоров и умножителя с получением функции $Y = \sum_{k=1}^N a_k x_k$, где a_k – действительные константы).

Уравнения Шеннона отличаются тем, что содержат лишь операции интегрирования, суммирования и умножения переменной на коэффициент. Отсутствие функциональных преобразований позволяет упростить и унифицировать процесс численного интегрирования дифференциальных уравнений в форме Шеннона. Для этого достаточно в исходных уравнениях, заданных в нормальной форме Коши, произвести замену переменных.

В отличие от уравнений в нормальной форме Коши, уравнения Шеннона не содержат общего для всех уравнений множителя, представляющего собой дифференциал независимой переменной. Отсутствие такого множителя указывает на то, что уравнения Шеннона требуют интегрирования не по Риману, а по Стильтьесу. Интегрирование по Стильтьесу отличается от интегрирования по Риману тем, что оно позволяет учитывать скачкообразные изменения в интегрируемой функции, что делает его более гибким и подходящим для обработки разрывных функций.

Исключением из теоремы являются довольно редко встречающиеся гипертрансцендентные (трансцендентно-трансцендентные) функции, например, гамма-функция или дзета-функция Римана [44].

Таким образом, можно провести аналогию между необходимыми элементами ГРАС К. Шеннона, реализациями в аналоговых вычислительных машинах (АВМ) и их вероятными биологическими аналогами в мозжечке (см. Таблица 5).

Таблица 5.

Устройство ГРАС по К.Шеннону и АВМ	Возможный аналог в мозжечке
<i>интеграторы, установщики вещественных констант</i> - в классических АВМ интеграторы реализуются с использованием конденсаторов и операционных усилителей, которые позволяют накапливать потенциал пропорционально входному сигналу и в последующем сохранять его	<i>гранулярные клетки (granule cells)</i> – накапливают или/и запоминают потенциал, поддерживают его уровень в параллельных волокнах
<i>сумматоры, умножители</i> – АВМ сумматоры реализуются с помощью резисторов и операционных усилителей генераторы функций	<i>клетки Гольджи (Golgi cells)</i> – получают сигналы от множества входов и суммируют их, выполняя функции, аналогичные сумматорам и умножителям в аналоговых компьютерах, функционируют во внутреннем контуре; <i>клетки Пуркинье (Purkinje cells)</i> – аналогичны <i>клетки Гольджи</i> , но являются единственными нейронами, «выходными» из мозжечка

Таким образом, в мозжечке содержатся все необходимые виды функциональных элементов для определения его в качестве универсального аналогового компьютера общего назначения.

В связи с некоторой предвзятостью в отношении аналоговых компьютеров следует отметить, что причиной их вытеснения цифровыми компьютерами стала специфика программирования – моделирование процессов реального мира было возможным при условии глубокого понимания физической природы явления и наличия математической интерпретации через аппарат дифференциальных уравнений. В результате аналоговые машины занимали нишу научных, инженерных или военных приборов с механическим процессом программирования, ограниченной сложностью моделей и невысокой скоростью обсчета. Однако, как будет показано далее на примере биологических организмов, эти ограничения могли быть преодолены в процессе органической эволюции.

Феноменальный прогресс в цифровой электронике привел к тому, что аналоговые вычисления гораздо эффективнее производят современные цифровые компьютеры, имеющие архитектуру фон Неймана. Более того, существует огромное множество программных решений, моделирующих механику сплошных сред, теплопередачу, распространения электромагнитных волн и пр. на основе решения описывающих их систем дифференциальных уравнений с использованием методов конечных элементов (Finite element method), конечных разностей (Finite-Difference Method, FDM) или конечных разностей во времени (Finite-Difference Time-Domain, FDTD) для моделирования, которые в некоторой степени схожи с предлагаемым подходом.

Настоящая работа предлагает альтернативную концепцию существующим системам искусственного интеллекта, формирование другого направления в архитектурах искусственных нейронных сетей. Учитывая гомеостатическую природу многоклеточных организмов, лежащую в основе структуры мозжечка, в отличие от традиционных нейронных сетей с однонаправленной передачей информации (от входных к выходным нейронам), предлагаемая архитектура больше соответствует машине Больцмана (Boltzmann machine) и сети Хопфилда (Hopfield model) [45]. Нейронные сети мозжечка также имеют многочисленные цепи обратной связи, что обеспечивает циркуляцию информации внутри сети до достижения устойчивого энергетического состояния. Математические интерпретации и программные приложения, разработанные на базе этой нейросетевой архитектуры, позволяют моделировать сложные физико-механические про-

цессы, которые традиционно описываются системами дифференциальных уравнений, что значительно расширяет возможности ИИ в управлении динамическими системами.

Для демонстрации возможностей предложенного подхода в области моделирования автором в опубликованных научных работах приведены примеры использования данной архитектуры для вычислительного решения задачи трех тел [46] и исследования процессов распространения волн в неоднородной среде [47] (рисунок 13).

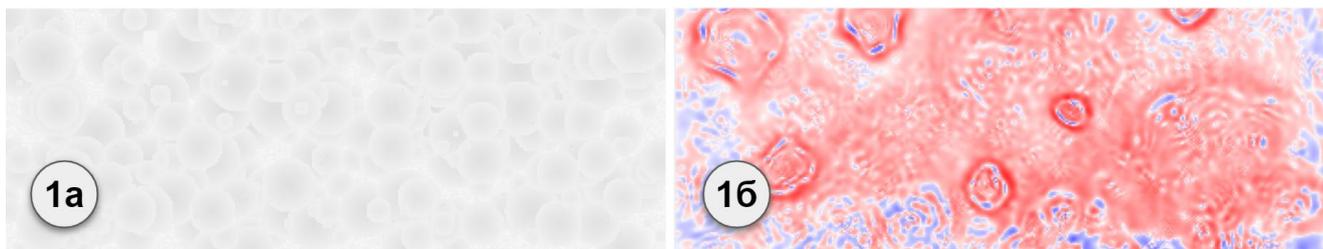


Рис. 13. Неоднородность среды (1a) и распространения волн в ней (16)

9. ФОРМИРОВАНИЕ МОДЕЛИ ФИЗИЧЕСКОГО ИНТЕЛЛЕКТА

По всей видимости, образование живых многоклеточных организмов сопровождалось острой необходимостью как внутренней координации частей тела, так и обеспечения взаимодействия с внешней средой. При этом стремительно растущая сложность самих организмов и необходимость локомоции потребовали создания нервной системы, способной сформировать точную кинестетическую модель частей тела, необходимую для эффективного управления мускулатурой.

Эволюция за последние 600 миллионов лет породила «бесконечное» число форм, начиная с древнего билатерального червя и заканчивая множеством разнообразных морфологий животных. Более того, все животные демонстрируют удивительную степень воплощенного интеллекта, используя свою развитую морфологию для реализации сложнейших задач. Область воплощенного познания утверждает, что разумному поведению могут быстро научиться агенты, морфология которых хорошо адаптирована к окружающей среде.

Воплощенный интеллект позволяет организмам управлять сложным ансамблем составных частей их тел, обеспечивая адекватное разумное поведение в окружающем их реальном мире [48]. Особенностью физико-анатомического

строения биологических организмов является возможность представления частей его тела в виде кинематических цепей (kinematic chain), см. рисунок 14. Соответственно, кинематические цепи играют ключевую роль в проектировании и управлении робототехническими системами. В классическом определении кинематическая цепь является совокупностью твердых тел, соединенных шарнирами для обеспечения ограниченного движения, которое является математической моделью механической системы.

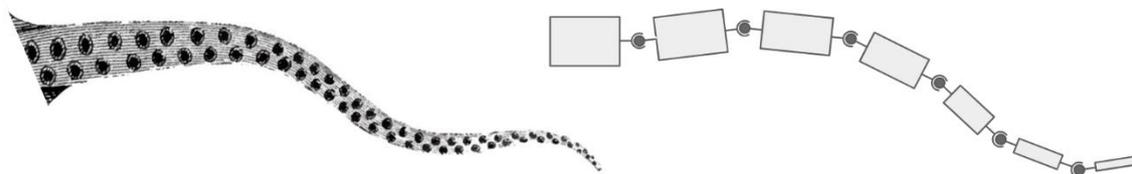


Рис. 14. Щупальце осьминога и представление в виде кинематической цепи

Ключевыми компонентами таких систем являются механические узлы - приводы (актуаторы), обеспечивающие движение частей объекта относительно друг друга, а также физический интеллект, обеспечивающий управление узлами с учетом их внутреннего и внешнего взаимодействия. Первая компонента была разработана уже много лет назад – конструкции приводов, по физическим параметрам равные и даже превосходящие мышцы биологических организмов. Современные промышленные высокоскоростные манипуляторы легко справляются с прецизионными операциями, такими как игра на музыкальных инструментах, теннис, снукер и т. д. Высокомоментные суставные двигатели, используемые сегодня, позволяют строить роботов, способных двигаться на четырех и даже двух конечностях, подобно большинству млекопитающих в природе.

Однако существующие системы управления актуаторами не только очень далеки от человеческих возможностей, но даже близко не подошли к способности конкурировать с биологическими организмами. Для человека требуется гемостатическое оперативное управление 640 мышцами, а количество мышц у насекомых еще больше.

Использование предложенной архитектуры нейронных сетей может приблизить решение этой ключевой задачи робототехнических систем. *Продемонстрируем использование данной архитектуры на примере управления простой моделью щупальца осьминога.*

Щупальце осьминога представляет собой удивительно сложную и высокоорганизованную структуру, состоящую из множества мышечных волокон, которые обеспечивают широкий спектр движений. При этом 180 млн нейронов в центральном мозге связаны с более чем 40 млн нейронов в каждом из восьми щупальцев через относительно небольшое количество эфферентов (~32 тыс.) и афферентов (~140 тыс.). Моторный контроль этого животного включает в себя уникальную воплощенную организацию взаимодействия между нервной системой, морфологией тела и средой выполнения задач животного [49].

Задачей настоящей главы являются определение и описание базовых элементов нейросети, использование которых позволяет построить простые механические аналоги биологических организмов. Наиболее простая аналогия нейронных сетей с глубокими обратными связями – это замкнутые энергетические системы, по этой причине оптимальнее всего строить модель механической системы через закон сохранения энергии.

Энергия механической системы может быть представлена в виде суммы двух существенно различных членов: кинетической энергии (T), зависящей от скоростей, и потенциальной энергии (U), зависящей только от координат частиц [50]:

$$E = T(q, \dot{q}) + U(q); \quad E = \sum_a \frac{m_a v_a^2}{2} + U(r_1, r_2, \dots)$$

Таким образом, любая сколь-либо сложная механическая система может быть представлена как набор двух видов универсальных элементов, отражающих кинетическую и потенциальную энергии.

Отталкиваясь от данного положения, можно ввести понятия двух видов элементов:

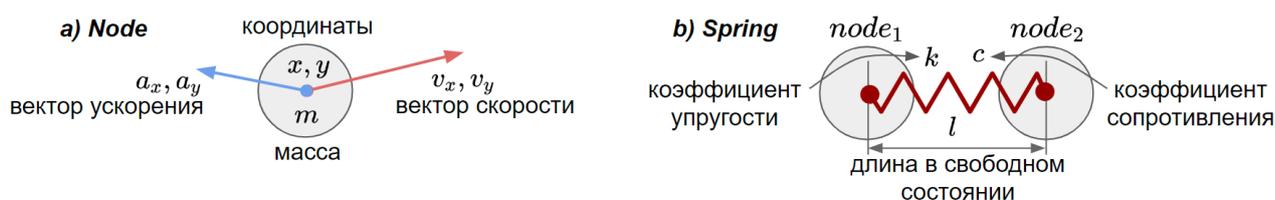


Рис. 15. Два вида базовых универсальных элементов механической системы

1. Узел (Node, рисунок 15а) – кинетический/механический элемент системы (T), выполняющий следующие роли: накопление кинетической энергии;

сочление, к которому крепятся пружины (spring); другие виды взаимодействия со средой, которые будут приведены ниже. Данный реактивный элемент отражает базовое свойство всех механических систем – наличие инерционной массы. Для рассматриваемого случая он обеспечивает учет, в том числе, массы и элементов spring.

Node для двухмерной модели может быть описан следующими параметрами: m – масса, x , y – координаты, \dot{x} , \dot{y} – скорости; \ddot{x} , \ddot{y} – ускорения.

2. Управляемая пружина/мышца (Spring, рисунок 15b) – пространственный элемент, позволяющий: накапливать потенциальную энергию (U); формировать пространственные структуры за счет упругого удерживания узлов (node) на определенном расстоянии. Является своего рода мышцей, то есть некоторые параметры элемента могут быть изменены нервной системой.

Это крайне универсальный – активный, реактивный и пассивный – элемент, то есть может создавать, поглощать и накапливать/отдавать энергию. Кроме того, именно с помощью него строится пространственная структура тела. Похож на мышцу, но способен не только сжиматься под влиянием нервных импульсов, но и разжиматься, а также при определенных параметрах заменять жесткие элементы, такие как кости.

Spring для двухмерной модели может быть описан следующими параметрами: l_0 – длина пружины в свободном состоянии, то есть когда Spring имеет заданную длину, но не оказывает воздействие на тела, которые он соединяет; k – коэффициент упругости, коэффициент, связывающий в законе Гука удлинение/сжатие упругого тела и возникающую вследствие этого силу воздействия; c – коэффициент сопротивления, устанавливающий соотношение между скоростью движения связываемых тел и возникающей при этом силой сопротивления,

позволяет демпфировать, то есть гасить колебания; $node_1$, $node_2$ – ссылки на тела, которые связывает spring между собой.

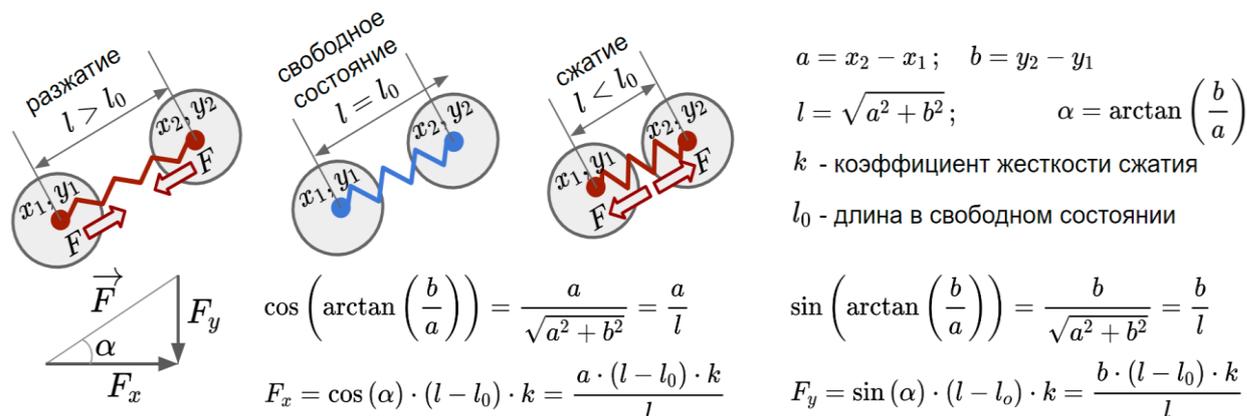


Рис. 16. Расчет силы воздействия spring на nodes

Элемент spring управляется нейронной сетью через задание величины l , фактически уменьшая эту величину, мы вызываем силу сжатия, то есть сближения между $node_1$ и $node_2$, и, наоборот: увеличивая значение l , мы вызовем силу, удаляющую $node_1$ и $node_2$. Силы, воздействующие на nodes, могут быть определены после оптимизации по довольно простой формуле, исключая ресурсоемкие тригонометрические операции (рисунок 16).

Таким образом, система дифференциальных уравнений для i -го элемента spring будет иметь вид

$$\begin{cases} \ddot{x}_{i,1} + \frac{c_i}{m_{i,1}} \dot{x}_{i,1} + \frac{F_{i,x}}{m_{i,1}} = 0 \\ \ddot{y}_{i,1} + \frac{c_i}{m_{i,1}} \dot{y}_{i,1} + \frac{F_{i,y}}{m_{i,1}} = 0 \\ \ddot{x}_{i,2} + \frac{c_i}{m_{i,2}} \dot{x}_{i,2} - \frac{F_{i,x}}{m_{i,2}} = 0 \\ \ddot{y}_{i,2} + \frac{c_i}{m_{i,2}} \dot{y}_{i,2} - \frac{F_{i,y}}{m_{i,2}} = 0 \end{cases} \quad \begin{cases} l = \sqrt{(x_{i,2} - x_{i,1})^2 + (y_{i,2} - y_{i,1})^2} \\ F_{i,x} = \cos(\alpha) \cdot (l - l_0) \cdot k = \frac{(x_{i,2} - x_{i,1}) \cdot (l - l_0) \cdot k}{l} \\ F_{i,y} = \sin(\alpha) \cdot (l - l_0) \cdot k = \frac{(y_{i,2} - y_{i,1}) \cdot (l - l_0) \cdot k}{l} \end{cases}$$

где индексы 1 и 2 – это элементы node, прикрепленные к i -й spring с обеих сторон.

С точки зрения цитоархитектурного представления, на рисунке 17 показана примерная схема биологического аналогового нейрокомпьютера, решающего вычислительным методом заданную систему дифференциальных уравнений.

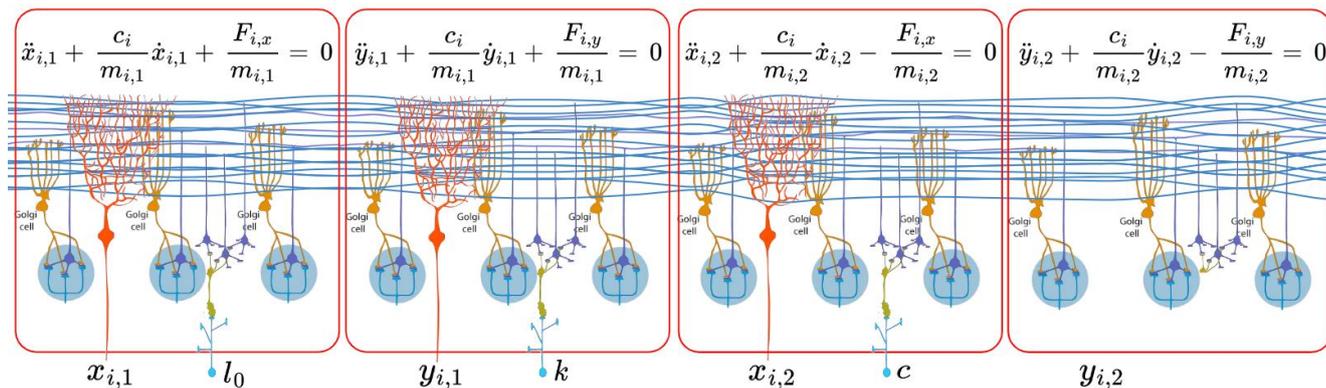


Рис. 17. Схема нейрокompьютера, вычисляющего взаимодействие двух nodes, связанных между собой spring

Параллельные волокна содержат параметры, определяющие модель и ее состояние, являющиеся аксонами гранулярных клеток. Для 2D-представления пара nodes требует $6 \times 2 + 1$ оперативных параметров ($x, y, \dot{x}, \dot{y}, \ddot{x}, \ddot{y}, lr$) и 1×2 фиксированных (m), а каждая spring, объединяющая пару, требует три управляемых параметра (l_0, k, c) и два фиксированных ($node_1, node_2$) – итого 20 гранулярных клеток. Как будет показано при дальнейшей детализации модели, стремительно растет число гранулярных клеток, по всей видимости, именно это является причиной наличия в биологических системах столь существенного количества этих видов нейронов. При этом параметры и, соответственно, гранулярные клетки можно разделить на три группы, определяющие:

– *морфологические (генетические) особенности* – признаки, приобретенные при формировании организма. В данном случае это nodes соединяющихся springs (b_1, b_2), то есть конструкция каркаса тела и системы органов;

– *параметры, формирующиеся в процессе адаптации* организма к локальным особенностям окружающей среды. Примером может служить нейромышечная память. Для рассматриваемой модели формируется через сенсорную обратную связь путем подстройки параметров, характеризующих свойства springs и частично nodes: l_0, k, c, m ;

– *текущие параметры* (параметры фазового пространства). Как правило, характеризуются быстрой динамикой изменения, для данного случая это $x, y, \dot{x}, \dot{y}, \ddot{x}, \ddot{y}$ и lr .

Таким образом, возникает альтернативная существующей гипотеза о процессе обучения не только через пластичность синаптических соединений в

нейронных сетях, но и через изменение потенциала в гранулярных клетках. Также стоит отметить роль клеток Пуркинье как главных эфферентных нейронов мозжечковой коры – трансформация информации, полученной от параллельных волокон, и передача обработанных сигналов к глубоким мозжечковым ядрам, которые затем влияют на моторные команды.

Программирование соответствующий модели можно представить как процесс, включающий в себя как минимум четыре уровня:

– *морфологическое программирование воплощенного интеллекта* – создания пространственной кинематической схемы, описывающей параметры механических элементов и их взаимосвязь друг с другом; для рассматриваемого случая это расположение nodes, а также определение элементов, которые соединяют springs;

– *определение базовых параметров*, элементов воплощенного интеллекта, то есть определение базового состояния параметров, позволяющих иметь оптимальную для существующей среды форму и состояние; для нашего случая это параметры, характеризующие свойства springs и частично nodes: l_0, k, c, m ;

– *формирование сенсорного восприятия элементами воплощенного интеллекта* – определение локальных петель обратной связи, позволяющих изменять базовые параметры элементов воплощенного интеллекта и обеспечивающих общую динамику базового уровня;

– *создания общего центра координации, существенно повышающего глубину возможной адаптации* – создание базы гомеостатических устойчивых состояний в зависимости от внутренних потребностей или внешнего влияния и затрагивающие большое количество элементов воплощенного интеллекта.

При этом нейронная архитектура позволяет осуществить эволюционную донстройку параметров.

10. МОРФОЛОГИЧЕСКИЙ УРОВЕНЬ МОДЕЛИ ФИЗИЧЕСКОГО ИНТЕЛЛЕКТА

Безусловно, структура биологических организмов гораздо сложнее, чем предложенное упрощение, а мозжечковая нейроархитектура, являясь универсальным аналоговым компьютером общего назначения, способна управлять гораздо более сложными физическими системами. Однако даже такое представ-

ление позволяет решать задачи физического моделирования комплексных объектов. Данное упрощение приводит предложенную модель к одной из интерпретаций метода конечных элементов (Finite element method) – численному методу решения дифференциальных уравнений в частных производных, широко используемых в инженерном и физическом моделировании. Метод включает дискретизацию сложной геометрии объекта на простые элементы, формирование уравнений для каждого из них и их последующую сборку в систему уравнений, которую затем решают для нахождения приближенных решений. Названный метод широко используется в анализе напряжений в конструкциях, теплопередачи, течения жидкостей, электромагнитных полей и других областях, где точное аналитическое решение сложно или невозможно получить. В свою очередь, рассматриваемый нейросетевой подход позволяет расширить данный метод и использовать его для создания динамических систем, адаптирующихся к внешней среде, реализации элементов воплощенного интеллекта, а также нахождения оптимальной конструкции через эволюционные алгоритмы.

Первый и второй уровни программирования модели – создание пространственной конструкции тела и подбор параметров, позволяющих иметь эластичную конструкцию. Данный этап требует использования инструментов, схожих по функционалу с графическими редакторами, оперирующими базовыми элементами, в данном случае это *nodes* и *springs*. Для этих целей автором написан на Python простой векторный редактор формирования каркаса с использованием *nodes* и *springs*, включающий *software engine* (программный движок), позволяющий в реальном времени моделировать деформацию конструкций под внешним воздействием, а также проводить настройку параметров [51]. На рисунке 18

представлены простейшие конструкции и процесс изменения их состояния во времени.

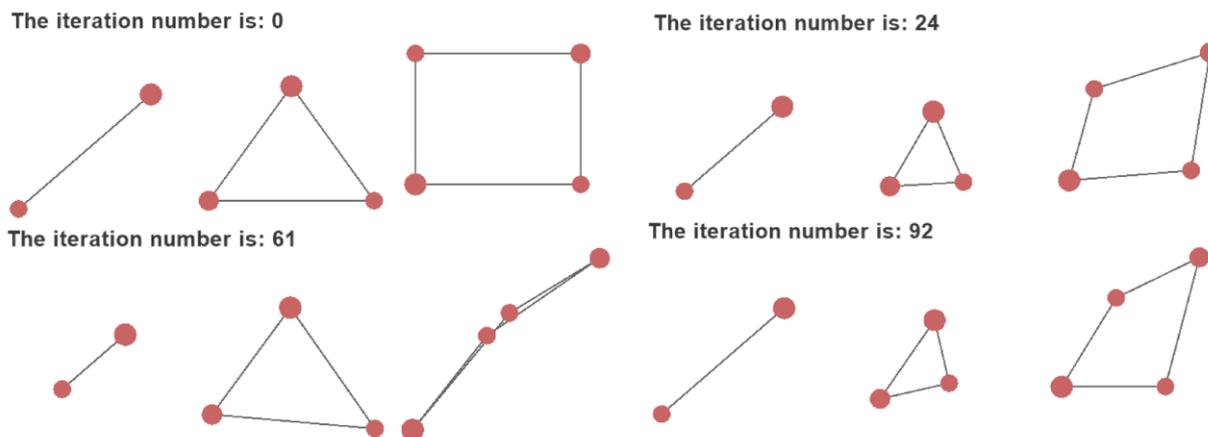


Рис. 18. 2–4-элементные базовые конструкции и их изменения во времени

Предложенная форма программирования динамических механических объектов позволяет конструировать и подвергать внешнему воздействию системы из десятков *springs* и *nodes*, то есть исследовать относительно сложные модели.

Задача формирования каркаса из *nodes* и *springs* в первом приближении совпадает с принципами конструирования ферм (лат. *firmus* – прочный), являющихся базовым разделом в строительной механике. Задача ферм схожа с биологической – оставаться геометрически неизменяемой после замены жестких узлов шарнирными в ее элементах. Для определенных нами элементов усилия растяжения – сжатия возникают только у *springs*, при этом *nodes* выполняют роль шарниров. Фермы также строятся из прямолинейных стержней, соединенных в узлах в геометрически неизменяемую систему, к которой нагрузка прикладывается только в узлах [52]. Фермы зародились в эпоху деревянного строительства, и металлические фермы являются наследниками многих инженерных находок той эпохи. Развитие железных дорог в первой половине XIX века привело к срочной потребности в мостах, способных нести тяжелые движущиеся грузы, и в новых формах зданий для вокзалов и системы обслуживания подвижного состава. Это послужило быстрому эволюционному прогрессу инженерии, появлению целого ряда новых конструктивных схем и формированию базовых конструкций

[53], таких как ферма Уоррена (Warren)  и ферма Пратта (Pratt) .

Для последней часто используется преднапряжение – сжатие раскосов. Данные фермы стали прародителями множества других мостовых конструкций, таких как фермы Bailey, Baltimore, Bollman, Bowstring, Brown, Cantilever.

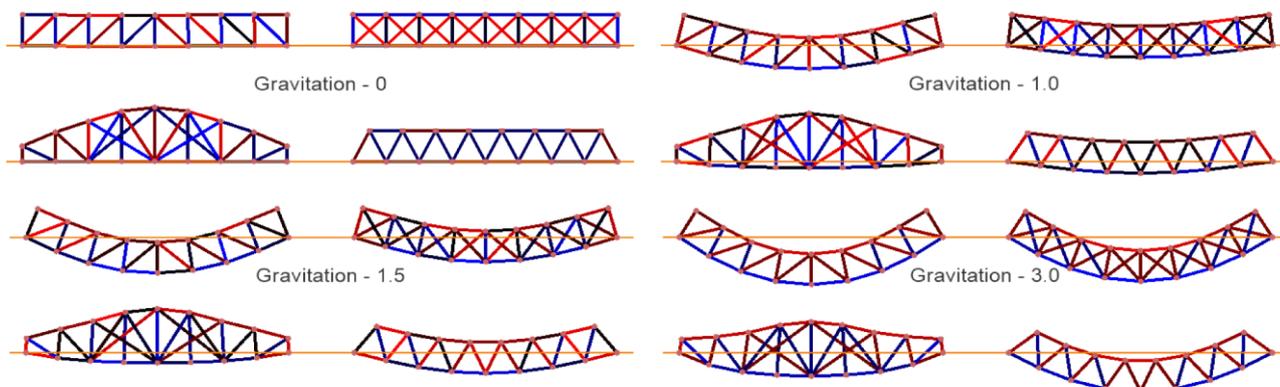


Рис. 19. Изменение формы фермы ферм при нагрузках [54]

Используя software engine, можно запрограммировать и смоделировать их деформацию под нагрузками любых конструкции ферм, на рисунке 19 показаны четыре вида ферм в разных гравитационных полях. Красным цветом отображается сжатие, а синим – растяжение springs, а интенсивность цвета отражает величину нагрузки с нормировкой на всю конструкцию.

Особый интерес в применении предложенного алгоритма при проектировании сложных механических конструкций вызывает потенциальная возможность своего рода обучения используемой в них нейронной сетям для решения оптимизационных задач. Метод градиентного спуска и генетические алгоритмы, примененные к параметрам настройки сети, а с математической точки зрения – к коэффициентам в системе дифференциальных уравнений, позволяют найти оптимальную форму фермы или предложить более простую конструкцию.

11. МОДЕЛЬ ЩУПАЛЬЦА И СЕНСОРНОЕ ВОСПРИЯТИЕ

По аналогии с биологическими организмами модель должна обеспечивать сочетание трех характеристик: жесткости конструкции, сохраняющей устойчивую форму, во избежание повреждения внутренних органов; минимального

количества несущих и управляющих элементов для экономии энергии на создание и поддержание структуры; гибкости и пластичности в процессе изменения формы для движений в среде, в том числе локомоции.

Применительно к выбранному в исследовании примеру синтетического щупальца, в рамках первых двух уровней программирования модели оптимально использовать существующий конструктив щупальцев *octopus rubescens*, продукт 500 млн лет эволюции. В нем присутствуют три основных вида мышц, расположенных определенным образом (рисунок 20) [55]:

- а) *поперечные мышцы (Transverse Muscles, TM)* – окружают осевую нервную цепочку и расположены перпендикулярно к осевой линии щупальца; сокращение приводит к удлинению щупальца за счет сокращения его поперечного сечения;
- б) *продольные мышцы (Longitudinal Muscles, LM)* – проходят параллельно осевой нервной цепочке вдоль всей длины щупальца; их сокращение приводит к укорочению и изгибу щупальца, что позволяет ему сгибаться и выполнять сложные движения;
- в) *косые мышцы (Oblique Muscles, OM)* – обвивают продольные и поперечные мышцы по спирали; сокращение косых мышц приводит к скручиванию щупальца, что позволяет ему закручиваться вокруг объектов и выполнять сложные трехмерные манипуляции.

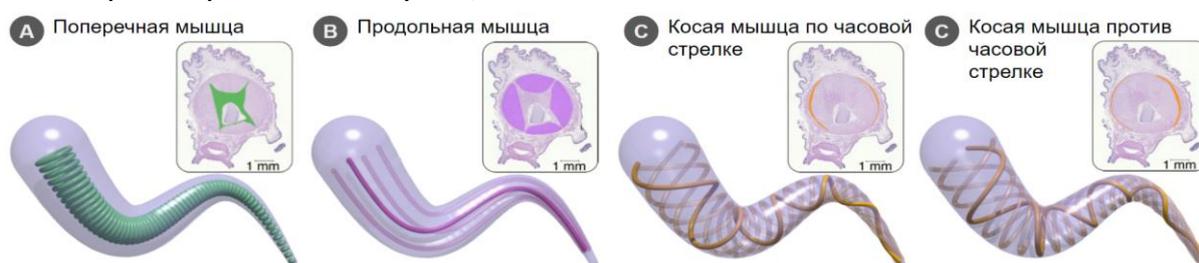


Рис. 20. Три вида мышц в щупальце *octopus rubescens*

Эти три группы мышц совместно обеспечивают осьминогу исключительную гибкость и возможность выполнения разнообразных движений и действий.

Осьминоги имеют гидростатический скелет, который использует несжимаемую жидкость внутри тела для поддержания его формы и создания структуры для движения. В отличие от твердых скелетов, таких как у позвоночных или насе-

комых, в гидростатическом скелете несущими элементами являются полости, заполненные несжимаемой жидкостью (водой), которая создает внутреннее давление, поддерживающее форму тела.

По этой причине кинетическая модель должна содержать каркас, являющийся своего рода аналогом цилиндрических полостей, заполненных жидкостью под давлением. Поиск возможных решений в редакторе приводит к следующему представлению пространственной программной модели (рисунок 21).

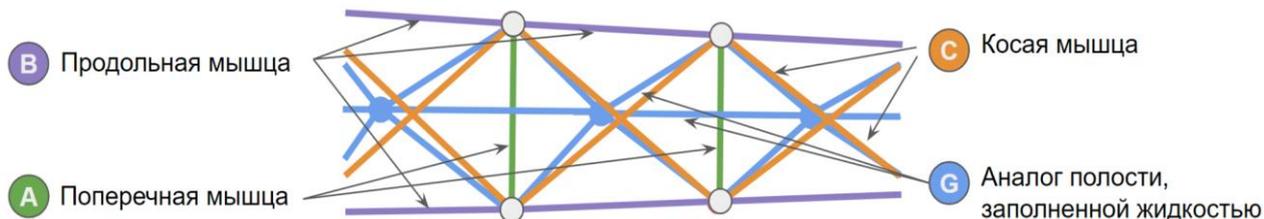


Рис. 21. Вариант аналога гидростатического скелета из nodes и springs

В полной мере возможности предлагаемой биоинспирированной нейрo-архитектуры воплощенного интеллекта раскрываются в случае реализации сенсорного восприятия на внешнее раздражение, с учетом физических особенностей реального мира.

Экзистенциальной особенностью предложенной нейросетевой архитектуры является способность проявлять свою активность с сохранением физической стабильности модели аналогично биологическим системам, поддерживающим гомеостаз. Гомеостатические сети способны адаптироваться к изменяющимся условиям и реагировать на внешние стимулы существенно гибче и эффективнее, так как подразумевают постоянное нахождение баланса между составными частями и внешней средой. В свою очередь, необходимость сенсорного восприятия для достижения требуемой реакции может находить разные формы влияния на параметры текущего гомеостаза. В процессе эволюции могло возникнуть множество видов влияния на нервную систему организма внешних раздражителей. В данной статье рассмотрена только одна из концепций – изменение параметров spring через локальную сенсорную чувствительность.

В категории параметров, формирующиеся в процессе адаптации, определен коэффициент характеризующих свободную длину springs – l_0 , его изменение полностью аналогично сокращению или расслаблению мышечных волокон, которые зависят от уровней поступающих эфферентных сигналов нервной системы. Соответственно, если сопрячь значение этого параметра с направлением нормали springs и удаленностью от источника раздражения, то мы получим модель, направляющую сегменты щупальца в сторону раздражителя (рисунок 22).

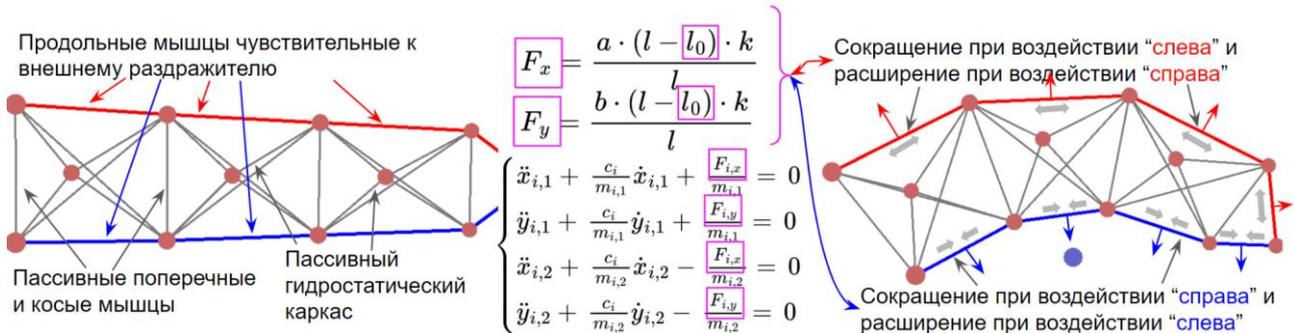


Рис. 22. Организация сенсорного восприятия springs

Стоит отметить, что модель является примером воплощенного или своего рода роевого интеллекта, у которой нет центрального аппарата, координирующего сокращения каждой мышцы, решение принимается локально, но в итоге мы получаем осмысленное общее поведение, при этом важнейшим фактором является учет физического мира и возможность синхронизации с реальным состоянием сегментов. Кроме того, здесь показано, насколько легко с эволюционной точки зрения могла сформироваться нервная система организмов, позволяющая оптимально реагировать на изменения в окружающей среде.

Ниже приведены скриншоты реагирования модели на внешний раздражитель в случае реализации в software engine рассмотренного локального сенсорного восприятия (рисунок 23). По указанной ссылке можно перейти к записи реакции модели щупальца во времени [56].

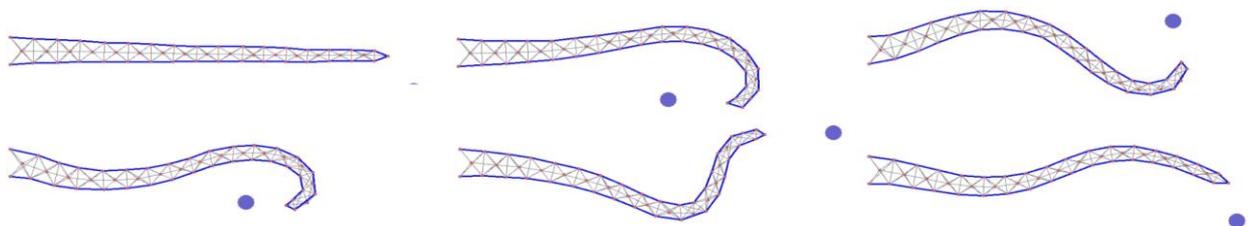


Рис. 26. Реакция нейромодели щупальца на раздражитель

В приведенном примере реализована самая простая модель сенсорного реагирования, по всей видимости, в случае наращивания сложности и дообучения нейросети можно получить гораздо более разнообразные и эффективные формы реагирования.

12. ЗАКЛЮЧЕНИЕ

Изначально интеллект формировался как воплощенный в части механизма управления телом, нервная система стала биологическим аналоговым компьютером, моделирующим механические процессы в нем с учетом особенностей окружающей среды.

Безусловно, частое упоминание аналоговых вычислительных машин ведет к вопросу об их итоговой неконкурентоспособности цифровым компьютерам, оказавшим революционное влияние на технический прогресс XXI века. Однако цифровые машины, несмотря на грандиозный прогресс, колоссальную вычислительную мощность и размеры памяти, до сих пор не могут конкурировать с биологическим интеллектом человека при решении подавляющего большинства прикладных задач и повсеместно замещать его, особенно в областях физического труда и социальных задач.

Воплощенное познание – концепция, предполагающая глубокую зависимость интеллекта от особенностей физического агента, то есть, когда аспекты тела за пределами мозга играют значительную причинную или физически конститутивную роль в когнитивной обработке [57]. В то же время на определенном этапе эволюции высшие разделы нервной системы обрели способности, которые мы совсем не отождествляем с физическим телом и воплощенным познанием, но в своей основе даже мозг человека остается все той же кибернетической системой адаптации тела к окружающей среде.

Обширная нервная система развивалась в первую очередь ради необходимости справляться с координацией тела, но в итоге усложнилась настолько, что возникли другие способности, по сути, как «побочный продукт». По всей видимости, некоторые способности, например распознавание человеческих лиц, могут быть своего рода таким «побочным продуктом». Тогда как прочие, например, решение задач, являются результатом эволюционных изменений мозга, связанных с занятием организмами высокорисковых экологических ниш [58].

По всей видимости, биологические аналоговые компьютеры в силу эволюционного развития способны быстро формировать модели, поведение которых аналогично сложным физическим системам как самого организма, так и внешним объектам. Так или иначе это означает, что они способны решать системы дифференциальных уравнений в миллионы раз большей размерности, чем лучшие из последних образцов аналоговых вычислительных машин. Кроме того, как показано в рассмотренной нейроархитектуре, модели могут легко автосинхронизировать свои параметры с состоянием реального объекта, прогнозируя его поведение с небольшой, но критически важной для выживания глубиной во времени.

Важнейшей частью создания воплощенного интеллекта является подбор ключевых, долгосрочных параметров, которые по сути и формируют окончательную модель, а с математической точки зрения, нахождение оптимальных коэффициентов в системах дифференциальных уравнений в процессе обучения. При этом процесс обучения не основан на использовании классических датасетов и предполагает использование симуляторов физической реальности, что требует совершенно других инструментов программирования воплощенного интеллекта, совмещающих в себе средства создания пространственных моделей, программные движки (software engines) и симуляторы среды. Совместно они позволяют осуществлять морфологическое программирование распределенных элементов, создавать цепи сенсорного восприятия, обучать модели с использованием методов градиентного спуска, проводить первичный и последующий эволюционный подбор их базовых параметров с использованием генетических алгоритмов [19].

На этапе эволюционного обучения моделей агентов воплощенного интеллекта программные движки и симуляторы создают виртуальные среды, максимально приближенные к реальным условиям, включая сопрягаемые электромеханические системы. Это позволяет агентам с минимальными затратами приобретать базовые навыки и адаптироваться к среде до их развертывания в физическом мире. Механизмы генетического эволюционного отбора должны позволять модифицировать физические формы и структуры агентов, анализируя их

адаптивность и приемлемость скорости оптимизации параметров. По всей видимости, ключевым фактором успеха является не столько сама архитектура, сколько наличие на всех этапах инструментов симуляции должного уровня для формирования и программирования воплощенного интеллекта. По сути, эти инструменты представляют собой вычислительные аналоги биологических экологических ниш, которые требуют создания аналоговых систем иерархически более высокого порядка, чем сами модели организмов, развивающиеся в них.

СПИСОК ЛИТЕРАТУРЫ

1. *Bubeck S. et al.* Sparks of artificial general intelligence: Early experiments with gpt-4 // arXiv preprint arXiv:2303.12712. 2023.
2. *Achiam J. et al.* Gpt-4 technical report // arXiv preprint arXiv:2303.08774. 2023.
3. *Brown T.B.* Language models are few-shot learners // arXiv preprint arXiv:2005.14165. 2020.
4. TOP 500. June 2024
URL: <https://www.top500.org/lists/top500/2024/06/>. 2024
5. *Paul A.* OpenAI argues it is 'impossible' to train ChatGPT without copyrighted work // Popular Science. 2024.
6. *Olivera F.* Memory systems in organizations: an empirical investigation of mechanisms for knowledge collection, storage and access // Journal of Management Studies. 2000. Vol. 37, No. 6. P. 811–832.
7. *Chen H., Vondrick C., Mao C.* Selfie: Self-interpretation of large language model embeddings // arXiv preprint arXiv:2403.10949. 2024.
8. *Marks S., Tegmark M.* The geometry of truth: Emergent linear structure in large language model representations of true/false datasets // arXiv preprint arXiv:2310.06824. 2023.
9. *Abramski K. et al.* Cognitive network science reveals bias in gpt-3, gpt-3.5 turbo, and gpt-4 mirroring math anxiety in high-school students // Big Data and Cognitive Computing. 2023. Vol. 7, No. 3. P. 124.
10. *Crowder J.A. et al.* Implicit learning in artificial intelligence // Artificial Psychology: Psychological Modeling and Testing of AI Systems. 2020. P. 139–147.

11. *Seger C.A.* Implicit learning // *Psychological bulletin*. 1994. Vol. 115, No. 2. P. 163–196.
12. *Olivera F.* Memory systems in organizations: an empirical investigation of mechanisms for knowledge collection, storage and access // *Journal of Management Studies*. 2000. Vol. 37, No. 6. P. 811–832.
13. *Rajan S., Pantelyat A.* Apraxias // *eLS*. 2018. P. 1–11.
14. *Goldenberg G.* Apraxia and the parietal lobes // *Neuropsychologia*. 2009. Vol. 47, No. 6. P. 1449–1459.
15. *Gardner H.E.* *Frames of mind: The theory of multiple intelligences*. Basic books, 2011.
16. *Sitti M.* Physical intelligence as a new paradigm // *Extreme Mechanics Letters*. 2021. Vol. 46. P. 101340.
17. Федеральная служба государственной статистики (2021) Рабочая сила, занятость и безработица в России в 2021. Статистический сборник. Таблица 2.33.
18. Nvidia CEO Jensen Huang says robots are the next wave of AI 2024
19. *Duan J. et al.* A survey of embodied ai: From simulators to research tasks // *IEEE Transactions on Emerging Topics in Computational Intelligence*. 2022. Vol. 6, No. 2. P. 230–244.
20. *Wilson R.A., Foglia L.* Embodied Cognition, in «*The Stanford Encyclopedia of Philosophy*». 2017.
21. *Cangelosi A. et al.* Embodied intelligence // *Springer handbook of computational intelligence*. 2015. P. 697–714.
22. *Jékely G., Keijzer F., Godfrey-Smith P.* An option space for early neural evolution // *Philosophical Transactions of the Royal Society B: Biological Sciences*. 2015. Vol. 370, No. 1684. P. 20150181.
23. *Иргер И.М.* *Клиника и хирургическое лечение опухолей мозжечка*. М.: Медгиз, 1959.
24. *Itō M.* *The cerebellum and neural control*. Raven press, 1984.
25. *Kawato M. et al.* 50 years since the Marr, Ito, and Albus models of the cerebellum // *Neuroscience*. 2021. Vol. 462. P. 151–174.
26. *Kalinichenko S.G., Motavkin P.A.* *Cerebellar Cortex*; Institute Marine Biology FEB RAS. Moscow: Nauka, 2005.

27. *Tanaka H. et al.* The cerebro-cerebellum as a locus of forward model: a review // *Frontiers in systems neuroscience*. 2020. Vol. 14. P. 19.
28. *Miall R.C., Wolpert D.M.* Forward models for physiological motor control // *Neural networks*. 1996. Vol. 9, No. 8. P. 1265–1279.
29. *Shadmehr R., Smith M.A., Krakauer J.W.* Error correction, sensory prediction, and adaptation in motor control // *Annual review of neuroscience*. 2010. Vol. 33, No. 1. P. 89–108.
30. *Habas C. et al.* Deep cerebellar nuclei (DCN) and language // *The linguistic cerebellum*. Academic Press, 2016. P. 337–353.
31. *Kalman R.E.* A New Approach to Linear Filtering and Prediction Problems // *Journal of Basic Engineering*. 1960. Vol. 82, No. 1. P. 35.
32. *Thomas A.* Cerebellar functions [Electronic resource] / André Thomas. New York: Journal of Nervous and Mental Disease Pub. Co. 1912. P. 244, Figure 67 (Monograph series; issue 12).
33. *Herculano-Houzel S.* Coordinated scaling of cortical and cerebellar numbers of neurons // *Frontiers in neuroanatomy*. 2010. Vol. 4. P. 952.
34. *Funahashi K., Nakamura Y.* Approximation of dynamical systems by continuous time recurrent neural networks // *Neural networks*. 1993. Vol. 6, No. 6. P. 801–806.
35. *Арнольд В.И.* Обыкновенные дифференциальные уравнения. МЦНМО. 2014. С. 5.
36. *Consalez G.G. et al.* Origins, development, and compartmentation of the granule cells of the cerebellum // *Frontiers in neural circuits*. 2021. Vol. 14. P. 611841.
37. Universidad Complutense de Madrid. Cuadros originales delineados por Santiago Ramón y Cajal y pintados por Ramón Padró. Corteza cerebelosa. URL: <https://www.ucm.es/seccionbiocel/historia>
38. *De Schutter E., Vos B., Maex R.* The function of cerebellar Golgi cells revisited // *Progress in brain research*. 2000. Vol. 124. P. 81–93.
39. *D'Angelo E. et al.* The cerebellar Golgi cell and spatiotemporal organization of granular layer activity // *Frontiers in neural circuits*. 2013. Vol. 7. P. 93.
40. *Maass W.* Networks of spiking neurons: the third generation of neural network models // *Neural networks*. 1997. Vol. 10, No. 9. P. 1659–1671.

41. *Izhikevich E.M., Hoppensteadt F.* Classification of bursting mappings // International Journal of Bifurcation and Chaos. 2004. Vol. 14, No. 11. P. 3847–3854.
42. *Izhikevich E.M.* Dynamical systems in neuroscience: The geometry of excitability and bursting. MIT press, 2007.
43. *Small J.S.* The analogue alternative: The electronic analogue computer in Britain and the USA, 1930–1975. Routledge, 2013.
44. *Чукуров Н.Г.* Численный метод решения обыкновенных дифференциальных уравнений с помощью приведения их к форме Шеннона // Матем. моделирование. 2020. Т. 32, № 8. С. 3–20.
45. *Marullo C., Agliari E.* Boltzmann machines as generalized Hopfield networks: a review of recent results and outlooks // Entropy. 2020. Vol. 23, No. 1. P. 34.
46. *Nurutdinov A.R., Latypov R.K.* A Novel Artificial Neural Networks Architecture Based on Cerebellum Model // Lobachevskii Journal of Mathematics. 2023. Vol. 44, No. 2. P. 733–746.
47. *Nurutdinov A., Latypov R.* Dynamically Predicting Wi-Fi Coverage Mapping Using Bioinspired Neural Networks // 2023 International Conference on Software, Telecommunications and Computer Networks (SoftCOM). IEEE, 2023. P. 1–6.
48. *Gupta A. et al.* Embodied intelligence via learning and evolution // Nature communications. 2021. Vol. 12, No. 1. P. 5721.
49. *Zullo L. et al.* Motor control pathways in the nervous system of Octopus vulgaris arm // Journal of Comparative Physiology A. 2019. Vol. 205. P. 271–279.
50. *Ландау Л.Д., Лифшиц Е.М.* Теоретическая физика. Издание 5-е, стереотипное. М.: Физматлит. 2004. Т. I. Механика. С. 26.
51. Cerebellum cytoarchitecture for solving the equation of harmonic oscillations. URL: <https://github.com/airatn/Cerebellum-cytoarchitecture-for-solving-the-equation-of-harmonic-oscillations>
52. *Муханов К.К.* Область применения и классификации ферм // Металлические конструкции. 1976. С. 287–293.
53. *Перельмутер А.В.* Очерки по истории металлических конструкций. Издательство АСВ, 2012.
54. ferms. URL: <https://youtu.be/GVK1D8-qoIE>

55. *Chang H.S. et al.* Energy-shaping control of a muscular octopus arm moving in three dimensions // *Proceedings of the Royal Society A*. 2023. Vol. 479, No. 2270. P. 20220593.

56. *tentacle*. URL: <https://youtu.be/en6EwpKJr6U>

57. *Wilson R.A., Foglia L.* Embodied Cognition, in «The Stanford Encyclopedia of Philosophy», 2017.

58. *Godfrey-Smith P.* Other minds: The octopus, the sea, and the deep origins of consciousness. Farrar, Straus and Giroux, 2016.

NEURAL NETWORK ARCHITECTURE OF EMBODIED INTELLIGENCE

A. R. Nurutdinov^[0009-0007-0918-0879]

PJSC "Tattelecom" (Embodied Artificial Intelligence Laboratory), Kazan, Russia
ayrat.nurutdinov@gmail.com

Abstract

In recent years, advances in artificial intelligence (AI) and machine learning have been driven by advances in the development of large language models (LLMs) based on deep neural networks. At the same time, despite its substantial capabilities, LLMs have fundamental limitations such as spontaneous unreliability in facts and judgments; making simple errors that are dissonant with high competence in general; credulity, manifested by a willingness to accept a user's knowingly false claims as true; and lack of knowledge about events that have occurred after training has been completed.

Probably the key reason is that bioinspired intelligence learning occurs through the assimilation of implicit knowledge by an embodied form of intelligence to solve interactive real-world physical problems. Bioinspired studies of the nervous systems of organisms suggest that the cerebellum, which coordinates movement and maintains balance, is a prime candidate for uncovering methods for realizing embodied physical intelligence. Its simple repetitive structure and ability to control complex movements offer hope for the possibility of creating an analog to adaptive neural networks.

This paper explores the bioinspired architecture of the cerebellum as a form of analog computational networks capable of modeling complex real-world physical systems. As a simple example, a realization of embodied AI in the form of a multi-component model of an octopus tentacle is presented, demonstrating the potential in creating adaptive physical systems that learn and interact with the environment.

Keywords: *artificial neural network, large language model, implicit learning, cerebellum model, analog computing, embodied cognition, soft robotics, octopus,.*

REFERENCES

1. *Bubeck S. et al.* Sparks of artificial general intelligence: Early experiments with gpt-4 // arXiv preprint arXiv:2303.12712. 2023.
2. *Achiam J. et al.* Gpt-4 technical report // arXiv preprint arXiv:2303.08774. 2023.
3. *Brown T.B.* Language models are few-shot learners // arXiv preprint arXiv:2005.14165. 2020.
4. TOP 500. June 2024
URL: <https://www.top500.org/lists/top500/2024/06/>. 2024
5. *Paul A.* OpenAI argues it is 'impossible' to train ChatGPT without copyrighted work // Popular Science. 2024.
6. *Olivera F.* Memory systems in organizations: an empirical investigation of mechanisms for knowledge collection, storage and access // Journal of Management Studies. 2000. Vol. 37, No. 6. P. 811–832.
7. *Chen H., Vondrick C., Mao C.* Selfie: Self-interpretation of large language model embeddings // arXiv preprint arXiv:2403.10949. 2024.
8. *Marks S., Tegmark M.* The geometry of truth: Emergent linear structure in large language model representations of true/false datasets // arXiv preprint arXiv:2310.06824. 2023.
9. *Abramski K. et al.* Cognitive network science reveals bias in gpt-3, gpt-3.5 turbo, and gpt-4 mirroring math anxiety in high-school students // Big Data and Cognitive Computing. 2023. Vol. 7, No. 3. P. 124.
10. *Crowder J.A. et al.* Implicit learning in artificial intelligence // Artificial Psychology: Psychological Modeling and Testing of AI Systems. 2020. P. 139–147.

11. *Seger C.A.* Implicit learning // *Psychological bulletin*. 1994. Vol. 115, No. 2. P. 163–196.
12. *Olivera F.* Memory systems in organizations: an empirical investigation of mechanisms for knowledge collection, storage and access // *Journal of Management Studies*. 2000. Vol. 37, No. 6. P. 811–832.
13. *Rajan S., Pantelyat A.* Apraxias // *eLS*. 2018. P. 1–11.
14. *Goldenberg G.* Apraxia and the parietal lobes // *Neuropsychologia*. 2009. Vol. 47, No. 6. P. 1449–1459.
15. *Gardner H.E.* *Frames of mind: The theory of multiple intelligences*. Basic books, 2011.
16. *Sitti M.* Physical intelligence as a new paradigm // *Extreme Mechanics Letters*. 2021. Vol. 46. P. 101340.
17. Federal'naya sluzhba gosudarstvennoj statistiki (2021) *Rabochaya sila, zanyatost' i bezrabotica v Rossii v 2021*. Statisticheskij sbornik. Tablica 2.33.
18. Nvidia CEO Jensen Huang says robots are the next wave of AI 2024
19. *Duan J. et al.* A survey of embodied ai: From simulators to research tasks // *IEEE Transactions on Emerging Topics in Computational Intelligence*. 2022. Vol. 6, No. 2. P. 230–244.
20. *Wilson R.A., Foglia L.* Embodied Cognition, in «*The Stanford Encyclopedia of Philosophy*». 2017.
21. *Cangelosi A. et al.* Embodied intelligence // *Springer handbook of computational intelligence*. 2015. P. 697–714.
22. *Jékely G., Keijzer F., Godfrey-Smith P.* An option space for early neural evolution // *Philosophical Transactions of the Royal Society B: Biological Sciences*. 2015. Vol. 370, No. 1684. P. 20150181.
23. *Irger I.M.* *Klinika i hirurgicheskoe lechenie opuholej mozzhechka*. M.: Medgiz, 1959.
24. *Itō M.* *The cerebellum and neural control*. Raven press, 1984.
25. *Kawato M. et al.* 50 years since the Marr, Ito, and Albus models of the cerebellum // *Neuroscience*. 2021. Vol. 462. P. 151–174.
26. *Kalinichenko S.G., Motavkin P.A.* *Cerebellar Cortex*; Institute Marine Biology FEB RAS. Moscow: Nauka, 2005.

27. *Tanaka H. et al.* The cerebro-cerebellum as a locus of forward model: a review // *Frontiers in systems neuroscience*. 2020. Vol. 14. P. 19.
28. *Miall R.C., Wolpert D.M.* Forward models for physiological motor control // *Neural networks*. 1996. Vol. 9, No. 8. P. 1265–1279.
29. *Shadmehr R., Smith M.A., Krakauer J.W.* Error correction, sensory prediction, and adaptation in motor control // *Annual review of neuroscience*. 2010. Vol. 33, No. 1. P. 89–108.
30. *Habas C. et al.* Deep cerebellar nuclei (DCN) and language // *The linguistic cerebellum*. Academic Press, 2016. P. 337–353.
31. *Kalman R.E.* A New Approach to Linear Filtering and Prediction Problems // *Journal of Basic Engineering*. 1960. Vol. 82, No. 1. P. 35.
32. *Thomas A.* Cerebellar functions [Electronic resource] / André Thomas. New York: Journal of Nervous and Mental Disease Pub. Co. 1912. P. 244, Figure 67 (Monograph series; issue 12).
33. *Herculano-Houzel S.* Coordinated scaling of cortical and cerebellar numbers of neurons // *Frontiers in neuroanatomy*. 2010. Vol. 4. P. 952.
34. *Funahashi K., Nakamura Y.* Approximation of dynamical systems by continuous time recurrent neural networks // *Neural networks*. 1993. Vol. 6, No. 6. P. 801–806.
35. *Arnol'd V.I.* Obyknovennye differencial'nye uravneniya. M.: MCNMO, 2014. S. 5.
36. *Consalez G.G. et al.* Origins, development, and compartmentation of the granule cells of the cerebellum // *Frontiers in neural circuits*. 2021. Vol. 14. P. 611841.
37. Universidad Complutense de Madrid. Cuadros originales delineados por Santiago Ramón y Cajal y pintados por Ramón Padró. Corteza cerebelosa. URL: <https://www.ucm.es/seccionbiocel/historia>
38. *De Schutter E., Vos B., Maex R.* The function of cerebellar Golgi cells revisited // *Progress in brain research*. 2000. Vol. 124. P. 81–93.
39. *D'Angelo E. et al.* The cerebellar Golgi cell and spatiotemporal organization of granular layer activity // *Frontiers in neural circuits*. 2013. Vol. 7. P. 93.
40. *Maass W.* Networks of spiking neurons: the third generation of neural network models // *Neural networks*. 1997. Vol. 10, No. 9. P. 1659–1671.

41. *Izhikevich E.M., Hoppensteadt F.* Classification of bursting mappings // International Journal of Bifurcation and Chaos. 2004. Vol. 14, No. 11. P. 3847–3854.
42. *Izhikevich E.M.* Dynamical systems in neuroscience: The geometry of excitability and bursting. MIT press, 2007.
43. *Small J.S.* The analogue alternative: The electronic analogue computer in Britain and the USA, 1930–1975. Routledge, 2013.
44. *Chikurov N.G.* Chislennyj metod resheniya obyknovennyh differencial'nyh uravnenij s pomoshch'yu privedeniya ih k forme SHennona // Matem. Modelirovanie. 2020. T.32, № 8. S. 3–20.
45. *Marullo C., Agliari E.* Boltzmann machines as generalized Hopfield networks: a review of recent results and outlooks // Entropy. 2020. Vol. 23, No. 1. P. 34.
46. *Nurutdinov A.R., Latypov R.K.* A Novel Artificial Neural Networks Architecture Based on Cerebellum Model // Lobachevskii Journal of Mathematics. 2023. Vol. 44, No. 2. P. 733–746.
47. *Nurutdinov A., Latypov R.* Dynamically Predicting Wi-Fi Coverage Mapping Using Bioinspired Neural Networks // 2023 International Conference on Software, Telecommunications and Computer Networks (SoftCOM). IEEE, 2023. P. 1–6.
48. *Gupta A. et al.* Embodied intelligence via learning and evolution // Nature communications. 2021. Vol. 12, No. 1. P. 5721.
49. *Zullo L. et al.* Motor control pathways in the nervous system of Octopus vulgaris arm // Journal of Comparative Physiology A. 2019. Vol. 205. P. 271–279.
50. *Landau L.D., Lifshic E.M.* Teoreticheskaya fizika. Izdanie 5-e, stereotipnoe. M.: Fizmatlit, 2004. T. I. Mekhanika. S. 26.
51. Cerebellum cytoarchitecture for solving the equation of harmonic oscillations. URL: <https://github.com/airatn/Cerebellum-cytoarchitecture-for-solving-the-equation-of-harmonic-oscillations>
52. *Muhanov K.K.* Oblast' primeneniya i klassifikacii ferm // Metallicheskie konstrukcii. 1976. S. 287–293.
53. *Perel'muter A.V.* Ocherki po istorii metallicheskih konstrukcij. Izdatel'stvo ASV, 2012.
54. ferm. URL: <https://youtu.be/GVK1D8-qoE>

55. *Chang H.S. et al.* Energy-shaping control of a muscular octopus arm moving in three dimensions // *Proceedings of the Royal Society A*. 2023. Vol. 479, No. 2270. P. 20220593.

56. tentacle. URL: <https://youtu.be/en6EwpKJr6U>

57. *Wilson R.A., Foglia L.* Embodied Cognition, in «*The Stanford Encyclopedia of Philosophy*», 2017.

58. *Godfrey-Smith P.* Other minds: The octopus, the sea, and the deep origins of consciousness. Farrar, Straus and Giroux, 2016.

СВЕДЕНИЯ ОБ АВТОРЕ



НУРУТДИНОВ Айрат Рафкатович – Генеральный директор ПАО «Таттелеком», руководитель лаборатории воплощенного искусственного интеллекта. Основные направления научных исследований: воплощенные формы искусственного интеллекта, архитектуры нейронных сетей, компьютерное моделирование.

Ayrat Rafkatovich NURUTDINOV – General Director of PJSC Tattelcom, Head of the Laboratory of embodied artificial intelligence. Main areas of research: embodied forms of artificial intelligence, neural network architectures and computer simulation.

email: ayrat.nurutdinov@gmail.com

ORCID: 0009-0007-0918-0879

Материал поступил в редакцию 12 июня 2024 года

АВТОМАТИЗАЦИЯ ЧТЕНИЯ СВЯЗАННЫХ ДАННЫХ ИЗ РЕЛЯЦИОННЫХ И НЕРЕЛЯЦИОННЫХ БАЗ ДАННЫХ В КОНТЕКСТЕ ИСПОЛЬЗОВАНИЯ СТАНДАРТА JPA

А. С. Савинчева¹ [0009-0005-0756-6001], А. А. Ференец² [0000-0002-7859-9901]

^{1, 2}Институт информационных технологий и интеллектуальных систем
Казанского федерального университета, ул. Кремлевская, 35, г. Казань, 420008

¹asanvlit@gmail.com, ²ist.kazan@gmail.com

Аннотация

Описан процесс автоматизации управления операцией чтения связанных данных из реляционных и нереляционных баз данных.

Разработанный программный инструмент основан на использовании стандарта JPA (Java Persistence API), который определяет возможности контроля жизненного цикла сущностей в Java-приложениях. Спроектирована архитектура встраивания в событийные процессы, позволяющая интегрировать решение в проекты вне зависимости от используемой реализации JPA. Реализована поддержка различных стратегий загрузки данных, типов и параметров отношений. Осуществлена оценка производительности инструмента.

Ключевые слова: JPA, ORM, Java, базы данных, реляционные базы данных, нереляционные базы данных.

ВВЕДЕНИЕ

В настоящее время одним из ключевых этапов разработки программного обеспечения является взаимодействие с базами данных. Применение разнообразных типов баз данных, таких как реляционные и нереляционные, позволяет использовать уникальные особенности и преимущества каждого типа для реализации различных задач, однако требует специализированных подходов к работе. На практике разработчики могут часто сталкиваться с необходимостью взаимодействия с реляционными и нереляционными базами данных одновременно в рамках одного проекта. Это требует реализации соответствующей программной логики для контроля отношений между данными из различных хранилищ.

В Java-проектах для управления отображением объектной модели программы на записи в реляционной базе данных введен свой стандарт – JPA [1]. Существующие реализации JPA не всегда обеспечивают поддержку работы с отношениями между сущностями из различных типов баз данных. Так, Hibernate, который является одним из самых популярных JPA-фреймворков [2], позволяет управлять связями между сущностями в рамках одной базы данных, но не предоставляет возможностей обработки отношений между сущностями, хранящимися в различных типах баз данных [3].

В данной работе описан процесс разработки программного инструмента на основе стандарта JPA, автоматизирующего процесс управления связанными сущностями из реляционных и нереляционных баз данных в Java-приложениях. Ценность следования стандарту заключается в обеспечении независимости от реализации JPA, применяющейся в пользовательском проекте, что позволяет использовать уникальные преимущества, предоставляемые поставщиком, и не осуществлять интеграцию для перехода на другой инструмент. Работа охватывает исследование возможностей JPA для управления жизненным циклом сущностей, проектирование стратегии встраивания инструмента в событийные процессы, разработку механизмов контроля отношений и оценку производительности программного решения.

1. ОБЪЕКТНО-РЕЛЯЦИОННОЕ ОТОБРАЖЕНИЕ: JPA И ЕГО РЕАЛИЗАЦИИ

Объектно-реляционное отображение (ORM) – методология управления данными, предназначенная для связи объектно-ориентированных моделей с реляционными базами данных, обеспечивающая при этом автоматическое отображение информации из одной формы в другую [4]. Согласно концепции, лежащей в основе ORM, классы сущностей программы сопоставляются с таблицами базы данных, объекты – с записями, а атрибуты классов – со столбцами.

JPA – это стандарт Java для ORM. Он не является реализацией ORM, а представляет собой набор интерфейсов и правил, которые должны быть реализованы в ORM-фреймворках [1].

JPA определяет правила управления жизненным циклом сущностей, вводя понятия контекста постоянства (Persistence Context) и менеджера сущностей

(Entity Manager). Контекст постоянства представляет собой набор управляемых объектов, где отслеживаются изменения, вносимые в сущности, а менеджер сущностей обеспечивает функциональность по выполнению операций над объектами контекста и синхронизации с базой данных [1].

Hibernate является одним из наиболее известных и широко используемых ORM-фреймворков в Java, его доля на рынке составляет около 72% [5]. Он реализует базовые возможности JPA и предоставляет продвинутые механизмы, расширяющие функционал, определяемый стандартом. Например, для выполнения запросов Hibernate вводит собственный объектно-ориентированный язык HQL, который поддерживает функции, специфичные для фреймворка [6].

Hibernate обеспечивает работу с различными реляционными базами данных, включая PostgreSQL, MySQL, Oracle, Microsoft SQL Server. Для работы с нереляционными базами данных существует механизм Hibernate Object/Grid Mapper (OGM), преобразующий концепцию управления объектами из Hibernate ORM для работы с NoSQL базами данных, такими как MongoDB, Couchbase и другими [6].

В JPA введено также понятие единицы постоянства (Persistence Unit), которая представляет собой логический контейнер, содержащий информацию о подключении к базе данных и описании сущностей, используемых в приложении [1]. Hibernate и Hibernate OGM предоставляют продвинутые инструменты для работы с данными внутри одного типа баз данных, однако в них не реализована возможность построения и контроля связей между сущностями из различных типов, то есть сущностей, хранящихся в различных единицах постоянства [3]. Это приводит к необходимости самостоятельной реализации дополнительной логики для контроля целостности таких связей, что ведёт к расходу дополнительных ресурсов на разработку и поддержку.

EclipseLink – еще один ORM-фреймворк с открытым исходным кодом, реализующий JPA, его доля на рынке составляет около 13% [5]. Уникальными возможностями EclipseLink являются поддержка различных типов кэширования с настраиваемыми параметрами [7], возможность вызова собственных SQL-функций в JPQL-запросах [8], связывание Java-классов с XML и JSON-форматами [9].

EclipseLink поддерживает множество реляционных баз данных. Также он совместим с NoSQL-хранилищами, такими как MongoDB, Oracle NoSQL, XML-

файлы, JMS, Oracle AQ и JCA [10, 11]. Одной из ключевых особенностей EclipseLink является поддержка работы со связанными сущностями из различных типов баз данных. Это достигается благодаря механизму Composite Persistence Units, позволяющему объединять несколько источников данных в одну логическую композитную единицу, которая может использоваться для выполнения операций с различными хранилищами [12]. Однако создание такой единицы в EclipseLink является многоступенчатым процессом, его настройка начинается с определения XML-конфигурации, включающей в себя узлы для каждого из имеющихся источников данных с описанием схем, классов сущностей, настроек соединения, транзакций и кэш-стратегий. Далее осуществляется сборка JAR-файлов каждого из проектов, отвечающих за свою единицу постоянства, а затем источники объединяются в один итоговый проект, соответствующий композитной единице [12].

2. ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО РЕШЕНИЯ

В предыдущем разделе были проанализированы две ключевые реализации JPA: Hibernate и EclipseLink. Отмечено, что в EclipseLink реализован механизм Composite Persistence Units, обеспечивающий управление связями между сущностями, хранящимися в различных типах баз данных. Однако, как было показано, настройка этого механизма является многоступенчатой, что может усложнить разработку и поддержку приложения.

Помимо многоступенчатой конфигурации, переход на EclipseLink с другого JPA-поставщика может быть затруднен наличием следующих факторов:

- использование уникального функционала JPA-фреймворков, расширяющего базовые возможности стандарта, может быть тесно связано со специфическими требованиями проекта; переход на другую реализацию может потребовать не только доработки кода, инициированной в результате поиска альтернативных подходов в других инструментах, но и пересмотра текущих архитектурных решений;
- сообщество разработчиков, применяющих данный фреймворк [5], является достаточно узким, что ограничивает круг доступной экспертизы для решения возникающих проблем;

- объём документации может оказаться недостаточным ввиду небольшого количества практических примеров настройки.

На основании сказанного возникает одно из главных требований к разрабатываемому инструменту: он должен быть реализован на основе стандарта JPA, то есть опираться на контракт, который соблюдает каждый поставщик, что позволит обеспечить его независимость от конкретного фреймворка. Таким образом, не возникает необходимости осуществлять миграцию проекта, так как программное решение будет способно интегрироваться с любым из поставщиков, расширяя его функциональность наличием контроля отношений между связанными сущностями, хранящимися в различных типах баз данных.

Основополагающей концепцией является единообразие взаимодействия со внешними и внутренними отношениями. Внешними будем называть связи между сущностями, хранящимися в различных типах баз данных, внутренними – внутри одной базы данных. На рис. 1 представлена концепция работы библиотеки.

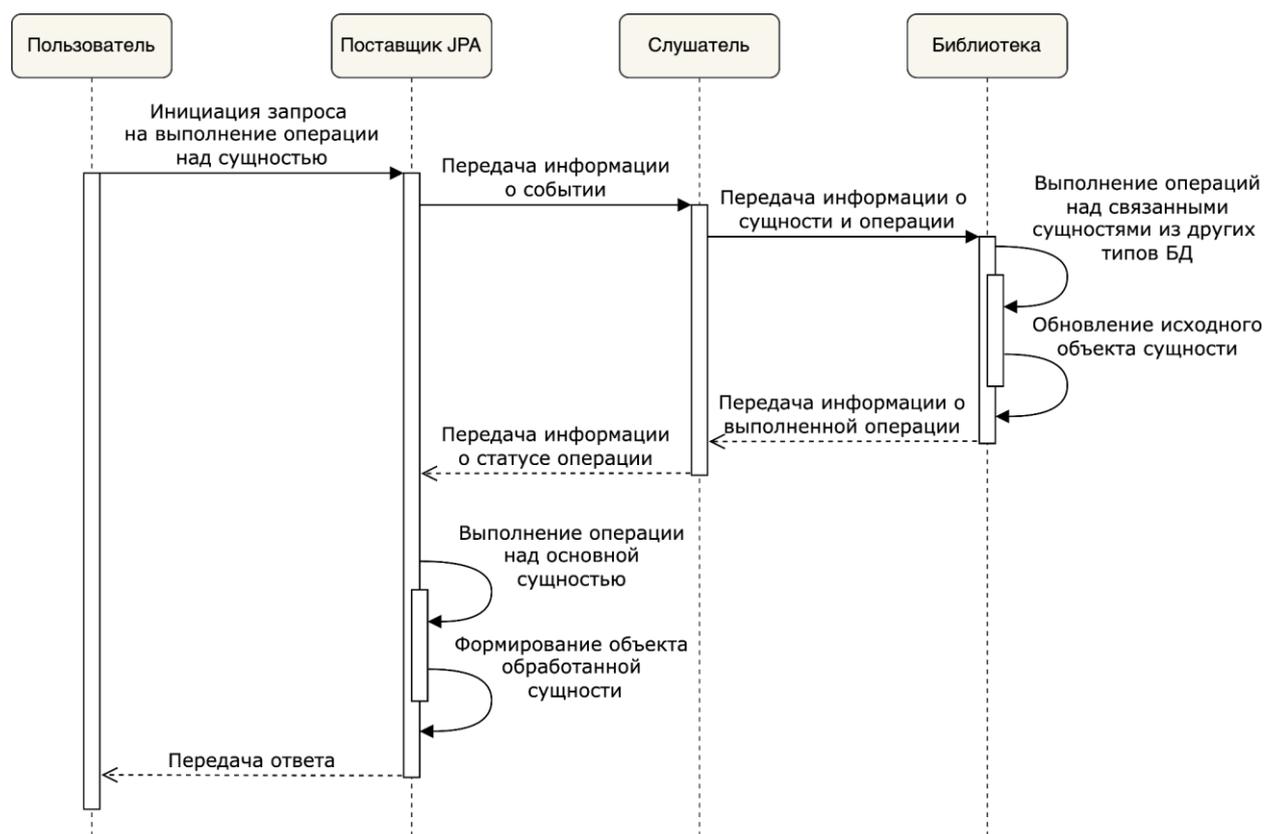


Рис. 1. Концепция работы библиотеки

Архитектура решения состоит из нескольких компонентов, осуществляющих контроль внешних отношений между сущностями. На рис. 2 изображён общий вид архитектуры библиотеки.

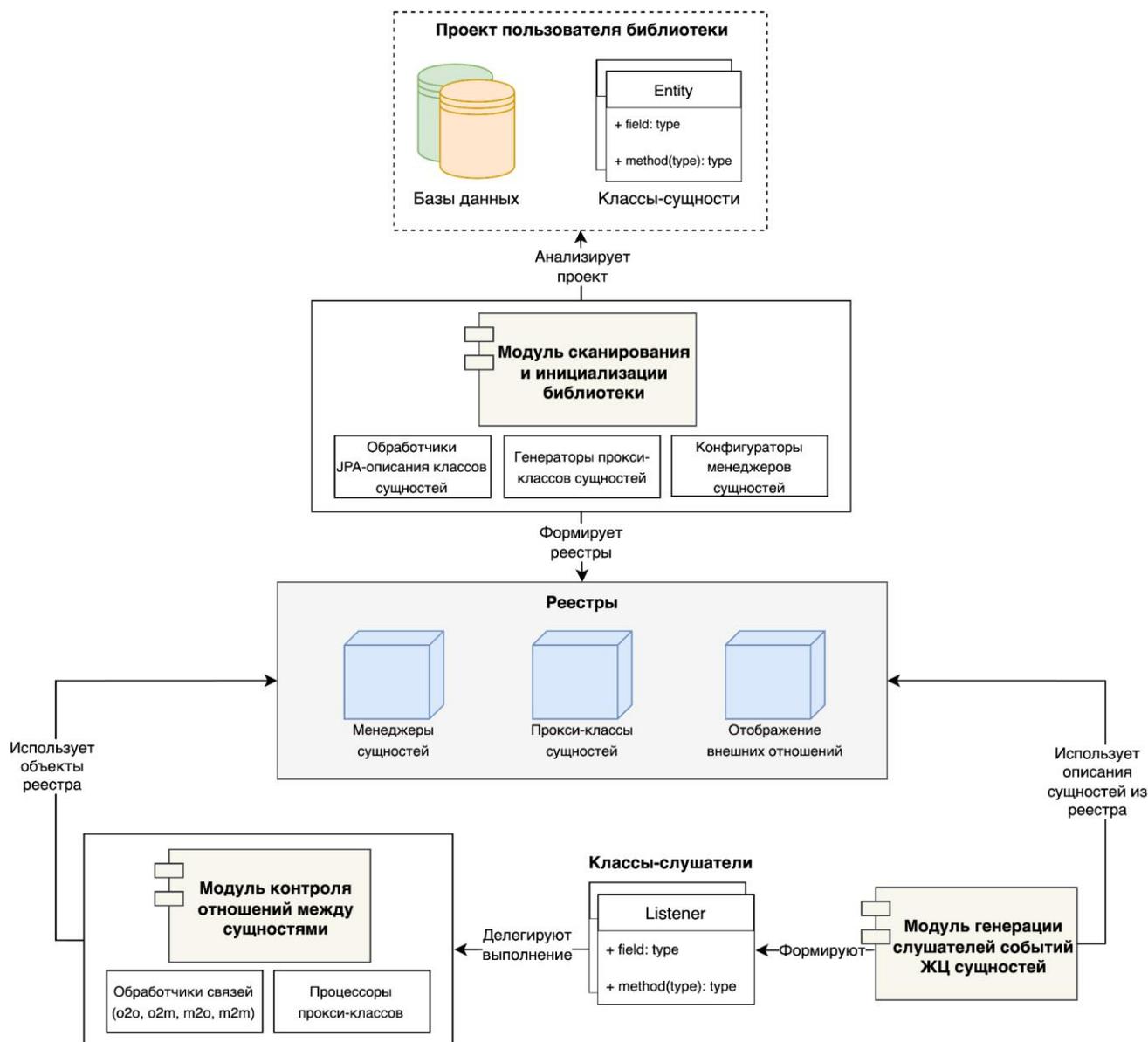


Рис. 2. Архитектура библиотеки

Модуль сканирования и инициализации анализирует структуру пользовательского проекта, определяет внешние связи между сущностями и устанавливает методы их контроля. Генераторы прокси-классов играют одну из ключевых ролей, создавая специальные классы-обёртки для сущностей из различных источников данных. Эти обёртки обеспечивают унифицированную структуру для ра-

боты с различными хранилищами данных и контроля изменений в сущностях. Результатом работы модуля являются заполненные реестры менеджеров сущностей, прокси-классов и отображений внешних связей, к которым происходит обращение в процессе жизненного цикла сущностей. На следующем этапе на основе данных из реестров генерируются классы-слушатели событий. Слушатели перехватывают вызов операций над сущностями, формируют информацию о событиях и передают её модулю контроля отношений, который является центральным узлом управления всеми внешними связями, учитывающим настройки пользовательской конфигурации. Он обеспечивает согласованность доступа к данным и выполнение операций над связанными сущностями.

3. МОДУЛЬ СКАНИРОВАНИЯ И ИНИЦИАЛИЗАЦИИ

Стандарт JPA предоставляет набор требований, определяющих объектно-ориентированное отображение сущностей на структуру в базе данных. Он вводит аннотации, такие как Entity, Table, Id и другие, используемые для обозначения JPA-сущностей и их настроек [1]. Для определения классов сущностей, хранящихся в нереляционных базах данных, в библиотеке введена аннотация NoSqlEntity.

В библиотеке реализованы сканеры, которые анализируют код проекта на предмет классов, соответствующих базовым критериям сущностей. На данном слое функционируют два таких компонента. Первый находит в проекте классы, соответствующие реляционным сущностям, второй – нереляционным, которые дополнительно помечены аннотацией NoSqlEntity, вводимой библиотекой.

Результат сканирования передаётся обработчикам соответствий. На этом этапе заполняются базовые свойства каждой сущности, такие как название таблицы, тип базы данных, уникальный ключ источника.

На последнем слое функционируют обработчики отношений, соответствующие каждому из четырех возможных типов связей. Их задача заключается в анализе отношений между сущностями и определении их конфигураций. Например, они сохраняют информацию о полях отношений, настройках чтения данных и определяют, какая из сущностей является владельцем отношения.

На рис. 3 показана схема данного процесса сканирования.

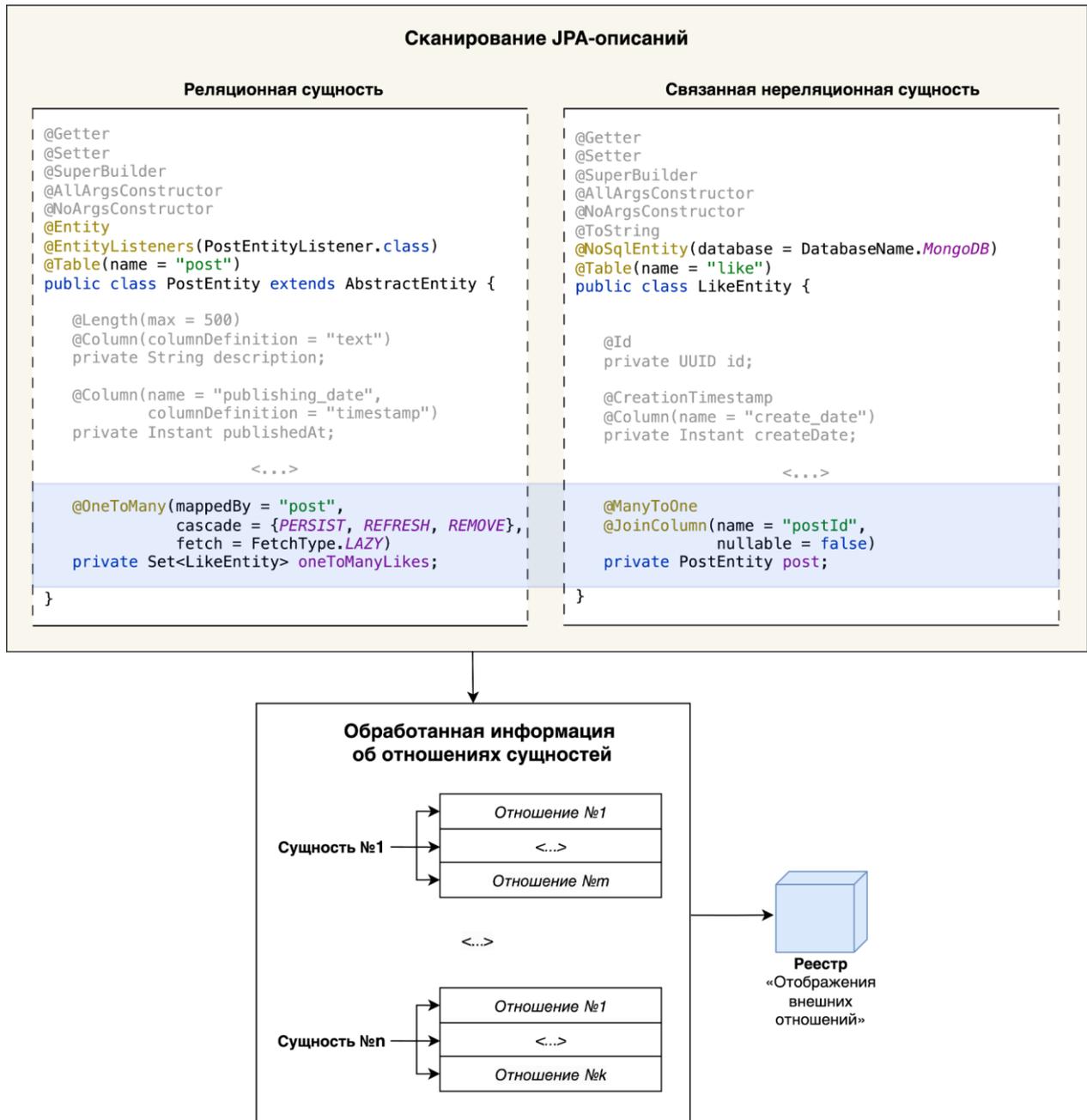


Рис. 3. Процесс сканирования JPA-описаний

На основе собранной информации формируется реестр, который служит централизованным хранилищем метаданных о всех обнаруженных сущностях, их полях, связях и конфигурации. Внутреннее устройство реестра содержит карту формата «ключ-значение», сложность операции извлечения из которой является константной. Благодаря данному первоначальному сканированию во время ра-

боты приложения происходит лишь обращение к реестру, что исключает необходимость повторного сканирования структуры этих сущностей при каждом обращении к ним.

Однако атрибуты исходных классов сущностей не соответствуют напрямую их хранимой структуре. Например, такое представление не содержит атрибутов, предназначенных для работы со внешними ключами отношения, вместо них в модели хранится объект целевого класса. Для решения этой проблемы в библиотеке реализован механизм генерации прокси-классов, которые наследуют исходную структуру сущностей, но дополнительно включают атрибуты, представляющие собой внешние ключи. Прокси является объектом, который выступает в качестве посредника между клиентским кодом и реальным объектом, тем самым обеспечивая дополнительный уровень контроля над доступом [13]. Расширенная структура прокси-класса позволяет менеджеру сущностей автоматически заполнять атрибуты внешних ключей при чтении данных, тем самым сокращая количество запросов к базе данных. Для создания прокси использовалась библиотека ByteBuddy.

На следующем этапе работы модуля происходит конфигурация менеджеров сущностей, которые представляют собой разработанный в рамках библиотеки слой абстракции, осуществляющий управление состоянием и жизненным циклом сущностей внутри контекста постоянства. Они инкапсулируют в себе уровень взаимодействия с базами данных, обеспечивая унифицированный интерфейс для работы с сущностями, независимо от типа используемой СУБД. Например, в случае MongoDB менеджер сущностей работает непосредственно с объектами `MongoCollection`, определяемыми драйвером MongoDB в Java, и реализует операции чтения, поиска и фильтрации данных посредством вызова соответствующих методов `MongoCollection`.

Созданные объекты менеджеров также регистрируются в специальном реестре для дальнейшего использования. Операции, которые требуется выполнить над связанными сущностями в течение их жизненного цикла, осуществляются посредством обращения к соответствующему менеджеру сущности из реестра.

На рис. 4 представлена схема описанного процесса создания менеджеров сущностей.

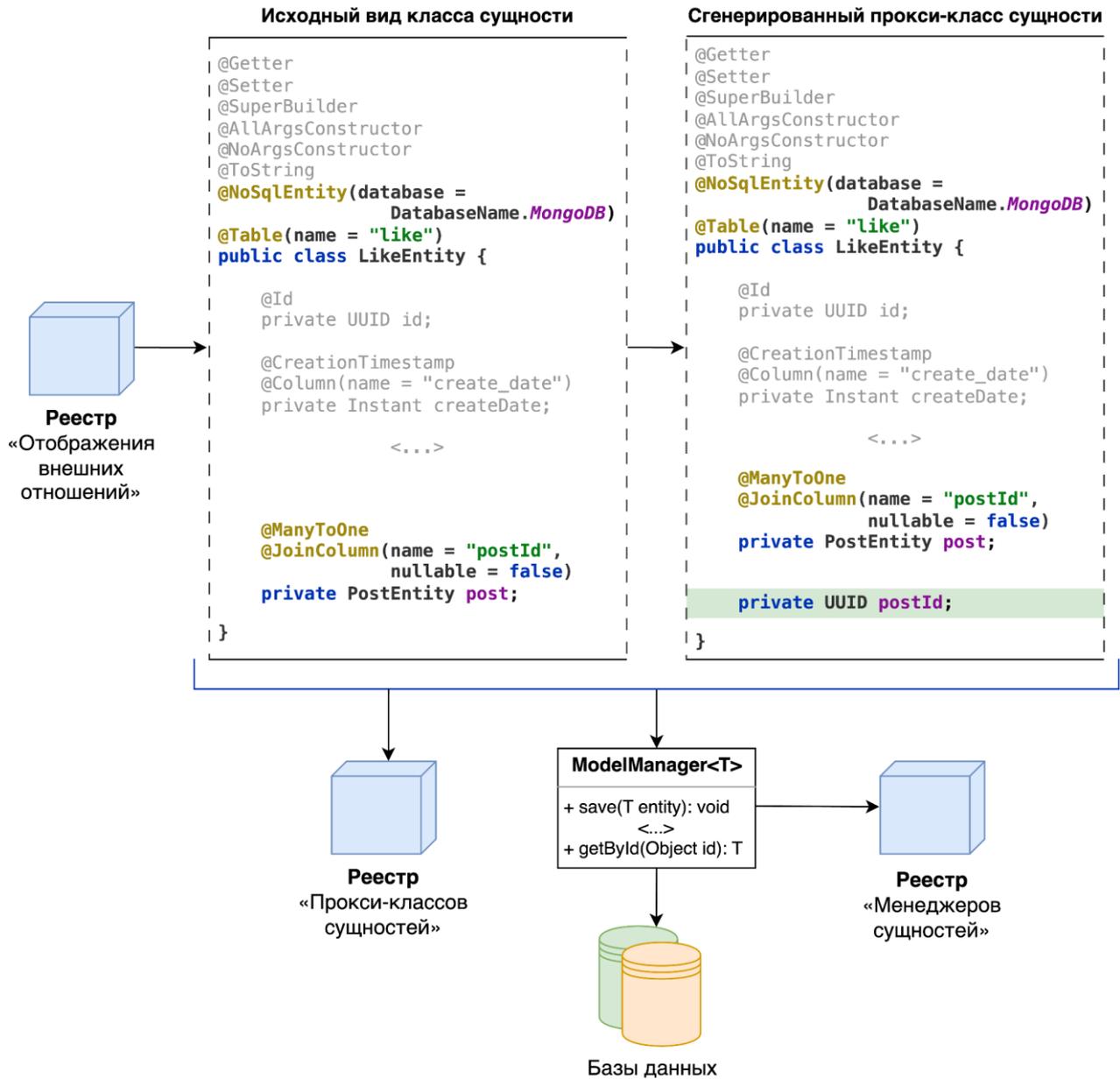


Рис. 4. Схема создания менеджеров сущностей

4. МОДУЛЬ ГЕНЕРАЦИИ СЛУШАТЕЛЕЙ СОБЫТИЙ

Слушатели – это программные компоненты, которые реагируют на наступление конкретных событий в системе и выполняют predetermined действия в ответ на них.

JPA также определяет средства для работы со слушателями событий жизненного цикла сущностей. Контроль этих событий осуществляется с помощью аннотаций и методов обратного вызова (callback-методов), привязанных к стадиям

жизненного цикла сущности [1]. В случае операции чтения данных используется аннотация `PostLoad`, позволяющая определить действие, которое будет выполняться после загрузки сущности из базы данных.

В листинге 1 приведен пример JPA-слушателя событий, содержащего метод для логирования информации о загруженной сущности.

Листинг 1. JPA-слушатель событий

```
public class EmployeeListener {  
  
    private static Log log = LogFactory.getLog(EmployeeListener.class);  
  
    @PostLoad  
    private void afterLoad(Employee employee) {  
        log.info(  
            "Employee with id loaded from database: " + employee.getId()  
        );  
    }  
}
```

В рамках решаемой задачи логика методов обратного вызова может содержать обращения к другим слоям абстракции, которые реализуют механизмы контроля внешних связей. Таким образом, встраивание в событийную модель с помощью слушателей, концепция которых определена стандартом, обеспечивает независимость от используемого в проекте поставщика JPA.

На следующем шаге работы модуля происходит непосредственная интеграция описанной модели в пользовательский проект. Существуют два подхода внедрения компонентов: на этапе компиляции и во время выполнения программы. Второй подход неприменим в данном случае, так как JPA не вводит механизмы регистрации обработчиков событий непосредственно во время работы программы. Подобные возможности могут быть предоставлены конкретными поставщиками JPA, однако их использование становится невозможным ввиду принципа независимости библиотеки от реализации стандарта. Таким образом, был выбран подход, основанный на предварительной генерации кода слушателей.

Для задачи кодогенерации было принято решение использовать `Freemarker` – библиотеку в Java, которая применяется для генерации текстовой информации на основе шаблонов и подставляемых значений параметров [14]. Основными преимуществами работы с шаблонизаторами являются:

- специализированный язык описания шаблона, ограничения в синтаксисе которого не позволяют выйти за пределы предназначения шаблонизатора, что снижает вероятность ошибок на уровне языка программирования;
- ограниченность контекста: в рамках шаблона разработчик имеет доступ только к тем данным, которые предназначены для работы шаблона, исключая риск несанкционированного доступа к частям системы;
- расширенная функциональность с набором встроенных операций, упрощёнными записями конструкций и выражений.

Freemarker также поддерживает работу с макросами – блоками кода, которые можно определить и затем переиспользовать в нескольких местах шаблона [15]. Для структурирования генерации кода слушателя были созданы следующие макросы:

- структура класса, который определяет общий вид класса слушателя, включая объявление класса и необходимые аннотации;
- метод инициализации, в котором осуществляется контроль добавления необходимых компонентов;
- метод обратного вызова для контроля операции чтения данных.

В качестве значений переменных в блоки шаблона подставляются значения параметров из реестра. Эти параметры предварительно обрабатываются в программе и затем передаются генератору слушателей.

В листинге 2 показан фрагмент макроса для генерации структуры слушателя событий.

Листинг 2. Фрагмент макроса структуры класса

```
<#ftl encoding="utf-8">
<#import 'initMethod.ftlh' as initMethod/>
<#import 'postLoad.ftlh' as postLoadMethod/>
<#assign entityClassName = entityClass.getSimpleName()>
<#assign entityObjectName = entityClassName?uncap_first>
package ${packageName};
<...>
import ${entityClass.getName()};
<#if managers??>
<#list managers as managerClass>
import ${managerClass.getName()};
</#list>
</#if>
```

```
import java.util.*;

public class ${className} {
    <@initMethod.initTemplate managers=managers entityClass=entityClass/>
    <@postLoadMethod.postLoadTemplate listenerGeneratorDto=listenerGeneratorDto
entityClassName=entityClassName entityObjectName=entityObjectName/>
}
```

Результат генерации метода контроля чтения данных зависит от настройки типа загрузки (Fetch type) для каждого из внешних отношений сущности. JPA вводит два режима загрузки: «жадный» и «ленивый» [1]. При установленной «жадной» стратегии связанные сущности загружаются немедленно при чтении исходной, а при «ленивой» – только при явном обращении к ним в приложении. Это позволяет оптимизировать количество запросов к базе данных и выполнять их только при необходимости обработки отношения.

В зависимости от выбранной стратегии загрузки код слушателя будет содержать обращение к модулю контроля внешних отношений для немедленной загрузки или для отложенной, с помощью прокси-обёрток.

Сгенерированные слушатели помещаются в структуру пользовательского проекта.

5. МОДУЛЬ КОНТРОЛЯ ОТНОШЕНИЙ МЕЖДУ СУЩНОСТЯМИ

На предыдущем этапе работы библиотеки были сформированы реестры и сгенерированы слушатели событий. Эти слушатели подписываются на события жизненного цикла сущностей, имеющих внешние связи с объектами из других источников данных. Задачей слушателей является формирование и перенаправление информации о событии в модуль контроля отношений между сущностями. На рис. 5 изображена схема данного процесса.

На первом шаге обработки модуль обращается к реестрам для получения информации о настройках отношений и соответствующих прокси-классах связанных сущностей. В зависимости от типа связи этими операциями управляет соответствующий процессор отношений. Каждый процессор реализован для работы с определённым типом связи и умеет обращаться к необходимому методу менеджера сущностей, который вернет один элемент или целевую коллекцию.

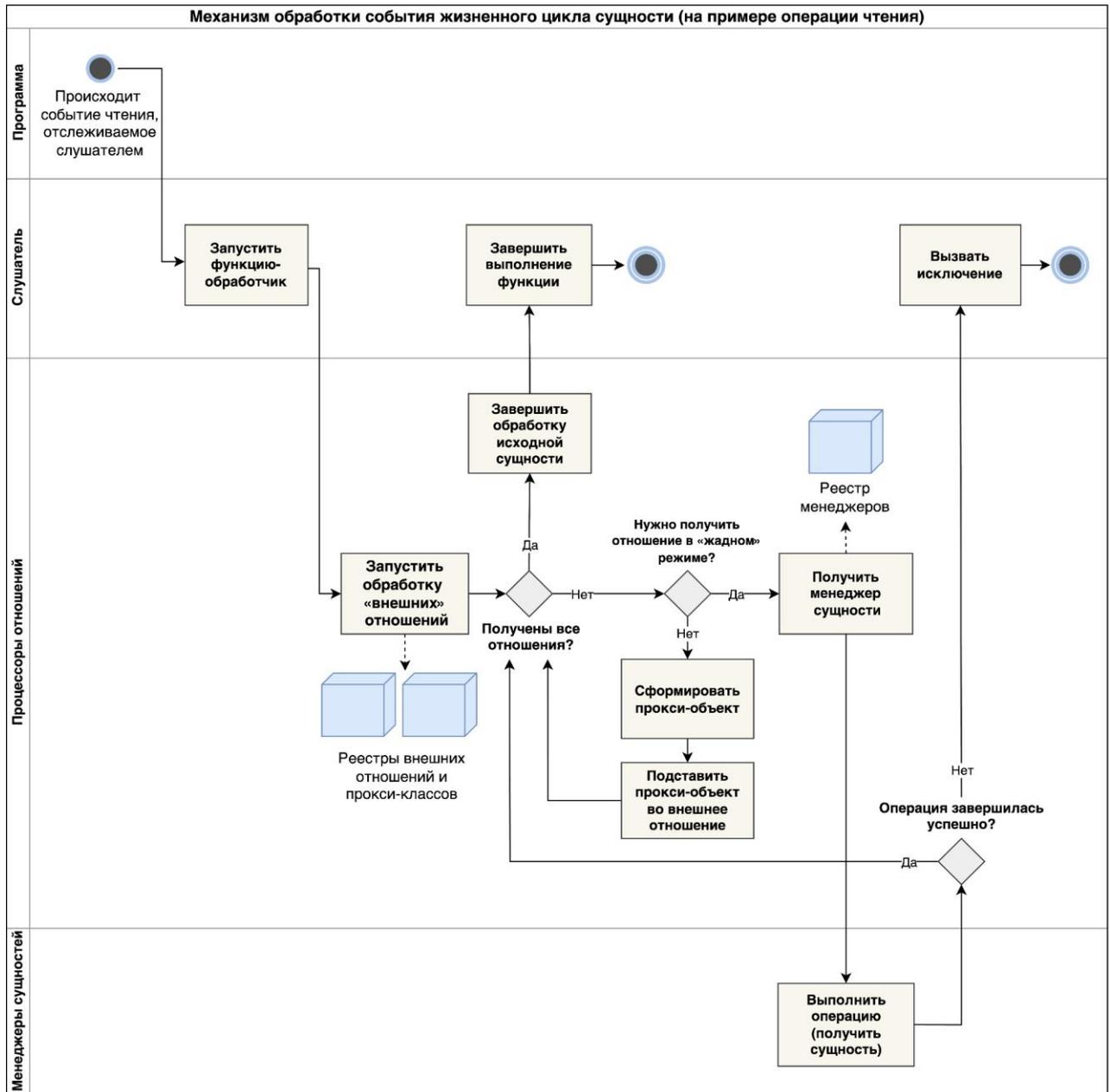


Рис 5. Механизм обработки внешних отношений

Одним из важнейших параметров конфигурации отношения является стратегия его загрузки.

При «жадной» загрузке процессор немедленно инициирует чтение связанных объектов из базы данных через менеджер сущностей.

В случае «ленивой» стратегии извлечение данных откладывается до момента фактического доступа к свойствам связанной сущности в коде приложения.

Для реализации подобной отложенной загрузки данных и идентификации осуществлённого обращения к атрибуту могут быть применены подходы, которые основаны на изменении логики вызываемых методов или подставляемых объектов. Одним из вариантов является модификация `get`-метода для получения данных об атрибуте отношения, но это противоречит концепции независимости библиотеки от реализации JPA, так как этот метод уже должен быть реализован в исходном классе сущности, поставщик JPA опирается на логику класса, определённую пользователем. Другой подход предполагает изменение алгоритма инициализации объекта целевого класса с использованием шаблона проектирования прокси. В Java для этой цели используется интерфейс `InvocationHandler`, который позволяет перехватывать вызовы методов целевого объекта [16]. Созданный прокси-объект связывается с экземпляром класса, реализующего интерфейс `InvocationHandler`, и делегирует выполнение методов данной логике. В контексте задачи определения момента обращения к атрибуту отношения метод `invoke`, принадлежащий данному интерфейсу, может использоваться для проверки состояния инициализации объекта и выполнения запроса к базе данных при первом обращении.

На рис. 6 показана схема данного взаимодействия описанных компонентов.

Также может возникнуть ситуация, когда у модели из внешнего отношения запрашиваемой исходной сущности есть другие внешние отношения, которые могут быть привязаны к иным источникам данных. Важным аспектом здесь является определение односторонних и двусторонних отношений. Одностороннее отношение характеризуется тем, что связь хранится только в одной из сущностей, в то время как двустороннее предполагает наличие ссылок обеих сущностей друг на друга. Для обеспечения корректного управления ссылками и предотвращения появления бесконечных циклов при обходе связей в двусторонних отношениях разработана соответствующая логика заполнения и проверок в процессорах отношений, учитывающая настройки и стратегию загрузки таких отношений.

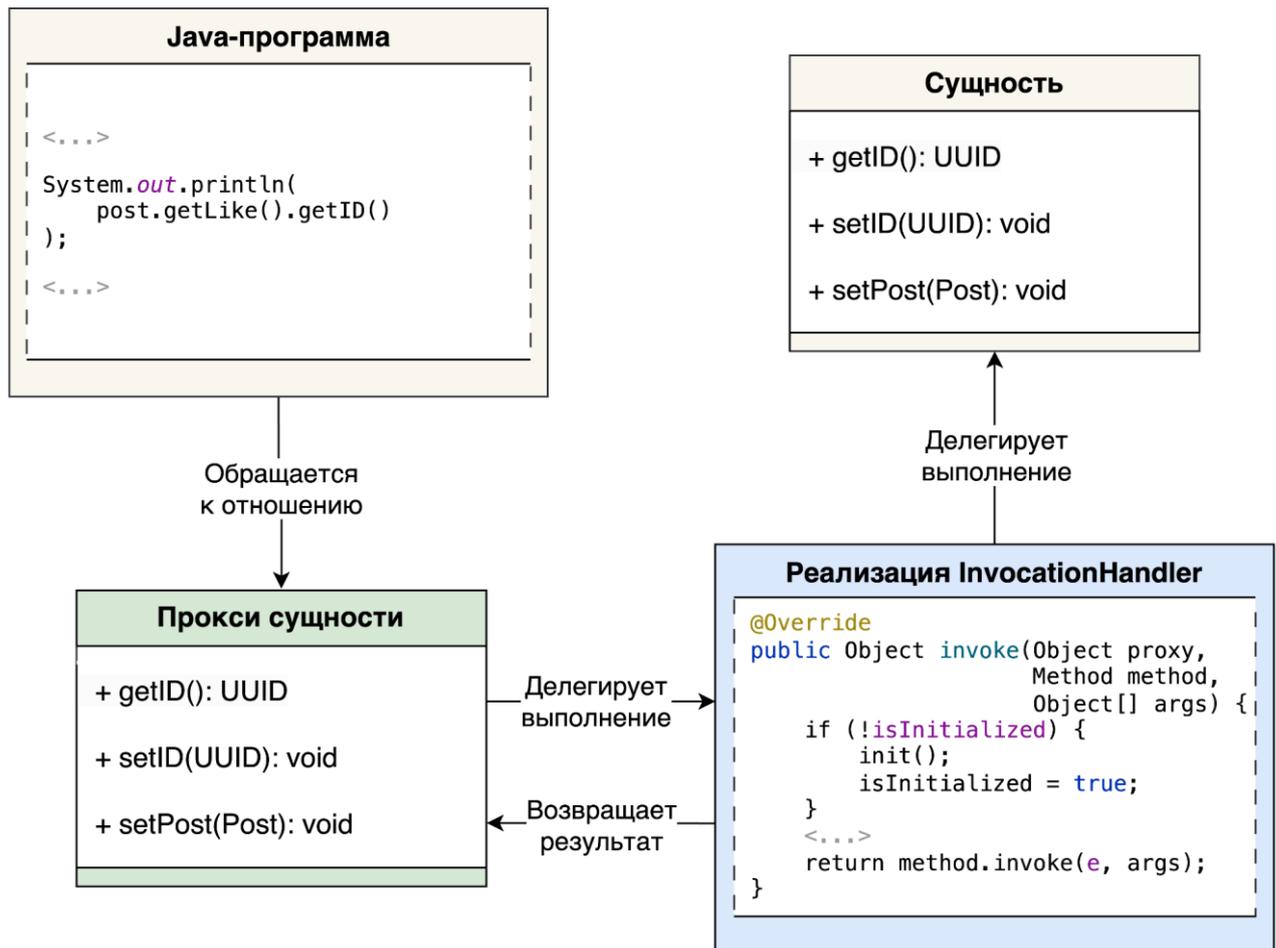


Рис. 6. Механизм взаимодействия с прокси в отношениях

6. ОЦЕНКА ПРОИЗВОДИТЕЛЬНОСТИ ИНСТРУМЕНТА

Оценка производительности проводилась путём сравнения времени выполнения операции чтения сущностей при двух подходах:

- осуществление контроля внешних отношений вручную: в данном случае разработчик самостоятельно реализует всю логику загрузки связанных сущностей;
- автоматизированный контроль с помощью разработанной библиотеки: управление связанными сущностями происходит с помощью предоставляемого функционала.

Отличие между этими подходами состоит в том, что при использовании библиотеки применяется рефлексия. Её рассмотренные механизмы позволяют исследовать и изменять внутреннее устройство объектов, а также модифициро-

вать их поведение во время работы программы, но требуют динамического разрешения типов, загрузки классов и информации, что влечёт дополнительные затраты во время выполнения [17].

Однако вся информация о сущностях, которая может быть получена с помощью рефлексии один раз, извлекается и сохраняется в реестры на этапе работы модуля сканирования и инициализации библиотеки. Это снижает влияние использования названных механизмов на производительность, что подтверждается проведёнными измерениями времени получения определённого количества сущностей.

Результат такого сравнения представлен на рис. 7.

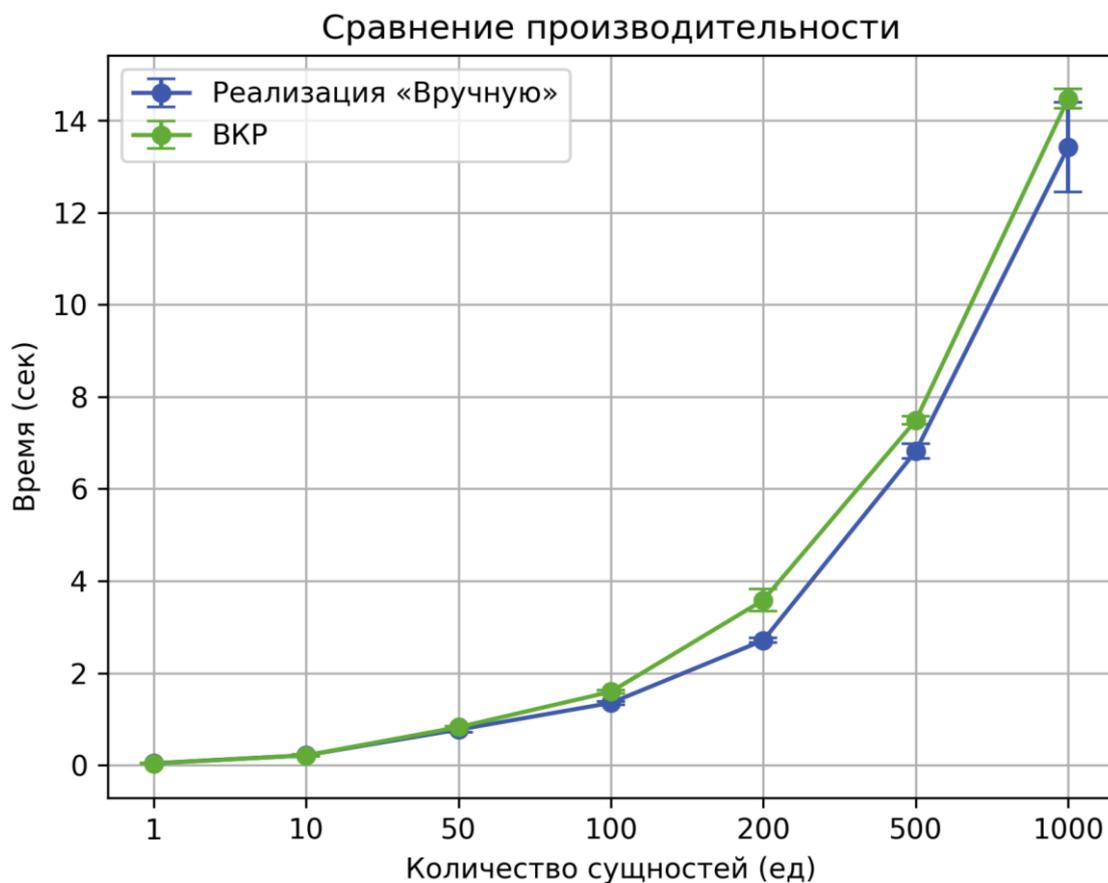


Рис. 7. Результаты измерения времени получения сущностей

Для тестирования были созданы две сущности: Entity1 (PostgreSQL) и MongoEntity (MongoDB). Сущности могли быть связаны между собой всеми типами отношений: «один-к-одному», «один-ко-многим», «многие-к-одному» и

«многие-ко-многим». Также эти структуры не содержали внутренних связей и не имели других внешних отношений, помимо перечисленных.

Генерация тестовых данных осуществлялась с помощью скрипта, который создавал экземпляры сущностей и в случайном порядке устанавливал между ними внешние отношения. Экземпляр мог не иметь внешних связей либо быть связан с другими сущностями посредством одного или нескольких (максимум четырёх) типов отношений. В случае имеющихся отношений «один-ко-многим» и «многие-ко-многим» скрипт генерировал от одной до десяти связей для каждого объекта.

Чтобы исключить влияние потенциальных выбросов, на графике, представленном на рис. 7, построены доверительные интервалы, которые были рассчитаны на основе 5 запусков с уровнем доверия 95% для каждой точки, соответствующей определённому количеству считываемых сущностей.

Согласно полученному результату, при небольшом количестве сущностей (1, 10, 50) время выполнения операций чтения сопоставимо для данных подходов, а с увеличением количества сущностей наблюдается рост времени выполнения в обоих случаях. Так, при обработке 1000 сущностей среднее время чтения для ручной реализации составило 13,42 секунды и 14,47 секунд для библиотеки. Однако при работе с сотнями и тысячами сущностей зачастую не требуется полная информация о каждой сущности. В таких случаях применяются другие подходы: пагинация, оптимизированные запросы получением исключительно основных свойств, «ленивая» загрузка. Таким образом, разработанная библиотека обеспечила сопоставимую с ручной реализацией производительность при работе с небольшим количеством сущностей, а в случае увеличения объемов данных демонстрирует приемлемую производительность, учитывая специфику практического применения.

ЗАКЛЮЧЕНИЕ

Описаны процесс и особенности разработки программного инструмента на основе стандарта JPA, автоматизирующего процесс управления связанными сущ-

ностями из реляционных и нереляционных баз данных в Java-приложениях, обеспечивающего при этом независимость от реализации JPA, используемой в проекте.

Исследованы определённые стандартом JPA возможности контроля жизненного цикла сущностей для обеспечения независимости от конкретной реализации, спроектирована стратегия встраивания программного решения в событийные процессы жизненного цикла сущностей, разработаны механизмы, автоматизирующие процесс управления связанными сущностями из реляционных и нереляционных баз данных, осуществлена оценка производительности разработанного решения. Инструмент обеспечивает работу с реляционными базами данных, перечень которых определяется JPA-фреймворком, используемым в проекте. В контексте нереляционных баз данных реализована поддержка MongoDB.

СПИСОК ЛИТЕРАТУРЫ

1. Jakarta Persistence. URL: <https://jakarta.ee/specifications/persistence/3.0/jakarta-persistence-spec-3.0.html/>.
2. Hibernate. URL: <https://hibernate.org/>.
3. Hibernate OGM. URL: <https://hibernate.org/ogm/faq/>.
4. What is Object-Relational Mapping (ORM) in DBMS?
URL: <https://www.geeksforgeeks.org/what-is-object-relational-mapping-orm-in-dbms/>.
5. JPA providers market share in 2016.
URL: <https://vladmihalcea.com/jpa-providers-market-share-in-2016/>.
6. Hibernate ORM User Guide.
URL: http://docs.jboss.org/hibernate/orm/6.5/userguide/html_single/Hibernate_User_Guide.html/.
7. EclipseLink/Examples/JPA/Caching.
URL: <https://wiki.eclipse.org/EclipseLink/Examples/JPA/Caching>.
8. JPQL.
URL: https://wiki.eclipse.org/EclipseLink/UserGuide/JPA/Basic_JPA_Development/Querying/JPQL.
9. Developing JAXB Applications Using EclipseLink MOXy.
URL: <https://eclipse.dev/eclipselink/documentation/2.4/moxy/overview001.htm/>.

10. Platform Specific Configurations.
URL: https://wiki.eclipse.org/EclipseLink/UserGuide/JPA/Advanced_JPA_Development/Platform_Specific_Configurations/.
11. EclipseLink/FAQ/NoSQL.
URL: <https://wiki.eclipse.org/EclipseLink/FAQ/NoSQL/>.
12. Composite Persistence Units.
URL: https://wiki.eclipse.org/EclipseLink/UserGuide/JPA/Advanced_JPA_Development/Composite_Persistence_Units/.
13. The Proxy Pattern in Java.
URL: <https://www.baeldung.com/java-proxy-pattern/>.
14. What is Apache FreeMarker™? URL: <https://freemarker.apache.org/>.
15. What are Macros?
URL: <https://academy.flowmailer.com/hc/en-gb/articles/9404831036306-What-are-Macros>
16. Dynamic Proxy Classes.
URL: <https://docs.oracle.com/javase/8/docs/technotes/guides/reflection/proxy.html/>.
17. The performance implications of Java reflection.
URL: <https://blogs.oracle.com/javamagazine/post/java-reflection-performance/>.

AUTOMATION OF READING RELATED DATA FROM RELATIONAL AND NON-RELATIONAL DATABASES IN THE CONTEXT OF USING THE JPA STANDARD

A. S. Savincheva¹ [0009-0005-0756-6001], A. A. Ferenets² [0000-0002-7859-9901]

^{1, 2}*Institute of Information Technologies and Intelligent Systems, Kazan (Volga Region) Federal University, ul. Kremlyovskaya, 35, Kazan, 420008*

¹asanvlit@gmail.com, ²ist.kazan@gmail.com

Abstract

The process of automating the management of the reading operation of related data from relational and non-relational databases is described.

The developed software tool is based on the use of the JPA (Java Persistence API) standard, which defines the capabilities of managing the lifecycle of entities in Java applications. An architecture for embedding in event processes has been designed, allowing the solution to be integrated into projects regardless of which JPA implementation is used. Support for various data loading strategies, types, and relationship parameters has been implemented. The performance of the tool has been evaluated.

Keywords: *JPA, ORM, Java, databases, relational databases, non-relational databases.*

REFERENCES

1. Jakarta Persistence. URL: <https://jakarta.ee/specifications/persistence/3.0/jakarta-persistence-spec-3.0.html/>.
2. Hibernate. URL: <https://hibernate.org/>.
3. Hibernate OGM. URL: <https://hibernate.org/ogm/faq/>.
4. What is Object-Relational Mapping (ORM) in DBMS?
URL: <https://www.geeksforgeeks.org/what-is-object-relational-mapping-orm-in-dbms/>.
5. JPA providers market share in 2016.
URL: <https://vladmihalcea.com/jpa-providers-market-share-in-2016/>.
6. Hibernate ORM User Guide.

URL: http://docs.jboss.org/hibernate/orm/6.5/userguide/html_single/Hibernate_User_Guide.html/.

7. EclipseLink/Examples/JPA/Caching.

URL: <https://wiki.eclipse.org/EclipseLink/Examples/JPA/Caching>.

8. JPQL.

URL: https://wiki.eclipse.org/EclipseLink/UserGuide/JPA/Basic_JPA_Development/Querying/JPQL.

9. Developing JAXB Applications Using EclipseLink MOXy.

URL: <https://eclipse.dev/eclipselink/documentation/2.4/moxy/overview001.htm/>.

10. Platform Specific Configurations.

URL: https://wiki.eclipse.org/EclipseLink/UserGuide/JPA/Advanced_JPA_Development/Platform_Specific_Configurations/.

11. EclipseLink/FAQ/NoSQL.

URL: <https://wiki.eclipse.org/EclipseLink/FAQ/NoSQL/>.

12. Composite Persistence Units.

URL: https://wiki.eclipse.org/EclipseLink/UserGuide/JPA/Advanced_JPA_Development/Composite_Persistence_Units/.

13. The Proxy Pattern in Java.

URL: <https://www.baeldung.com/java-proxy-pattern/>.

14. What is Apache FreeMarker™? URL: <https://freemarker.apache.org/>.

15. What are Macros?

URL: <https://academy.flowmailer.com/hc/en-gb/articles/9404831036306-What-are-Macros>

16. Dynamic Proxy Classes.

URL: <https://docs.oracle.com/javase/8/docs/technotes/guides/reflection/proxy.html/>

17. The performance implications of Java reflection.

URL: <https://blogs.oracle.com/javamagazine/post/java-reflection-performance/>

СВЕДЕНИЯ ОБ АВТОРАХ



САВИНЧЕВА Ангелина Сергеевна – выпускник бакалавриата Института информационных технологий и интеллектуальных систем Казанского (Приволжского) федерального университета, г. Казань.

Angelina SAVINCHEVA – graduate student of the Institute of Information Technologies and Intelligent Systems, Kazan (Volga region) Federal University, Kazan.

email: asanvlit@gmail.com

ORCID: 0009-0005-0756-6001



ФЕРЕНЕЦ Александр Андреевич – старший преподаватель кафедры программной инженерии Института информационных технологий и интеллектуальных систем Казанского (Приволжского) федерального университета, г. Казань.

Alexander FERENETS – senior lecturer of Software Engineering of Institute of Information Technologies and Intelligent Systems KFU.

email: ist.kazan@gmail.com

ORCID: 0000-0002-7859-9901

Материал поступил в редакцию 7 июля 2024 года

УДК 004.056.523

ПРИМЕНЕНИЕ АЛГОРИТМА ДУГЛАСА–ПЕККЕРА В ВОПРОСАХ ОНЛАЙН-АУТЕНТИФИКАЦИИ ИНСТРУМЕНТОВ УДАЛЁННОЙ РАБОТЫ ПРИ ПОДГОТОВКЕ СПЕЦИАЛИСТОВ УКРУПНЁННОЙ ГРУППЫ СПЕЦИАЛЬНОСТЕЙ 10.00.00 «ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ»

А. Г. Уймин¹ [0000-0003-1572-5488], **В. С. Греков**² [0009-0006-4067-4976]

^{1,2} *Российский государственный университет нефти и газа (Национальный исследовательский университет) имени И.М. Губкина*

¹au-mail@ya.ru, ²grekov.vs.work@gmail.com

Аннотация

В условиях перехода образовательных систем на дистанционное обучение, а также развития тренда на удалённую работу, возникла острая потребность в разработке надежных технологий биометрической идентификации и аутентификации для верификации исполнителей работ в режиме удаленной работы. Такие технологии позволяют обеспечить высокую степень защиты и удобство использования, что делает вопросы их разработки и оптимизации крайне важными.

Проблема заключается в необходимости повышения точности и эффективности систем распознавания движений манипулятора «мышь» без использования специализированных устройств в максимально короткий промежуток времени. Для ее решения требуется эффективная предобработка таких движений, чтобы упростить их траектории, сохранив при этом их ключевые особенности.

В статье предложено использование алгоритма Дугласа–Пеккера для предварительной обработки данных траекторий движений «мыши». Этот алгоритм позволяет значительно уменьшить количество точек в траекториях, упрощая их при сохранении основной формы движений. Данные с упрощенными траекториями затем используются для обучения нейронных сетей.

Экспериментальная часть работы показала, что применение алгоритма Дугласа–Пеккера позволяет сократить количество точек в траекториях на 60%, что приводит к увеличению точности распознавания движений с 70% до 82%. Такое упрощение данных способствует ускорению процесса обучения нейронных сетей и повышению их операционной эффективности.

Проведенное исследование подтвердило эффективность использования алгоритма Дугласа–Пеккера для предварительной обработки данных в задачах распознавания движений «мыши». Полученные результаты могут найти применение в разработке более интуитивно понятных и адаптивных пользовательских интерфейсов.

Предложены также направления для дальнейших исследований, включая оптимизацию параметров алгоритма для различных типов движений и исследование возможности его комбинирования с другими методами машинного обучения.

Ключевые слова: аутентификация, биометрическая идентификация, удалённая работа, дистанционное обучение, алгоритм Дугласа–Пеккера, предобработка данных, нейросеть, HID-устройство, траектория движений «мыши», оптимизация данных.

ВВЕДЕНИЕ

В современном мире цифровые технологии проникают во все аспекты человеческой деятельности, включая образование и трудовую деятельность. С 2019 года, когда в ответ на глобальные вызовы мировые образовательные системы активно начали переходить на дистанционные формы обучения, возникла острая необходимость в разработке и внедрении надежных технологий идентификации и аутентификации. Эти технологии необходимы для обеспечения подлинности выполнения работ и защиты от фальсификации академических достижений, особенно в контексте получения высшего образования в соответствии с укрупненной группой специальностей и направлений (УГСН) 10.00.00 – Информационная безопасность, где лабораторные и практические работы играют ключевую роль в учебном процессе.

В ответ на увеличивающуюся потребность в удаленной аутентификации пользователей особое внимание было уделено разработке методов биометрической идентификации. Прокторинг с использованием видеотрансляций и распознавание HID-устройств (Human Interface Devices) стали наиболее популярными подходами. Среди HID-устройств манипуляторы типа «мышь» выделяются благодаря их устойчивости к попыткам подделки и подмены, что делает их предпочтительным выбором для систем аутентификации. Тем не менее, проблема оптимизации потока входных данных от этих устройств остается актуальной. Во-первых, большой объем данных может привести к переобучению нейронных сетей, являющихся ядром системы распознавания. Во-вторых, высокие требования к пропускной способности сети могут ограничить широкое внедрение технологий биометрической аутентификации в дистанционное образование.

Одним из решений указанных проблем является использование алгоритма Дугласа–Пеккера¹ для предварительной обработки данных траекторий жестов мыши. Этот алгоритм позволяет существенно уменьшить количество точек в траекториях, тем самым упрощая их и сохраняя при этом ключевые особенности движений. Такой подход не только облегчает нагрузку на сетевые и вычислительные ресурсы, но и повышает точность и эффективность последующего распознавания движений с помощью нейронных сетей.

В статье представлен обзор существующих методов аутентификации, проанализирована актуальность проблемы биометрической идентификации в условиях дистанционного образования и работы и обоснован выбор алгоритма Дугласа–Пеккера как оптимального решения для предварительной обработки данных движений «мыши». Особое внимание уделено экспериментальной проверке эффективности предложенного подхода и его потенциалу для повышения надежности и удобства систем дистанционной аутентификации.

¹ https://rosettacode.org/wiki/Ramer-Douglas-Peucker_line_simplification

ОБЗОР ЛИТЕРАТУРЫ

В области биометрической аутентификации и распознавания проведено множество исследований, охватывающих различные аспекты от общих подходов к биометрии до конкретных приложений. В данном обзоре литературы мы рассмотрим научные работы, которые представляют значимый вклад в развитие названной области, начиная от общих обзоров биометрических систем безопасности и заканчивая специфическими методами обработки данных и их применением.

В статье [1] приведён обзор биометрических подходов к аутентификации, охватывающий широкий спектр технологий и подходов. Эта работа служит введением в область биометрической аутентификации, представляя различные методы и их применение.

Обзор, выполненный в [2], представляет собой всесторонний анализ производительности, безопасности и схем защиты шаблонов биометрии в системах аутентификации. Это исследование содержит критический взгляд на существующие подходы и предлагает направления для усиления безопасности биометрических систем.

В работе [3] представлен обширный обзор безопасности биометрической аутентификации, рассмотрены основные концепции и вызовы, стоящие перед системами биометрической аутентификации. Это исследование формирует фундаментальную базу для понимания текущего состояния технологий в области биометрии и подчеркивает важность разработки надежных методов аутентификации.

В [4] проведено исследование уязвимости биометрических систем аутентификации на основе данных с клавиатуры и компьютерной «мыши» перед атаками с использованием техник «черного ящика» и специфических для домена. Это исследование подчеркивает важность разработки устойчивых к атакам систем биометрической аутентификации.

Авторы исследования [5] предлагают обширный обзор методов ненавязчивой активной аутентификации пользователя в биометрии, акцентируя внимание на комфорте пользователя и невмешательстве в его деятельность при обеспечении безопасности.

Далее, в [6, 7] проведён углубленный анализ методов непрерывной аутентификации, использующих биометрические данные. Эти работы акцентируют внимание на важность разработки систем, способных обеспечивать постоянную и надежную аутентификацию пользователя в реальном времени.

В области анализа поведенческих биометрических данных, особенно в контексте использования данных компьютерной «мыши», значительный вклад внесли работы [8, 9]. Эти исследования фокусируются на анализе потоковых данных кликов «мыши» как средства для непрерывной аутентификации пользователя.

Исследования [10, 11] дополнительно подчеркивают потенциал поведенческих биометрических данных, полученных с помощью «мыши», для создания надежных систем аутентификации.

Работы [12, 13] иллюстрируют применение машинного и глубокого обучения для анализа динамики движений «мыши», подчеркивая значимость этих подходов для улучшения точности и надежности биометрической аутентификации.

В контексте образовательных технологий исследования [14–16] раскрывают возможности использования биометрической аутентификации для обеспечения честности и подлинности образовательного процесса в дистанционных условиях.

Исследование [17] представляет собой углубленный анализ алгоритма Дугласа–Пеккера, нацеленный на определение оптимальных значений его параметров для систем распознавания онлайн-рукописного ввода. Эта работа значительно обогащает понимание применения алгоритма для упрощения данных, что непосредственно связано с областью биометрической аутентификации.

В статье [18] рассмотрена разработка гибкой системы для отслеживания движений компьютерной «мыши» и оценки поведения, что предоставляет ценные данные для анализа поведенческих биометрий. Это исследование расширяет горизонты использования биометрических данных, полученных с помощью устройств ввода, для аутентификации.

Отметим работы [19, 20], которые специализируются на разработке и анализе инструментов для верификации студентов и предварительной обработки данных с манипуляторов «мышь» для использования в анализе поведенческих биометрий. Эти исследования предлагают конкретные методы и инструменты для улучшения точности и эффективности систем аутентификации в образовательной среде.

Приведенный обзор литературы подчеркивает многоуровневый подход к разработке и внедрению биометрических систем аутентификации, от теоретических основ до конкретных технологических решений, и иллюстрирует широкий спектр исследований, направленных на повышение безопасности, надежности и доступности биометрической аутентификации в различных областях применения.

МАТЕРИАЛЫ И МЕТОДЫ

Для проведения исследования использовались стандартные HID-устройства («мыши»), подключаемые к персональным компьютерам испытуемых. Данные движений «мыши» собирались в реальном времени в ходе выполнения типовых задач, имитирующих действия пользователя в процессах дистанционной работы или обучения. Сбор данных осуществлялся с помощью специально разработанного программного обеспечения RemoteTopology, регистрирующего движения «мыши» с высокой точностью.

Алгоритм Дугласа–Пеккера — это метод уменьшения количества точек в ломаной линии без значительной потери формы. Алгоритм определяет точки, которые можно исключить, минимизируя при этом искажение исходной траектории [17]. В контексте данного исследования алгоритм применялся для упрощения траекторий жестов «мыши» с целью снижения объема данных перед их обработкой нейронными сетями.

Предобработка данных включала несколько этапов:

- Фильтрация шумов и случайных движений, не относящихся к целенаправленным.
- Применение алгоритма Дугласа–Пеккера для уменьшения количества точек в траекториях движений.

- Нормализация данных для унификации входных параметров для обучения нейронных сетей.

В качестве ядра системы распознавания использовались нейронные сети, обученные на данных с упрощенными и неупрощенными траекториями жестов, для сравнения результатов. Подбор архитектуры нейронных сетей, их параметров и гиперпараметров осуществлялся на основе предварительных экспериментов с целью максимизации точности распознавания движений.

Эффективность алгоритма Дугласа–Пеккера оценивалась по следующим критериям:

- Уменьшение объема данных: измерялось процентное сокращение количества точек в траекториях движений после применения алгоритма.
- Точность распознавания жестов: сравнивалась точность идентификации движений нейронными сетями до и после упрощения траекторий.
- Время обучения нейронных сетей: измерялось время, необходимое для обучения сетей на упрощенных и неупрощенных данных.

Эксперименты проводились в контролируемой среде с активным участием группы испытуемых. Участникам были предложены задания, специально разработанные для имитации реальных сценариев использования. Проведение сравнительного анализа результатов обучения нейронных сетей на исходных и обработанных алгоритмом Дугласа–Пеккера данных позволило детально оценить влияние этого метода на эффективность системы биометрической аутентификации, подчеркивая его значимость в реальных сценариях применения.

РЕЗУЛЬТАТЫ

Эксперимент проводился с участием студентов и молодых профессионалов, активно вовлеченных в использование компьютерных технологий в своей повседневной деятельности. Всего было 50 испытуемых, среди кото-

рых 30 мужчин и 20 женщин в возрасте от 18 до 35 лет. Перед началом эксперимента каждому участнику была предоставлена информация о целях исследования, и они дали свое согласие на участие.

Модель нейронной сети была выбрана на основе экспериментальной оценки нескольких архитектур: KNN (k-nearest neighbours, k-ближайших соседей), CNN (convolutional neural networks, свёрточные нейронные сети), RNN (recurrent neural network, рекуррентные нейронные сети).

Результаты эксперимента представлены в таблице 1.

Таблица 1. Результаты эксперимента

Параметр	До применения алгоритма	После применения алгоритма
Среднее количество точек в траектории	1584	634
Средняя точность распознавания, %	70	82
Среднее время обучения, мин	120	84

Результаты демонстрируют значительное улучшение в эффективности системы распознавания жестов «мыши» после применения алгоритма Дугласа–Пеккера. Уменьшение количества точек в траекториях на 60% привело к сокращению времени обучения на 30% и увеличению точности распознавания на 12%.

ОБСУЖДЕНИЕ

Результаты проведенных экспериментов ясно показывают, что применение алгоритма Дугласа–Пеккера для предварительной обработки данных траекторий «мыши» в рамках систем биометрической аутентификации имеет огромные преимущества. Уменьшение объема данных не только облегчает их обработку и сокращает время обучения нейронных сетей, но и

существенно повышает точность распознавания, что крайне важно для эффективности таких систем.

Точность распознавания движений играет ключевую роль в системах онлайн-аутентификации, и увеличение этого показателя на 12% может значительно повлиять на надежность и безопасность процессов идентификации. Это подчеркивает важность выбора подходящих методов предварительной обработки данных при разработке биометрических систем идентификации.

Кроме того, уменьшение времени обучения нейронных сетей также является важным результатом, поскольку позволяет более эффективно использовать вычислительные ресурсы. Это особенно актуально в условиях ограниченных сетевых и временных ресурсов, которые характерны для дистанционного обучения и работы.

Однако важно также обсудить потенциальные ограничения проведенного исследования. Например, необходимо изучить влияние разнообразия жестов и их сложности на эффективность применения алгоритма Дугласа–Пеккера. Также важно рассмотреть возможности дальнейшего улучшения точности распознавания путем оптимизации параметров алгоритма и комбинирования его с другими методами предварительной обработки данных и машинного обучения.

Результаты, приведенные выше, открывают новые горизонты для дальнейших исследований в области биометрической аутентификации и разработки пользовательских интерфейсов, направленных на улучшение взаимодействия между пользователем и системой в условиях дистанционной работы и обучения.

СПИСОК ЛИТЕРАТУРЫ

1. *Yusuf N. et al.* A survey of biometric approaches of authentication // International Journal of Advanced Computer Research. 2020. Vol. 10. No. 47. P. 96–104.

2. *Sarkar A., Singh B.K.* A review on performance, security and various biometric template protection schemes for biometric authentication systems // *Multimedia Tools and Applications*. 2020. Vol. 79. No. 37. P. 27721–27776.
3. *Alwahaishi S., Zdrálek J.* Biometric authentication security: an overview // *2020 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM)*. IEEE, 2020. P. 87–91.
4. *López C. et al.* Adversarial attacks against mouse-and keyboard-based biometric authentication: black-box versus domain-specific techniques // *International Journal of Information Security*. 2023. Vol. 22. No. 6. P. 1665–1685.
5. *Thomas P.A., Preetha Mathew K.* A broad review on non-intrusive active user authentication in biometrics // *Journal of Ambient Intelligence and Humanized Computing*. 2023. Vol. 14. No. 1. P. 339–360.
6. *Dahia G., Jesus L., Pamplona Segundo M.* Continuous authentication using biometrics: An advanced review // *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*. 2020. Vol. 10. No. 4. P. e1365.
7. *Ryu R. et al.* Continuous multimodal biometric authentication schemes: a systematic review // *IEEE Access*. 2021. Vol. 9. P. 34541–34557.
8. *Almalki S., Assery N., Roy K.* An empirical evaluation of online continuous authentication and anomaly detection using mouse clickstream data analysis // *Applied Sciences*. 2021. Vol. 11. No. 13. P. 6083.
9. *Almalki S., Chatterjee P., Roy K.* Continuous authentication using mouse clickstream data analysis // *Security, Privacy, and Anonymity in Computation, Communication, and Storage: SpaCCS 2019 International Workshops, Atlanta, GA, USA, July 14–17, 2019, Proceedings 12*. Springer International Publishing, 2019. P. 76–85.
10. *Jorgensen Z., Yu T.* On mouse dynamics as a behavioral biometric for authentication // *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*. 2011. P. 476–482.
11. *Sayed B. et al.* Biometric authentication using mouse gesture dynamics // *IEEE systems journal*. 2013. Vol. 7. No. 2. P. 262–274.

12. *Siddiqui N. et al.* Machine and deep learning applications to mouse dynamics for continuous user authentication // *Machine Learning and Knowledge Extraction*. 2022. Vol. 4. No. 2. P. 502–518.
13. *Zhang Y.G. et al.* Trustworthy interaction model: continuous authentication using time–frequency joint analysis of mouse biometrics // *Behaviour & Information Technology*. 2024. P. 1–18.
14. *Fenu G., Marras M., Boratto L.* A multi-biometric system for continuous student authentication in e-learning platforms // *Pattern Recognition Letters*. 2018. Vol. 113. P. 83–92.
15. *Gao Q.* Biometric authentication to prevent e-cheating // *International Journal of Instructional Technology and Distance Learning*. 2012. Vol. 9. No. 2. P. 3–14.
16. *Hernandez-Ortega J. et al.* edBB: Biometrics and behavior for assessing remote education // *arXiv preprint arXiv:1912.04786*. 2019.
17. *Abuzaraida M.A., Jebriel S.M.* The detection of the suitable reduction value of Douglas–Peucker algorithm in online handwritten recognition systems // *2015 IEEE International Conference on Service Operations and Logistics, And Informatics (SOLI)*. IEEE, 2015. P. 82–87.
18. *Fong T. et al.* PyMouseTracks: Flexible Computer Vision and RFID-Based System for Multiple Mouse Tracking and Behavioral Assessment // *eNeuro*. 2023. Vol. 10. No. 5.
19. *Uymin A.* Instruments for student verification and assessment of his emotional and psychological state during remote work // *Norwegian Journal of development of the International Science*. 2022. No. 96. P. 98–101.
20. *Uymin A.* Preprocessing data from the mouse manipulator for use in behavioral biometrics analysis // *Norwegian Journal of Development of the International Science*. 2022. No. 85. P. 53–58.

APPLICATION OF THE DOUGLAS-PEUCKER ALGORITHM IN ONLINE AUTHENTICATION OF REMOTE WORK TOOLS FOR SPECIALIST TRAINING IN HIGHER EDUCATION GROUP OF SCIENTIFIC SPECIALTIES (UGSN) 10.00.00

A. G. Uymin¹ [0000-0003-1572-5488], **V. S. Grekov**² [0009-0006-4067-4976]

^{1,2}*National University of Oil and Gas «Gubkin University»*

¹au-mail@ya.ru, ²grekov.vs.work@gmail.com

Abstract

In today's world, digital technologies are penetrating all aspects of human activity, including education and labor. Since 2019, when, in response to global challenges, the world's educational systems have actively started to shift to distance learning, there has been an urgent need to develop and implement reliable identification and authentication technologies. These technologies are necessary to ensure the authenticity of work and protection from falsification of academic achievements, especially in the context of higher education in accordance with the group of specialties and directions (USGS) 10.00.00 - Information Security, where laboratory and practical work play a key role in the educational process.

The problem lies in the need to optimize the flow of incoming data, which, first, can affect the retraining of the neural network core of the recognition system, and second, impose excessive requirements on the network's bandwidth. To solve this problem, efficient preprocessing of gesture data is required to simplify their trajectories while preserving the key features of the gestures.

This article proposes the use of the Douglas–Peucker algorithm for preliminary processing of mouse gesture trajectory data. This algorithm significantly reduces the number of points in the trajectories, simplifying them while preserving the main shape of the gestures. The data with simplified trajectories are then used to train neural networks.

The experimental part of the work showed that the application of the Douglas–Peucker algorithm allows for a 60% reduction in the number of points in the trajectories, leading to an increase in gesture recognition accuracy from

70% to 82%. Such data simplification contributes to speeding up the neural networks' training process and improving their operational efficiency.

The study confirmed the effectiveness of using the Douglas–Peucker algorithm for preliminary data processing in mouse gesture recognition tasks. The article suggests directions for further research, including the optimization of the algorithm's parameters for different types of gestures and exploring the possibility of combining it with other machine learning methods. The obtained results can be applied to developing more intuitive and adaptive user interfaces.

Keywords: *authentication, biometric identification, remote work, distance learning, Douglas–Peucker algorithm, data preprocessing, neural network, HID devices, mouse gesture trajectories, data optimization.*

REFERENCES

1. Yusuf N. et al. A survey of biometric approaches of authentication // International Journal of Advanced Computer Research. 2020. Vol. 10. No. 47. P. 96–104.
2. Sarkar A., Singh B.K. A review on performance, security and various biometric template protection schemes for biometric authentication systems // Multimedia Tools and Applications. 2020. Vol. 79. No. 37. P. 27721–27776.
3. Alwahaishi S., Zdrálek J. Biometric authentication security: an overview // 2020 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM). IEEE, 2020. P. 87–91.
4. López C. et al. Adversarial attacks against mouse-and keyboard-based biometric authentication: black-box versus domain-specific techniques // International Journal of Information Security. 2023. Vol. 22. No. 6. P. 1665–1685.
5. Thomas P.A., Preetha Mathew K. A broad review on non-intrusive active user authentication in biometrics // Journal of Ambient Intelligence and Humanized Computing. 2023. Vol. 14. No. 1. P. 339–360.
6. Dahia G., Jesus L., Pamplona Segundo M. Continuous authentication using biometrics: An advanced review // Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. 2020. Vol. 10. No. 4. P. e1365.

7. *Ryu R. et al.* Continuous multimodal biometric authentication schemes: a systematic review // *IEEE Access*. 2021. Vol. 9. P. 34541–34557.
8. *Almalki S., Assery N., Roy K.* An empirical evaluation of online continuous authentication and anomaly detection using mouse clickstream data analysis // *Applied Sciences*. 2021. Vol. 11. No. 13. P. 6083.
9. *Almalki S., Chatterjee P., Roy K.* Continuous authentication using mouse clickstream data analysis // *Security, Privacy, and Anonymity in Computation, Communication, and Storage: SpaCCS 2019 International Workshops, Atlanta, GA, USA, July 14–17, 2019, Proceedings 12*. Springer International Publishing, 2019. P. 76–85.
10. *Jorgensen Z., Yu T.* On mouse dynamics as a behavioral biometric for authentication // *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*. 2011. P. 476–482.
11. *Sayed B. et al.* Biometric authentication using mouse gesture dynamics // *IEEE systems journal*. 2013. Vol. 7. No. 2. P. 262–274.
12. *Siddiqui N. et al.* Machine and deep learning applications to mouse dynamics for continuous user authentication // *Machine Learning and Knowledge Extraction*. 2022. Vol. 4. No. 2. P. 502–518.
13. *Zhang Y.G. et al.* Trustworthy interaction model: continuous authentication using time–frequency joint analysis of mouse biometrics // *Behaviour & Information Technology*. 2024. P. 1–18.
14. *Fenu G., Marras M., Boratto L.* A multi-biometric system for continuous student authentication in e-learning platforms // *Pattern Recognition Letters*. 2018. Vol. 113. P. 83–92.
15. *Gao Q.* Biometric authentication to prevent e-cheating // *International Journal of Instructional Technology and Distance Learning*. 2012. Vol. 9. No. 2. P. 3–14.
16. *Hernandez-Ortega J. et al.* edBB: Biometrics and behavior for assessing remote education // *arXiv preprint arXiv:1912.04786*. 2019.
17. *Abuzaraida M.A., Jebriel S.M.* The detection of the suitable reduction value of Douglas–Peucker algorithm in online handwritten recognition systems // *2015 IEEE International Conference on Service Operations and Logistics, And Informatics (SOLI)*. IEEE, 2015. P. 82–87.

18. *Fong T. et al.* PyMouseTracks: Flexible Computer Vision and RFID-Based System for Multiple Mouse Tracking and Behavioral Assessment // *eNeuro*. 2023. Vol. 10. No. 5.

19. *Uymin A.* Instruments for student verification and assessment of his emotional and psychological state during remote work // *Norwegian Journal of development of the International Science*. 2022. No. 96. P. 98–101.

20. *Uymin A.* Preprocessing data from the mouse manipulator for use in behavioral biometrics analysis // *Norwegian Journal of Development of the International Science*. 2022. No. 85. P. 53–58.

СВЕДЕНИЯ ОБ АВТОРАХ



УЙМИН Антон Григорьевич – ст. преподаватель, федеральное государственное автономное образовательное учреждение высшего образования «Российский государственный университет нефти и газа (Национальный исследовательский университет) имени И.М. Губкина»

Anton Grigorievich UYMIN – Senior Lecturer, Federal State Autonomous Educational Institution of Higher Education "Gubkin Russian State University of Oil and Gas (National Research University)".

email: au-mail@ya.ru

ORCID: 0000-0003-1572-5488



ГРЕКОВ Владимир Сергеевич – ассистент, Федеральное государственное автономное образовательное учреждение высшего образования «Российский государственный университет нефти и газа (национальный исследовательский университет) имени И.М. Губкина»

Vladimir Sergeevich GREKOV – Assistant, Federal State Autonomous Educational Institution of Higher Education "Gubkin Russian State University of Oil and Gas (National Research University)".

email: grekov.vs.work@gmail.com

ORCID: 0009-0006-4067-4976

Материал поступил в редакцию 30 марта 2024 г.